SCRATCH. ПРОГРАМУЄМО ІГРИ
25 уроків для дітей

Версія уроків 1.20 (2 лютого 2021)
Автор збірника: Сергій Курінний

Пропоную вашій увазі уроки Scratch 3 для дітей, за допомогою яких можна навчитися програмувати ігри. Уроки для дітей 8+.

За допомогою Scratch ви можете програмувати інтерактивні історії, ігри, анімацію та ділитися своїми проектами з іншими в Інтернет.

Програма Scratch працює у браузері. Також її можна встановити на комп'ютер. Посилання: https://scratch.mit.edu/projects/editor

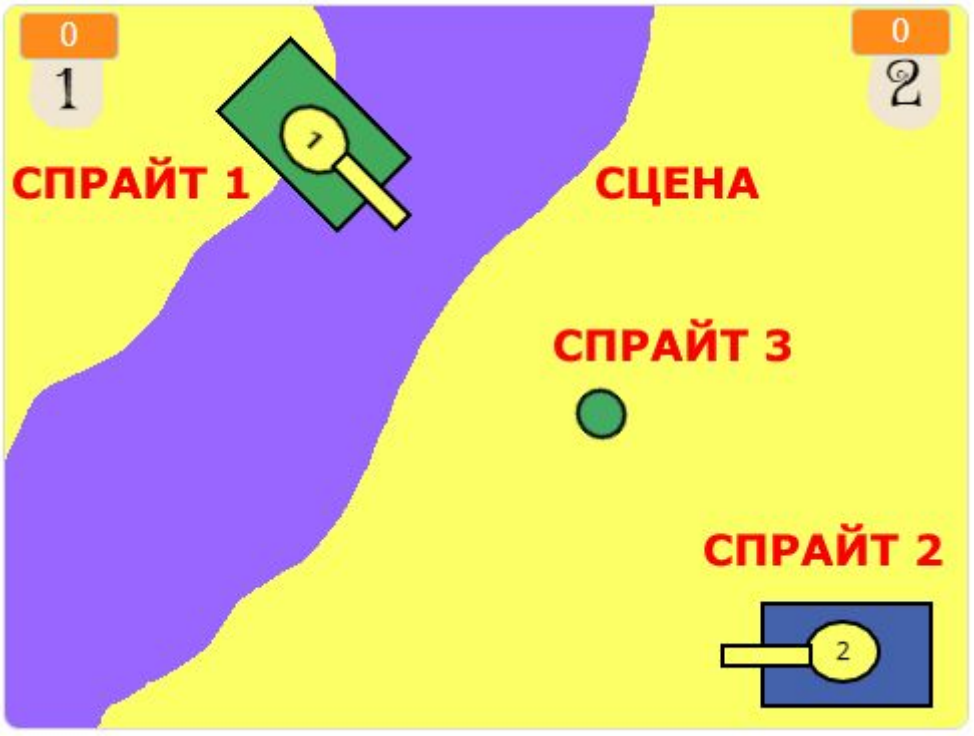
Деякі уроки переведені з російської мови, деякі - з англійської, деякі я розробив самостійно. Для деяких уроків я зняв відео (посилання в тексті уроку). Уроки зроблено для себе (я викладаю на гуртку), але буду радий, якщо вони стануть в нагоді й вам. В кінці - довідники блоків Scratch та додаткові завдання.

ЗВОРОТНІЙ ЗВ'ЯЗОК

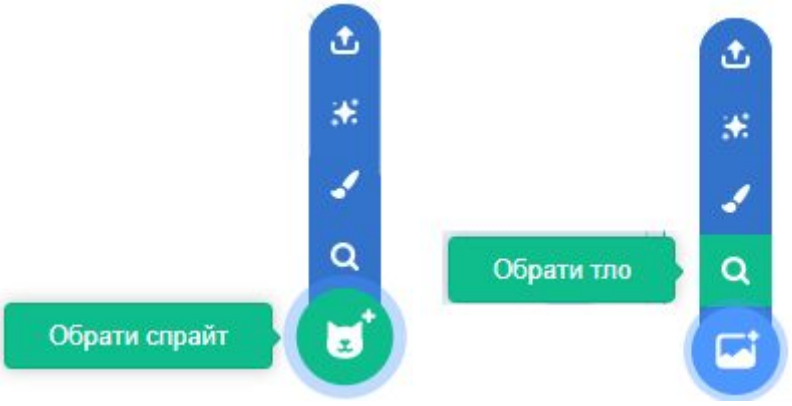
Побажання, коментарі, повідомлення про помилки можна надсилати:
Сергій Курінний
Email: general.loki.2018@gmail.com
FACEBOOK: fb.com/sergiy.kurinny
ВАЙБЕР/телеграм: +38 (096) 311-54-55

СПРАЙТИ ТА СЦЕНА

- **СПРАЙТ** у казках - це міфологічна істота, яка схожа на фею чи привида. Спрайт у програмуванні - це двовимірний персонаж або об’єкт гри, яким ви можете керувати за допомогою команд.
- **СЦЕНА** – об’єкт гри, якому ви можете давати інструкції. Сцена дозволяє керувати **ТЛОМ** та іншими речами, які стосуються всієї гри. Тло – не те саме, що сцена.

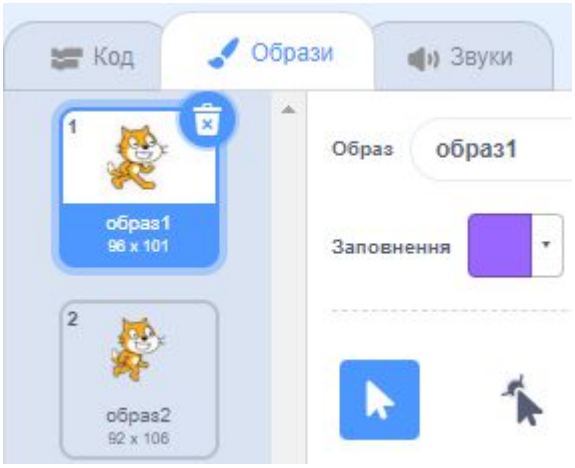


- Ви можете намалювати спрайт і тло сцени, завантажити їх з комп’ютера або обрати з вбудованої бібліотеки.



- **ОБРАЗ СПРАЙТА** – це його зовнішній вигляд, якого він набуває. Спрайт має активний образ, якій відображається на екрані.

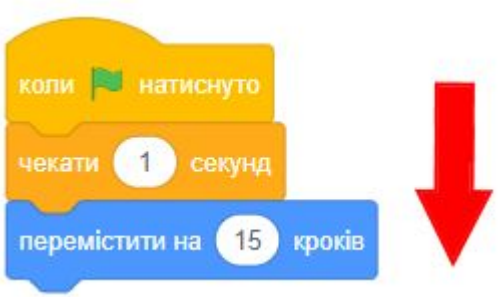
- Спрайт може мати приховані образи, які можна активувати за потребою. Якщо активний образ змінюється, спрайт все одно залишається собою.



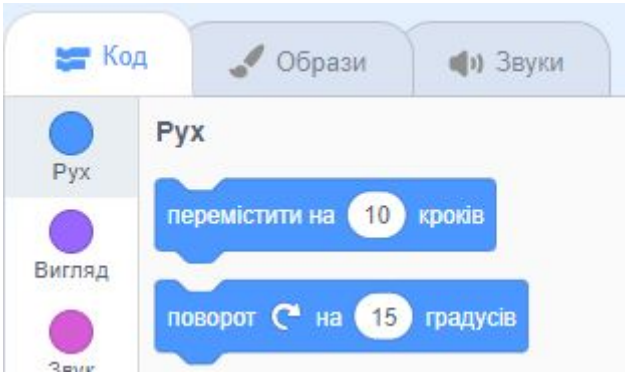
- Сцена також має активне тло, яке відображається на екрані. Сцена може мати приховані зображення, які можна активувати.
- Немає жодних обмежень щодо образів для спрайту та їх перетворень. За вашим бажанням людина може легко перетворитися на яблуко або казкову істоту.
- **АНІМАЦІЯ** – це ілюзія руху спрайта та швидкої зміни його образів. Анімація полягає у поєднанні декількох зображень з невеликими відмінностями. Під час анімації образи спрайта швидко змінюють один одного, створюючи ефект мультиплікації.

СКРИПТИ ТА БЛОКИ

- **СКРИПТ** у перекладі з англійської мови означає сценарій, у якому пишуть слова та дії для акторів.
- Скрипт у програмуванні – це послідовність команд та інструкцій, за допомогою яких ви можете керувати грою, спрайтами та сценою. Комп’ютер виконує всі команди зверху вниз послідовно, одну за одною.



- Уявіть, що всі персонажі вашої гри, - актори, які грають певні ролі, а ви - режисер та сценарист, якій створює ці ролі, сцену, слова та дії.
- У Scratch інструкції та команди називають **БЛОКАМИ**. Блоки бувають дуже різними, кожен з них має свої властивості, поведінку та належить до певної категорії.
- Всі блоки забарвлені в різні кольори. Колір блоку точно відповідає категорії, до якої він належить. Наприклад, команди, які рухають спрайт, забарвлені в синій колір. Їх можна знайти в категорії **РУХ**. Щоб побачити блоки іншої категорії, треба натиснути на неї.

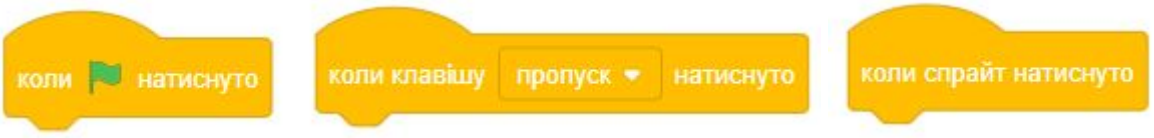


- Скрипт починається зі стартової події, виконується комп'ютером послідовно, та зупиняється на останній інструкції.
- Кожен спрайт та сцена можуть мати власні скрипти.
- До кожного спрайту можна додавати багато скриптів, вони будуть виконуватися одночасно. Наприклад, ці два скрипти працюють одночасно. Перший - пересуває спрайт на випадкову позицію. Другий - змінює образи для анімації руху спрайта.

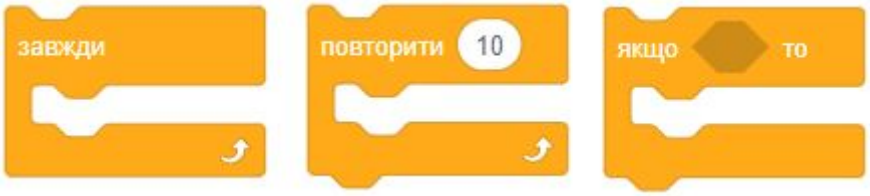
- Коли виконується скрипт, він підсвічується.

ЯКІ БУВАЮТЬ БЛОКИ?

- **СТАРТОВІ БЛОКИ** розташовані на початку скрипта, свідчать про початок його виконання. Зверху над ними не можна розміщувати інші блоки. Є декілька різних стартових блоків, які спрацьовують за певних умов.

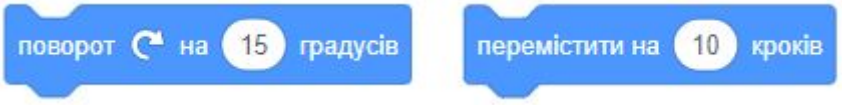


- **БЛОКИ КЕРУВАННЯ** вказують, коли та скільки разів потрібно виконати якусь певну дію. Ці блоки обгортають інші блоки та мають С-подібну форму.



- **ЛОГІЧНІ БЛОКИ** мають шестикутну форму. При виконанні цих блоків повертається тільки два значення: ІСТИНА (true) або БРЕХНЯ (false).

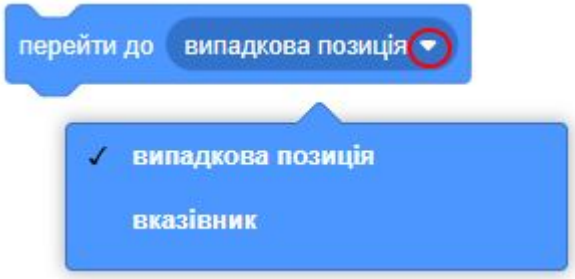
- **КОМАНДНІ БЛОКИ** використовують частіше за все. Ці блоки повідомляють компьютеру, що і як потрібно робити. Зверху та знизу до таких блоків можна під’єднувати інші блоки.



- **ЗАВЕРШАЛЬНІ БЛОКИ** розміщуються в кінці скрипта та свідчать про завершення його дії. Під цими блоками не можна розташовувати інші блоки.

- **БЛОКИ-РЕПОРТЕРИ** мають закруглені краї. Ці блоки можуть містити значення різних типів (числа, рядки).

- Деякі блоки містять **МАЛЕНЬКИЙ ТРИКУТНИК**. Коли натиснути на цей трикутник, то з’явиться меню, що випадає.
За допомогою цього меню можна обрати будь-яке доступне в ньому значення та змінити поведінку блока.



- Блоки можна з'єднувати (вставляти один в другий), якщо їхня форма дозволяє це.

КАТЕГОРІЇ БЛОКІВ

Рух	**РУХ** містить блоки, що керують рухами спрайтів.
Вигляд	**ВИГЛЯД** містить блоки, що змінюють зовнішній вигляд спрайтів та сцени.
Звук	**ЗВУК** містить блоки, за допомогою яких можна додати звуки та музику до вашої гри.

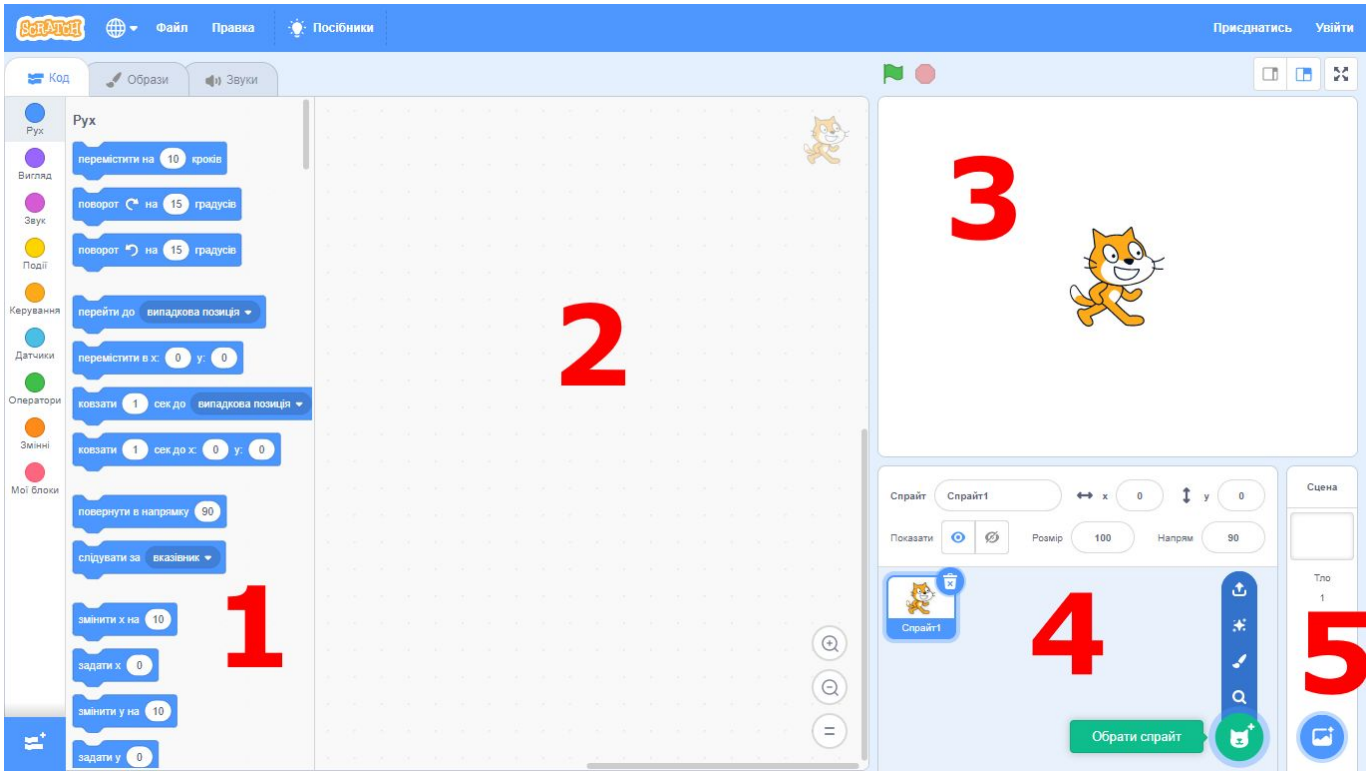
	**ПОДІЇ** містить блоки, що представляють події або ситуації.
	**КЕРУВАННЯ** містить блоки, які пояснюють грі, яким чином і коли активувати скрипти.
	**ДАТЧИКИ** містить блоки, щоб стежити, чи торкаються спрайти, чи натиснута клавіша або мишка.
	**ОПЕРАТОРИ** містить блоки для обчислення та порівняння чисел між собою.
	**ЗМІННІ** містить блоки для створення змінних і списків, та для роботи з ними.
	**МОЇ БЛОКИ** дає можливість створювати свої власні блоки, якими можна потім користуватися в скриптах.
	**ОЛІВЕЦЬ** містить блоки для малювання на сцені та керування олівцем. Щоб ця категорія з'явилася, треба додати розширення.
	**МУЗИКА** містить блоки для керування музичними інструментами та барабанами. Щоб ця категорія з'явилася, треба додати розширення.

ОБЛАСТІ ТА ІНСТРУМЕНТИ ПРОГРАМИ

Вікно програми розділено на п’ять областей.

1	**ПАЛІТРА БЛОКІВ** містить усі можливі блоки, розподілені по різних категоріях.
2	**ПОЛЕ СКРИПТІВ** дає можливість командувати об’єктами і тлом вашої гри. Сюди треба перетягувати блоки з палітри блоків.
3	**СЦЕНА ГРИ** - місце, де оживають ваші ігри та інші проекти.
4	**СПИСОК СПРАЙТІВ** містить усі персонажі та об'єкти вашої гри.

5	**ОБЛАСТЬ НАЛАШТУВАННЯ СЦЕНИ** дає можливість додавати та редагувати тло проекту.



	Почати та зупинити гру.
	Увімкнути повноекранний режим, в ньому ви можете тільки грати. Щоб вийти з цього режиму та редагувати проект, натисніть цю кнопку ще раз.
 добавити спрайт  добавити тло	Додати спрайт або тло. Показує додаткове меню з командами:  Завантажити спрайт або тло з комп'ютера. Підтримуються картинки в популярних графічних форматах (svg, png, jpg, gif та інші).  Сюрприз. Обирає спрайт або тло з бібліотеки випадковим чином.  Додає порожній спрайт або порожнє тло, щоб була можливість намалювати їх самостійно.  Вибрати спрайт або тло з вбудованої бібліотеки.

	Натисніть сюди, щоб обрати потрібну мову для інтерфейсу програми.
	Додати розширення. Дає можливість обрати потрібне розширення із понад 10 додаткових можливостей, та додати його до проекту. Увага: Команди олівця та музики в новому Scratch тепер перенесені до розширень.
	За допомогою цієї вкладки можна зберегти проект, створити новий проект та завантажити проект з вашого комп'ютера.
	За допомогою цієї вкладки можна скасувати останню дію та увімкнути турбо-режим.

ПАНЕЛІ СПРАЙТІВ І СЦЕНИ

- Кожен спрайт має три панелі.

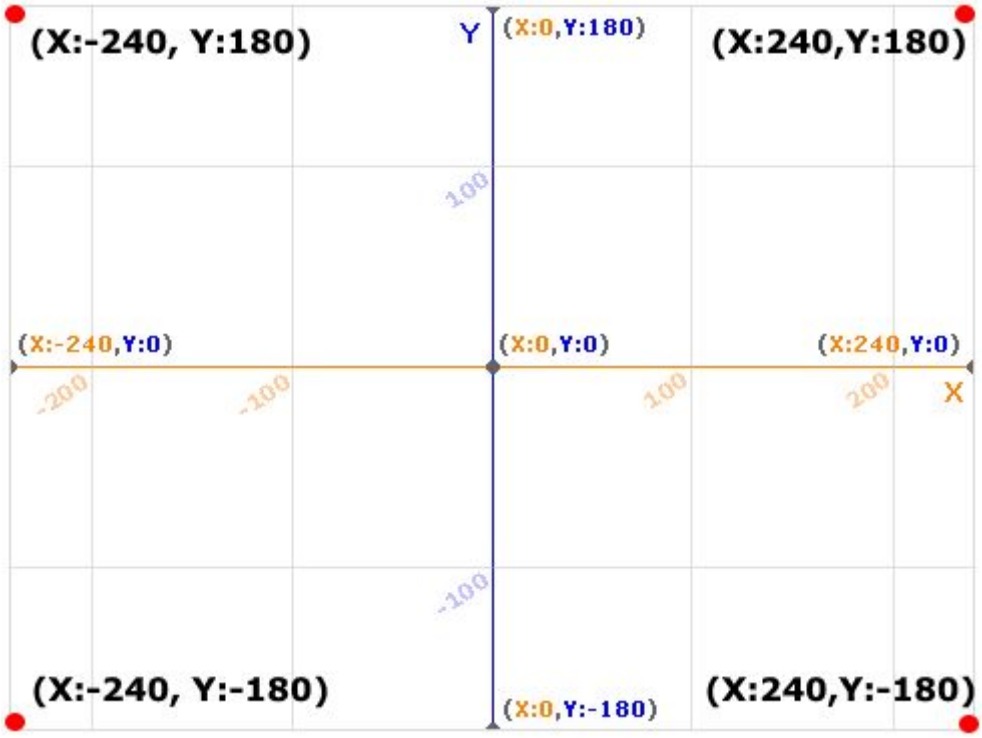
- **КОД** переключає вікно програми на палітру блоків та область для створення скриптів.
- **ОБРАЗИ** містить усі подоби, в яких може з’являтися спрайт. Тут розташований редактор зображень, в якому ви можете редагувати зовнішній вигляд спрайтів.
- **ЗВУКИ** дає можливість додати звук із вбудованої бібліотеки, завантажити його з вашого комп’ютера або зробити власний аудіо запис з мікрофону.
- Сцена також має три панелі.

- **КОД** та **ЗВУКИ** працюють таким же чином, як і для спрайтів.
- **ТЛО** відрізняється. Відкривши цю панель, ви побачите всі фонові зображення, які були додані до проекту. Ви можете додати інші зображення, редагувати або видалити їх.

КООРДИНАТИ ТА НАПРЯМОК

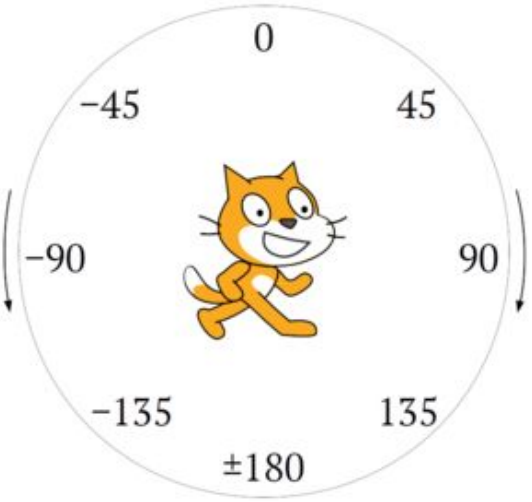
- Кожна точка на екрані компьютера позначається числами **X** та **Y**. Вони мають назву **КООРДИНАТИ**.
- Число **X** задає положення спрайта на горизонтальній лінії. Якщо X має значення від -240 до 240, спрайт розташований на сцені. При інших значеннях він знаходиться поза сценою.
- Число **Y** задає положення спрайта на вертикальній лінії. Якщо координата **Y** має значення від -180 до 180, спрайт розташований на сцені. При інших значеннях він знаходиться поза сценою.
- Центр екрана має координати: **X**=0, **Y**=0.
- Поточні координати спрайта відображені в інформаційному вікні, яке розташоване під сценою над списком спрайтів.

- Від'ємні координати (позначені з мінусом) знаходяться внизу або ліворуч від центру екрану. Позитивні - зверху або праворуч.

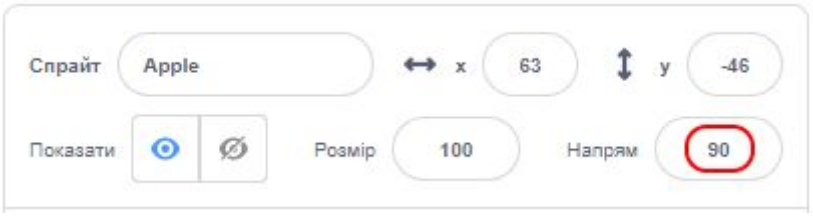


- Напрямок спрайта вимірюють у градусах (від 0 до 360).
- Якщо спрайт повернутий вгору, це напрямок 0.
- Якщо спрайт повернутий праворуч, його напрямок дорівнює 90.
- Якщо спрайт повернутий вниз, його напрямок дорівнює 180.

- Отже: **90** - праворуч, **-90** - ліворуч, **0** - вгору, **180** - вниз.

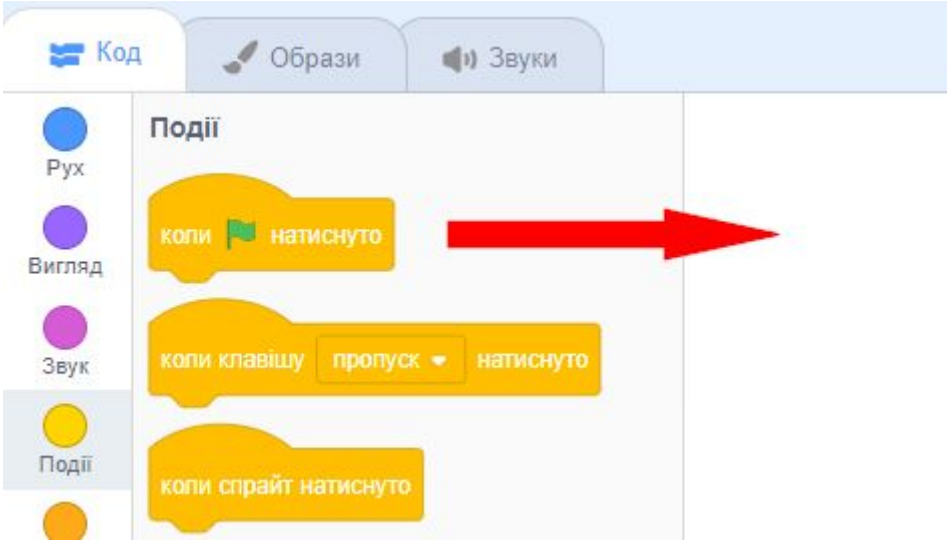


- Поточний напрямок спрайту можна подивитись, а також змінити в інформаційному вікні спрайту. Натисніть мишкою, щоб обрати напрям і стиль обертання у додатковому вікні.



МИ ПОЧИНАЄМО

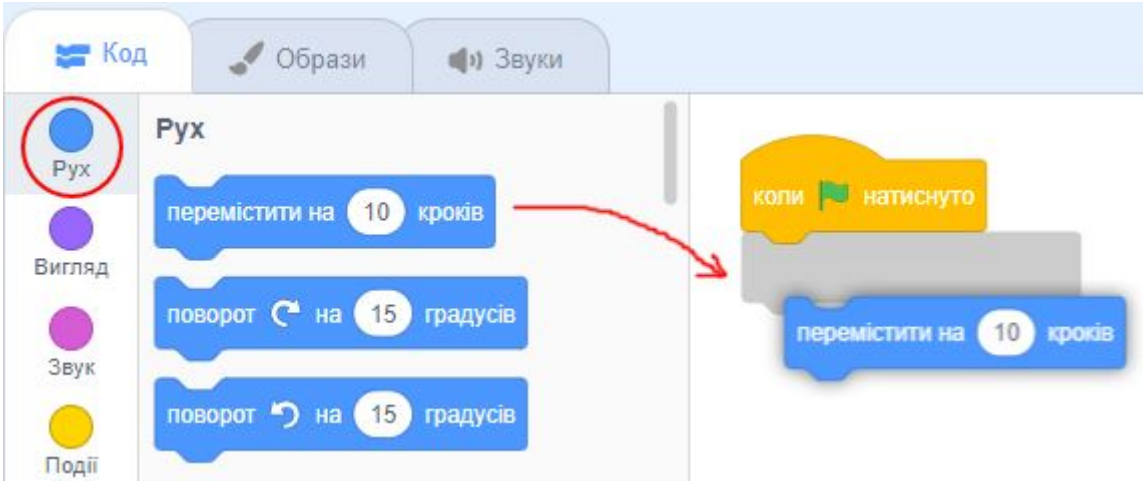
- Виберіть категорію команд ПОДІЇ і перетягніть за допомогою мишки блок із зеленим прапором в область скриптів.



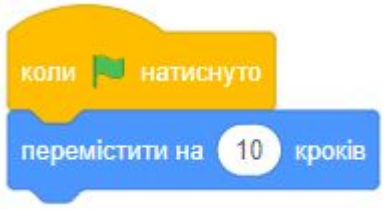
- Як зробити перетягування? В комп'ютерних іграх ви часто перетягували об’єкти за допомогою мишки або пальця. Перетягування складається з трьох етапів.

1. Об’єкт натиснути лівою кнопкою мишки.
2. Об’єкт рухається разом з вказівником, поки натиснута кнопка мишки.
3. Відпускаємо кнопку мишки. Після цього об’єкт припиняє слідувати за вказівником.

- Виберіть категорію блоків РУХ і перетягніть за допомогою мишки блок ПЕРЕМІСТИТИ в область скриптів, приєднавши його знизу до попереднього блоку.

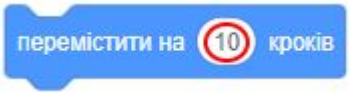


- В результаті ми створили найпростіший скрипт з двох блоків.



- Для запуску програми натисніть на кнопку із зеленим прапором.

- При натисканні на зелений прапор, кіт буде пересуватись вперед на 10 кроків. Щоб змінити число кроків, натисніть мишкою всередині білого кола і введіть інше число.



- Вітаю! Ви написали першу програму.

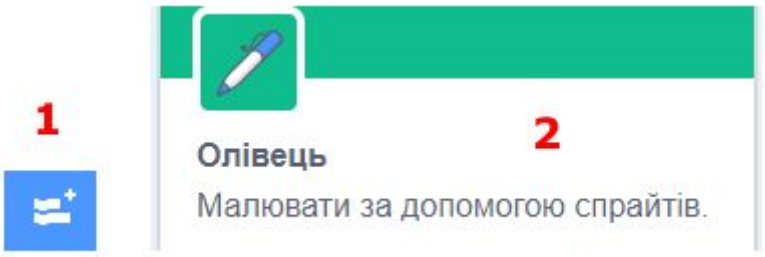
УРОК 1. МАЛЮВАННЯ ФІГУР

Відео: https://youtu.be/1tAm5gm1zek

На цьому уроці ми навчимося малювати різні геометричні фігури (квадрат, трикутник, коло та інші) за допомогою блоків олівця та пересування спрайта.

КРОК 1. МАЛЮВАННЯ КВАДРАТА

- У новому Scratch категорія блоків управління олівцем захована в розширеннях. Для того щоб її додати, необхідно натиснути кнопку додавання розширення та вибрати розширення ОЛІВЕЦЬ.



- Нам треба намалювати квадрат за допомогою блоків руху та олівця. Нам допоможе скрипт, показаний нижче.

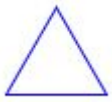
- Перші шість блоків однакові для всіх проектів цього уроку і виконують початкові налаштування (очистити сцену від слідів малювання, перемістити в центр екрану, повернути в потрібному напрямку і опустити олівець).

- **Завдання.** Перепишіть скрипт, щоб він спрацьовував після натискання цифри 1 на клавіатурі.

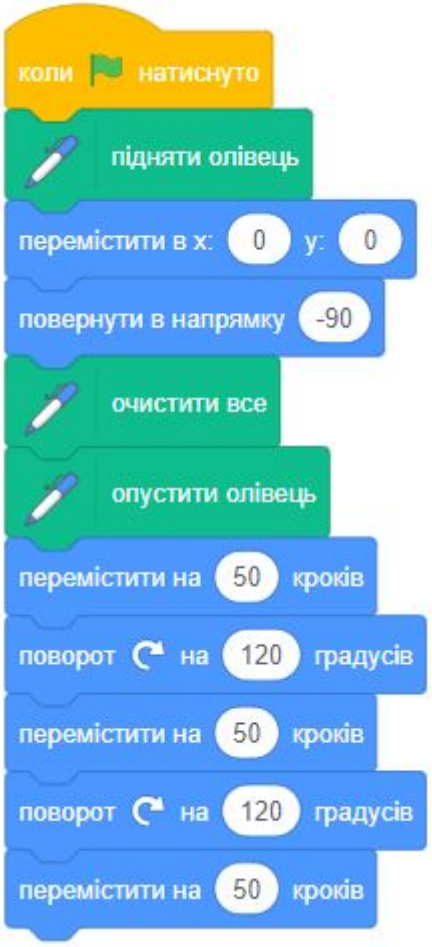
- **Завдання.** Намалюйте квадрат, який буде в два рази більший. Запуск скрипта після натискання цифри 2.
- **Завдання.** Намалюйте прямокутник, в якому ширина буде більше висоти. Запуск скрипта після натискання цифри 3.
- **Завдання.** Намалюйте прямокутник, в якому висота буде більше ширини. Запуск скрипта після натискання цифри 4.
- **Завдання.** Перепишіть скрипт за допомогою блока ПОВТОРИТИ. У вас має бути менше блоків ПЕРЕМІСТИТИ та ПОВОРОТ за рахунок користування блоком ПОВТОРИТИ. Запуск скрипта після натискання цифри 5.

КРОК 2. МАЛЮВАННЯ ТРИКУТНИКА

- Давайте намалюємо рівносторонній трикутник за допомогою блоків руху та олівця.



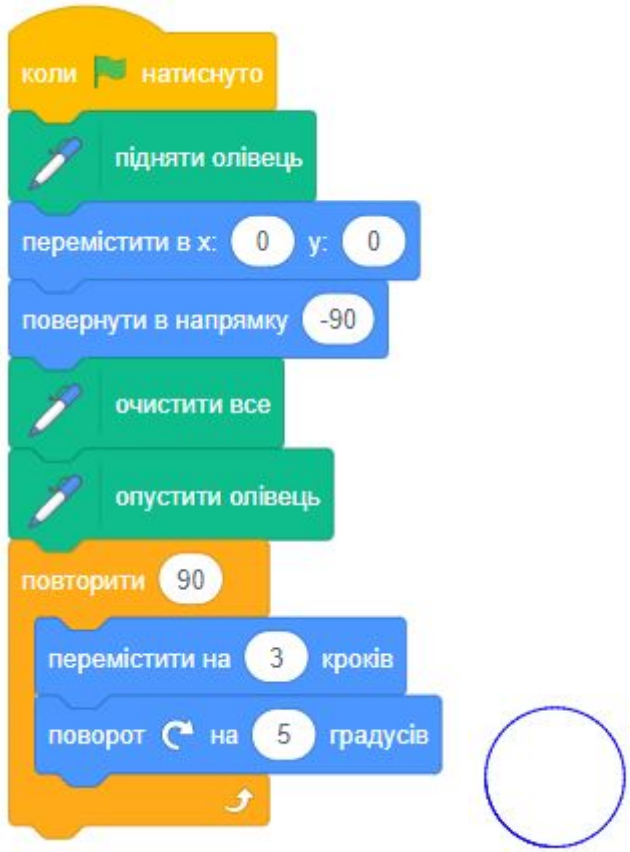
- Скрипт малювання схожий на попередній, тільки повертаємо на інше число градусів і малюємо менше сторін (3 замість 4).



- **Завдання.** Перепишіть скрипт, щоб він спрацьовував коли натиснута цифра 6 (шість) на клавіатурі.
- **Завдання.** Перепишіть скрипт за допомогою блока ПОВТОРИТИ. У вас має бути менше блоків ПЕРЕМІСТИТИ та ПОВОРОТ за рахунок користування блоком ПОВТОРИТИ. Запуск скрипта після натискання цифри 7.
- **Завдання.** Намалюйте перевернутий трикутник. Запуск скрипта після натискання цифри 8.

КРОК 3. МАЛЮВАННЯ КОЛА

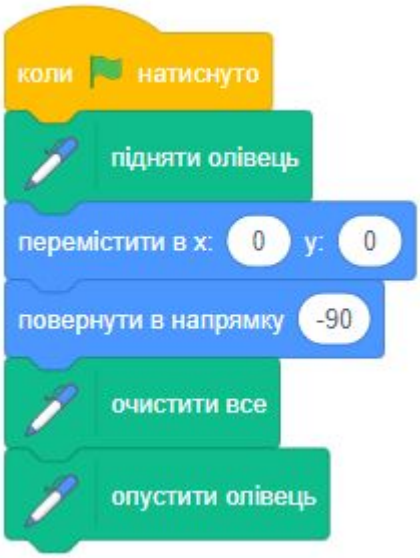
- Коло малюється трохи складніше, за допомогою блоку ПОВТОРИТИ. Нам треба повторити команди ПЕРЕМІСТИТИ і ПОВОРОТ кілька разів.



- **Завдання.** Перепишіть скрипт, щоб він спрацьовував, коли натиснута клавіша 9 (дев'ять) на клавіатурі.
- **Завдання.** Спробуйте змінити кількість повторень, число кроків для переміщення спрайту або число градусів для повороту, подивіться, що вийшло. Запуск після натискання цифри 0.

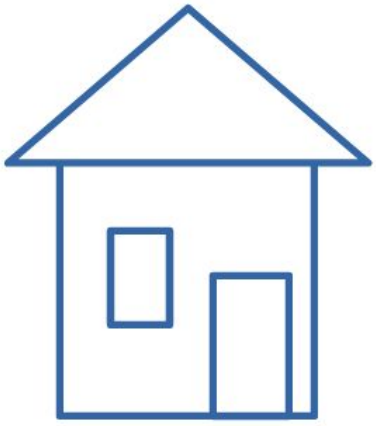
КРОК 4. МАЛЮВАННЯ СКЛАДНИХ ФІГУР

- Давайте намалюємо більш складні фігури. Треба за допомогою команд руху і олівця намалювати картинки, які показано нижче. Кожну фігуру робити в окремому файлі.
- Перш за все очистіть область скриптів, створивши новий проект.
- Додайте ці блоки для встановлення початкових значень перед малюванням фігур:

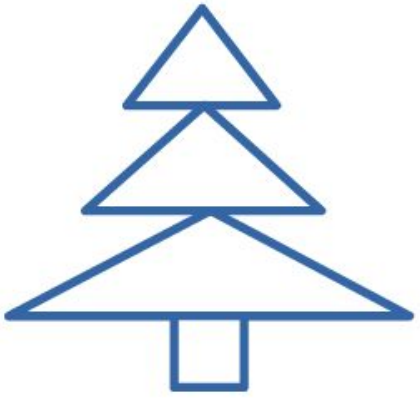


- Тепер самостійно виконайте завдання з малювання.

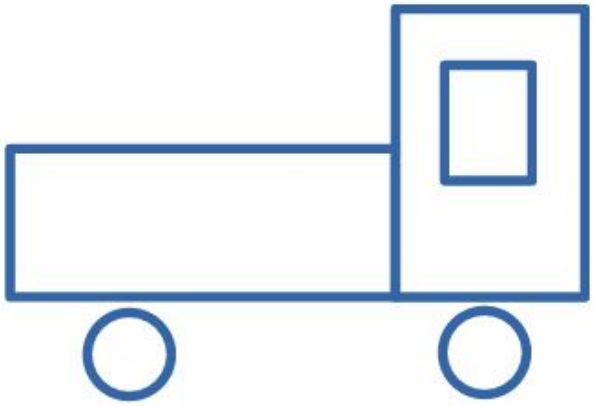
МАЛЮЄМО БУДИНОЧОК



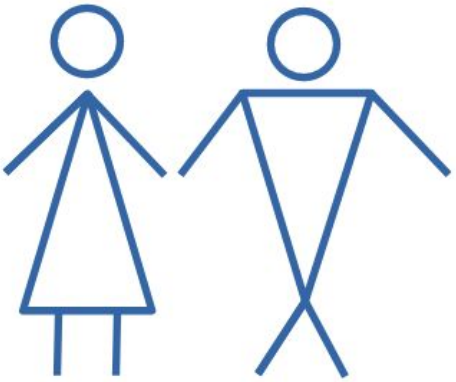
МАЛЮЄМО ЯЛИНКУ



МАЛЮЄМО МАШИНКУ



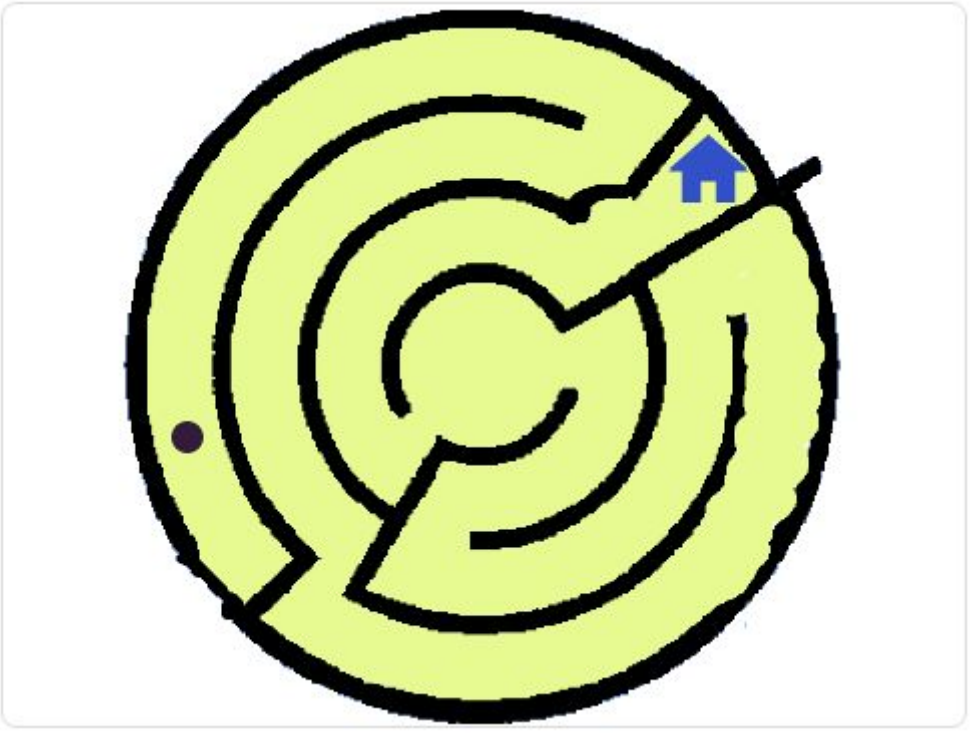
МАЛЮЄМО ДІТЕЙ



УРОК 2. ЛАБІРИНТ

Відео: https://youtu.be/D0Ud6egbAqU

На цьому уроці ми створимо гру лабіринт. Гравцеві треба пройти весь шлях по лабіринту, не торкаючись стін, і знайти вихід.



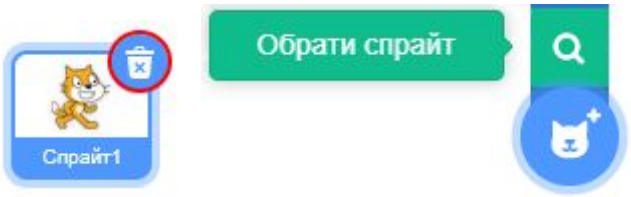
На цьому уроці ми будемо використовувати:

1. Розгалуження.
2. Цикли.

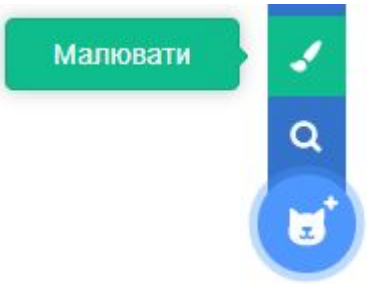
КРОК 1. СПРАЙТИ ТА СЦЕНА

- Нам потрібно створити три спрайта:
  - Labirint - лабіринт.
  - Exit - вихід з лабіринту.
  - Ball - шар, який буде рухатися по лабіринту.

- Шар керується за допомогою клавіатури та прямує лабіринтом на вихід. Якщо він торкається стіни лабіринту, то це буде поразка, і все починається знову. Якщо спрайт торкається виходу, це буде перемога. В обох випадках спрайт переміщується в початок лабіринту. Також спрайт доповідає переміг він чи ні.
- Створюємо новий проект, видаляємо звідти кота. Далі натисніть ОБРАТИ СПРАЙТ, щоб відкрити список усіх спрайтів із вбудованої бібліотеки.

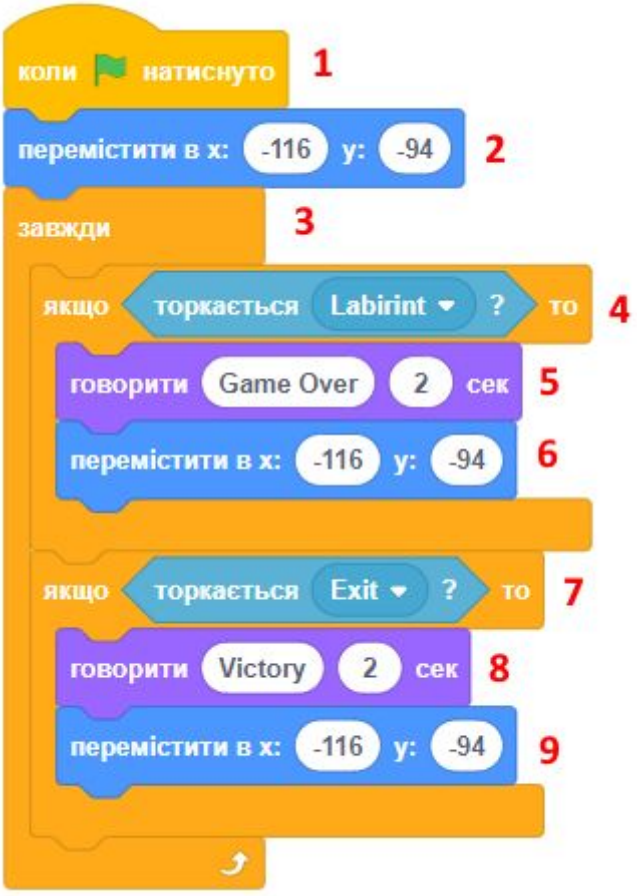


- Знайдіть спрайт, який вам подобається, і виберіть його.
- Таким же чином додайте спрайт, який буде метою нашої подорожі по лабіринту.
- Ще нам потрібно намалювати спрайт лабіринта.
- Натисніть кнопку олівця, щоб додати новий порожній спрайт. Потім за допомогою інструментів малювання створіть лабіринт.



КРОК 2. СКРИПТ ПЕРЕВІРКИ ТОРКАННЯ

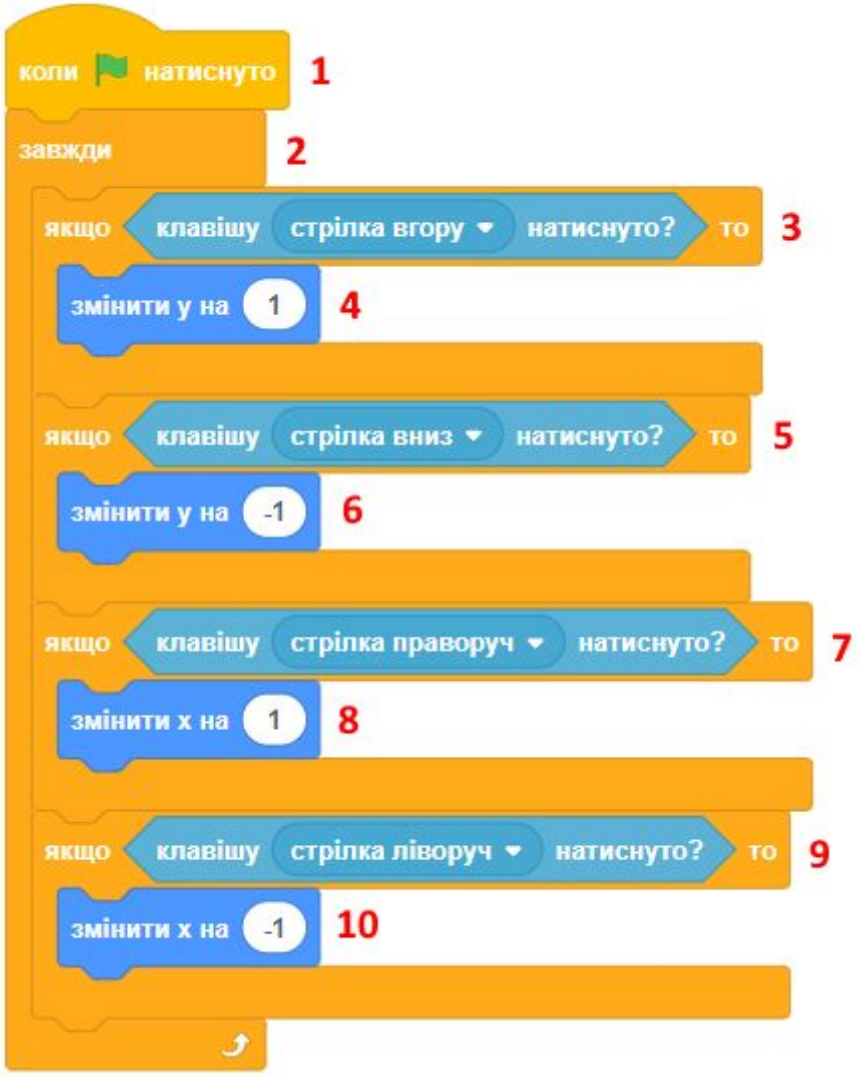
- Цей скрипт дозволяє спрайту перевірити, чи він торкається стін лабіринту (поразка) або спрайта виходу з лабіринта (перемога).



1	Скрипт починає свою роботу на початку гри. Початковий блок знаходиться в категорії ПОДІЇ.
2	Точка (-116, -94) в блоках ПЕРЕМІСТИТИ (блоки 2, 6, 9) вибрана для початкової позиції спрайту. В вашому випадку вона буде інша, тому підберіть та задайте її координати.
3	Впродовж гри треба робити перевірки торкання з іншими спрайтами.
4	Якщо спрайт торкається стін лабіринту, треба повідомити гравця та перейти в початок лабіринту. Блок ЯКЩО знаходиться в категорії КЕРУВАННЯ, а блок ТОРКАЄТЬСЯ - в категорії ДАТЧИКИ.
5	Блок ГОВОРИТИ знаходиться в категорії ВИГЛЯД.
7	Якщо спрайт торкається виходу, треба обробити це.

КРОК 3. СКРИПТ РЕАГУВАННЯ НА КЛАВІШІ

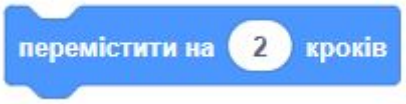
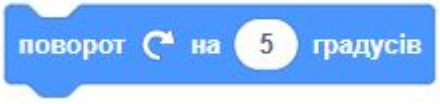
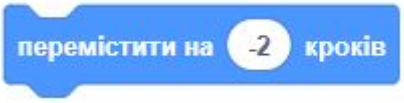
- Цей скрипт дозволяє спрайту реагувати на натискання клавіш клавіатури та ходити по лабіринту. Додавши його, ви можете перевірити, як спрайт переміщується, за допомогою клавіш ВНИЗ, ВГОРУ, ПРАВОРУЧ, ЛІВОРУЧ.



- Перевірте, що ви можете пройти лабіринт.

КРОК 4. ЗАВДАННЯ

- **Завдання.** Додайте керування спрайтом за допомогою клавіш W (вгору), S (вниз), A (ліворуч), D (праворуч).
- **Завдання.** Зробіть так, щоб спрайт рухався в два рази швидше.
- **Завдання.** Зробіть так, щоб клавіші 1, 2, 3, 4 рухали спрайт іншим чином, як показано в таблиці:

КЛАВІША	БЛОК
1	
2	
3	
4	

- **Завдання.** Зробіть так, щоб клавіша  ПРОПУСК переносила спрайт у початок лабірінту.
- **Завдання.** Зробіть так, щоб клавіша  ENTER переносила спрайт майже до виходу з лабіринту.
- **Завдання.** Додайте інші спрайти до гри. Коли спрайт, яким ви керуєте, торкається цих нових спрайтів, він повинен щось казати. Також треба ховати ці спрайти, немов ваш герой їх збирає. На початку гри показуйте усі спрайти гри на сцені за допомогою блока ПОКАЗАТИ.
- **Завдання.** Додайте спрайт ключа та розмістить його у лабіринті. Якщо ваш основний спрайт торкається ключа, той зникає. Змініть код торкання виходу. Вихід відчиняється, тільки якщо ви підібрали ключ. Якщо ключа у вас немає, ваш спрайт скаже, що вихід зачинено, коли його торкнеться.
- **Завдання.** Намалюйте другий рівень лабіринту. Переключайте образ лабіринту на другий рівень, коли ваш основний спрайт торкається виходу.

УРОК 3. ПОЛІТ КАЖАНА

Відео: https://youtu.be/CraZTa4r72c

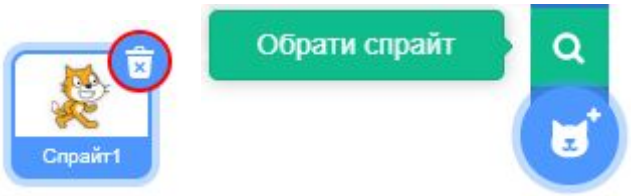
На цьому уроці ми створимо гру, в якій кажан буде літати і збирати різні фрукти та інші продукти.

На цьому уроці ми будемо використовувати:

1. Клони.
2. Цикли.

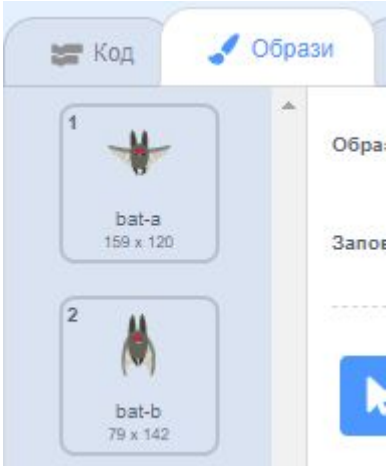
КРОК 1. СПРАЙТИ ТА СЦЕНА

- Створюємо новий проект, видаляємо звідти кота. Далі натисніть ОБРАТИ СПРАЙТ, щоб відкрити список усіх спрайтів із вбудованої бібліотеки.



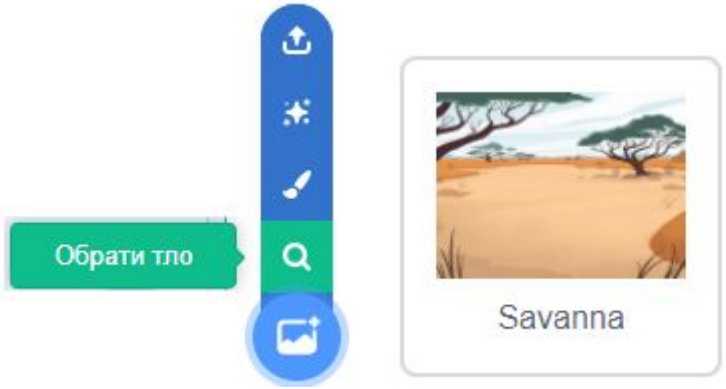
- Знайдіть спрайт кажана і виберіть його за допомогою мишки.

- Виберіть сторінку **ОБРАЗИ**. Як ви бачите, у спрайта є декілька образів. Давайте зробимо так, щоб кажан махав крилами. Для цього нам потрібно активувати перші два образи по черзі.

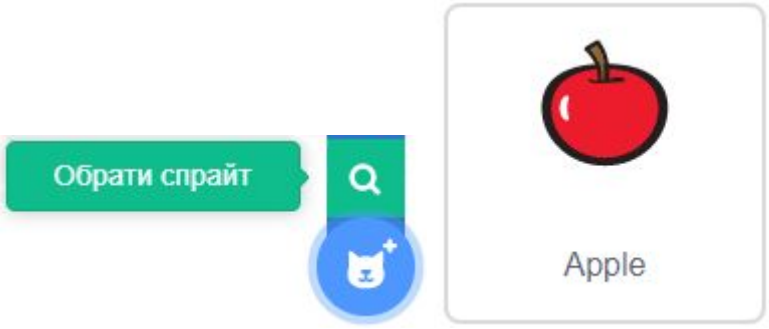


- Видаліть інші образи, вони нам не потрібні. Для цього натисніть на корзину з хрестиком поряд із образами, які не потрібні.

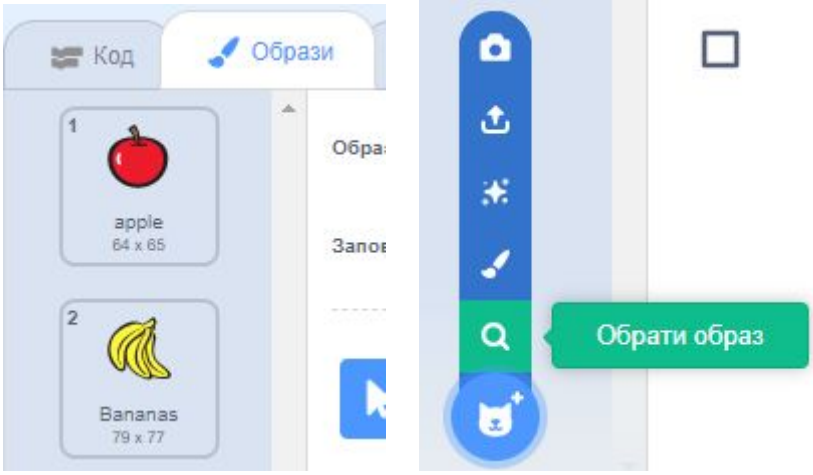
- Натисніть на кнопку завантаження нового тла. У вікні, що з'явилося, виберіть тло, яке вам подобається.



- Додайте новий спрайт **яблуко** з бібліотеки спрайтів.

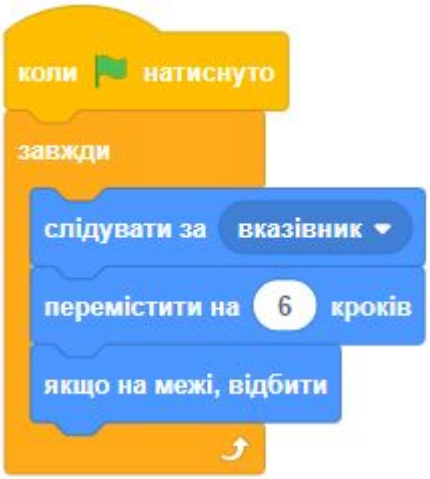


- Тепер для спрайта **яблука** додайте три (або більше) нових образа фруктів та інших продуктів.

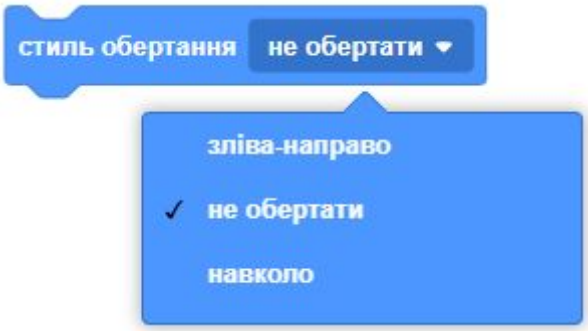


КРОК 2. СКРИПТИ КАЖАНА

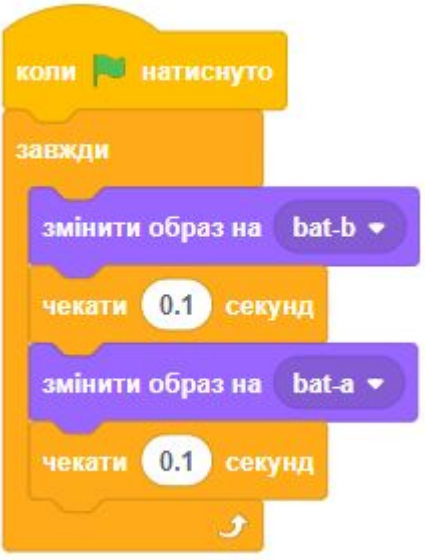
- Давайте зробимо так, щоб кажан рухався за покажчиком миши. Нам треба щоразу повертатися до покажчика мишки та переміщатися в цьому напрямку на декілька кроків.
- Додайте скрипт до **кажана**:



- Важливо налаштувати стиль обертання спрайта, щоб кажан обертався, як вам потрібно. Це можно зробити за допомогою блоку СТИЛЬ ОБЕРТАННЯ із категорії РУХ. Додайте цей блок до початку вашого скрипта.



- Давайте зробимо, щоб кажан махав крилами. Анімація робиться за допомогою швидкої зміни образу кажана. Додайте новий скрипт до **кажана**:



КРОК 3. СКРИПТИ ЯБЛУКА

- Додайте новий скрипт у **спрайт яблуко**.

1	Спочатку треба заховати спрайт яблука, нам потрібні тільки його клони, а не він сам.
2	Нам треба постійно створювати копію (клон) спрайта на сцені, тому створюємо клон всередині цикла ЗАВЖДИ.
3	Клони створюйте у кількості по 10 штук.
4	Блок створення нового клону.
5	Чекати 2 секунди між створеннями клонів, щоб вони не створювалися занадто швидко.
6	Фрукти створюються у кількості по 10 штук, а потім ми знову чекаємо 2 секунди.

- Клони спрайтів з'являються, тільки коли гра виконується (після натискання зеленого прапора).
- Щоб знищити все створені клони, треба зупинити гру, натиснувши на червоний значок СТОП поруч із зеленим прапором.

- Додайте скрипт в **спрайт яблуко**.

1	Скрипт буде виконуватися тільки для клонів.
2	Змінюємо образ клону, вибираючи випадковий (в образах можуть бути яблуко, банан, клубника, молоко, донат або інші продукти). Скільки ви додали образів, таке число треба й поставити замість цифри 5.
3	Переміщуємо спрайт у випадкове місце на екрані.
4	Якщо ми вийшли за межі екрану, змінимо координати спрайта.
5	Показуємо клон спрайта на сцені.
6	Протягом гри ми чекаємо, чи торкається кажан яблука.
7	Якщо кажан торкається яблука, знищуємо клон (для гравця фрукт зникає з екрану).

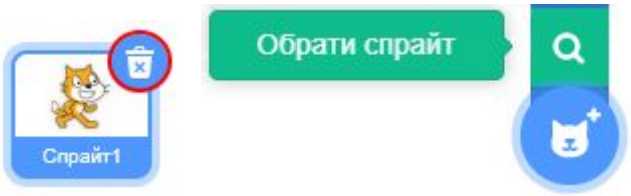
УРОК 4. РОК ГРУПА

Відео: https://youtu.be/HFy0om-shW0

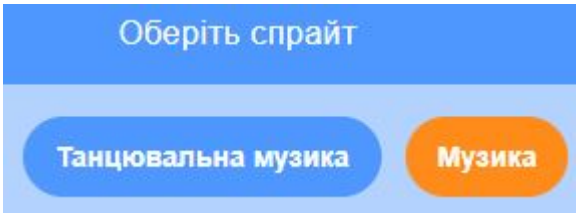
На цьому уроці ми будемо програмувати звуки і музику. На сцені виступить співачка, гратимуть барабани та інші музичні інструменти.

КРОК 1: СПРАЙТИ ТА СЦЕНА

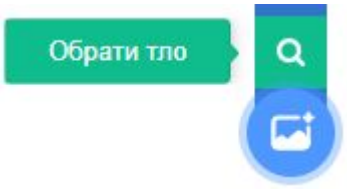
- Створюємо новий проект, видаляємо звідти кота. Далі натисніть ОБРАТИ СПРАЙТ, щоб відкрити список спрайтів із бібліотеки.



- Барабани знаходяться в категорії МУЗИКА, клацніть на неї, щоб фільтрувати список спрайтів та обрати тільки музичні інструменти. Натисніть на барабан, який вам подобається, щоб додати його до вашого проекту.



- Додайте на сцену тло. Для цього натисніть на кнопку ОБРАТИ ТЛО в правому нижньому кутку вікна програми.



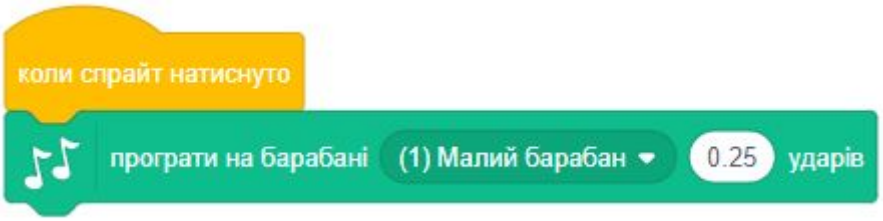
- Виберіть тло, яке вам подобається, щоб додати його до проекту.

- Додайте розширення **МУЗИКА**, натиснув кнопку  в лівому нижньому кутку вікна програми.

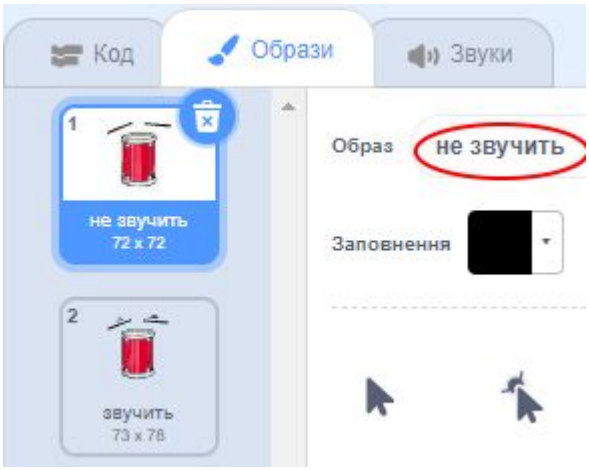
КРОК 2: СТВОРЮЄМО БАРАБАН

Давайте запрограмуємо барабан, щоб він звучав при натисканні.

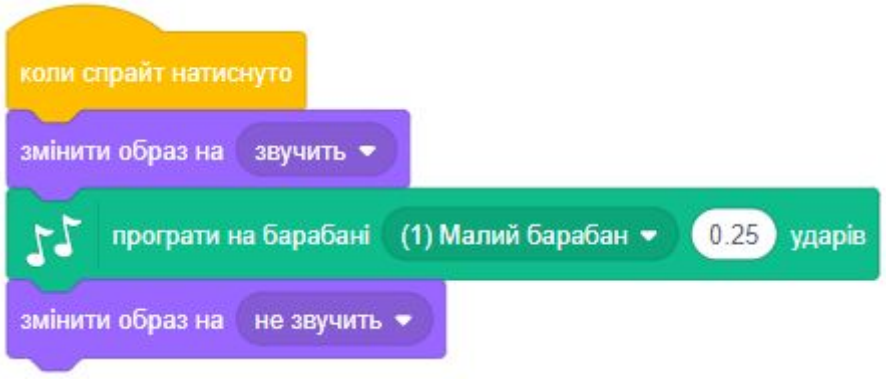
- Знайдіть блоки у вкладці КОД. Виберіть спрайт барабана і перетягніть ці два блоки в поле для скриптів праворуч. Переконайтесь, що блоки з'єднані між собою.



- Клацніть по барабану - випробуйте його!
- Ви можете змінити вигляд барабана під час натискання. Для цього потрібно створити новий образ, якщо його немає. Натисніть на вкладку ОБРАЗИ, і ви побачите картинку барабана. Натисніть правою кнопкою мишки на образ і виберіть інструмент 'дублювати', щоб створити його копію.
- Назви образів зараз не дуже зручні для використання. Перейменуйте обидва образи, перший в '**не звучить**', а другий в '**звучить**'. Назви можна ввести в текстовому полі.



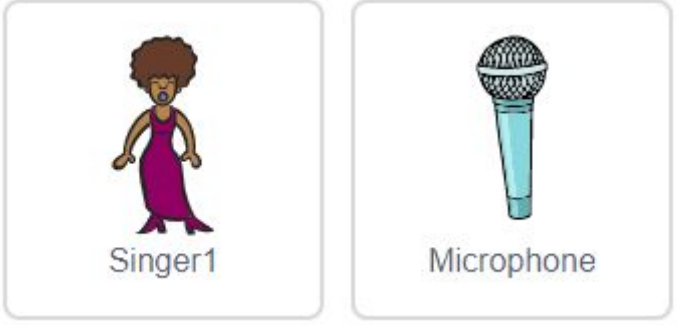
- Тепер у вас є два різних образи для барабана, і ви можете вибрати, який буде відображатися! Додайте два нових блоки до скрипта барабану, щоб переключати ці образи. Блоки для зміни образа розташовані в категорії ВИГЛЯД.



- Спробуйте свій барабан. При натисканні барабан повинен виглядати ніби по ньому вдарили!

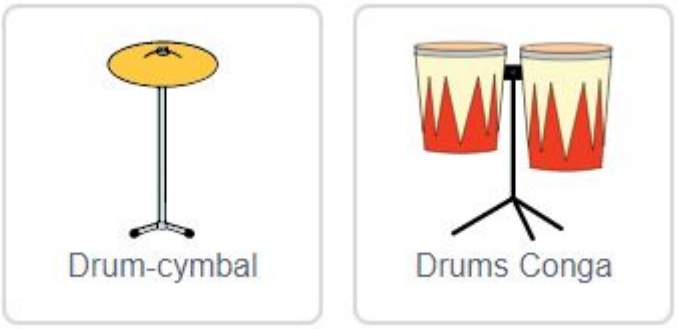
КРОК 3: СТВОРЮЄМО СПІВАКА

- Додайте ще два спрайта із бібліотеки спрайтів: соліст і мікрофон.

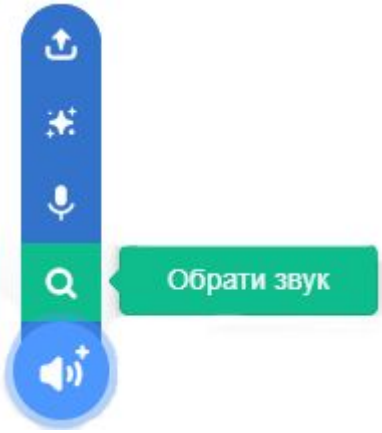


Ці спрайти знаходяться в категорії МУЗИКА.

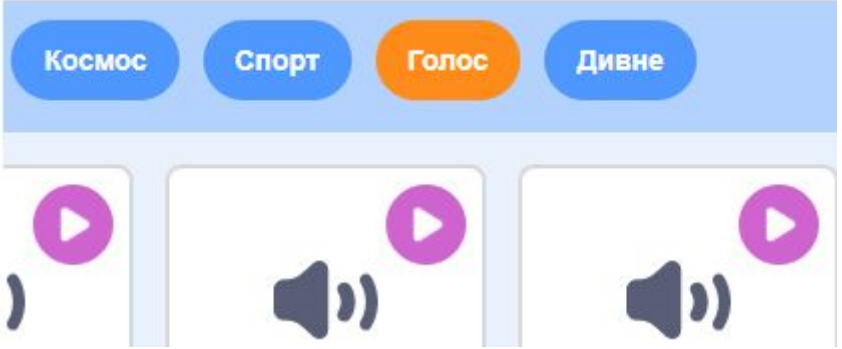
- Мікрофон треба домалювати, додавши до нього мікрофонну стійку, щоб він не висів у повітрі. Якщо не хочете малювати самостійно, ви можете скористатися частиною зображень інших спрайтів:



- Перед тим як змусити соліста заспівати, вам потрібно додати звук у ваш спрайт. Переконайтесь, що ви вибрали співака, потім натисніть на вкладку ЗВУКИ та виберіть ОБРАТИ ЗВУК:



- Виберіть категорію ГОЛОС у вікні, яке з'явилося, та знайдіть найкращий звук для соліста.

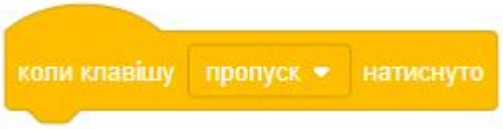


- Тепер, коли звук вибрано, ви можете додати скрипт для **соліста**. Після цього клацніть по солісту, перевірте, співає він чи ні.

КРОК 4. ЗАВДАННЯ

ЗАВДАННЯ: ПОКРАЩИМО БАРАБАН

- Спробуйте зробити так, щоб барабан теж звучав, коли натиснута клавіша ПРОПУСК. Вам для цього потрібен блок:



- Ви можете копіювати вже існуючі скрипти натисканням правої клавіші мишки на них та вибираючи інструмент ДУБЛЮВАТИ.
- Чи зможете ви замінити звук барабана, щоб він звучав по-іншому при натисканні? Блок ПРОГРАТИ НА БАРАБАНІ має маленький трикутник, клацніть на ньому та виберіть інший інструмент, який вам до вподоби.

ЗАВДАННЯ: ЗМІНИТИ КОСТЮМ СПІВАКА

- Зробить так, щоб при натисканні мишкою по солісту здавалося, що той співає. Також ви можете зробити так, щоб він підстрибував, рухався або танцював.
- Якщо вам знадобиться допомога, ви можете скористатися наведеними вище інструкціями для створення барабану.

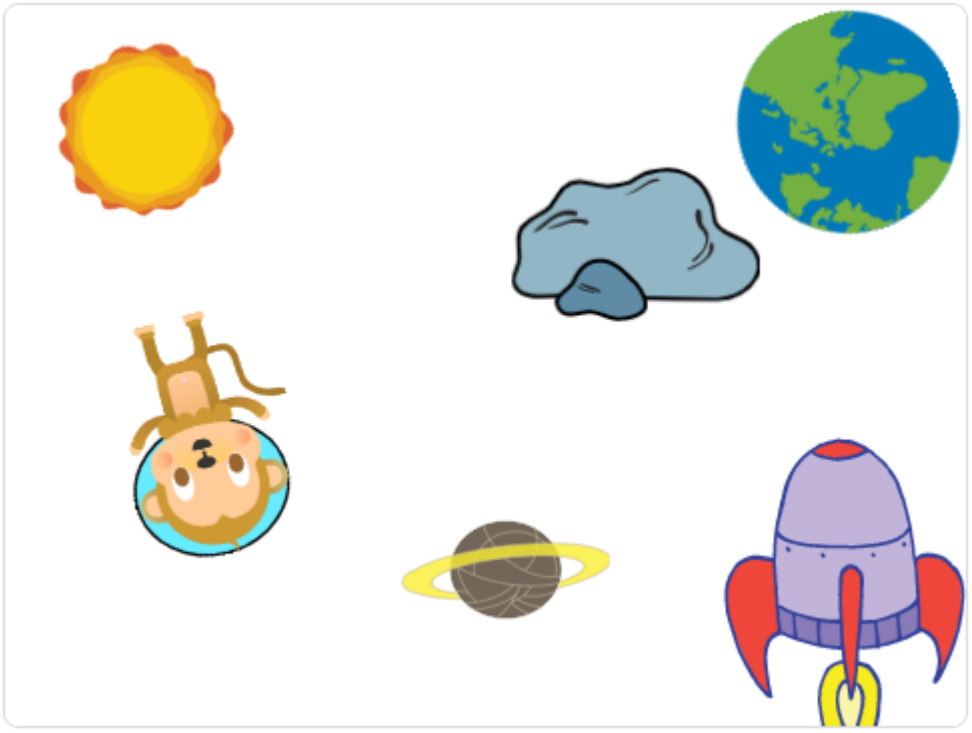
ЗАВДАННЯ: СТВОРИТИ ВАШУ ВЛАСНУ ГРУПУ

- Використовуйте отримані знання, щоб створити свою власну групу! Ви можете створити будь-які інструменти за своїм бажанням.
- Перегляньте доступні інструменти та звуки в бібліотеці, щоб краще орієнтуватися.
- Ваш інструмент не зобов'язаний бути реальним. Наприклад, ви можете зробити піаніно з пиріжків або щось інше!
- Замість використання існуючих спрайтів, ви можете намалювати свої. Якщо у вас є мікрофон - ви можете записати свої звуки.
- Додайте декілька спрайтів на сцені та у глядацькому залі, які будуть танцювати та рухатися під музику.

УРОК 5. ЗАГУБЛЕНІ В КОСМОСІ

Відео: https://youtu.be/segImxXJkWE

На цьому уроці ми створимо гру про космос. Ми будемо рятувати мавпочку-космонавта, яку випадково забули.



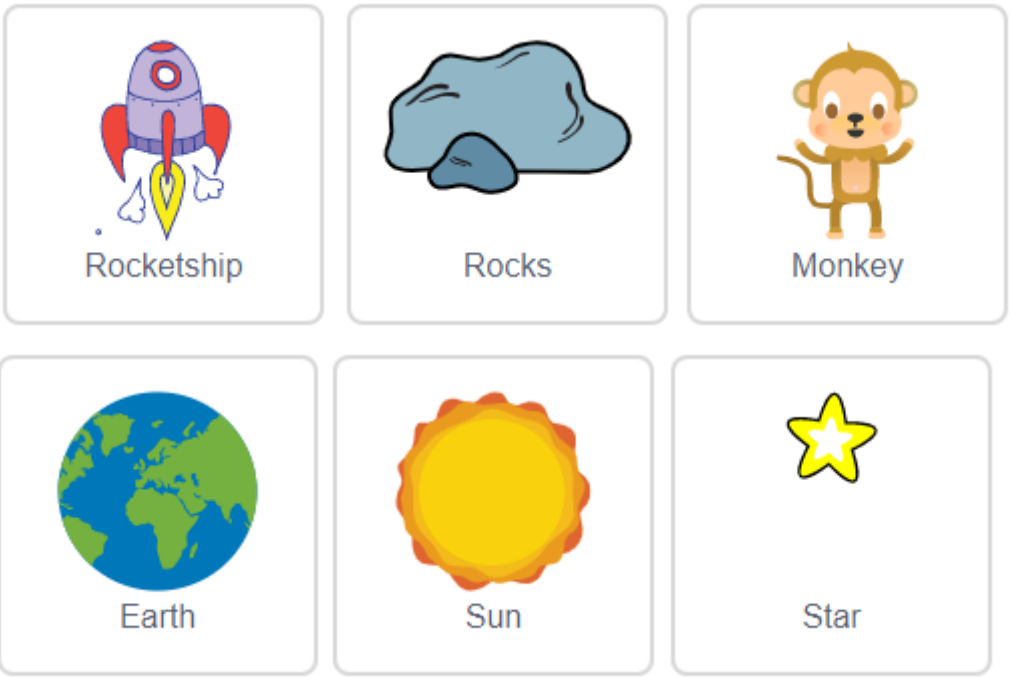
На цьому уроці ми будемо використовувати:

1. Цикли.
2. Анімацію.

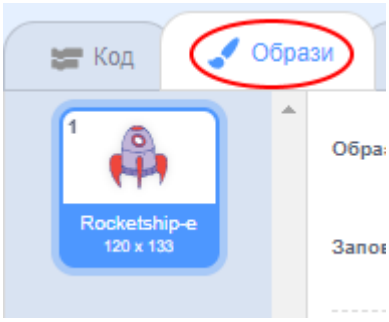
КРОК 1: АНІМАЦІЯ КОСМІЧНОГО КОРАБЛЯ

Давайте зробимо так, щоб космічний корабель летів до Землі!

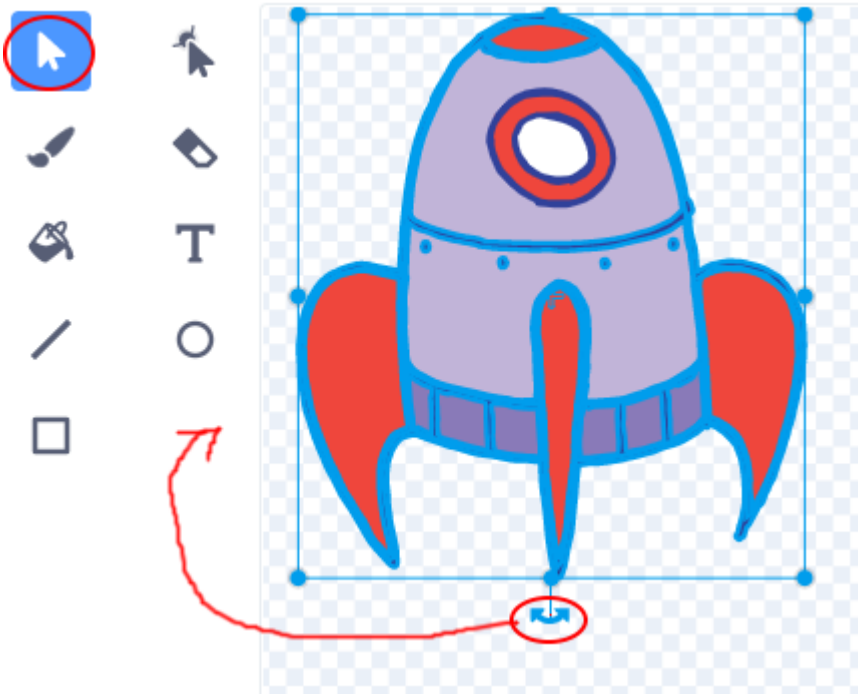
- Створюємо новий проект, видаляємо звідти кота.
- Додайте спрайти «Корабля», «Землі» та інші потрібні спрайти на сцену. Також додаємо тло «Зірки» для сцени.



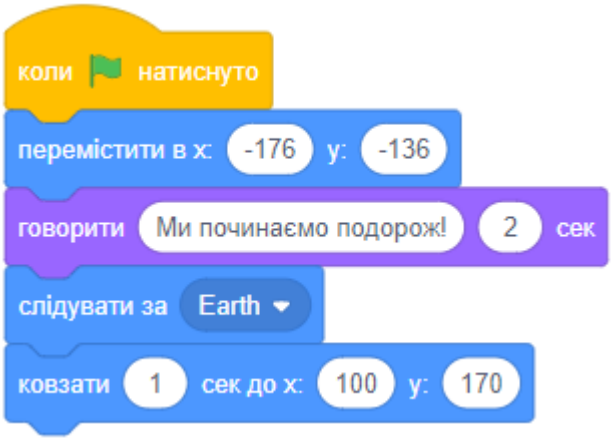
- Натискаємо на спрайт корабля у списку спрайтів потім - на вкладку «Образи».



- Використовуючи інструмент «курсор», вибираємо зображення корабля і далі натискаємо на кругову стрілку для повороту зображення. Повертаємо, допоки він не ляже на бік.



- Додайте наступний скрипт до **спрайта корабля**. Поміняйте координати (X, У) в блоках пересування спрайта таким чином, щоб вони відповідали тому, що знаходиться на сцені.



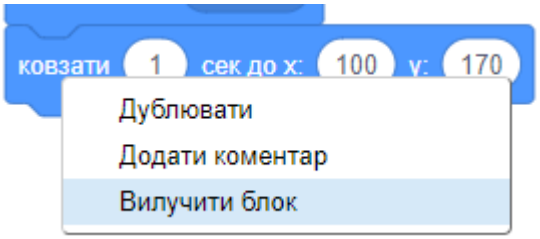
- Натискаючи на зелений прапор, щоб запустити гру, ми маємо побачити, як з корабля доноситься привітання, після чого він нахиляється і летить у бік Землі.

- **ЗАВДАННЯ.** Змінити числа в блоках анімації так, щоб: Корабель мчав до Землі до тих пір, поки не торкнеться її; Корабель мчав до Землі трохи повільніше. Вам треба буде замінити числа в блоці КОВЗАТИ.

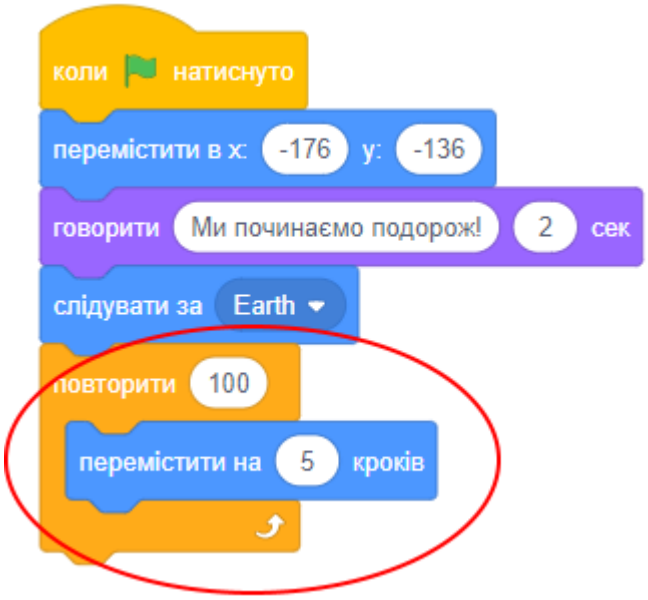
КРОК 2: АНІМАЦІЯ З ВИКОРИСТАННЯМ ЦИКЛІВ

Ще один спосіб змусити корабель летіти - рухати його невеликими кроками багато разів.

- Видаліть КОВЗАТИ з вашого коду, клацнувши правою кнопкою мишки на блоці і натиснувши «Вилучити».

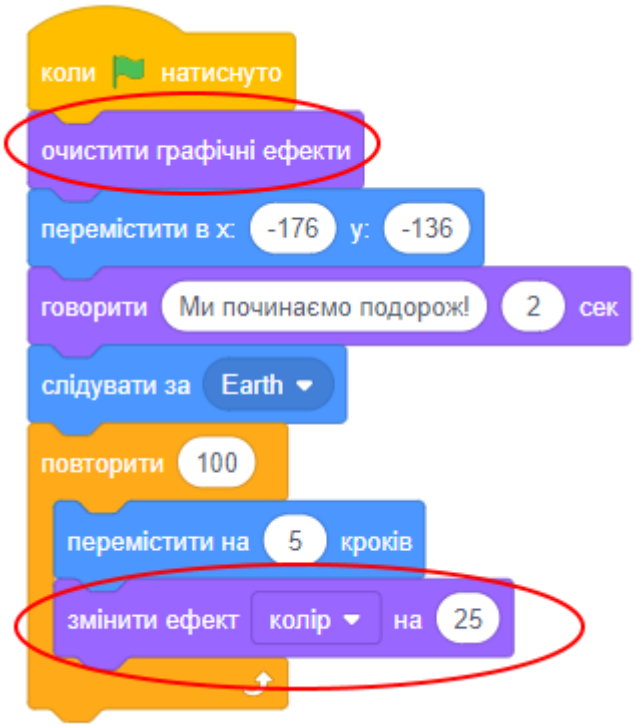


- Також можна видалити блоки, просто протягнувши їх за межі вікна скриптів в ту частину, де знаходиться палітра блоків.
- Додайте замість блока КОВЗАТИ блок ПОВТОРИТИ (вкладка КЕРУВАННЯ). Він використовується для повторення чогось задану кількість разів, також його можна назвати **циклом.**

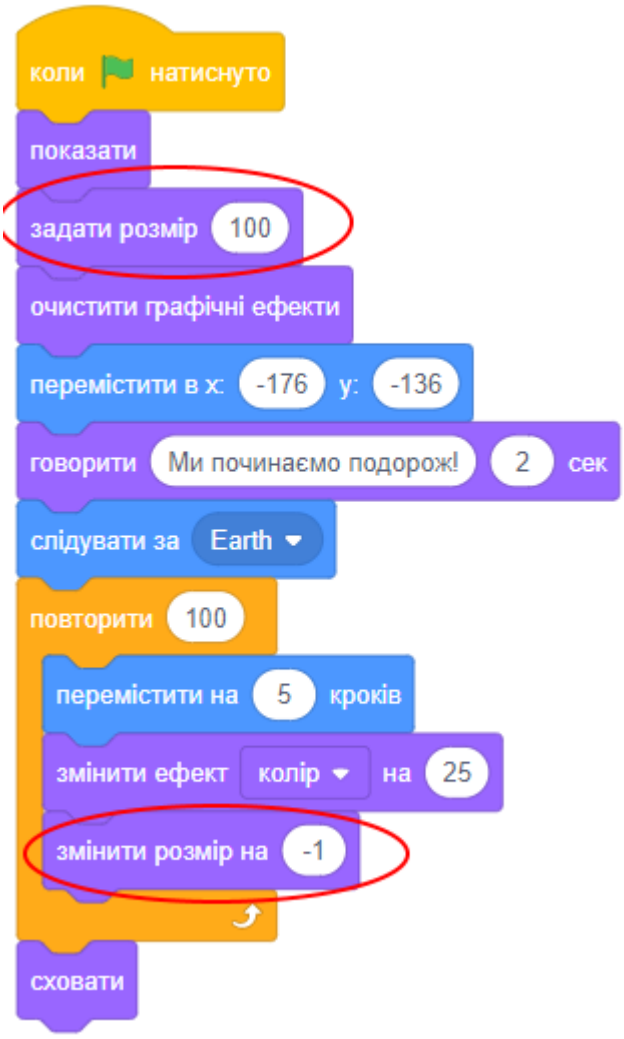


- Натисніть на зелений прапор, щоб протестувати, що вийшло.
- Схоже, в поведінці нічого не змінилося, хоча скрипт інший.
- В цикл можна додати багато цікавих речей. Додайте блок ЗМІНИТИ ЕФЕКТ КОЛІР у цикл (вкладка ВИГЛЯД), щоб кілька разів змінився колір корабля по мірі його руху.

Також додайте блок ОЧИСТИТИ ГРАФІЧНІ ЕФЕКТИ у початок скрипта, щоб відмінити дію зміни кольору на початку гри.



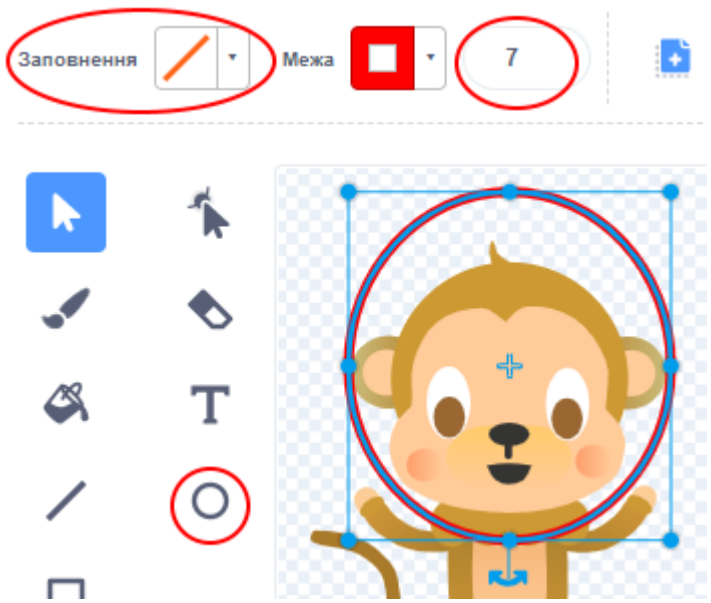
- Подивіться, що вийшло, натиснувши зелений прапор.
- Можна також зменшувати розмір корабля, коли він рухається ближче до Землі і віддаляється від нас. Для цього можна використати блок ЗМІНИТИ РОЗМІР.
- Протестуйте анімацію. Що станеться, якщо натиснути на зелений прапор двічі? Чи стартує корабель з правильної позиції і з правильним розміром?
- Можна використовувати наступний блок, щоб виправити цю проблему: ЗАДАТИ РОЗМІР 100%. Також ми додали блок ПОКАЗАТИ на початку скрипта та блок СХОВАТИ наприкінці скрипта, щоб корабель зникав, коли долітає до Землі.
- Фінальна версія скрипта для корабля показана нижче:



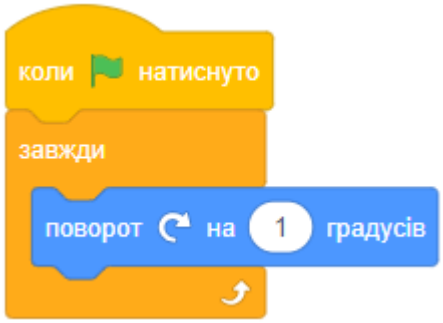
КРОК 3: ЛІТАЮЧА МАВПОЧКА

Хто може літати в космосі? У нас це буде мавпочка, яка загубилася.

- Додайте з бібліотеки спрайт мавпочки на сцену, якщо ще не зробили це на початку.
- Якщо натиснути на мавпочку і перейти у вкладку «Образи», ми можемо редагувати вигляд мавпочки.
- Натисніть на інструмент **КОЛО**, потім намалюйте скафандр навколо голови мавпочки. Тепер вона не задихнеться, може перебувати в космосі. Вам треба встановити прозоре заповнення та потрібну товщину лінії.



- Переходимо знову в область скриптів для мавпочки і додаємо скрипт, щоб вона повільно оберталася по колу, поки її не врятують. Блок ЗАВЖДИ - це ще один вид циклу, цього разу без умов, тому він буде працювати весь час.

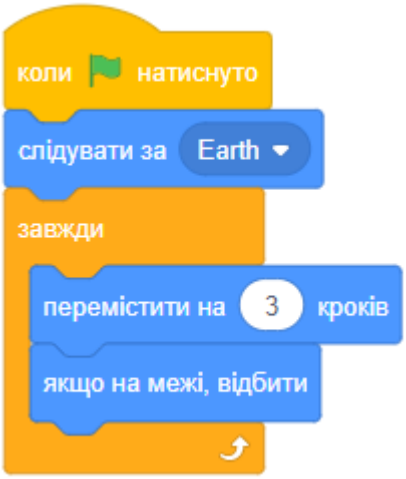


- Давайте протестуємо, що вийшло. Щоб зупинити анімацію, слід натиснути на червону кнопку поряд із зеленим прапором.

КРОК 4: ЛІТАЮЧІ АСТЕРОЇДИ

Як ми могли забути про астероїди? Який космос обходиться без них?

- Додайте спрайт астероїдів на сцену, якщо ще не зробили цього.
- Додайте **скрипт для астероїдів**, щоб вони літали:

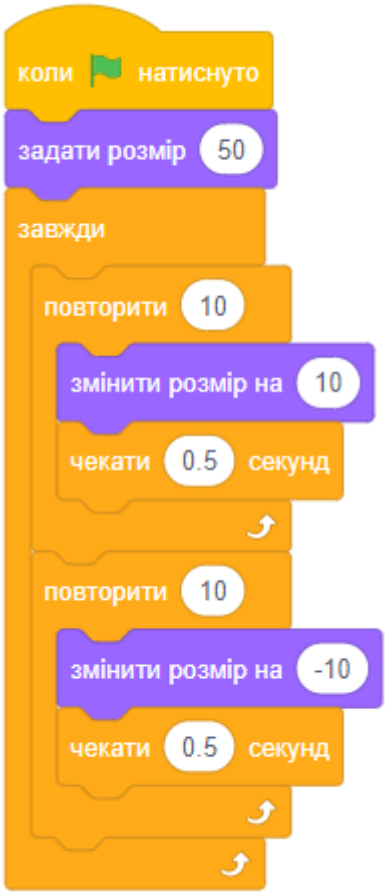


- Перевіримо, що вийшло. Чи літають камені по сцені, відштовхуючись від країв?

КРОК 5: БЛИСКУЧА ЗІРКА

Тепер зробимо деяку анімацію, щоб у нас засвітилася зірка.

- Додайте спрайт "Зірка" на сцену, якщо ще не зробили цього.
- Додайте **скрипт для зірки**.



- Натисніть зелений прапор, щоб перевірити анімацію зірки.

- Що робить цей скрипт? Насправді, ми зробили спочатку зірку трохи більше - в 10 разів, а потім менше в 10 разів, встановивши її початковий розмір. Ці два цикли знаходяться всередині циклу ЗАВЖДИ, тому анімація буде повторюватися весь час.

КРОК 6. ЗАВДАННЯ

ЗАВДАННЯ: КОРАБЕЛЬ ДЛЯ ПОРЯТУНКУ

- До проекту додамо новий космічний корабель, яким будемо керувати з клавіатури (ВГОРУ, ВНИЗ, ВПРАВО, ВЛІВО).
- Цей корабель повинен бути іншого кольору.
- Якщо цей корабель торкається мавпочки - він бере її на борт (мавпочка зникає з екрану).
- Якщо цей корабель торкається планети - він висаджує мавпочку (мавпочка з'являється на Землі).
- Користуйтеся досвідом інших уроків (керування клавіатурою, як перевірити торкання та інше).

ЗАВДАННЯ: ШАЛЕНИЙ КОРАБЕЛЬ

- Створити новий проект.
- Додати тло на вибір.
- Додати спрайт на вибір.
- При натисканні на зелений прапор спрайт повинен нескінченно рухатися в будь-який бік, відбиваючись від меж екрану. При цьому він повинен зменшуватися або збільшуватися.

ЗАВДАННЯ: ЗРОБІТЬ СВОЮ ВЛАСНУ АНІМАЦІЮ

- Зупиніть проект і створіть новий.
- Використовуйте ті навички, які ви тільки що отримали, щоб створити свою власну анімацію.
- Це може бути все що завгодно, але щоб воно було більш-менш реалістичним.

УРОК 6. МИСЛИВЦІ НА ПРИВИДІВ

Відео: https://youtu.be/I9d01obabWk

Ми створимо гру, в якій буде треба полювати на привидів. Тільки від нашої влучності залежить - чи зможемо ми врятувати людей від привидів, або вони й надалі будуть лякати всіх.

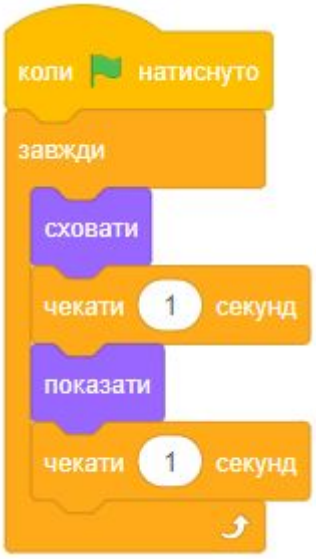
На цьому уроці ми будемо використовувати:

1. Змінні.
2. Оператори порівняння.
3. Блоки звука та керування.

КРОК 1: АНІМАЦІЯ ПРИВИДА

- Створюємо новий проект, видаляємо звідти кота, щоб проект став порожнім.
- Додайте новий спрайт привида і відповідне тло для сцени.

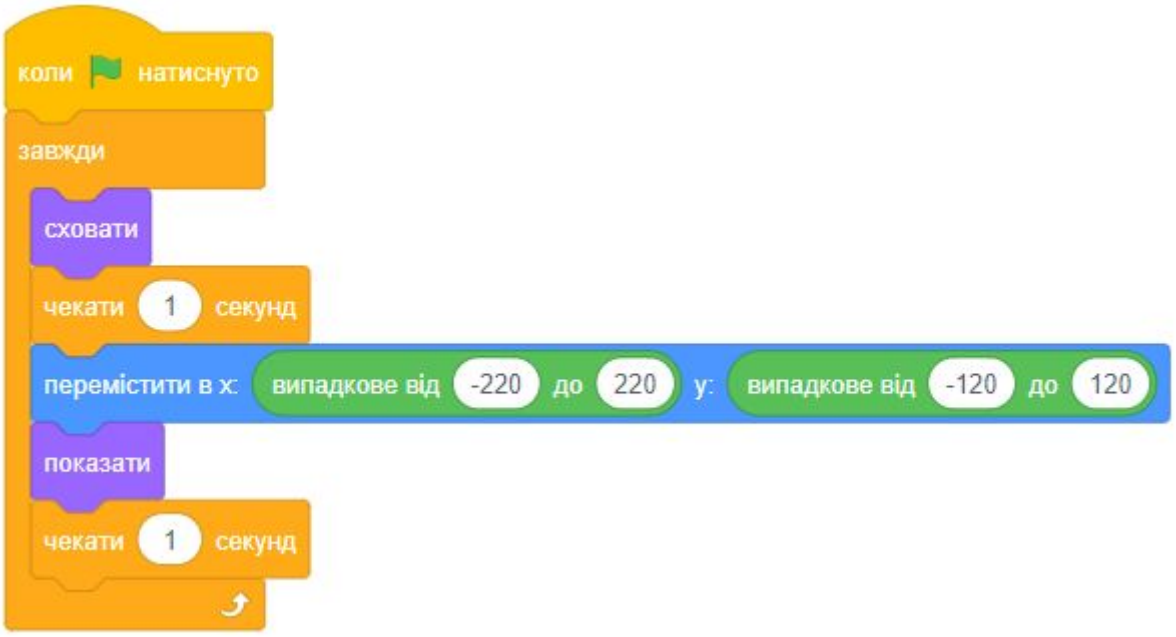
- Додайте цей скрипт до **привида** так, щоб він неодноразово з'являвся і зникав. Потім перевірте поведінку привида, натиснувши на зелений прапор.



КРОК 2: ВИПАДКОВІ ПРИВИДИ

- Привида легко зловити, тому що він не рухається! Замість того, щоб залишатися на тому ж місці, він повинен переміщатися у випадкові координати.

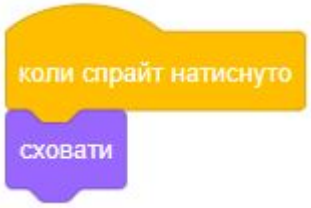
- Додайте блок ПЕРЕМІСТИТИ до скрипта **привида**, щоб він виглядав так:



- Знову перевірте поведінку привида. Ви можете побачити, що кожен раз привид з'являється в новому місці.

КРОК 3: ПОЛЮВАННЯ НА ПРИВИДІВ

- Щоб дозволити гравцеві зловити привида, додайте цей скрипт:



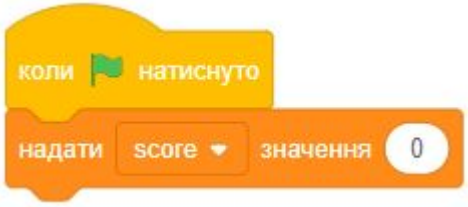
- Перевірте роботу гри. Чи можете ви упіймати привидів, коли вони з'являються? Якщо вам важко це зробити, ви можете грати в повноекранному режимі, натиснувши на цю кнопку:

КРОК 4: ДОДАЄМО РАХУНОК

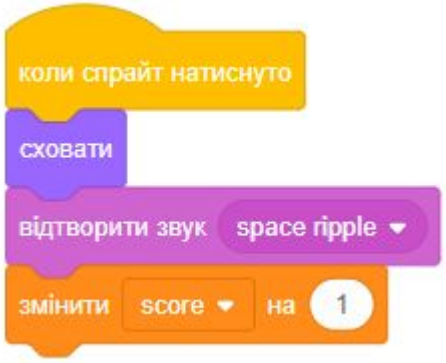
Давайте зробимо гру більш цікавою шляхом додавання рахунку.

- Щоб зберегти рахунок гравця, вам потрібно його кудись помістити. Додайте нову змінну **score** (рахунок). Ви можете побачити цю змінну у верхньому лівому кутку сцени.

- Коли почнеться нова гра (після натискання на зелений прапор), вам потрібно встановити початкове значення **рахунку** = 0. Додайте скрипт до **сцени**.



- Кожен раз, коли привид буде спіймано, вам потрібно додати один бал до **рахунку**. Додайте скрипт до **привида**.



- Запустіть гру та зловить кілька привидів. Рахунок змінився?

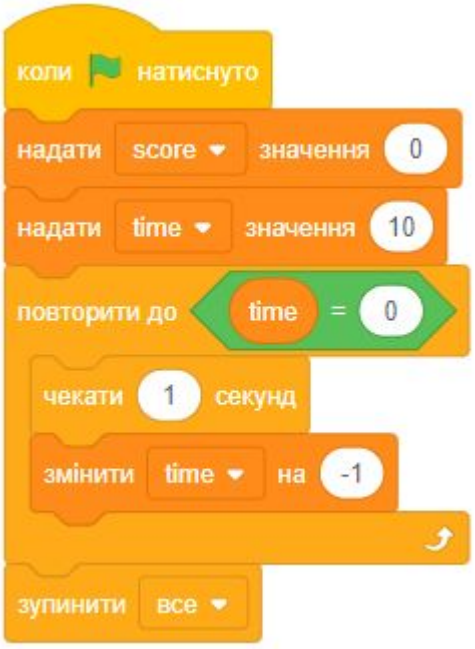
КРОК 5: ДОДАВАННЯ ТАЙМЕРА

Ви можете зробити гру більш цікавою, якщо дасте гравцеві тільки 10-15 секунд, щоб він зловив якомога більше привидів.

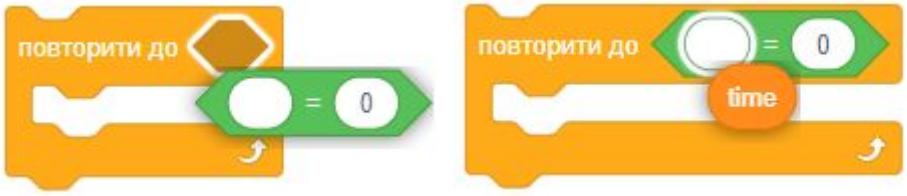
- Ви можете використовувати іншу змінну для часу, що залишається. Натисніть на сцену і створіть нову змінну **time** (час).

- Таймер повинен працювати так:
  - Таймер стартує з 10 секунд;
  - Кожну секунду значення таймера зменшується;
  - Гра закінчується, коли таймер дійде до 0.

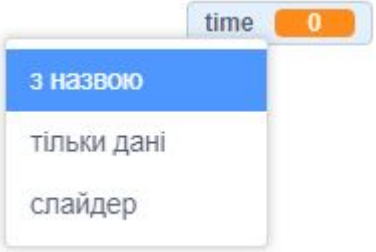
- Додайте скрипт до **сцени**:



- Ось так потрібно додавати блок ПОВТОРИТИ ДО «**time**» = 0:



- Перетягніть змінну **time** в правий (або лівий) куток верхньої частини сцени. Також ви можете натиснути правою кнопкою мишки на змінну і вибрати стиль її відображення (з назвою або без неї).



- Змінні на сцені повинні виглядати гарно. Додайте два спрайта, які будуть інформувати гравця, в якому місці на сцені розташований рахунок, а де відображено час.

- Протестуйте гру. Скільки балів вам вдалося набрати? Якщо гра занадто проста, ви можете:
  - Надати гравцеві менше часу;
  - Зробити так, щоб привиди з'являлися рідше;
  - Зробити привидів меншими за розміром.

- Зіграйте кілька разів, поки вам не підійде рівень складності.

КРОК 6: ЗАВДАННЯ

ЗАВДАННЯ: БІЛЬШЕ ВИПАДКОВОСТЕЙ

Ви можете змусити чекати привид випадкову кількість часу, перш ніж він з'явиться? Чи зможете ви використовувати блок ЗМІНИТИ РОЗМІР, щоб кожен раз, коли з'являється привид, він змінював свій розмір?

ЗАВДАННЯ: ДОДАВАННЯ ЗВУКУ

- Чи можете ви додати звук кожен раз, коли привида спіймано?
- Щоразу коли привид з'являється на сцені, треба програвати страшний звук.
- Додайте фонову музику, яка буде звучати впродовж всієї гри.
- Кожну секунду має звучати короткий тихий звук.

ЗАВДАННЯ: БІЛЬШЕ ОБ'ЄКТІВ

Ви можете додати інші об'єкти у вашу гру?

Подумайте про об'єкти, які ви хочете додати:

- Якого розміру вони будуть?
- Вони будуть з'являтися частіше або рідше за привидів?
- Як ваші об'єкти будуть виглядати та звучати, коли їх упіймають?
- Скільки балів ви отримаєте (або втратите), коли їх упіймаєте?

ЗАВДАННЯ: ДОДАТКОВІ

- Нехай деякі з доданих об'єктів після того, як з'являться, рухаються по сцені у випадковому напрямку з випадковою швидкістю. Напрямок і швидкість руху задаються один раз при появі і зберігаються, допоки гравець не «зловить» об'єкт.
- Додайте рівні складності. Спочатку об'єкти з'являються на сцені і не зникають дві секунди. З кожним новим спійманим об'єктом, час появи наступних скорочується, що призводить до того, що їх складніше буде зловити. Також можна змінювати їх розмір.
- Нехай кількість балів збільшується пропорційно до складності (швидкість, час появи, розмір).
- Після закінчення гри на сцені з'являється кіт або інший спрайт, який оголошує набрану кількість очок.

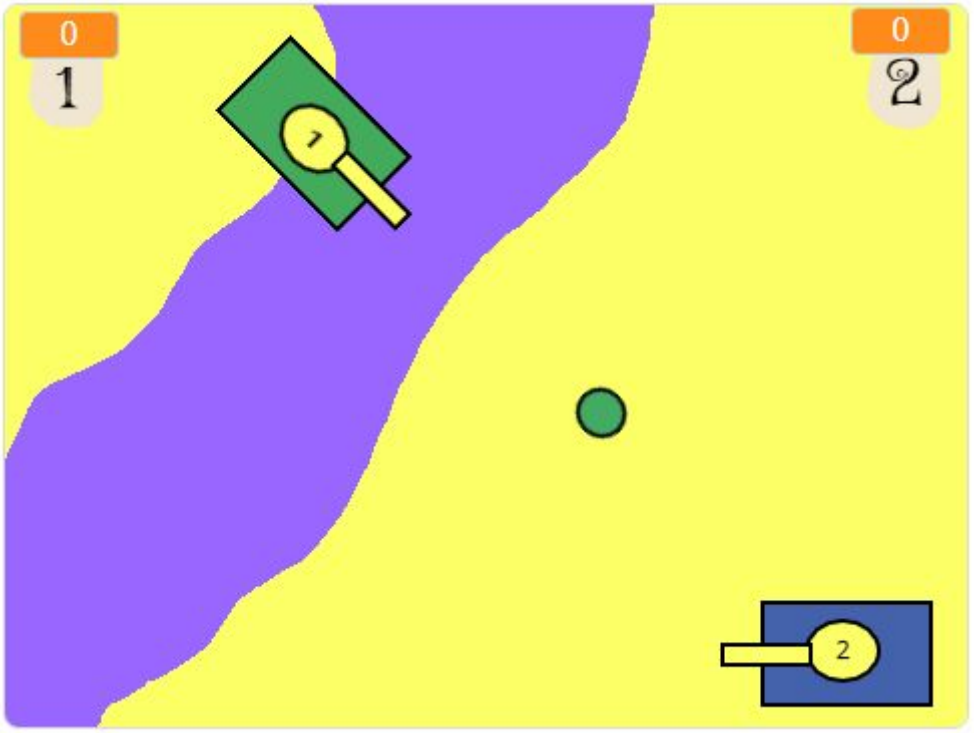
ЗАВДАННЯ: ДОДАТКОВІ 2

- Деякі привиди можуть зменшувати рахунок гри при натисканні.
- Зробіть кілька образів для привидів, вибирайте їх випадково.
- Деякі привиди будуть ховатися тільки після другого натискання.
- Додайте кілька різних фонів на сцену, щоб вони змінювалися через кожні 5-10 секунд.
- Анімувати привидів, щоб вони рухалися у випадкову область на сцені. При русі вони повинні виглядати трохи по-іншому.
- Намалюйте спрайт прицілу, який буде рухатися разом з вказівником мишки. Якщо вказівник мишки торкається привида, приціл мусить змінювати свой образ або колір.
- Придумайте своє вдосконалення для гри.

УРОК 7. ВІЙНА ТАНКІВ

Відео: https://youtu.be/excA13NGIqc

Ми зробимо гру, в якій два танка будуть воювати. Гра розрахована на двох гравців, кожен з яких керує власним танком.



На цьому уроці ми будемо використовувати:

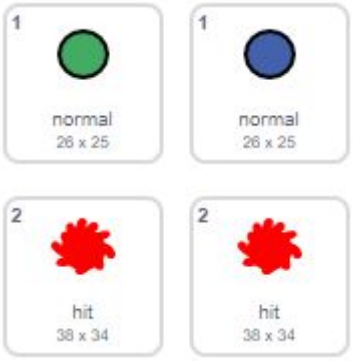
1. Змінні.
2. Повідомлення.
3. Арифметичні та логічні оператори.

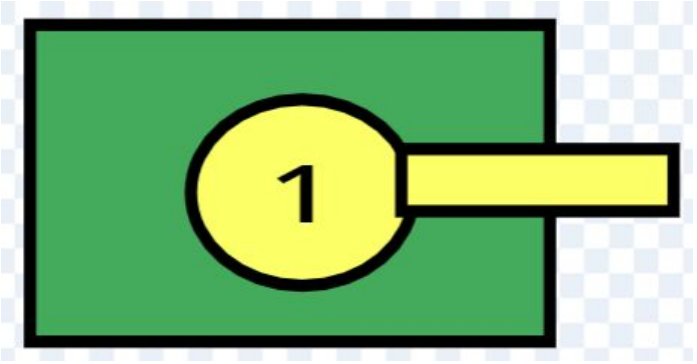
КРОК 1: СПРАЙТИ ТА СЦЕНА

- Намалюйте сцену, на якій буде відбуватися битва (або виберіть її з вбудованої бібліотеки).
- Додайте дві нові змінні **tank1** і **tank2** для відображення результатів гри та розташуйте їх зверху на сцені.

- Намалюйте спрайти танків різного кольору та спрайти снарядів.

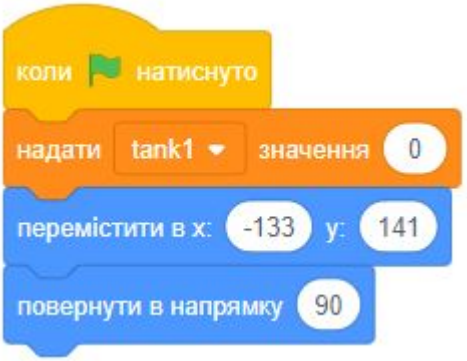
- У кожного снаряда має бути два образи (в звичайному стані і в стані вибуху). У танка має бути номер зверху.



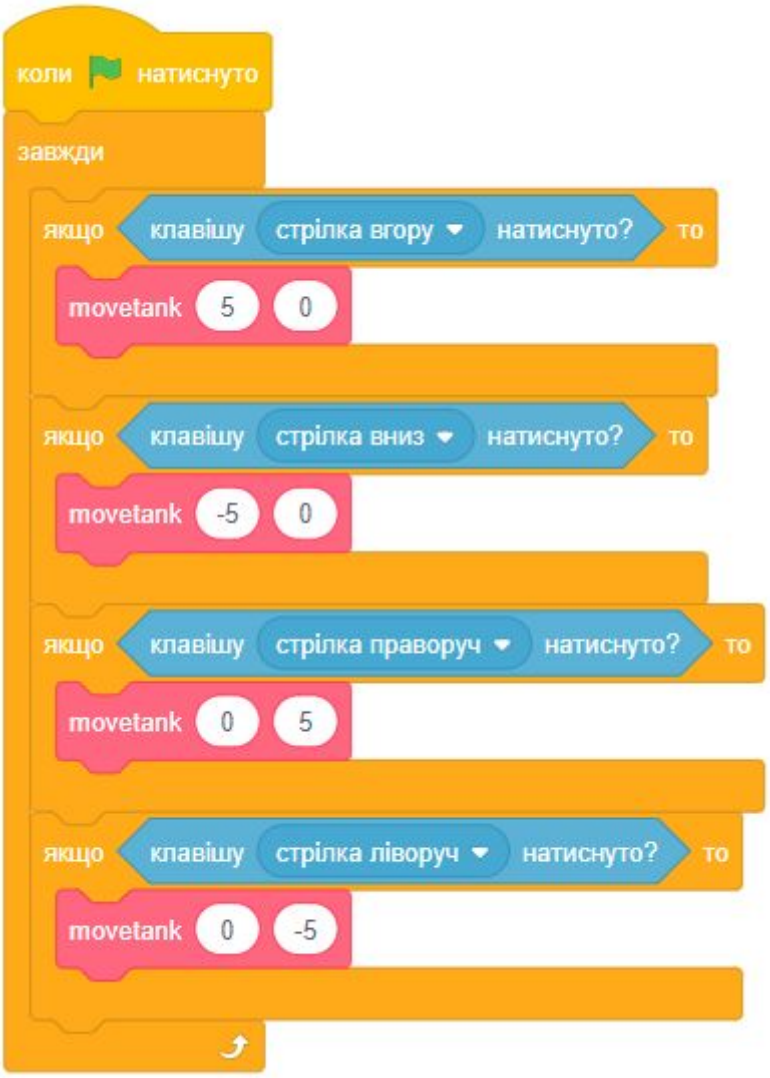


КРОК 2: ПЕРШИЙ ТАНК

- Цей скрипт першого танка встановлює початкове положення спрайта та його напрямок. Також цей скрипт надає початкове значення для змінної рахунку гри.



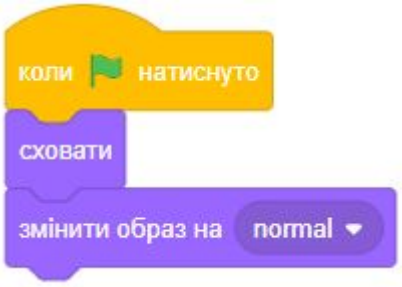
- Кожен танк має власні початкові координати та напрямок, тому що танки розміщуються в різних куточках сцени та повернуті в протилежних напрямках.
- Цей скрипт танка обробляє потрібні нам клавіші клавіатури та викликає блок для пересування і повороту танка.



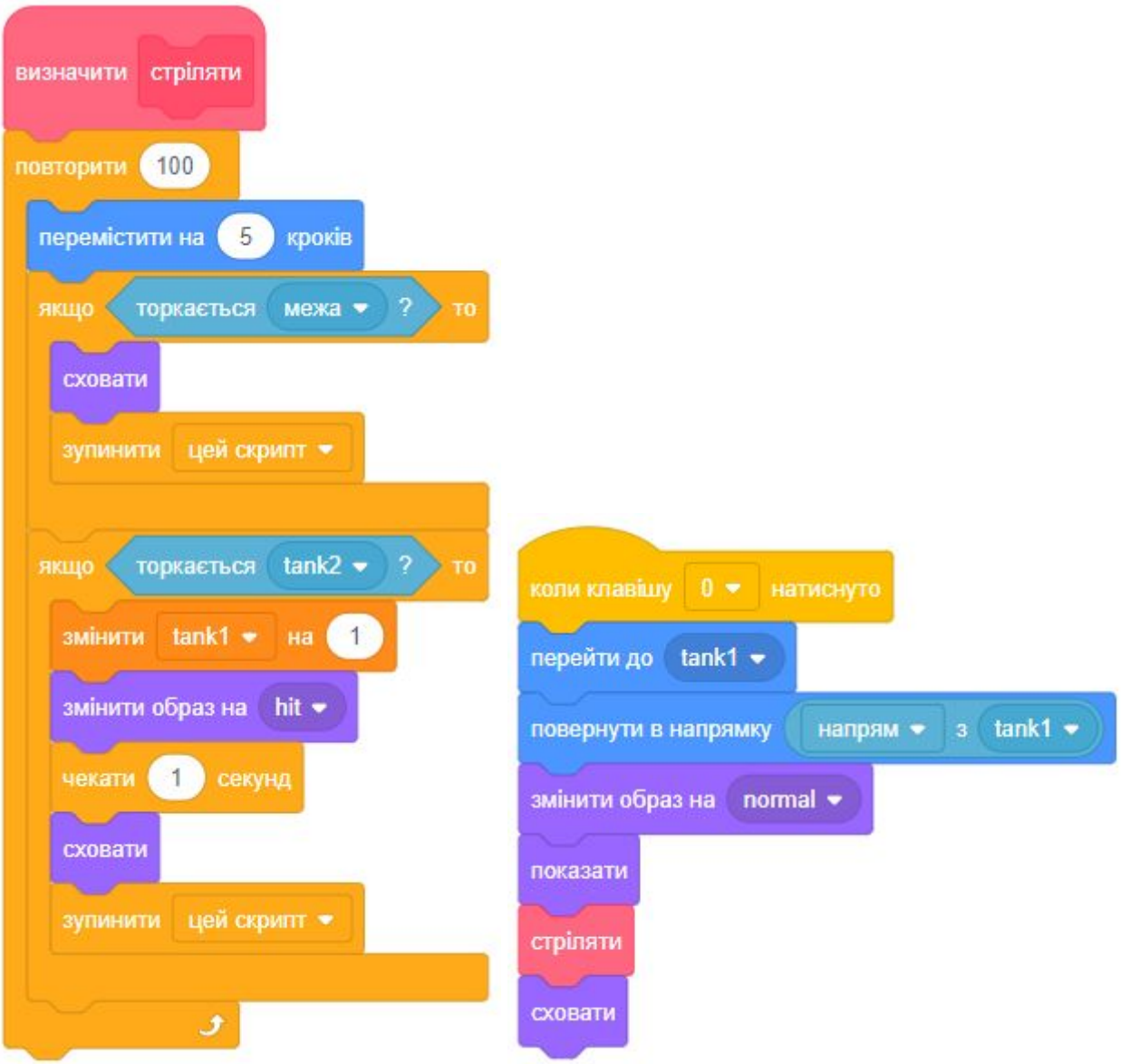
- Цей допоміжний скрипт переміщає танк на потрібне число кроків (перший параметр блока) та здійснює поворот на вказане число градусів (другий параметр блока). Також скрипт перевіряє, чи торкається танк іншого танка або межі екрану. Якщо танк торкається, переміщення і поворот танку відміняється.

КРОК 3: ПЕРШИЙ СНАРЯД

- На початку гри снаряд схований, він показується тільки коли танк стріляє. Також нам треба встановити його початковий образ.

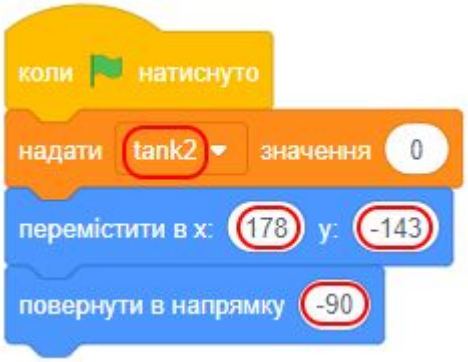


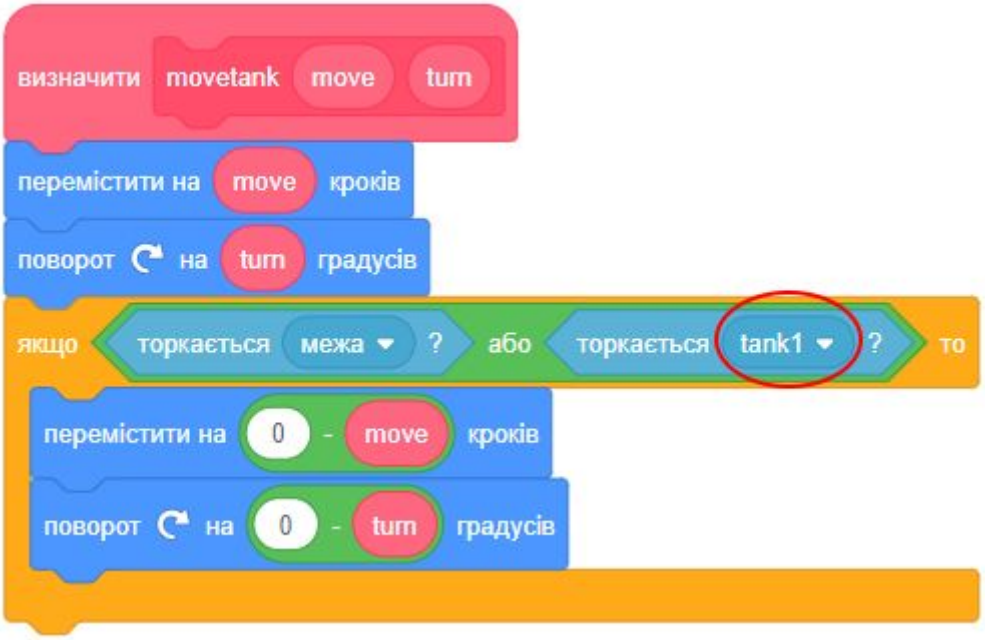
- Коли натиснута клавіша пострілу, снаряд переміщується по координатам танка та повертається в напрямку його руху. Потім снаряд показується та переміщується вперед. Після цього снаряд знову ховається до наступного пострілу.
- Під час переміщення постійно перевіряється торкання до інших об'єктів. Якщо снаряд торкнувся межі екрана, він ховається. Якщо снаряд торкається іншого танка, його образ змінюється на вибух та збільшується рахунок гри.



КРОК 4: ДРУГИЙ ТАНК

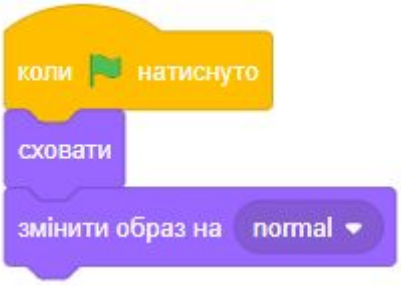
Ви можете дублювати скрипти першого танка з деякими змінами.



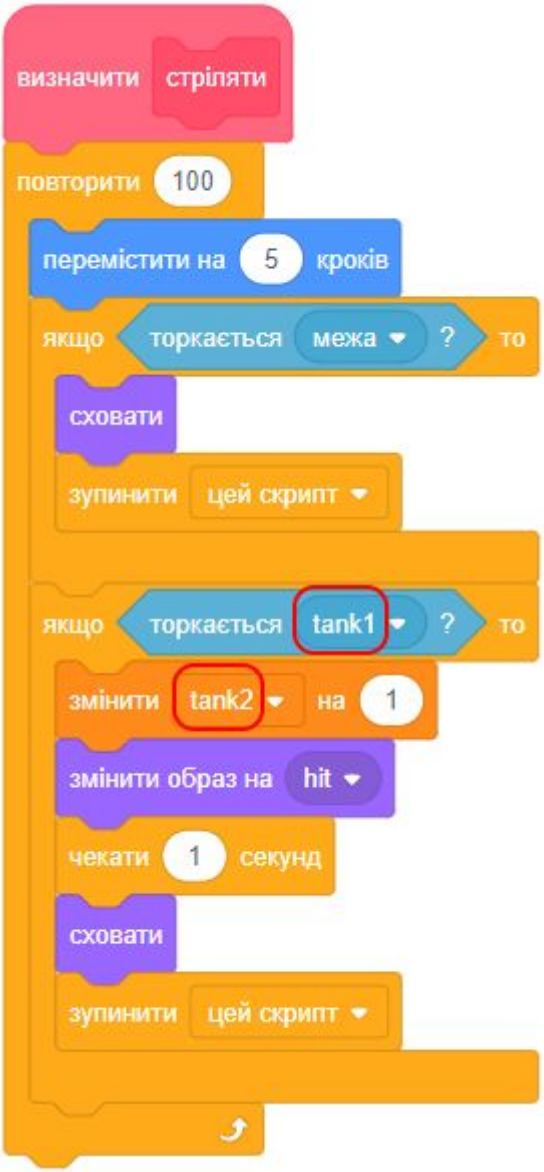


КРОК 5: ДРУГИЙ СНАРЯД

Можливо дублювати скрипти першого снаряду з деякими змінами.



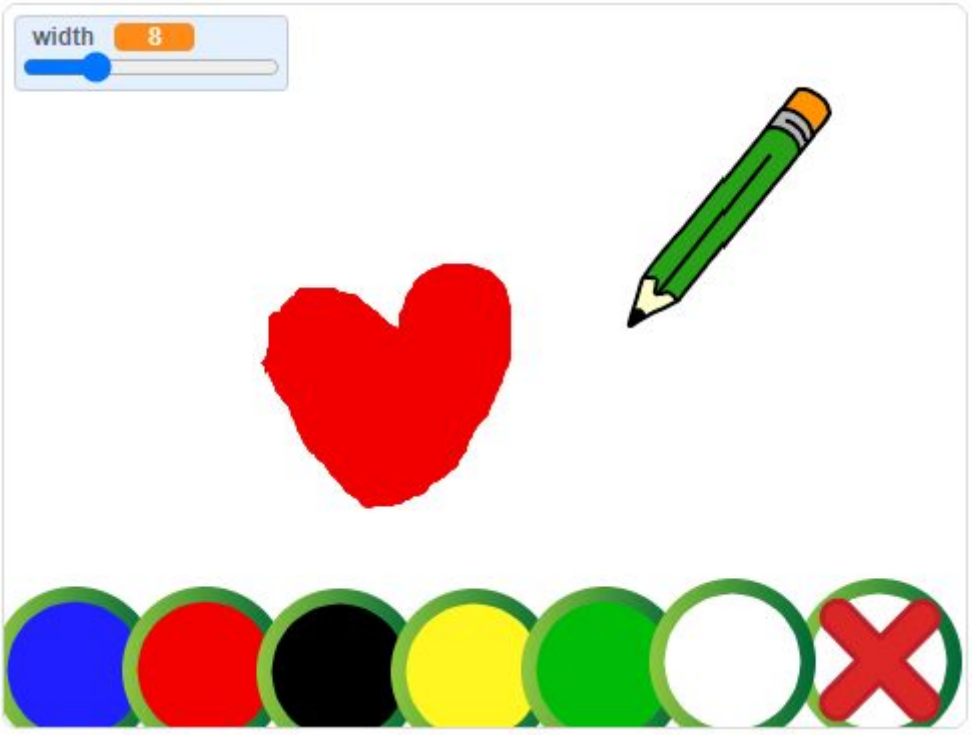
Другий снаряд також має власну версію визначення блока СТРІЛЯТИ:



УРОК 8. МАЙЖЕ ФОТОШОП

Відео: https://youtu.be/2D0-rRBp-o4

На цьому уроці ми зробимо графічний редактор, в якому можна буде малювати олівцями різних кольорів.



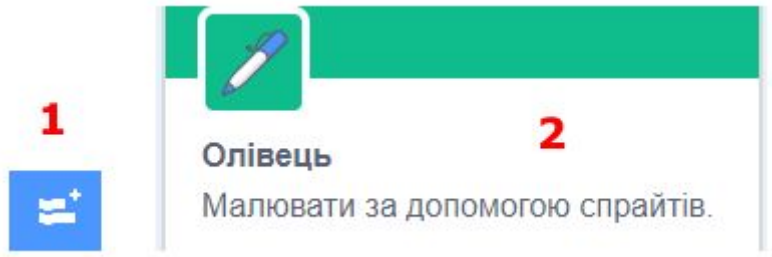
На цьому уроці ми будемо використовувати:

1. Змінні.
2. Повідомлення.

КРОК 1: СТВОРЮЄМО ОЛІВЕЦЬ

Давайте почнемо зі створення олівця, який буде малювати на сцені.

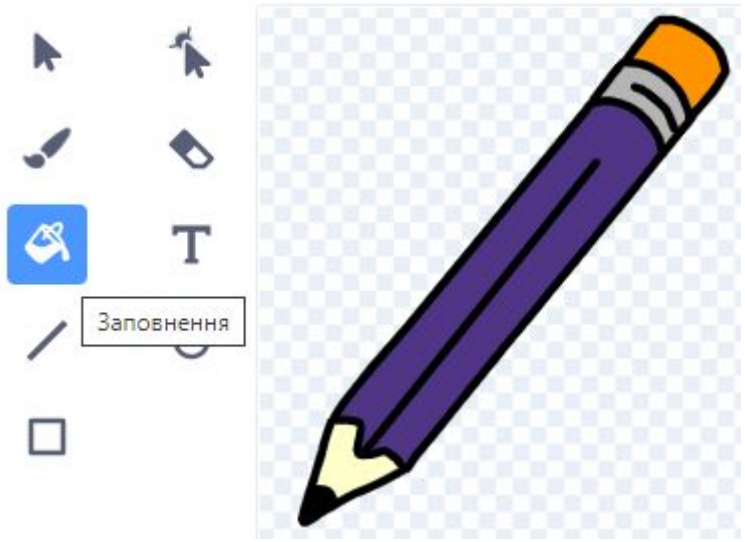
- Створюємо новий проект, видаляємо звідти кота.
- У новому Scratch категорія блоків управління олівцем захована в розширеннях. Для того щоб її додати, необхідно натиснути кнопку додавання розширення та вибрати розширення ОЛІВЕЦЬ.



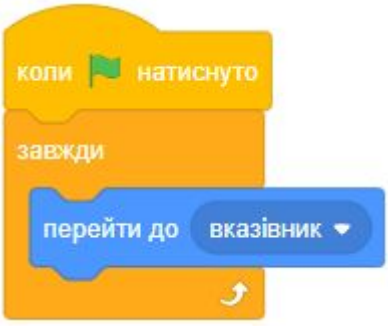
- Додайте спрайт олівця в проект.

- Зайдіть у вкладку ОБРАЗИ та видаліть другий костюм.

- Змініть назву образу на "blue" (синій). За допомогою інструмента ЗАПОВНЕННЯ розфарбуйте олівець синім кольором.

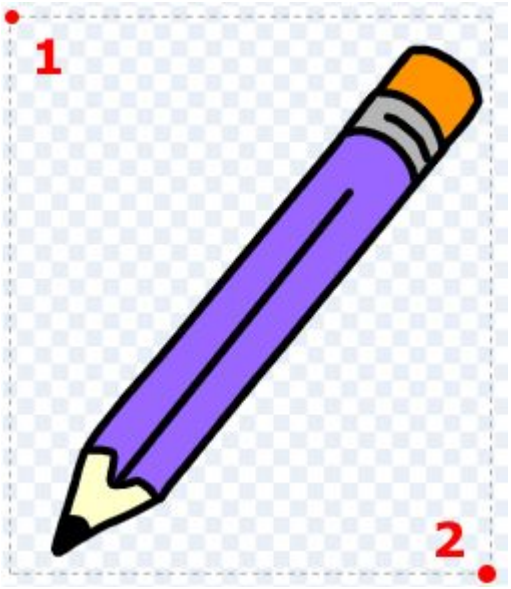


- Оскільки ми використовуємо мишку для малювання на сцені, нам необхідно, щоб олівець слідував за вказівником мишки ЗАВЖДИ. Додайте **скрипт для спрайта олівця**:

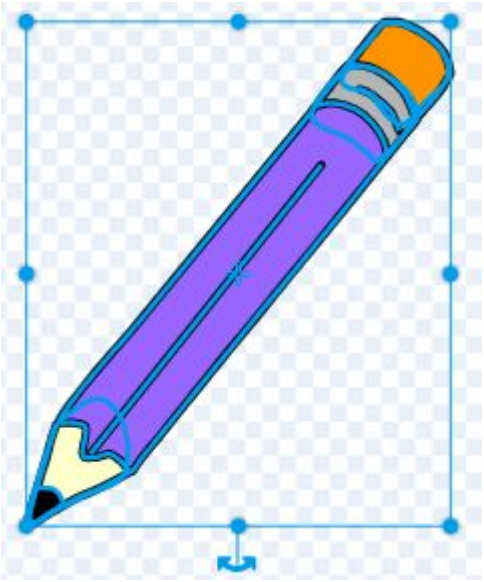


- Протестуйте скрипт натисканням на зелений прапор. Порухайте вказівником мишки навколо сцени. Перевірте, як все працює.
- Напевно, ви помітили, що олівець слідує за мишкою, але малювати він збирається не кінцем грифеля, а серединою.
- Щоб це виправити, зайдемо на образи спрайту олівця, виберемо наш синій образ і перенесемо олівець трохи правіше та вище центру спрайта. Як це зробити?

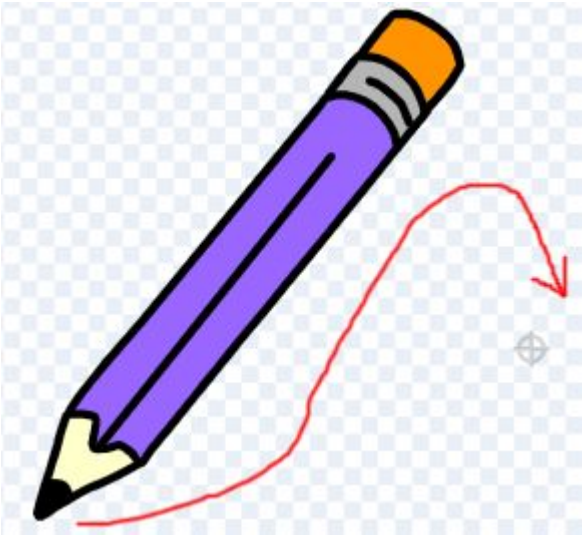
1. Натисніть на інструмент ОБРАТИ  .
2. Натисніть лівою кнопкою мишки на порожньому місці в точці, що розташована зверху та зліва від олівця (точка 1).
3. Не відпускаючи кнопку мишки, тягніть прямокутник вибору до правої нижньої точки, яка розташована за олівцем (точка 2).

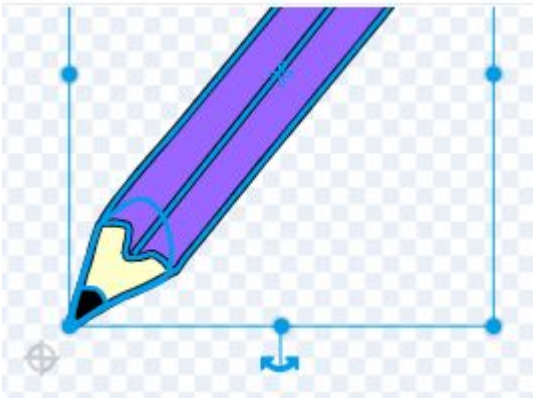


4. Тепер відпустіть кнопку мишки, всі частини олівця будуть зібрані в группу. Це виглядає так:

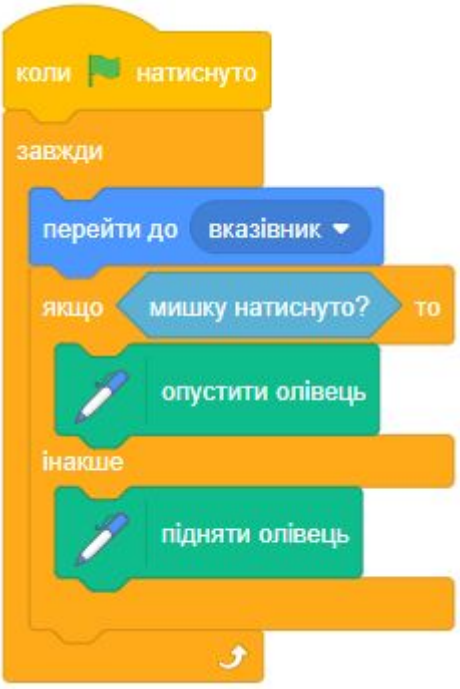


5. Натисніть на кнопку ГРУПУВАТИ  , яка знаходиться зверху на палітрі інструментів.
6. Тепер ми можемо перетягнути олівець за допомогою мишки куди завгодно, залишаючи всі його частини якими вони мають бути. Перетягніть образ олівця правіше та вище за центр спрайта.





- Перейдемо знову на скрипти і протестуємо проект. Так краще?
- Тепер зробимо так, щоб олівець малював, якщо була натиснута кнопка мишки. Додайте **скрипт для олівця**:

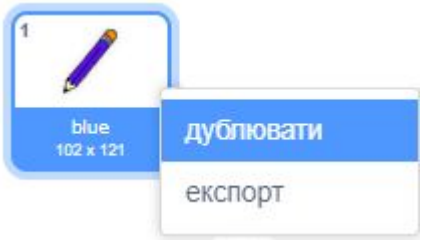


- Рухайте олівцем по сцені, натискаючи кнопку мишки. Малює?

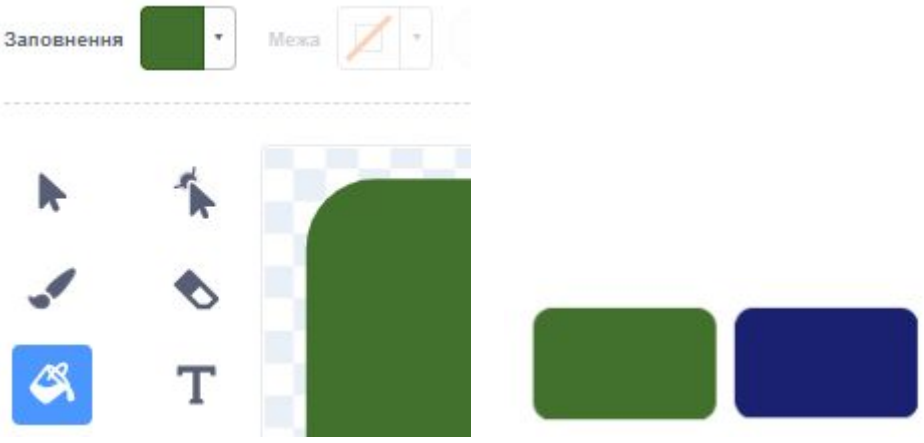
КРОК 2: КОЛЬОРОВІ ОЛІВЦІ

Додамо ще один колір й дозволимо користувачеві вибирати його.

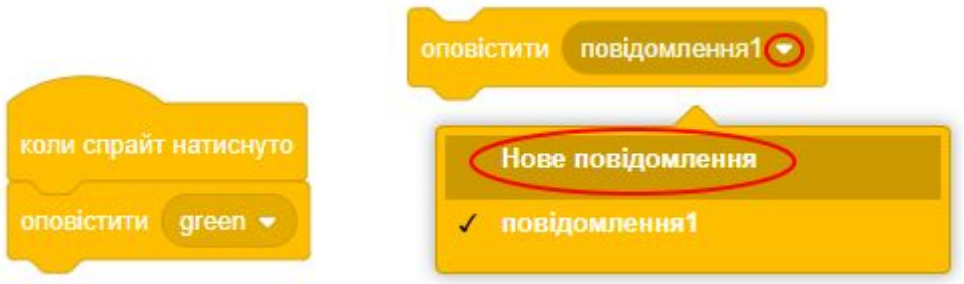
- Натисніть на спрайт олівця, зайдіть в ОБРАЗИ і продублюйте образ синього олівця.



- Перейменуйте новий образ в "green" (зелений) і розфарбуйте його зеленим кольором.
- Створіть два нових спрайта, які ми будемо використовувати як кнопки для вибору синього або зеленого олівця. У бібліотеці спрайтів є кнопки різної форми, оберіть, які вам подобаються, та змініть їх колір.

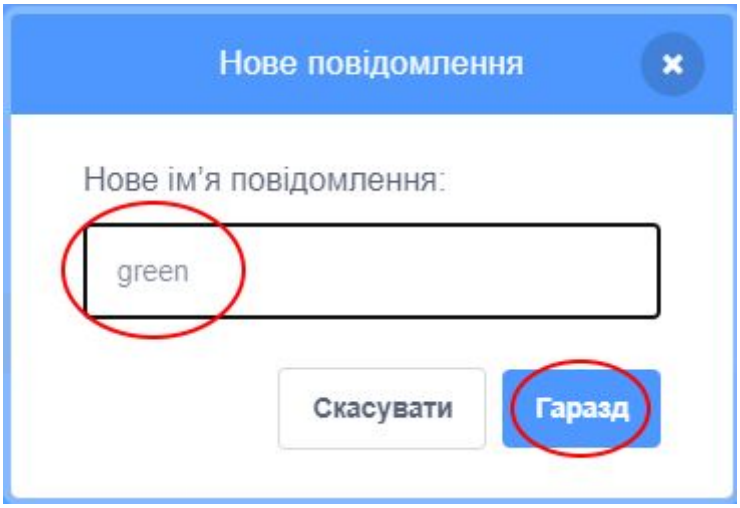


- Коли натиснута зелена кнопка, ми повинні передати повідомлення спрайту олівця, щоб він змінив свій образ. Додайте **скрипт для зеленої кнопки**:

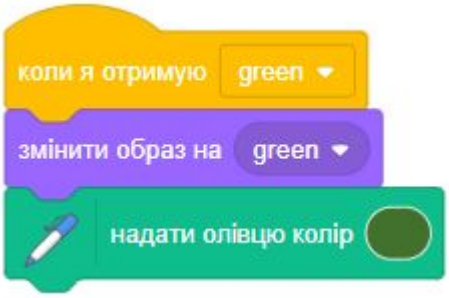


- Щоб створити в блоці ОПОВІСТИТИ нове повідомлення, натисніть трикутник і виберіть пункт НОВЕ ПОВІДОМЛЕННЯ.

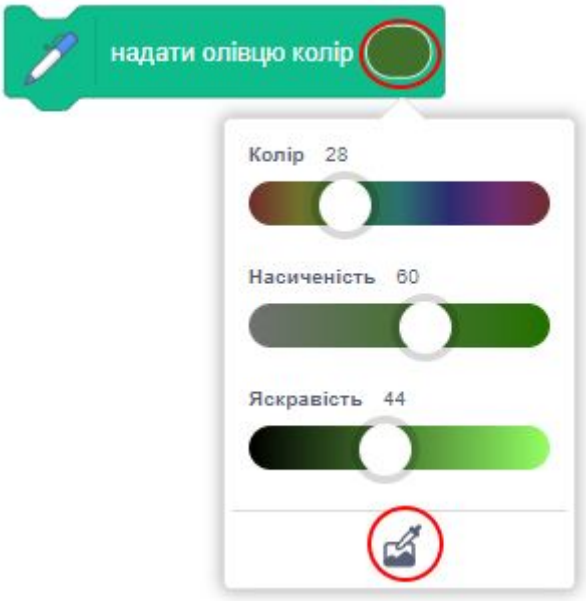
- У вікні створення нового повідомлення напишіть слово "green" та натисніть кнопку ГАРАЗД. Нове повідомлення готове.



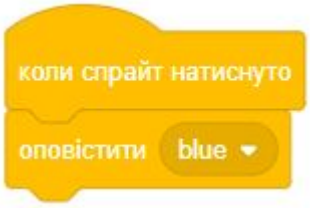
- Тепер олівець буде знати, що робити, коли отримає це повідомлення. Додайте **скрипт для спрайта олівця**:



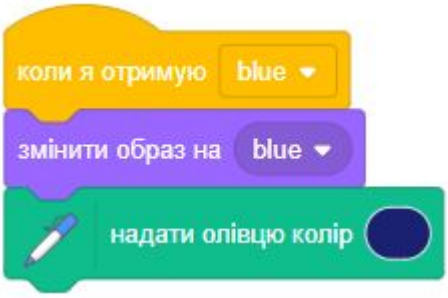
- Щоб отримати зелений колір, клацніть поле кольору в блоці КОЛІР ОЛІВЦЯ і натисніть на значок вибору кольору. Потім виберіть потрібний колір за допомогою мишки.



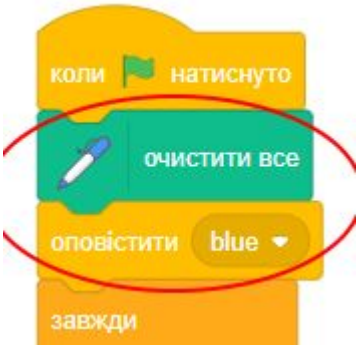
- Додайте **скрипт для синьої кнопки**:



- Додайте **скрипт для спрайта олівця**, що обробляє синій колір:



- Тепер при старті проекту нам необхідно вказати спрайту олівця, який костюм і який колір повинен бути обраний, а також очистити сцену від попередніх малюнків.
- Додайте два блоки в початок скрипта для **спрайта олівця**. Зрозуміло, можна стартувати і з іншого кольору за бажанням!



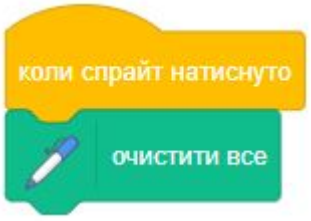
- Протестуйте проект. Чи можете ви перемикатися між кольорами?

КРОК 3. ВИДАЛЕННЯ ПОМИЛОК

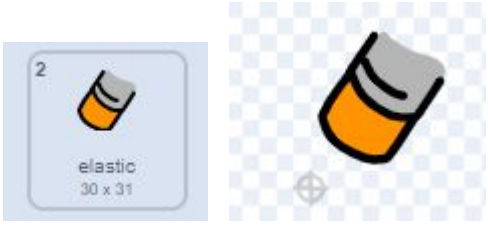
Коли малюєш на папері, іноді трапляються плями. Так хочеться їх стерти гумкою або взагалі стартувати з чистого аркуша! Давайте запрограмуємо кнопку, яка надасть нам новий чистий аркуш.

- Додайте спрайт з буквою "X" з бібліотеки та видаліть його перший образ, щоб залишився тільки "X" червоного кольору.

- Додайте **скрипт для "X" кнопки**, щоб очистити всю сцену:

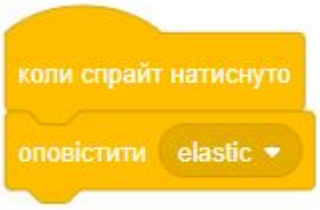


- Нам знадобиться гумка для більш дрібної роботи. Намалюйте її самостійно, створивши **новий образ для спрайта олівця**, або продублюйте існуючий, повернувши та перефарбувавши його.

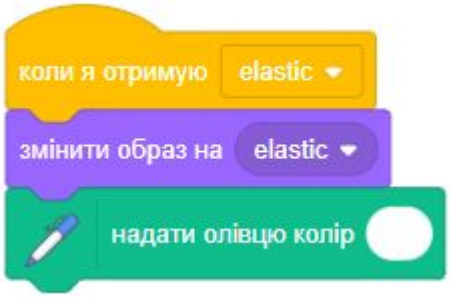


- Також нам потрібно додати кнопку, після натискання на яку буде обраний інструмент "гумка".

- Щоб олівець міг змінювати свій образ на гумку, потрібно додати цей **скрипт для кнопки гумки**.



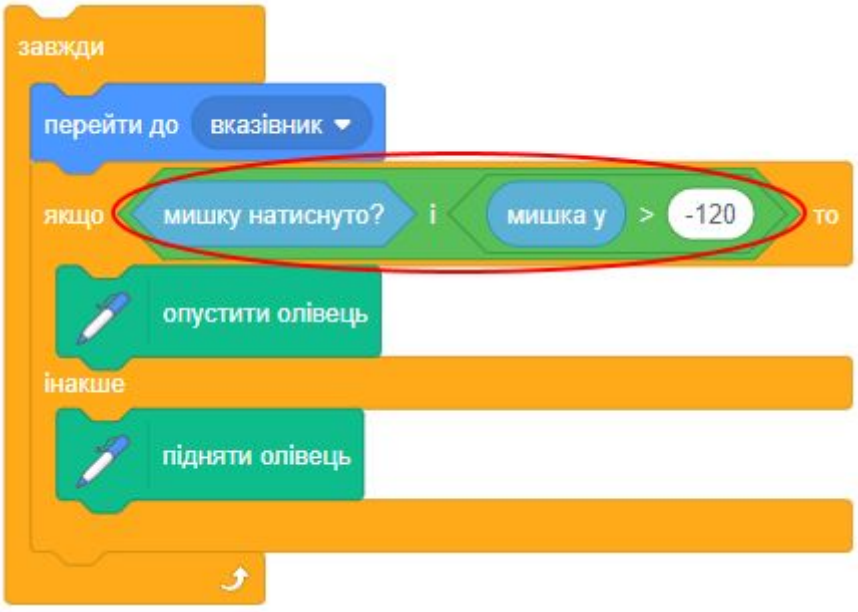
- Коли олівець отримує це повідомлення, його образ повинен змінитися на гумку і встановити колір малювання в колір сцени. Додайте **скрипт для олівця**:



- Протестуйте проект, щоб побачити, чи можете ви стерти повністю все зі сцени, а також як працює гумка.

- Є ще одна невелика проблема з олівцем - він може малювати будь-де на сцені, включаючи область, де розташовані кнопки вибору кольорів!

- Щоб поліпшити стан справ, ви повинні дозволити олівцю малювати, тільки якщо положення миші по осі Y більше -120. Змініть умову блока ЯКЩО в **скрипті олівця**, щоб він виглядав:

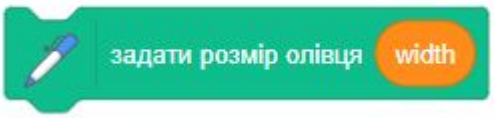


- Протестуйте проект. Тепер малювати в області наших кнопок буде заборонено.

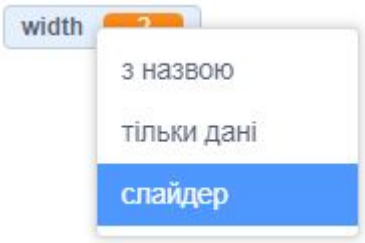
КРОК 4: ЗМІНА ТОВЩИНИ ОЛІВЦЯ

Давайте зробимо так, щоб олівець малював грифелем різної товщини.

- Додайте нову змінну **width** (товщина). Зайдіть у вкладку ЗМІННІ та натисніть на кнопку "Створити змінну".
- Додайте цей блок всередину циклу ЗАВЖДИ для скрипта олівця. Тепер олівець буде постійно малювати розміром, що зазначений у змінній "width".



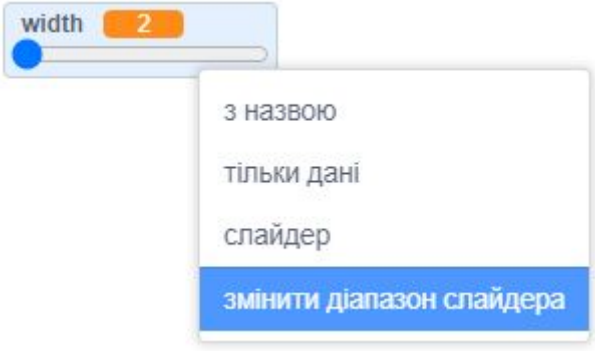
- Натисніть правою кнопкою мишки на змінну у вікні сцени і виберіть пункт "Слайдер".



- Тепер можливо керувати товщиною олівця за допомогою важеля.

- Протестуйте проект, подивіться, як малює олівець, якщо змінити товщину його грифеля.

- Можна встановити дозволений мінімум і максимум для значення "width". Щоб це зробити, натисніть правою кнопкою мишки на змінну та виберіть "Змінити діапазон слайдера". Встановіть діапазон у межах розумного, від 1 до 20.



- Продовжуйте тестувати і експериментувати зі змінною "width" до тих пір, поки ви не досягнете бажаного ефекту.

КРОК 5: ЗАВДАННЯ

ЗАВДАННЯ: БІЛЬШЕ ОЛІВЦІВ

Додайте до проекту червоний, жовтий і чорний олівці. Не забудьте також додати кнопки для цих олівців! Чи зможете ви тепер намалювати красиву картину, використовуючи всі олівці?

ЗАВДАННЯ: ШВИДКИЙ ДОСТУП З КЛАВІАТУРИ

Чи можете ви створити ярлики для найчастіше використовуваних команд малювання? Наприклад, клавіші:

- 1 - перемикаємо олівець на **синій** колір.
- 2 - перемикаємо олівець на **зелений** колір.
- 3 - перемикаємо олівець на **червоний** колір.
- 4 - перемикаємо олівець на **жовтий** колір.
- 5 - перемикаємо олівець на **чорний** колір.
- 0 - перемикаємо олівець в **режим гумки**.
- ПЛЮС - **збільшити товщину** олівця.
- МІНУС - **зменшити товщину** олівця.
- ПРОПУСК - **стерти все**, що на екрані.

- Також підпишіть кнопки зміни кольорів цифрами.

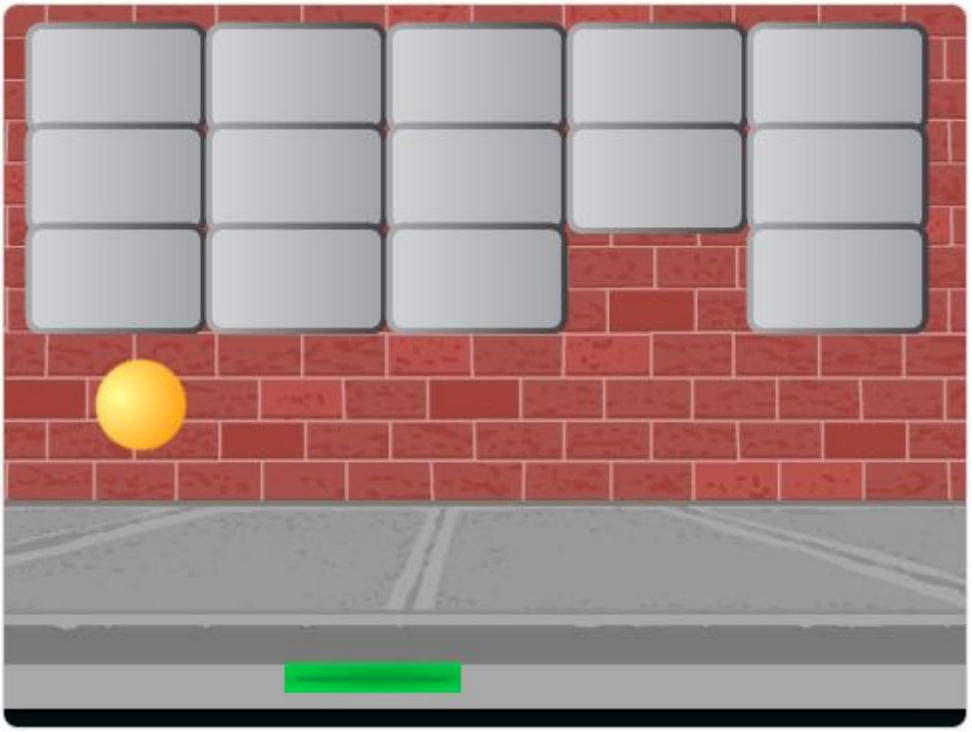
ЗАВДАННЯ: ЩО ЩЕ МОЖНА ДОДАТИ?

- Додайте кнопки (+) та (-), щоб при натисканні вони збільшували та зменшували товщину олівця .
- Додайте нову змінну **elasticWidth** (товщина гумки) та використовуйте її замість **width**, коли обрано інструмент гумки.
- Додайте малювання квадрату, трикутника та кола при натисканні на якісь клавіші. Вам буде потрібно додати нові змінні для розміру фігур, які будете малювати. Щоб згадати малювання фігур, дивіться Урок 1.
- Що ще можна додати в цей проект?

УРОК 9. АРКАНОЇД

Відео: https://youtu.be/_iobsYM68vE

Ми будемо робити гру арканоїд. Гравець контролює ракетку, яку можна пересувати горизонтально, підставляючи її під кульку, щоб запобігти її падіння. Удар кульки по цеглині призводить до її руйнування.

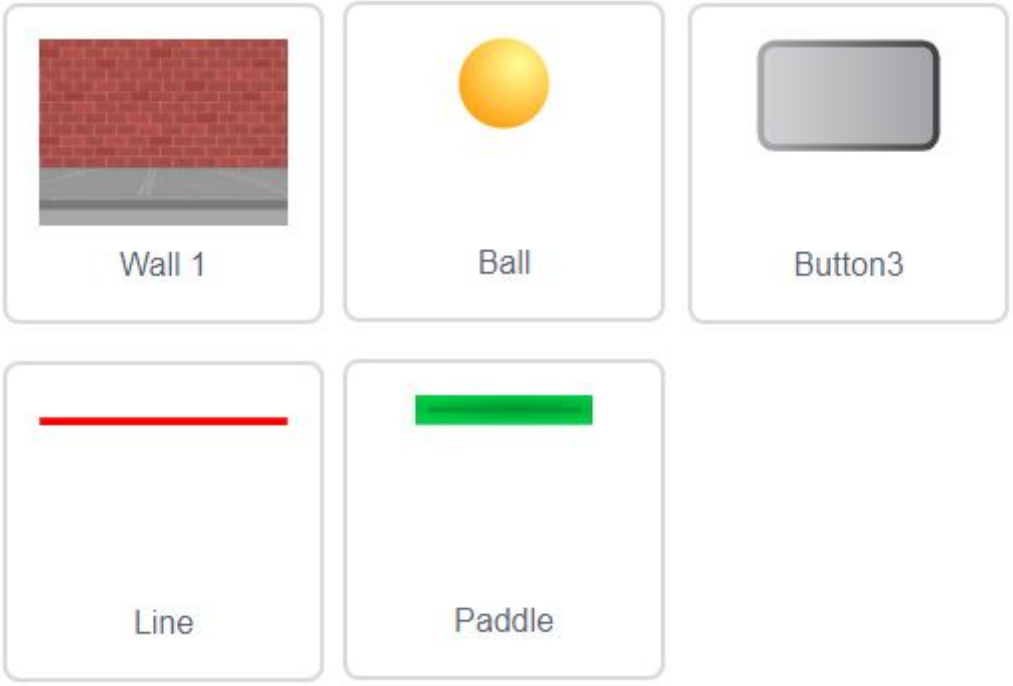


На цьому уроці ми будемо використовувати:

1. Клони
2. Повідомлення.

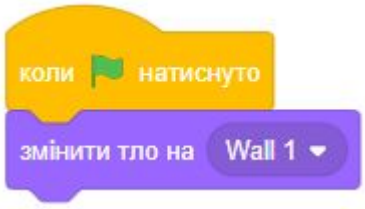
КРОК 1: СПРАЙТИ ТА СЦЕНА

- Додайте з бібліотеки або намалюйте чотири спрайта та виберіть будь-яке тло для сцени. Наприклад:



Ball - кулька. Вона літає по сцені, відштовхується від ракетки та меж екрану, розбиває цеглу.
Paddle - ракетка. Вона може рухатися тільки вправо та вліво. Відбиває кульку, щоб та не впала на лінію знизу.
Line - лінія. Вона розташована знизу сцени. Якщо кулька торкається її, гра закінчується.
Button3 - цегла. Клони цього спрайта створюються та показуються на сцені. Основний спрайт прихований.

- Додайте **скрипт для сцени**. Він встановлює початкове тло.



- Додайте тло **gameOver**, воно буде показуватися на сцені, коли гра закінчується. Створити красивий напис вам допоможе сайт https://cooltext.com/

КРОК 2: РАКЕТКА

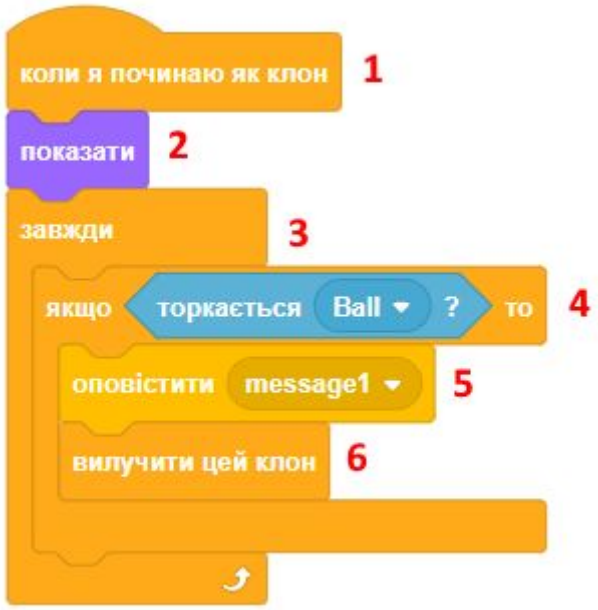
- Ракетка повинна рухатися вліво і вправо в межах сцени. На початку ще потрібно встановити стиль обертання ракетки.

1	Скрипт запускається на початку гри.
2	Встановлюємо стиль обертання ракетки. Нам не потрібно, щоб вона оберталася.
3	Ми під час гри виконуємо потрібні нам перевірки.
4	Перевірка, чи натиснута клавіша праворуч.
5	Якщо клавіша натиснута, нам потрібно збільшити X, тобто перемістити ракетку праворуч.
6	Перевірка, чи натиснута клавіша ліворуч.
7	Якщо клавіша натиснута, нам потрібно зменшити X, тобто перемістити ракетку ліворуч.

8	Треба перевірити, чи торкається ракетка межі, та змінити напрямок її руху в разі потреби.

КРОК 3: ЦЕГЛА

- Цей **скрипт для цегли** під час гри перевіряє зіткнення з кулькою, прибирає цеглу зі сцени, а також посилає повідомлення, щоб кулька змінила напрямок руху.



1	Скрипт починає свою роботу тільки для клонів.
2	Основний спрайт завжди схований, тому нам треба показати клон на сцені.
3	Ми під час гри виконуємо потрібні нам перевірки.
4	Перевірка, чи торкається кулька цього клону цегли.
5	Якщо торкається, потрібно повідомити кульку, що їй треба змінити напрямок руху.
6	Кулька розбиває цеглину, тому нам цей клон більше не потрібен.

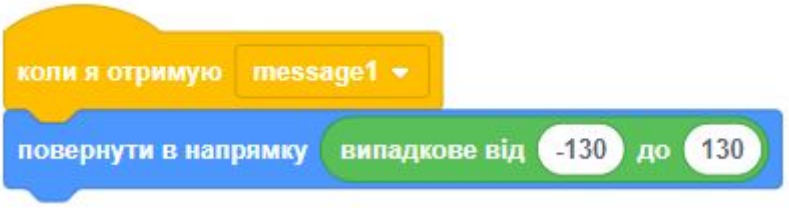
- Скрипт для зображення **цеглин** на сцені за допомогою створення клонів. Початкові координати і величини, на які змінюються X і У, залежать від розміру та кількості цеглин на сцені.

1	Скрипт запускається на початку гри.
2	Основний спрайт нам не потрібен, ми ховаємо його.
3	Задаємо початкове значення У для появи клонів цегли. Це координата цеглини, яка розташована зліва та зверху. Значення залежить від розміру цеглини та кількості клонів по вертикалі.
4	У нас буде 3 рядки цеглин по вертікалі. Змініть це значення, якщо вам потрібно більше або менше цеглин.
5	Задаємо початкове значення Х для появи клонів. Воно залежить від розміру цеглин та кількості клонів по горізонталі.
6	У нас буде 5 цеглин по горізонталі. Змінить параметри цього блоку, якщо вам потрібно більше або менше цеглин.
7	Створюємо клон цеглини з Х та У основного спрайта.
8	Чекаємо деякий час, щоб клони створювались поступово.

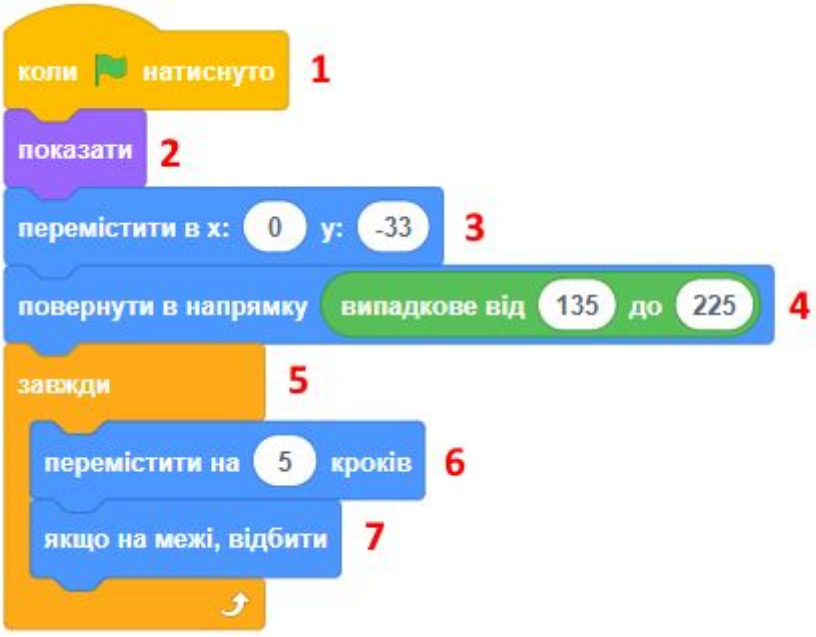
9	Змінюємо X на розмір цеглини, щоб клони не перекривали одне одного.
10	Змінюємо У на висоту цеглини, щоб додавати клони у наступний рядок цеглин.

КРОК 4: КУЛЬКА

- Скрипт змінює напрямок руху кульки, після зіткнення з цеглою.



- Кулька буде завжди літати по сцені. Також ми встановлюємо початкове положення кульки та випадковий напрямок, в якому вона буде рухатися на початку.



2	Нам треба показати кульку на початку гри.
3	Встановлюємо початкове положення кульки на сцені.
4	Встановлюємо випадковий напрямок, в якому кулька буде рухатися на початку гри.
5	Ми під час гри виконуємо потрібні нам дії.
6	Перемістити кульку у напрямку руху.

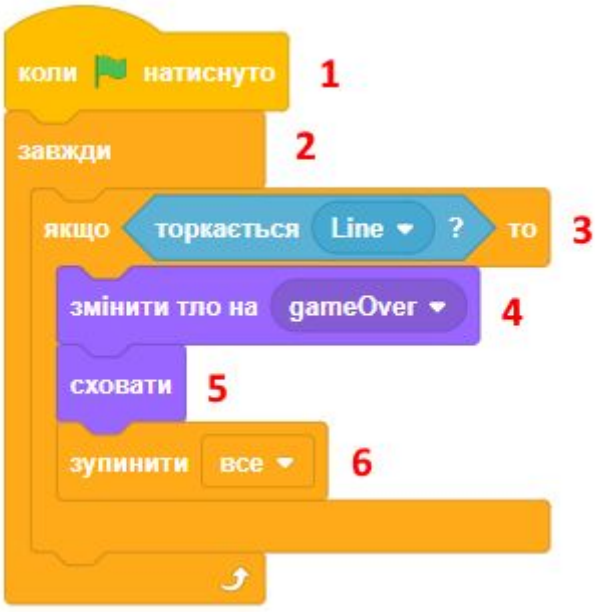
	Зменьшіть або збільшить параметр цього блоку, якщо вам потрібно змінити швидкість кульки.
7	Треба перевірити, чи торкається кулька межі, та змінити напрямок її руху, якщо потрібно.

- Цей скрипт перевіряє **торкання ракетки кулькою**.

2	Ми під час гри виконуємо потрібну нам перевірку.
3	Перевірка, чи торкається кулька ракетки.
4	Якщо кулька торкається ракетки, нам потрібно змінити напрямок її руху. Новий напрямок руху вибирається випадковим чином.
5	Нам потрібно почекати, поки кулька не відлетить від ракетки та не перестане її торкатися.

- Скрипт перевіряє **торкання кульки і лінії внизу**.

3	Перевірка, чи торкається кулька лінії внизу.
4	Якщо кулька торкається лінії, ми переключаємо тло на малюнок з написом Game Over.
5	В кінці гри нам потрібно сховати кульку.
6	Зупиняємо гру. Якщо хочете зробити гру нескінченною, цей блок ви маєте замінити на інші.



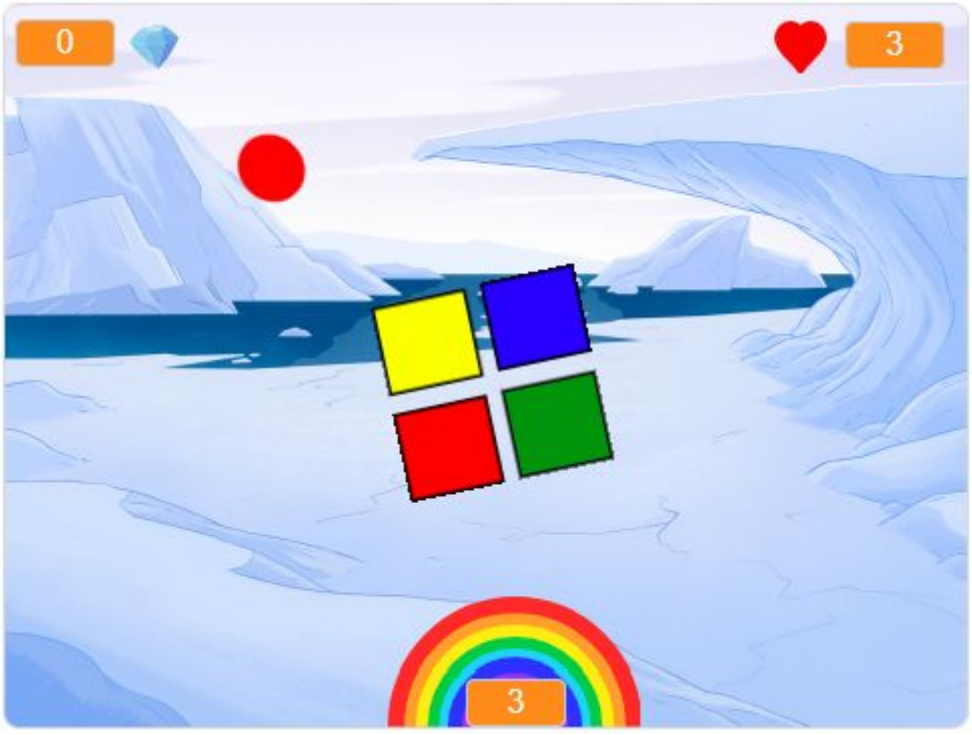
КРОК 5: ЗАВДАННЯ

- Зробіть гру нескінченною. Щоб коли всі цеглини пропали після зіткнення з кулькою, вони знову з'являлися на сцені.
- Змінюйте образ кульки, коли вона стикається з цеглою.
- Зробіть цеглу різних кольорів і показуйте їх випадковим чином.
- Якщо кулька потрапляє в цеглину червоного кольору, змініть колір цегли на звичайний, але не ховайте цеглу зі сцени.
- Рахуйте кількість збитих цеглин, показуйте це число на сцені.
- Зробіть так, щоб у ракетки було три життя. Якщо не всі життя витрачені, гру не зупиняти. При попаданні кульки в лінію знизу, віднімати одне життя. Показуйте кількість життів на сцені.
- Додайте спрайти, які будуть рухатися випадковим чином по сцені. Ракетка повинна їх теж збивати.
- Зробіть спеціальні цеглини іншого кольору, які прискорюють і сповільнюють рух кульки та ракетки.
- Зробіть спеціальні цеглини іншого кольору, які збільшують і зменшують розмір ракетки.
- Змінюйте тло сцени через деякий проміжок часу.
- На початку гри показуйте назву гри та інструкцію.
- Додайте звуки та музику до вашої гри.
- Подивіться гру арканоїд на смартфоні або комп'ютері, запозичте якусь ідею звідти.
- Що ще можна зробити? Запропонуйте своє поліпшення гри.

УРОК 10. ЛОВЕЦЬ ТОЧОК

Відео: https://youtu.be/s_dqsVWA9FI

У цьому проекті ми навчимося, як можна створити гру, в якій доведеться підібрати правильну частину контролера, щоб вона відповідала за кольором випадаючим кольоровим точкам.



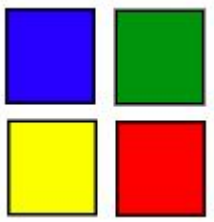
На цьому уроці ми будемо використовувати:

1. Змінні.
2. Списки.
3. Створення власних блоків.

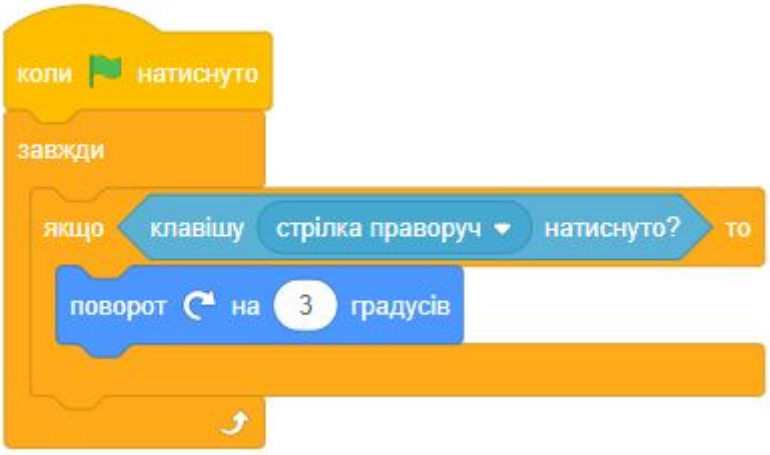
КРОК 1: СТВОРЕННЯ КОНТРОЛЕРА

Давайте почнемо зі створення контролера. Це такий круглий або квадратний різнокольоровий об'єкт, який буде збирати точки.

- Створіть новий проект, видаліть кота, щоб проект став порожнім.
- Намалюйте спрайт контролера. Кожна частина контролера має свій колір (синій, зелений, жовтий, красний). Наведіть спрайт так, щоб він виявився в центрі сцени.



- Повернемо контролер вправо, коли натиснута СТРІЛКА ВПРАВО. Додайте цей скрипт до контролера та протестуйте, що вийшло.
- Контролер повинен повертатися вправо після натискання на стрілку вправо.



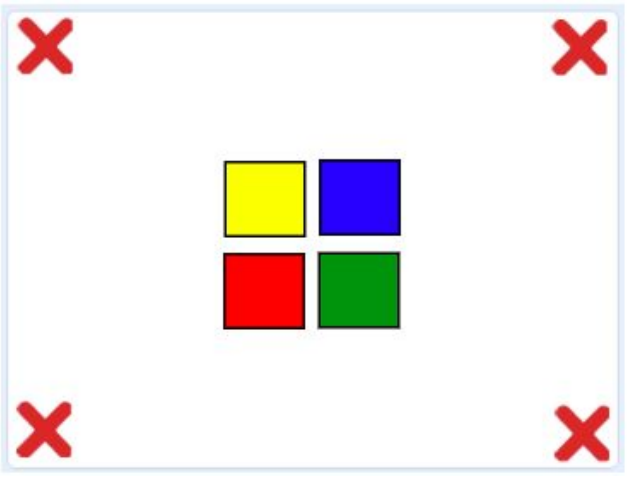
- **Завдання.** Чи можете ви змусити контролер крутитися вліво, якщо натиснута СТРІЛКА ВЛІВО?

КРОК 2: ЗБИРАННЯ ТОЧОК

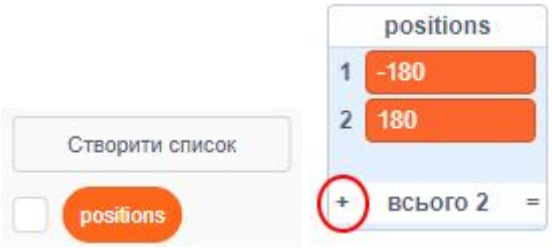
Додаємо в гру точки, які будуть збиратися контролером. Створіть новий спрайт, назвавши його **red** (червоний). Намалюйте образ для спрайта, це буде маленьке червоне коло.

- Додайте **скрипт для червоної точки**, який буде створювати новий клон спрайту через певну кількість секунд.

- Клон повинен з'являтися в одному з чотирьох кутів сцени.

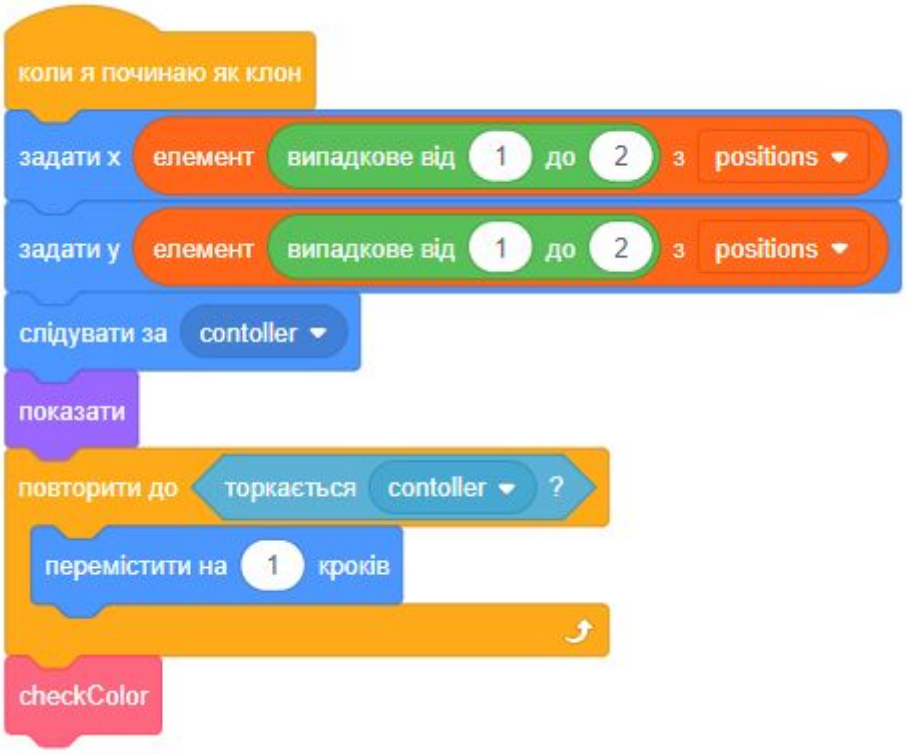


- Щоб це запрограмувати, **створіть новий список**, назвіть його **positions** (позиції) і натисніть кнопку (+), щоб додати до списку значення -180 і 180.



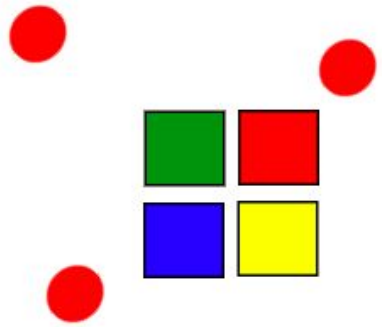
Тепер можна використовувати наш список, щоб випадково вибрати будь-який з чотирьох кутів сцени.

- Додайте **скрипт для спрайта точки.** Кожен новий клон з'являється в кутку сцени і рухається у напрямку до контролера.

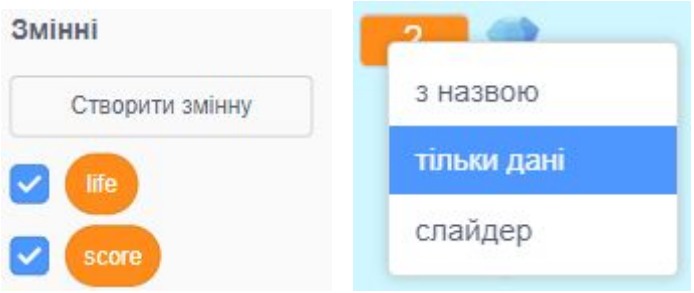


- Блок **checkColor** буде перевіряти, чи спіймали ми точку правильною стороною контролера. Створіть цей блок у категорії МОЇ БЛОКИ. Визначення цього блоку поки залиште порожнім.
- Наведений вище скрипт вибирає значення -180 або 180 для позиції по осях x і y. Таким чином кожен клон починає рух з одного із чотирьох кутів сцени.

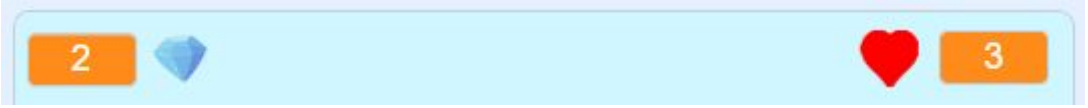
- Протестуйте проект. Ми повинні побачити купу червоних точок, що з'являються в різних кутах сцени і рухаються повільно у напрямку до контролера.



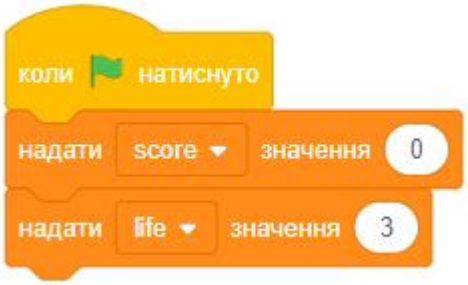
- **Створіть дві нові змінні** з назвою **life** (життя) і **score** (рахунок). Покажіть змінні на сцені в режимі - ТІЛЬКИ ДАНІ.



- Розмістить змінні рахунку та життя зверху у протилежних кутках сцени. Додайте ще маленьки спрайти поруч зі змінними, щоб гравцю було все легко зрозуміти.



- Додайте **скрипт в сцену**, щоб надати початкові значення змінних, коли гра запускається. Початкова кількість життів - на ваш розсуд, це може бути 3, 5 або інше число.



- Додайте визначення для блоку **checkColor** спрайта червоної точки. В блоці ТОРКАЄТЬСЯ КОЛЬОРУ точно вкажіть колір частини контролера за допомогою піпетки. Це важливо!

- В скрипті ми додаємо одиницю до рахунку, якщо збігаються кольори точки та частини контролера, якої вона торкається. Ще ми зменшуємо кількість життів, якщо кольори не збігаються.

- Додайте в **скрипт для сцени** нові блоки таким чином, щоб гра зупинялась, як тільки кількість життів дорівнює нулю:

- Протестуйте гру, щоб упевнитися, що все працює.

КРОК 3: БІЛЬШЕ ТОЧОК

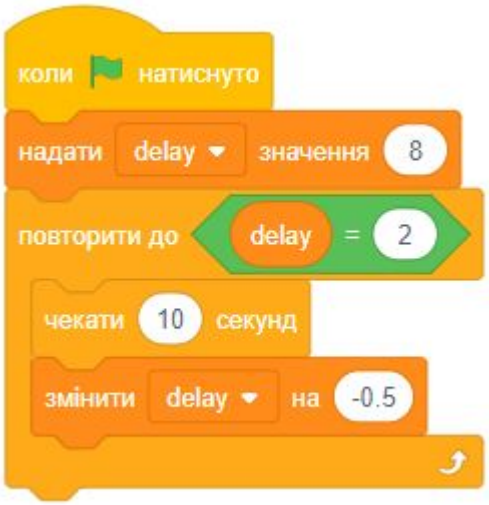
- Продублюйте двічі спрайт червоної точки і назвіть ці два спрайта, відповідно yellow (жовтий) і blue (синій). Також ви можете зробити green (зелений) спрайт.

- Відредагуйте створені спрайти разом із скриптами всередині них. Важливо, щоб колір точки збігався з кольором однієї з частин контролера. Вам потрібно відредагувати колір у блоках ТОРКАЄТЬСЯ КОЛЬОРУ.
- Перевірте проект, щоб впевнитись, що ви заробляєте бали, коли кольори однакові, або втрачаєте життя, коли вони не збігаються.
- Також необхідно впевнитися в тому, що гра не дуже складна і одночасно не дуже легка!

КРОК 4: ЗБІЛЬШУЄМО СКЛАДНІСТЬ

Давайте трохи ускладнимо гру. Чим довше йде гра, тим коротшим буде проміжок часу для появи кольорових точок.

- Створіть нову змінну з ім'ям **delay** (затримка).
- Додайте **скрипт для сцени**, який надає змінній **delay** початкове значення (досить велике число), а потім буде зменшувати її.



- Нарешті, ми можемо використовувати змінну **delay** в скриптах для наших червоної, жовтої і синьої точок.

- Видаліть блок, який чекає випадкову кількість секунд між створенням клонів кольорових точок, і замініть його на нову змінну **delay**.

- Протестуйте гру та переконайтесь, що все працює.
  - Чи зменшується поступово затримка при появі нових точок?
  - Чи все працює правильно для всіх кольорів точок?
  - Чи можете ви відобразити значення змінної **delay** на сцені?
  - Чи зменшується значення змінної **delay**?

КРОК 5: ТОЧКИ РУХАЮТЬСЯ ШВИДШЕ

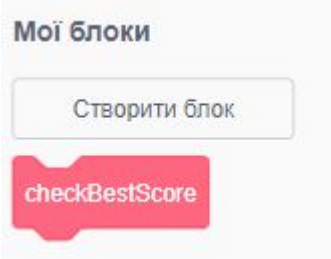
Давайте покращимо гру, додавши туди змінну **speed** (швидкість) і змусивши точки рухатися з часом все швидше і швидше! Це буде працювати так само, як і зі змінною **delay,** тому в скрипт можна підглянути і використати.

КРОК 6: КРАЩІ РЕЗУЛЬТАТИ

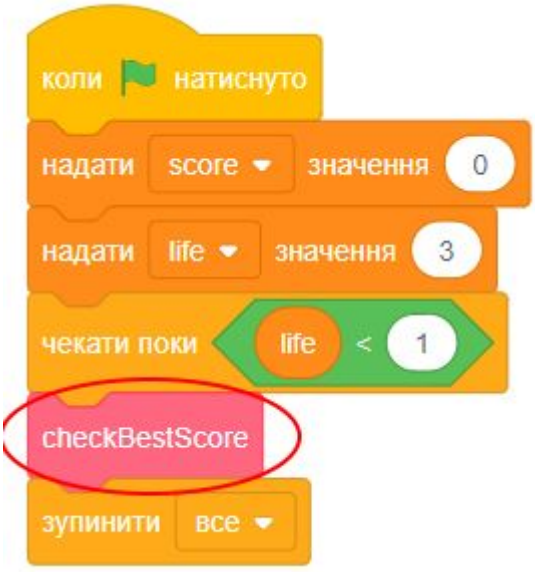
Давайте тепер зберігати найкращі результати, щоб гравці мали змогу побачити свій прогрес.

- Створіть нову змінну з ім'ям **bestScore** (найкращий рахунок).

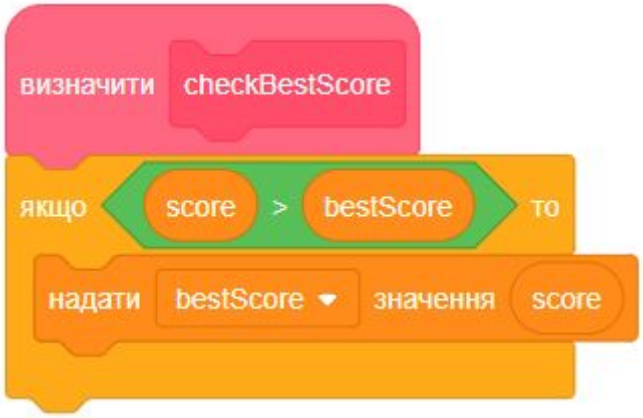
- Натисніть на **сцені** і створіть **новий користувальницький блок** під назвою **checkBestScore** (перевірити найкращий рахунок).



- Додайте виклик нового блоку **в кінці скрипта сцени** перед зупинкою гри.



- Додайте визначення цього блоку для надання поточного значення **score** в **bestScore,** якщо це були найвищі бали.



- Тепер перевіримо, чи все вийшло, як ми очікували. Пограйте кілька разів, щоб переконатися, що **bestScore** змінюється відповідно до того, як ви досягаєте кращих результатів.

КРОК 7: ДОДАТКОВІ ЗАВДАННЯ

- В кінці гри показуйте тло GAME OVER (кінець гри).
- Перед початком гри показуйте тло привітання з назвою гри, інструкціями та кнопкой BEGIN GAME (почати гру).
- Зробіть так, щоб кольорові точки летіли не з чотирьох кутів, а з випадкових місць на межах сцени. Це може бути зроблено не з початку гри, а коли гравець перейдете на складний рівень через деякий час.
- Зробіть так, щоб інколи до контролера летіли спеціальні точки, які додають йому життя.
- Зробіть підрахунок часу гри, та показуйте найдовший час гри, як ви це зробили з найкращим рахунком.
- Якщо ще не зробили точки зеленого кольору, додайте їх до гри.
- Клавіша ПРОПУСК має запускати якусь ракету, яка збіває точки. Кількість ракет обмежена і додається через деякий час з анімацією підльота космічного корабля з боєприпасами.
- Що ще можна зробити?

УРОК 11. ПЕРЕВІРКА ПАМ'ЯТІ

У цьому проекті ми створимо гру для перевірки пам'яті, в якій потрібно буде запам'ятовувати і повторювати послідовність кольорів.

КРОК 1: ВИПАДКОВІ КОЛЬОРИ

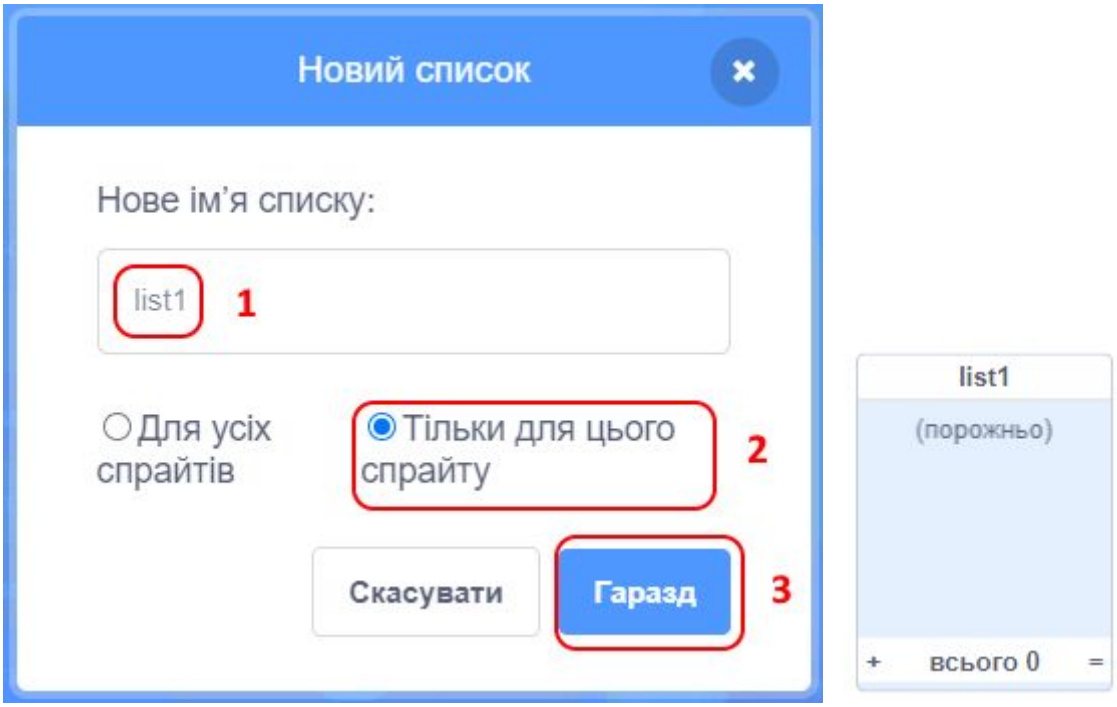
Для початку давайте створимо персонаж, який може змінювати послідовність кольорів для того, щоб гравець міг їх запам'ятати.

- Створюємо новий проект, видаляємо звідти кота.
- Виберіть персонаж і сцену з бібліотеки. Персонаж не обов'язково повинен бути людиною, але потрібно, щоб він міг змінювати свої кольори. Наприклад, нам підходить Casey:

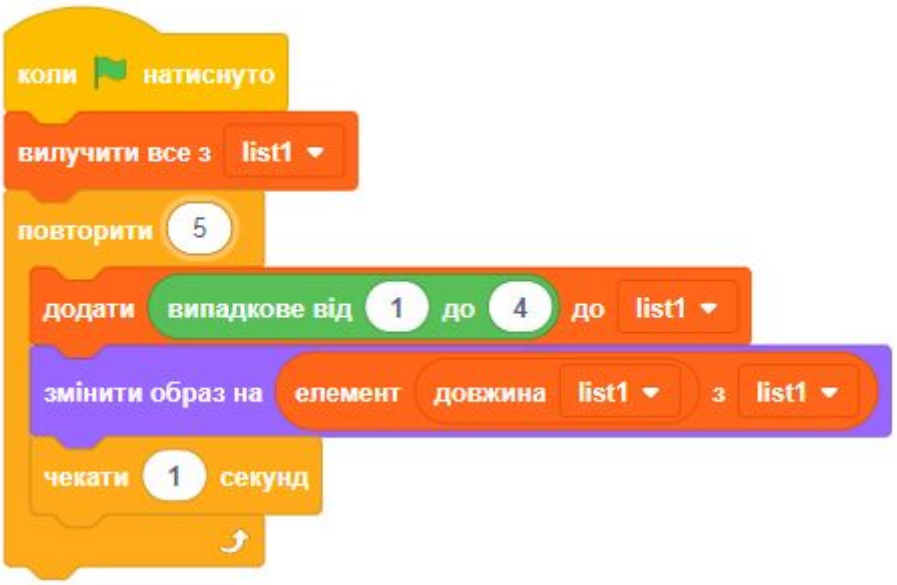
- У вашій грі кожна цифра відповідає певному кольору:
  - 1 = червоний (red).
  - 2 = синій (blue).
  - 3 = зелений (green).
  - 4 = жовтий (yellow).

- Додайте персонажу чотири образа, кожен з яких відповідає чотирьом кольорам, наведеним вище. Якщо у спрайта вже є потрібні образи, розфарбуйте їх. Переконайтеся, що кольори ваших образів розташовані в правильному порядку.

- Щоб додати випадкову послідовність, необхідно створити список. Список - це змінна, яка зберігає безліч даних по порядку. Створіть новий список **list1.** Змінна потрібна тільки для одного спрайта, треба відзначити 'Тільки для цього спрайта'.



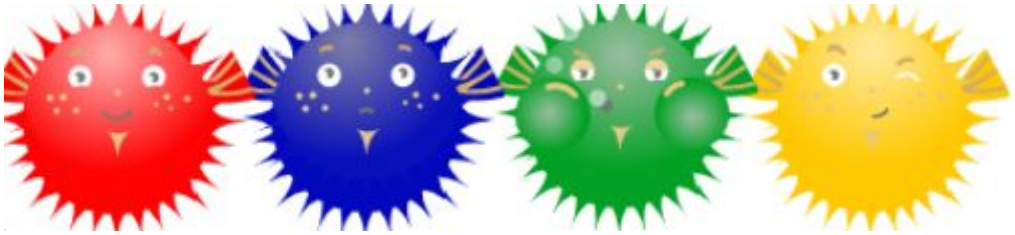
- Тепер ви маєте побачити порожній список у кутку сцени, а також блоки для використання списку на палітрі блоків.
- Щоб додати випадкове число до списку (і показати відповідний костюм) п'ять разів, додайте цей скрипт до спрайта. Зверніть увагу, що ви очистили список на початку гри.



КРОК 2: ПОВТОРЕННЯ ПОСЛІДОВНОСТІ

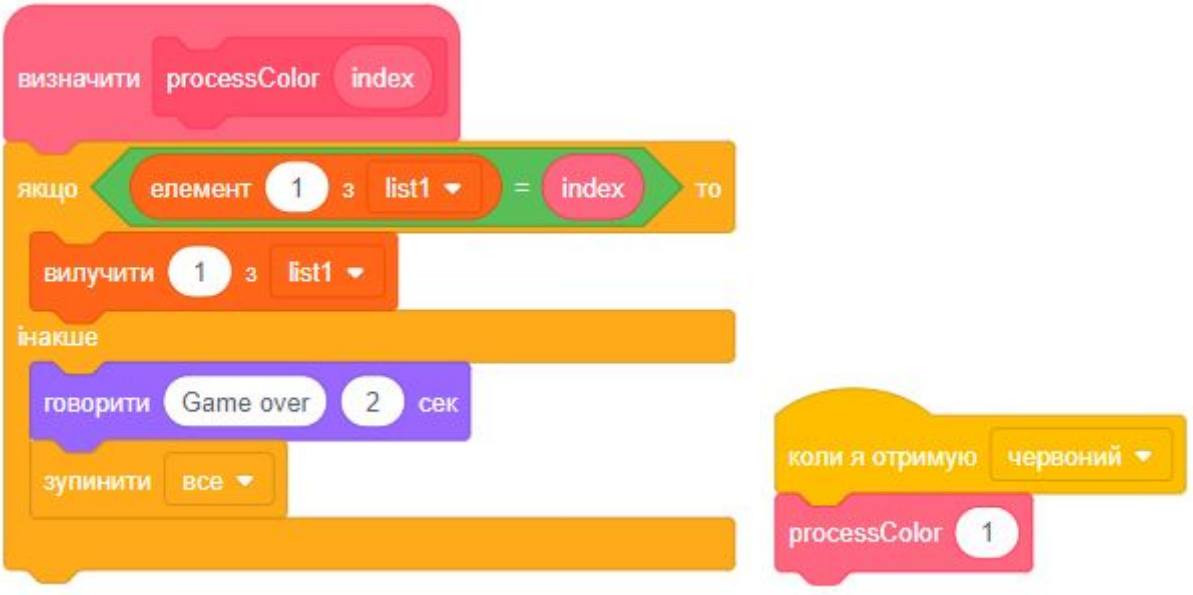
Давайте додамо чотири кнопки, за допомогою яких гравець зможе повторити показану послідовність кольорів.

- Додайте чотири спрайти, які будуть служити в якості кнопок. Відредагуйте кожен так, щоб він відповідав кольору.



- Коли натиснута червона кнопка, нам потрібно передати повідомлення, яке сповіщає про це. Додайте скрипт для **червоної кнопки**:

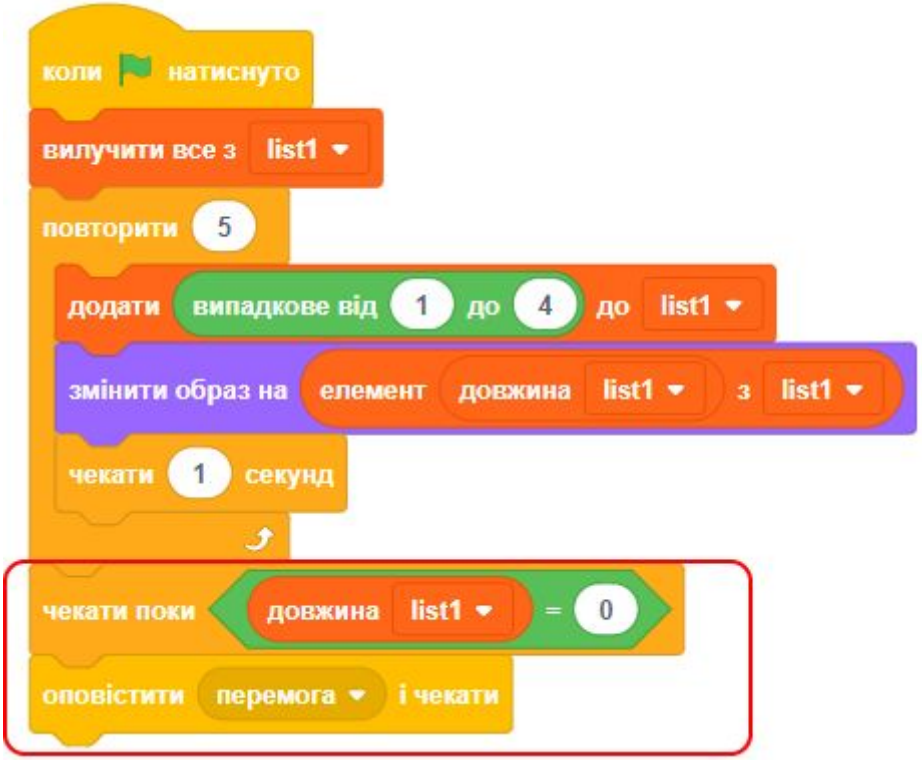
- Коли **персонаж** отримує повідомлення, він повинен перевірити, а чи дійсно номер 1 був на початку списку (тобто в послідовності показаних кольорів червоний був першим). Якщо так, ми можемо видалити цей номер зі списку. Якщо ні, ну що ж - кінець гри!



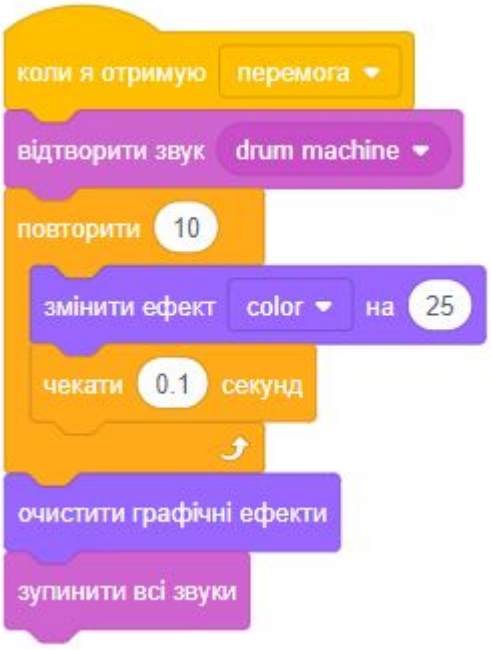
- Ми скористалися власним блоком **processColor** з параметром, в якому вказується номер кольору для опрацювання.

- Повторіть ці ж кроки для синьої, зеленої і жовтої кнопок. Ви можете також додати звук при натисканні на кнопки.
- Не забувайте перевірити гру, як тільки все зроблено! Чи можете ви запам'ятати п'ять кольорів? Послідовність різна кожен раз?

- Якщо раптом трапилося, що гравець вгадав правильно всю послідовність кольорів, можна яскраво сповістити його про перемогу. Це станеться тоді, коли наш список стане порожнім. Додайте цей код в кінець скрипта КОЛИ НАТИСНУТО ЗЕЛЕНИЙ ПРАПОР для нашого персонажа:



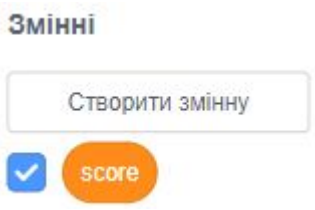
- Додайте скрипт **сцені**, щоб тимчасово змінити тло, коли гравець виграв.



КРОК 3: БАГАТО РІВНІВ

Запам'ятати п'ять кольорів - це будь-кому під силу. Давайте трохи покращимо гру таким чином, щоб довжина послідовності зростала.

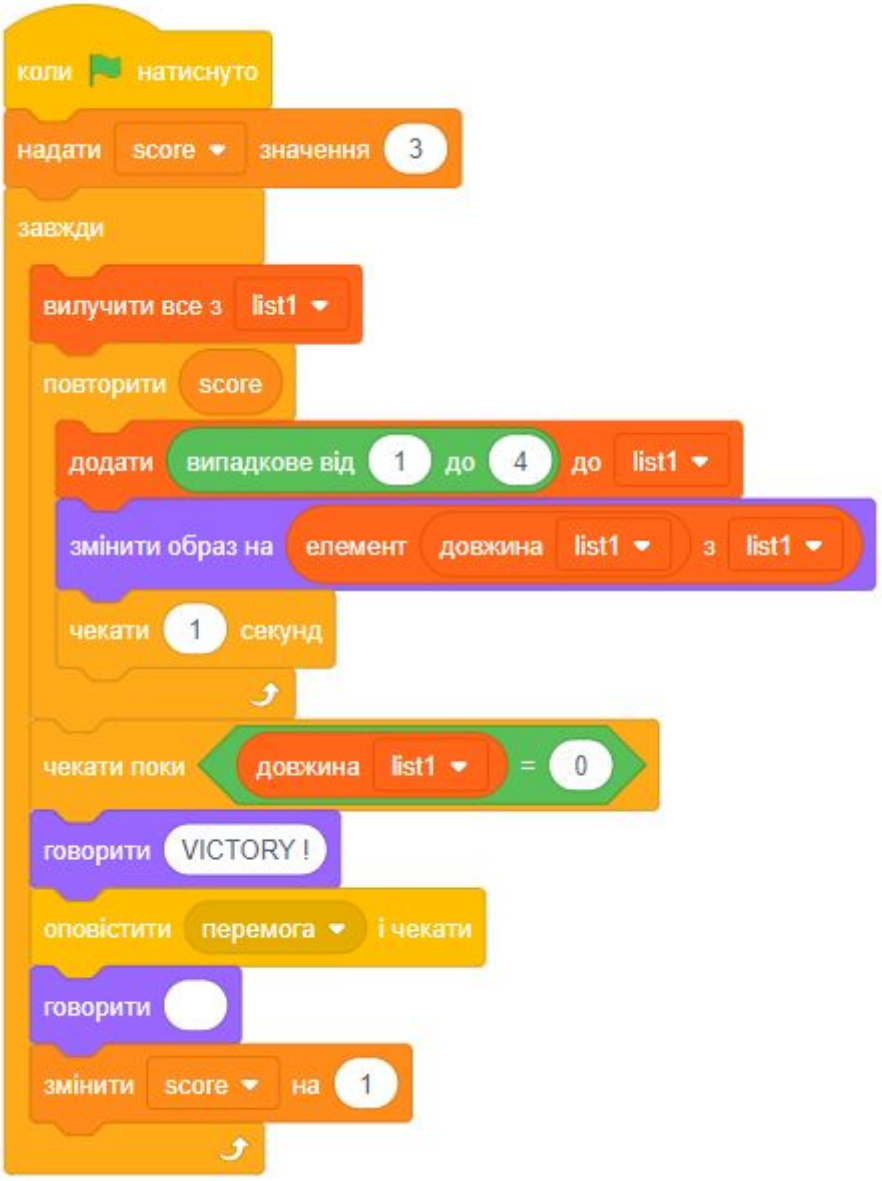
- Створіть нову змінну і назвіть її **score** (рахунок).



- Ця змінна буде використовуватися для визначення довжини послідовності, яку гравець повинен запам'ятати. З самого початку нехай наша довжина послідовності (і значення змінної) становитиме **три**. Замість того, щоб завжди створювати послідовність з п'яти кольорів, тепер для визначення довжини ми будемо посилатися на змінну **score**.

- Якщо вся послідовність вгадана правильно, ми на наступному етапі додаємо одиницю до змінної **score**. Таким чином ми поступово збільшуємо кількість кольорів у послідовності.

- Нарешті, необхідно додати цикл ЗАВЖДИ, обгорнувши ним код для створення послідовності, так щоб нова послідовність створювалася для кожного рівня. Ось як тепер повинен виглядати скрипт для персонажа:



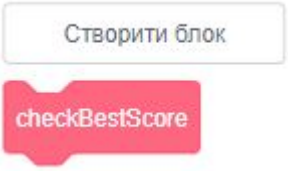
- Покличте друзів і перевірте, наскільки гарна у них пам'ять. І ваша гра. Сховайте список, перш ніж вони почнуть грати!

КРОК 4: КРАЩІ РЕЗУЛЬТАТИ

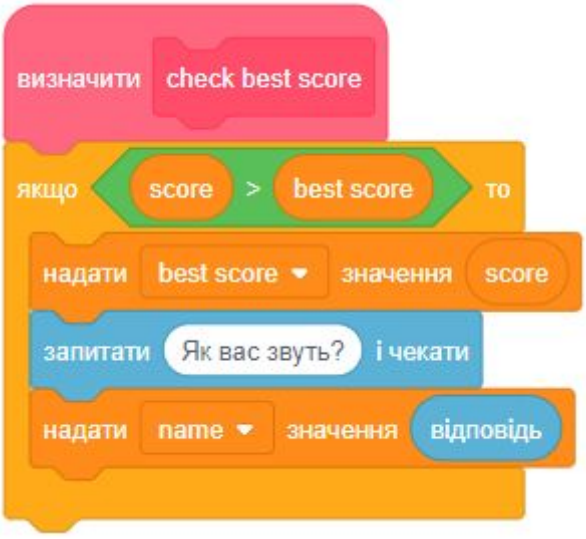
Давайте тепер будемо зберігати рекордні бали.

- Додайте дві нові змінні в проект:
bestScore (найкращий рахунок) і **name** (ім'я).

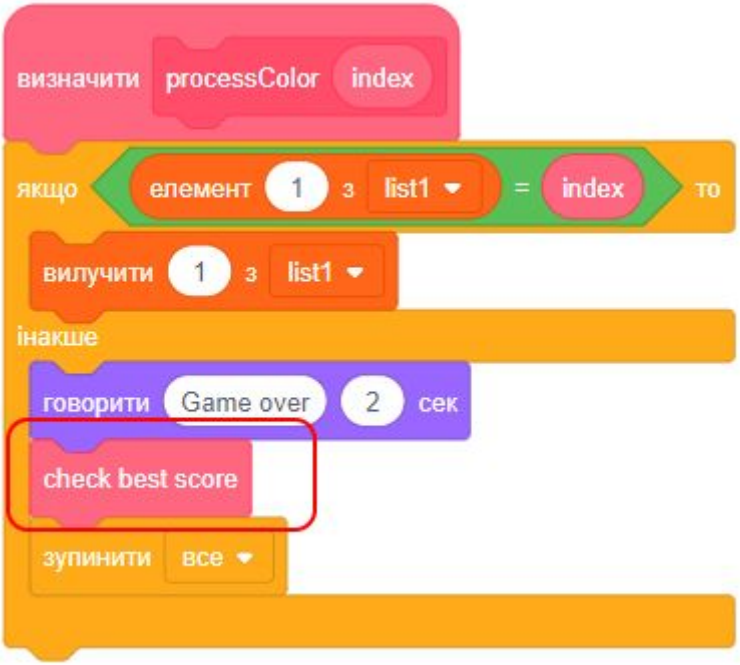
- Якщо гра закінчується (коли не вгадали колір), нам треба буде перевірити, чи перевищують результати гравця найвищий на даний момент бал. Якщо так, нам потрібно зберегти ім'я гравця та новий рахунок як найвищий бал.
- Створіть свій власний блок **checkBestScore** (перевірити найкращий рахунок).



- Ось як повинен виглядати скрипт:



- Ще потрібно додати його виклик до блоку **processColor**:



КРОК 5: ЗАВДАННЯ

ЗАВДАННЯ: ДОДАВАННЯ ЗВУКУ

Перевірте ваш проект кілька разів. Можливо, ви помітили, що один і той же номер трапляється два (або більше) рази поспіль, через це запам'ятати послідовність складніше. Ви можете зробити так, щоб кожен раз, коли персонаж змінює образ, звучали барабани? Як щодо того, щоб програвався різний звук в залежності від обраного числа? Це буде схоже на код, який змінює образ персонажа.

ЗАВДАННЯ: ІНШІ КОСТЮМИ

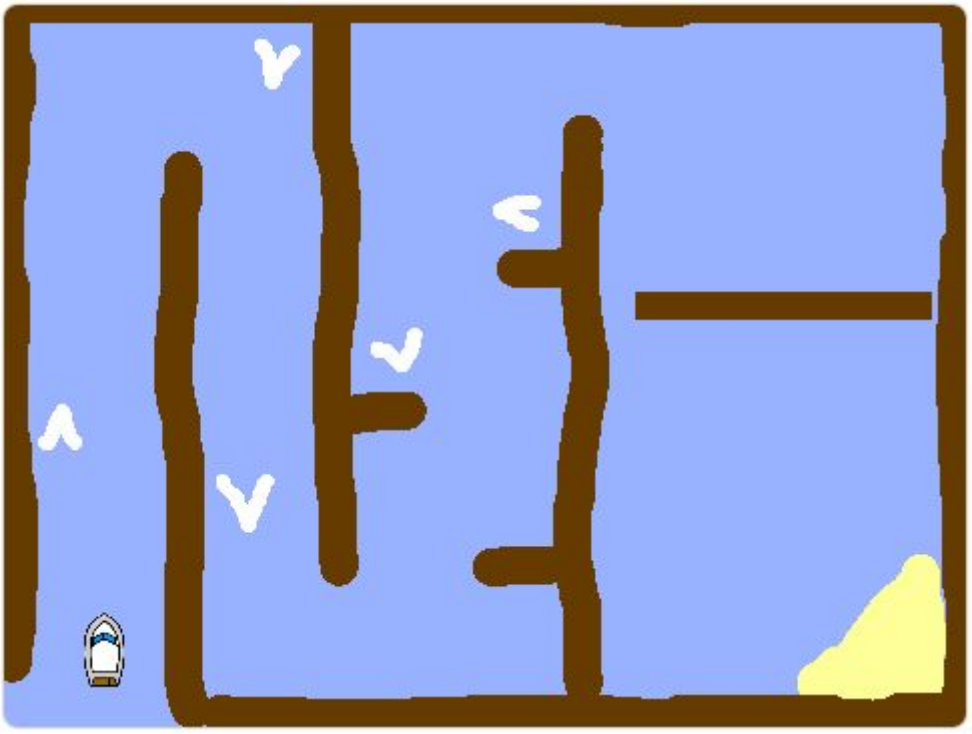
Чи помітили ви, що на початку гри з'являється один з чотирьох кольорів, який відображався останнім, коли для гравця була актуальна послідовність? Чи можете ви додати ще один образ (нейтральний) для персонажа, який буде відображатися на старті гри і коли гравець повторює послідовність на кнопках?

ЗАВДАННЯ: УСКЛАДНЕННЯ РІВНІВ

Чи можете ви дати гравцеві можливість вибрати між рівнями складності гри: легкий (використовуються червоний і синій кольори), і звичайний (використовуються всі 4 кольори)? Чи можете ви додати ще один (5) колір?

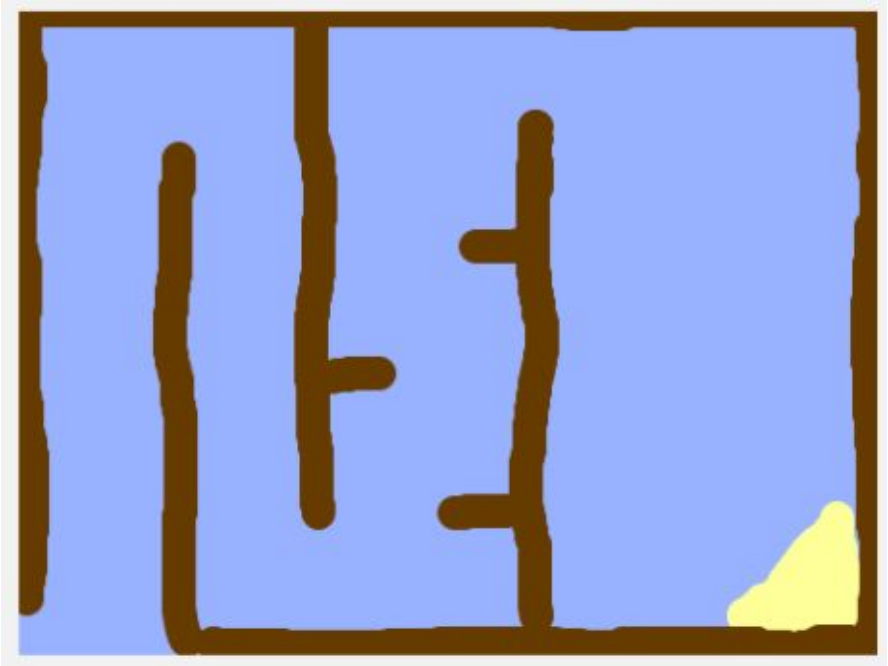
УРОК 12. ГОНКА ЧОВНІВ

Розробимо гру, в якій ми можемо стати капітаном човна, що пливе до безлюдного острова!

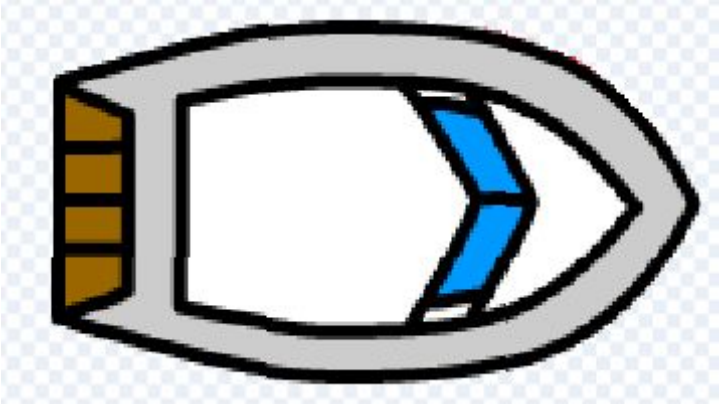


КРОК 1: СПРАЙТИ ТА СЦЕНА

- Створюємо новий проект, видаляємо звідти кота.
- Натисніть на сцені і сплануйте рівень гри. Ви повинні додати:
    - Сміття і колоди, які плавають і заважають човну. Ваш човен повинен уникнути зіткнення з ними.
    - Острів, до якого човен повинен буде дістатися.

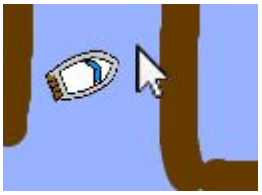


- Намалюйте човен і розмістіть його у початкову позицію.

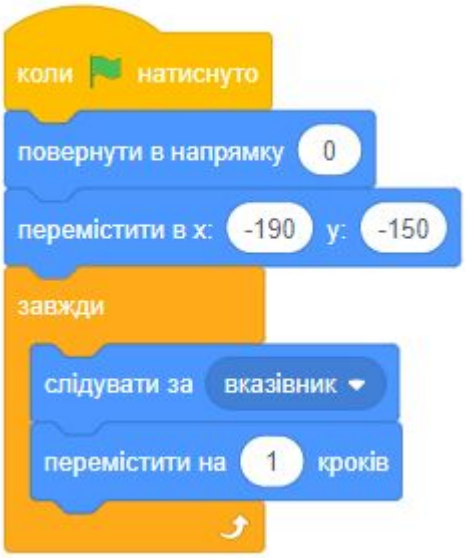


КРОК 2: УПРАВЛІННЯ ЧОВНОМ

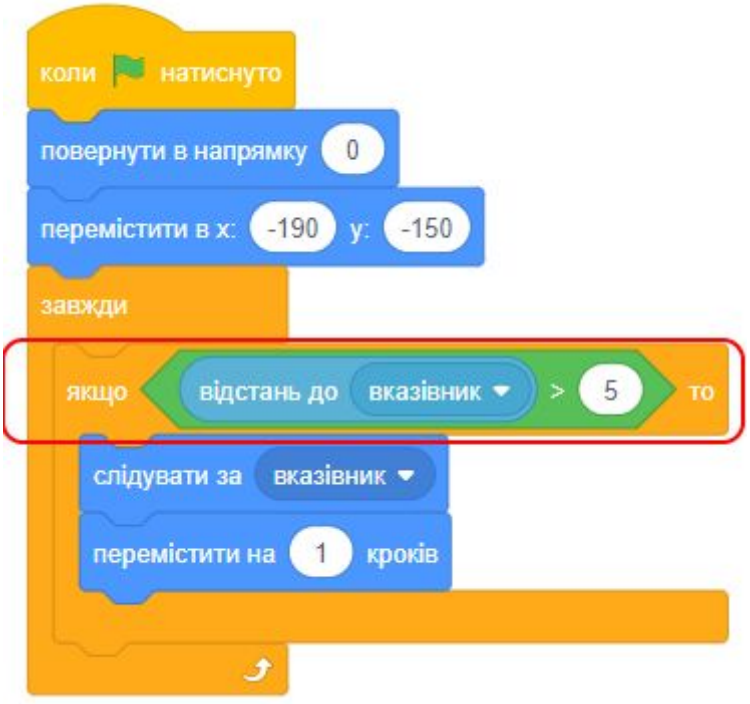
- Човном ми будемо управляти за допомогою мишки.



- Додайте такий скрипт для спрайта **човна**:



- Протестуйте човен, натиснувши на зелений прапорець. Посовайте мишкою. Човен пливе в напрямку вказівника мишки? Що відбувається, коли човен досягає вказівника мишки? Жах відбувається. Щоб це виправити, необхідно додати блок ЯКЩО і вказати в умові, щоб човен рухався тільки тоді, коли відстань від мишки більше 5 пікселів. Перевірте, чи вирішена ця проблема.



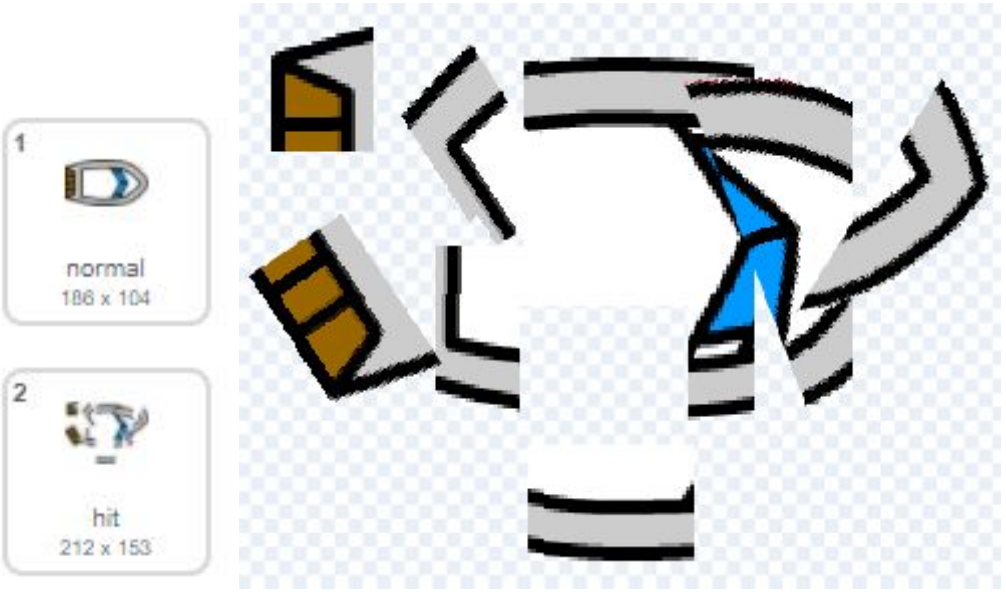
КРОК 3: АВАРІЯ

Те, заради чого ми тут всі зібралися - видовище! Якщо човен чіпляється за колоду, він повинен одразу потонути.

Але поки що, схоже, наш човен на повітряних подушках і перелітає через всі колоди! Давайте це виправимо.

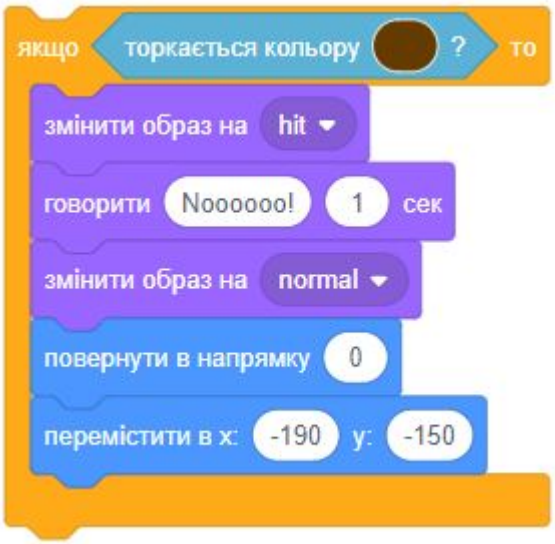
- Нам знадобляться два образи для човна, один звичайний і другий, коли човен тоне. Продублюйте звичайний образ і назвіть один - «normal», другий - «hit».
- Натисніть на «hit» образ, потім переключіться на растровий редактор зображень.

- Інструментом "Вибрати" виділяйте дрібні шматочки човна і перетягуйте їх у різні боки, повертаючи на різні кути. Спрайт повинен виглядати, ніби човен розвалюється на частини.



- Коли човен зачіпає колоду, повинна статися аварія.

- Додайте код для **човна** всередині циклу ЗАВЖДИ. У нас весь час буде відбуватися перевірка, чи торкається човен колоди.

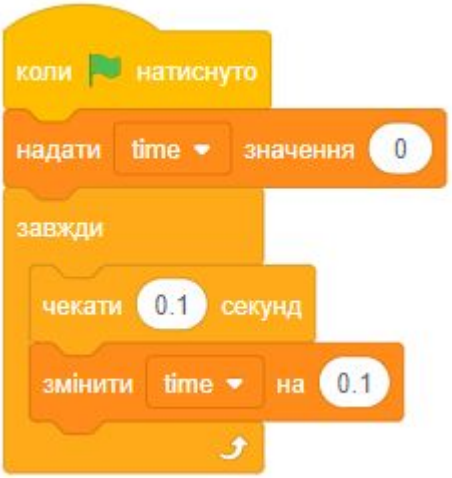


- Також ми повинні переконатися, що під час старту гри механіки все полагодили, і ми стартуємо на новенькому блискучому, без жодної дірки човні, тобто повинен бути обраний образ"boat".
- Тепер ми точно не зможемо перестрибнути через колоди, ви побачите, як відбувається трагедія, всі тонуть, а потім повертаються на старт.

КРОК 4: ГОНКИ НА ЧАС

Зараз у нас гравець по дорозі до острова милується заходом сонця. Давайте додамо таймер у гру таким чином, щоб гравець плив до острова якомога швидше.

- Додайте нову змінну **time** (час) для сцени. Можна також змінити відображення вашої нової змінної.
- Додайте в **сцену** скрипт, який буде відраховувати таймер до тих пір, поки човен не досягне острова.

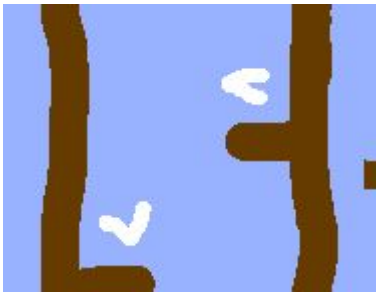


- Відмінно! Протестуйте гру і перевірте, наскільки швидко ви можете тепер дістатися до острова!

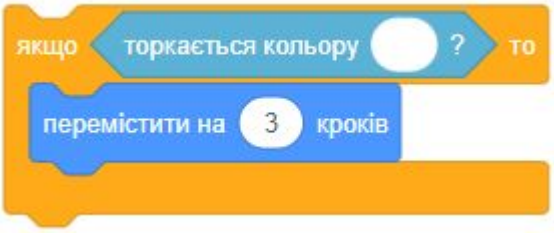
КРОК 5: ПЕРЕШКОДИ ТА ПРИСКОРЮВАЧІ

Наша гра зараз дуже легка. Додайте кілька штрихів, щоб зробити її більш цікавою.

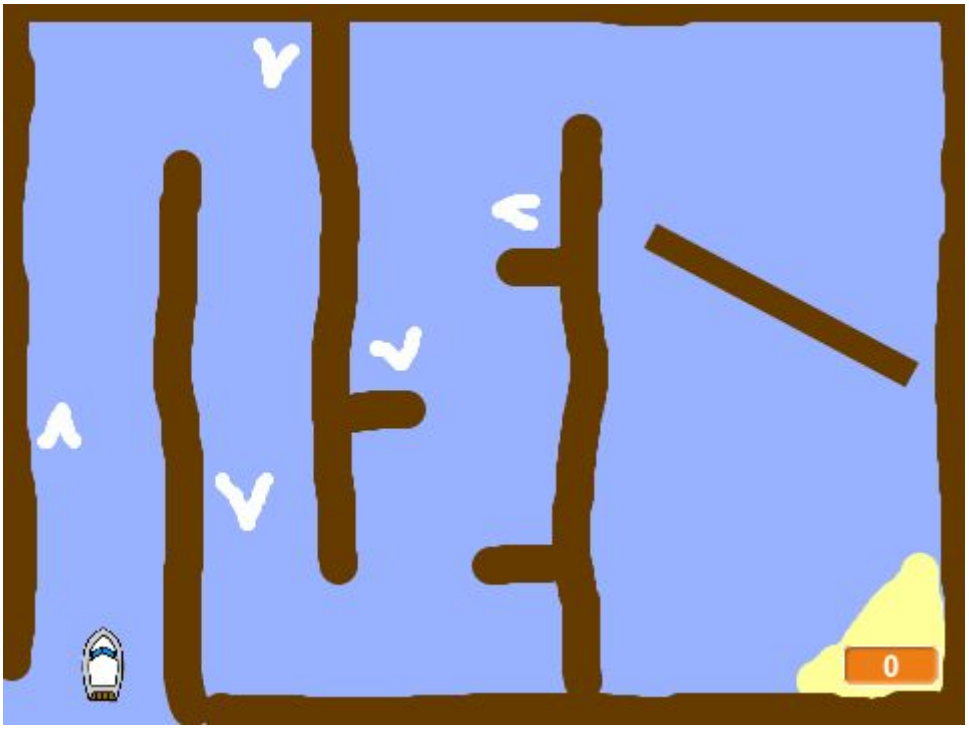
- Спершу додайте так звані прискорювачі, які будуть підштовхувати човен. На річці часто зустрічаються вири, які можуть таке зробити. Зайдіть на тло сцени і додайте білі стрілки.



- Тепер можна додати трохи ходу для спрайта човна в циклі ЗАВЖДИ. Коли він рухається і натикається на білий колір, то робить **3 додаткових** кроки.

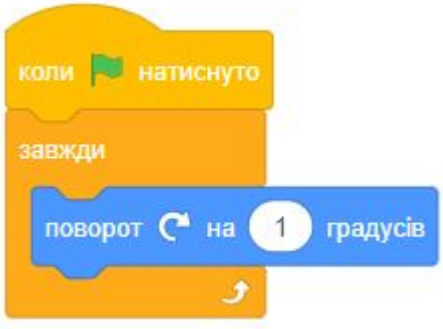


- Тепер ще одна перешкода: крутяться ворота! Ваш човен повинен уникнути зіткнення з ними!



- Додайте новий спрайт «ворота», який буде виглядати так:

- Зробіть ворота такого ж кольору, як і колоди.
- Встановіть центр для спрайта воріт.
- Додайте скрипт для воріт, який їх буде повільно крутити за допомогою циклу ЗАВЖДИ. Протестуйте гру. Тепер у вас цілий набір перешкод, яких ми повинні побоюватися.

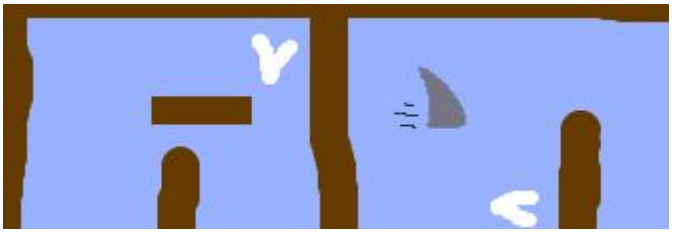


КРОК 6: ЗАВДАННЯ

ЗАВДАННЯ: БІЛЬШЕ ПЕРЕШКОД

- На тлі у воду можна додати трохи зеленої рідини, яка буде сповільнювати гравця, як тільки він туди потрапить. Для цього можна використовувати блок ЧЕКАТИ:

- Ви також можете додати рухомі об'єкти - акулу і колоди, що плавають по воді!



ЗАВДАННЯ: ЗВУКОВІ ЕФЕКТИ

Тихо все якось відбувається. Давайте додамо трохи звукових ефектів: коли човен торкається колоди і коли досягає острова. Можна також додати фонову музику.

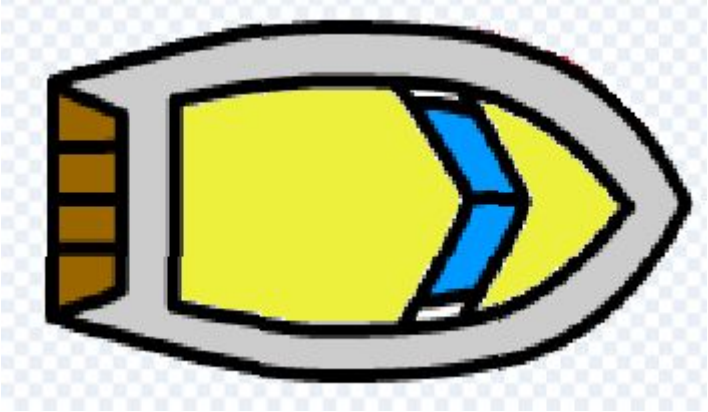
ЗАВДАННЯ: ПЕРЕМОГА

Зачекайте, поки ще не перемога, але ми до неї повільно та впевнено йдемо! Достатньо лише додати ще один оператор умови ЯКЩО в код до спрайта човна так, щоб гравець вигравав, як тільки він допливає до безлюдного острова. Острів у нас жовтого кольору - багато піску, жодного дерева. Як тільки човен доторкнеться до цього жовтого кольору - ми з радістю кричимо "Так!", і гра зупиняється.

ЗАВДАННЯ: БІЛЬШЕ ЧОВНІВ

Давайте перетворимо нашу гру в гонку для двох!

- Продублюйте човен і розфарбуйте в інший колір.



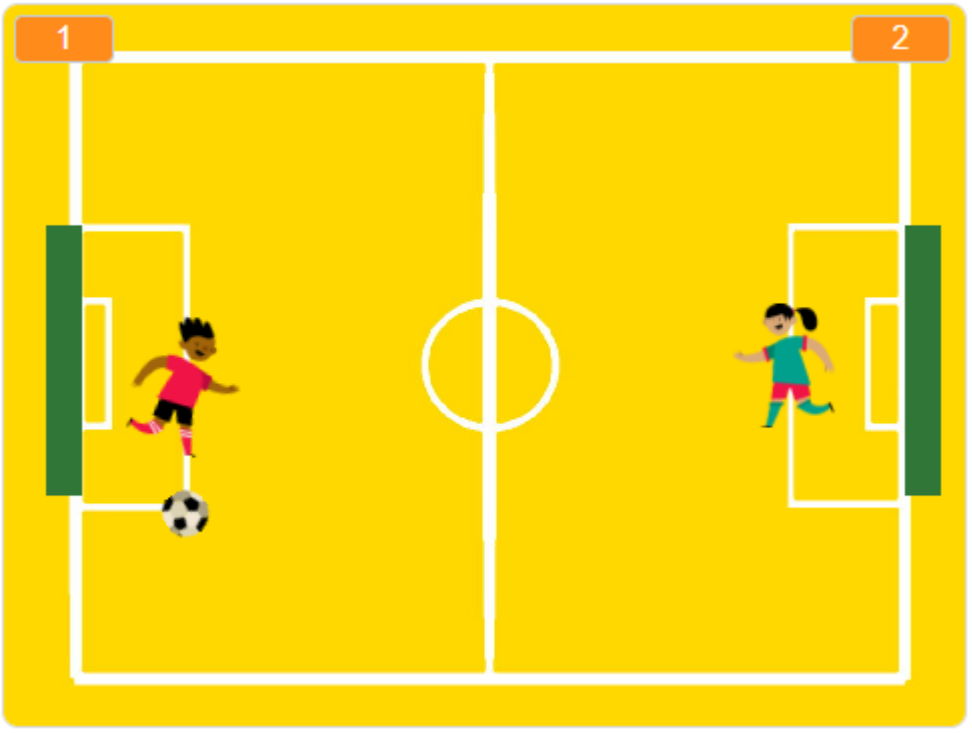
- Треба змінити стартову позицію другого гравця.
- Видаліть скрипт у нового гравця, який використовувався для керування човном за допомогою мишки. Замініть його на скрипт, який буде контролювати човен за допомогою клавіатури.

ЗАВДАННЯ: БІЛЬШЕ РІВНІВ

Чи можете ви створити додаткові рівні, намалювавши кілька фонів, а також дати гравцеві можливість вибирати між ними?

УРОК 13. ФУТБОЛ

Ми будемо програмувати гру для двох гравців, у якій вони керуватимуть головними героями та гратимуть у футбол.



На цьому уроці ми будемо використовувати:

1. Змінні.
2. Повідомлення.

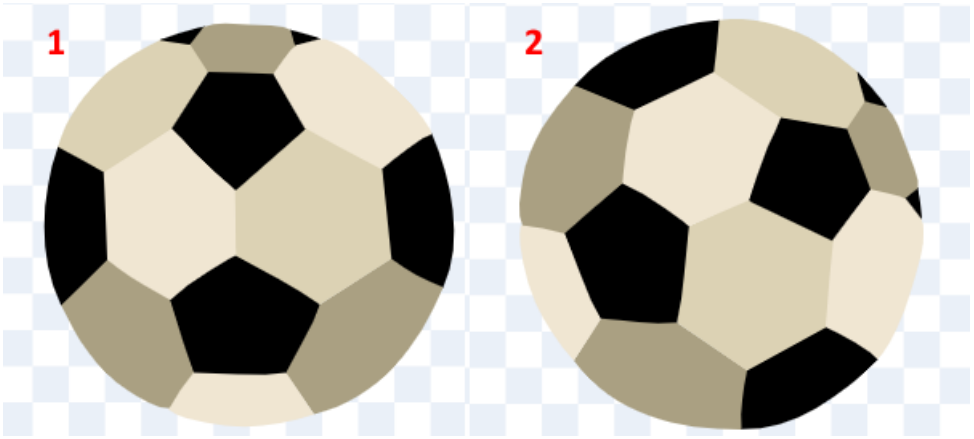
КРОК 1: СПРАЙТИ ТА СЦЕНА

- Почніть новий проект і видаліть з нього спрайт кота.
- Намалюйте або завантажте тло для вашої гри.
- Я знайшов безкоштовний малюнок футбольного поля через google. Треба перейти на пошук зображень, показати Інструменти, вибрати ліцензію Creative Commons (безкоштовні) та ввести пошуковий запит «soccer field» (футбольне поле).
- Додайте два спрайта футболістів та спрайт м'яча. Це можуть бути будь-які спрайти з бібліотеки. Наприклад:

- Назвіть правого футболіста **rightPlayer**, а лівого - **leftPlayer**.
- Якщо потрібно, ви можете повернути образи футболістів, щоб вони дивилися у центр поля.

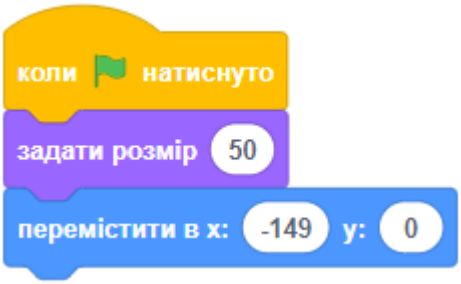
- Ще нам потрібно намалювати два спрайта воріт. Назвіть праві ворота **rightGoal**, ліві - **leftGoal**. Ворота можуть бути просто розфарбованими прямокутниками.

- Спрайту м'яч додайте другий образ з деякими відмінностями. Це можна зробити за допомогою дублювання першого образа і повороту на певний кут.

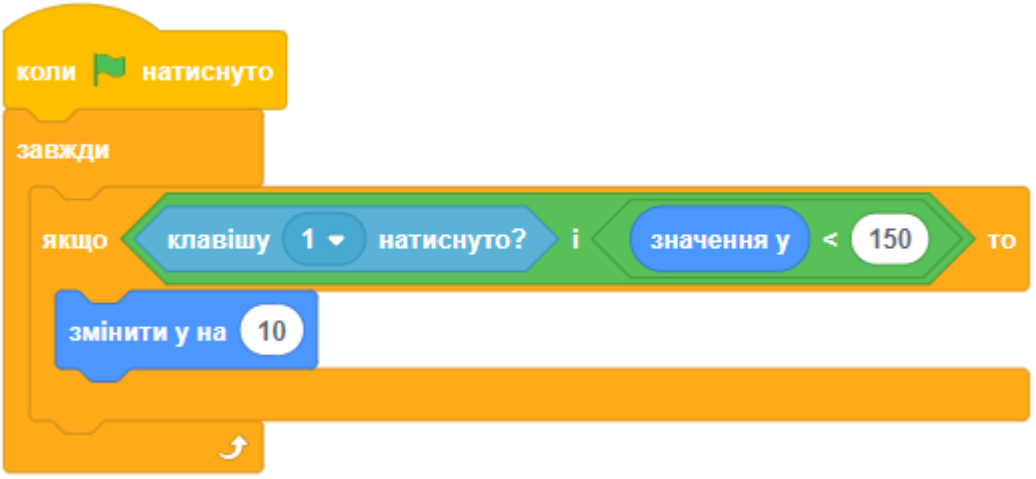


КРОК 2: ФУТБОЛІСТИ

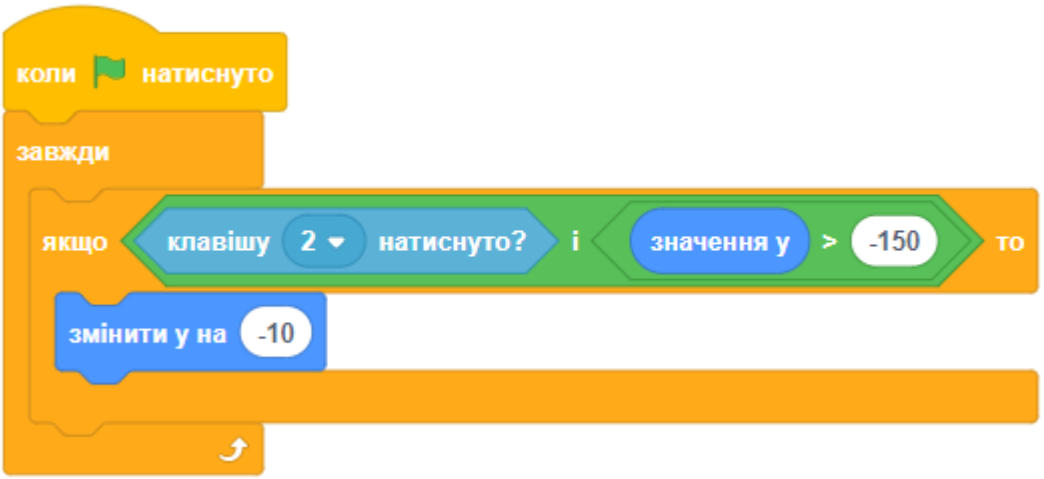
- Додайте скрипт, який перемістить **лівого футболіста** у початкове положення з лівої сторони ігрового поля та трохи зменшить його.



- Додайте скрипт для пересування лівого футболіста вгору.
Ми постійно перевіряємо, чи натиснута потрібна клавіша та чи футболіст не вийшов за межі сцени.



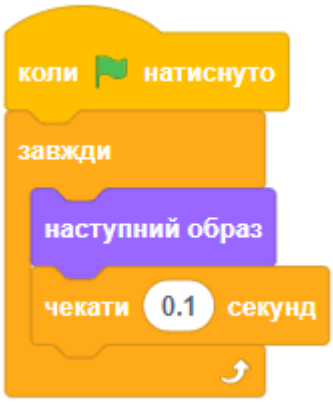
- Додайте скрипт для пересування лівого футболіста донизу.



- **ЗАВДАННЯ 1.** Спробуйте самостійно налаштувати спрайт правого футболіста, лише з тими відмінностями, що його стартова позиція – з правої сторони сцени, а керування відбувається за допомогою клавіш «стрілка вгору» та «стрілка вниз».
- Якщо вам це не вдалося – дивіться відповіді в кінці уроку.

КРОК 3: М’ЯЧ

- Перейдемо до програмування **м’яча**. Нам треба зробити так, щоб він постійно рухався по сцені з певною швидкістю, а гравці мали змогу відбивати його. Швидкість постійно зростає, а у разі забиття голу змінюється на початкову.
- Додайте анімацію **м’яча**. Для цього нам необхідно змінювати образи кожну десяту частину секунди.

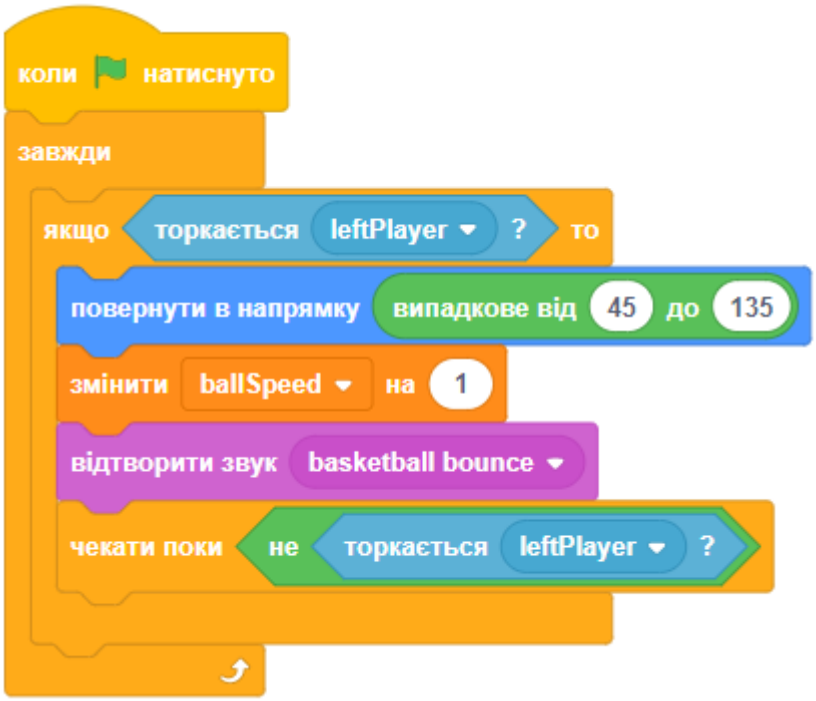


- Створіть змінну **ballSpeed** (швидкість м'яча).
- Додайте скрипт до **м’яча** для встановлення початкових значень. Скрипт надає початкові швидкість, розмір, напрямок та координати спрайта, переміщає його на передній план.

- Ще один скрипт програмує **м’яч** так, щоб він постійно рухався вперед з певною швидкістю та відскакував від меж сцени.

- Перейдемо до створення скриптів **м’яча**, які відповідають за відбиття м’яча гравцями.

- Спочатку запрограмуємо відбиття м’яча лівим футболістом. Додайте до **м’яча** такий скрипт.
    - За допомогою блоків ЗАВЖДИ, ЯКЩО та ТОРКАЄТЬСЯ постійно перевіряйте, чи не торкається м’яч лівого футболіста.
    - При відбитті м’яча він буде постійно прискорюватися, тому змініть значення змінної ballSpeed на одиницю. Ще нам треба змінити напрямок руху м’яча.
    - Відтворіть звук «удар по м’ячу».
    - Створіть затримку роботи скрипта, поки м’яч не перестане торкатися лівого футболіста. Це робиться для того, щоб гравець не змінив напрям руху м’яча декілька разів поспіль.



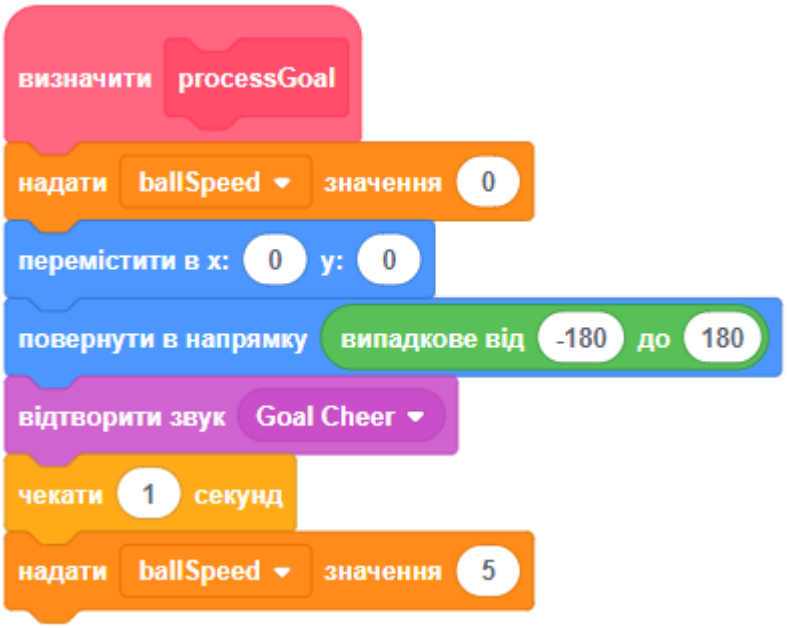
- **ЗАВДАННЯ 2.** Спробуйте самостійно створити ще один скрипт **м’яча**, який відповідатиме за відбиття від правого футболіста. Будьте уважні при обранні кутів відбиття.
- Якщо вам це не вдалося – дивіться відповіді в кінці уроку.

КРОК 4: ВЗАЄМОДІЯ ВОРІТ І М’ЯЧА

Нарешті нам треба запрограмувати ворота.

- Створіть дві змінні: **leftScore** (рахунок лівого футболіста) та **rightScore** (рахунок правого футболіста). Розмістить їх гарно на сцені в режимі відображення - тільки дані.
- Додайте скрипт **м’яча** для взаємодії з лівими воротами.

- У скрипті ми надаємо початкове значення змінної рахунку та виконуємо постійну перевірку, чи не торкається м’яч лівих воріт.
- Якщо м’яч доторкнувся до воріт, це означає, що правий футболіст забив лівому футболісту гол. Тому нам потрібно змінити рахунок на одиницю та викликати додатковий блок **processGoal**.
- Новий блок **processGoal** змінює швидкість м’яча на початкову, відтворює звук «забив гол» та переміщає м’яч у центр поля. Також ми очікуємо одну секунду перед початком нового раунду.

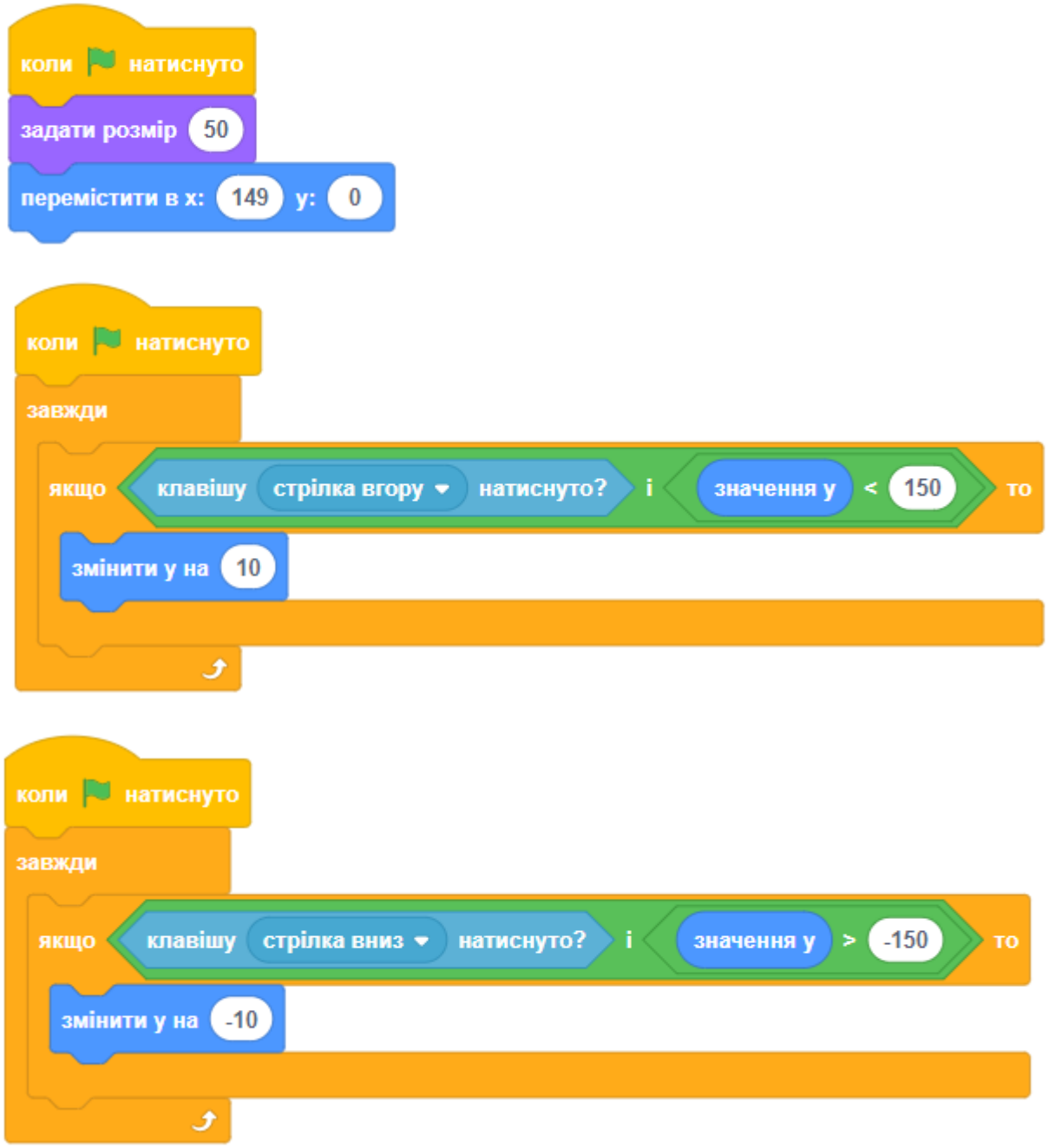


- **ЗАВДАННЯ 3.** Самостійно додайте скрипт **м’яча** для взаємодії з правими воротами. Якщо вам це не вдалося – дивіться відповіді в кінці уроку.

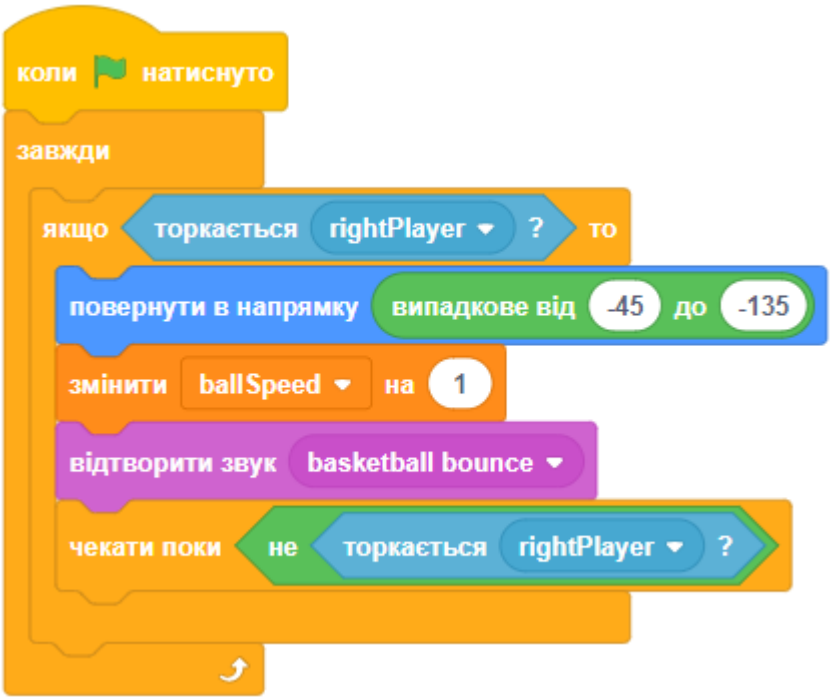
- Відмінно! Протестуйте гру і перевірте, як все працює.

ВІДПОВІДІ НА ЗАВДАННЯ

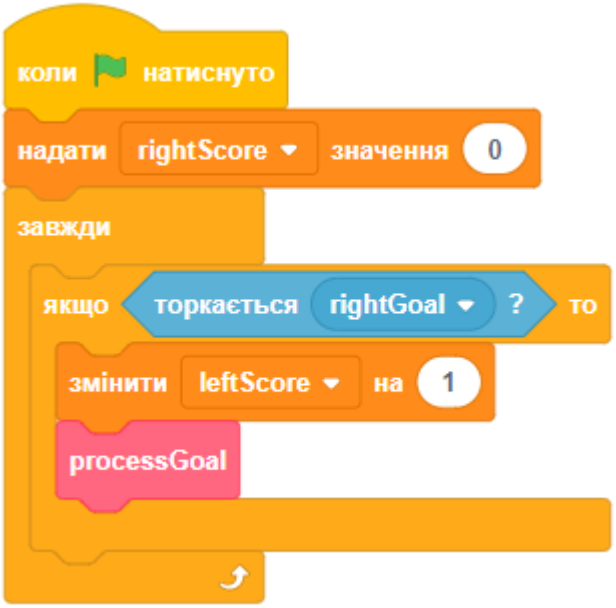
ВІДПОВІДЬ 1: ПРАВИЙ ФУТБОЛІСТ



ВІДПОВІДЬ 2: ВІДБИТТЯ М'ЯЧА ВІД ПРАВОГО ФУТБОЛІСТА

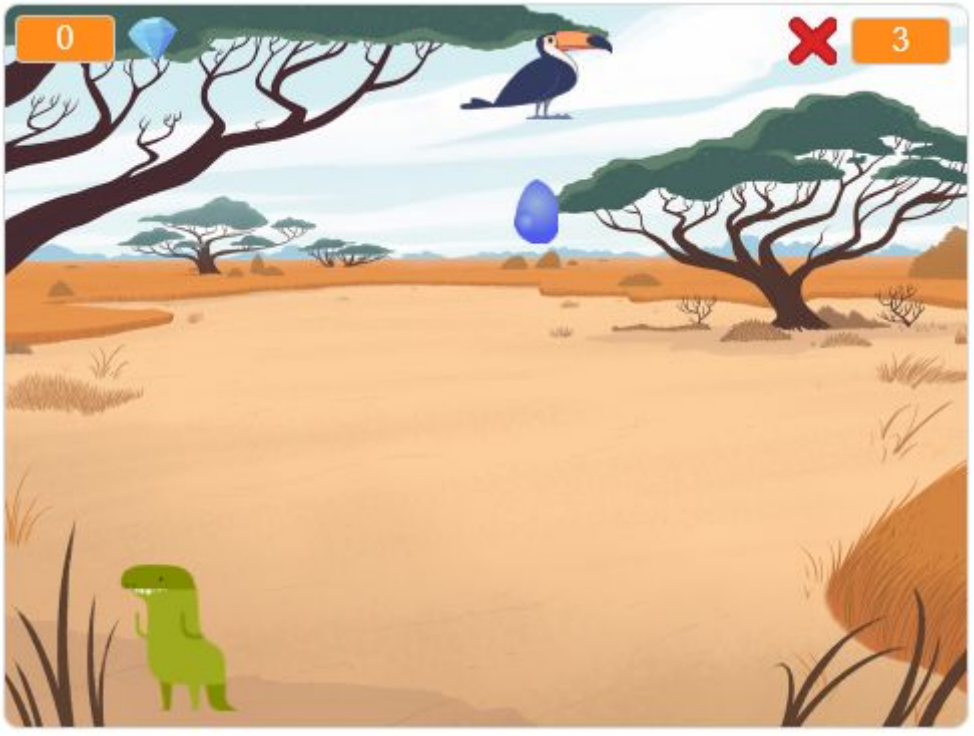


ВІДПОВІДЬ 3: ВЗАЄМОДІЯ М’ЯЧА З ПРАВИМИ ВОРОТАМИ



УРОК 14. ДИНОЗАВР ПРОТИ ПЕЛІКАНА

Ми будемо програмувати гру, в якій гравець буде збирати яйця пеліканів. Будьте уважні, не дозволяйте яйцям падати на землю, бо вони розіб'ються. Якщо розіб'ються 11 яєць, гра закінчується. За кожне зібране яйце гравець отримує діамант.

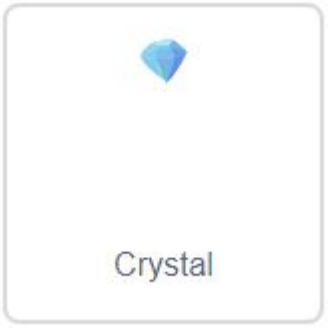


На цьому уроці ми будемо використовувати:

1. Цикли.	3. Повідомлення.
2. Змінні.	4. Створення власних блоків.

КРОК 1: СПРАЙТИ ТА СЦЕНА

- Почніть новий проект і видаліть з нього спрайт кота.
- Додайте такі спрайти з бібліотеки.

	Це наш головний герой. Ви можете обрати інший спрайт, але в нього мають бути щонайменше два образи, бо ми будемо робити анімацію.
	Цей спрайт потрібно розмістити у лівому верхньому кутку сцени поряд зі змінною рахунка гри. Ви можете вибрати або намалювати будь-який інший спрайт.
	Наш пелікан, який кидає яйця. Він з'являється у випадкових місцях верхньої частини сцени, трохи нахиляється та кидає яйце. Ви можете вибрати або намалювати будь-який інший спрайт замість пелікана.
	Яйця будуть падати з верхньої частини сцени та розбиватися внизу. Нам потрібні тільки два образи цього спрайта (у нормальному і у розбитому стані). Також можете пофарбувати яйце в інший колір, щоб воно було більш помітним. Замість яєць ви можете використати будь-який спрайт з бібліотеки або намалювати його самостійно.

	Цей спрайт потрібно розмістити у правому верхньому кутку сцени поряд зі змінною рахунка розбитих яєць. Нам потрібен другий образ цього спрайта (він червоного кольору). Також треба трохи зменшити спрайт, щоб він гарно виглядав поряд зі значенням змінної.

- Додайте тло сцени, яке вам подобається. Наприклад:

- Додайте змінні до вашої гри.

score	Рахунок гри. Збільшується, якщо динозавр з'їдає яйце. Показуйте цю змінну на сцені без назви, розмістить її у правому верхньому кутку.
badScore	Рахунок розбитих яєць. Якщо розбилося більше 11 яєць - кінець гри.
bestScore	Найкращий рахунок гри. Наприкінці гри рахунок записується сюди, якщо він більший за поточне значення цієї змінної.

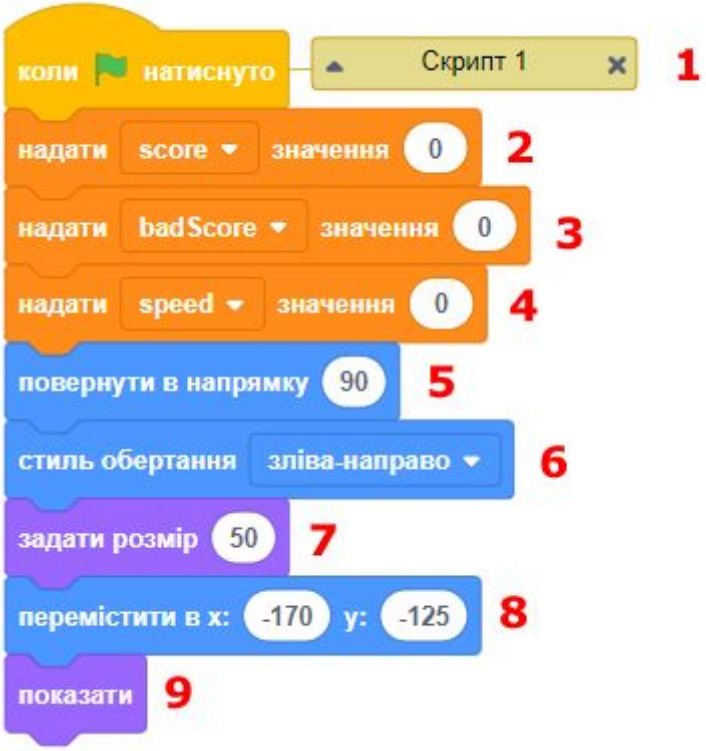
- Додайте ці локальні змінні до динозавра. Вкажіть «**Тільки для цього спрайта**», коли створюєте змінну.

speed	Швидкість руху динозавра. Постійно збільшується до певного значення, якщо динозавр не повертається та не торкається меж сцени.
jump	Змінна використовується під час стрибка.

КРОК 2: ДИНОЗАВР

СКРИПТ 1. ВСТАНОВЛЮЄМО ПОЧАТКОВІ ЗНАЧЕННЯ

- Додайте новий скрипт до динозавра. Цей скрипт задає початкові значення змінних, стиль обертання, образ, початкове положення та напрям руху.



1	Скрипт запускається на початку гри.
2-4	Встановлюємо початкові значення змінних рахунка зібраних яєць, рахунка розбитих яєць і швидкості руху динозавра.
5	Повертаємо спрайт у початковому напрямку.
6	Нам потрібно обертати динозавра тільки зліва направо.
7	Трохи зменшуємо динозавра. Якщо у вас інший спрайт, значення параметра цього блоку треба буде змінити.
8	Переміщуємо динозавра у початкове положення.
9	Показуємо спрайт динозавра на сцені.

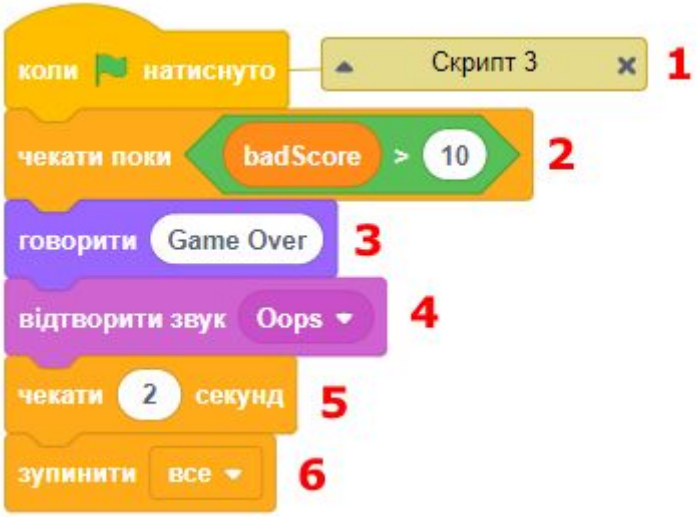
СКРИПТ 2. ДИНОЗАВР РУХАЄТЬСЯ

- Додайте динозавру скрипт, який буде відповідати за переміщення ігровим полем.

1	Скрипт запускається на початку гри.
2	Весь час гри ми будемо виконувати певні дії.
3	Переміщуємо динозавра з поточною швидкістю.
4	Якщо динозавр торкається межі, нам треба виконати певні дії.
5	Відтворюємо потрібний звук. Ви можете використати будь-який звук з бібліотеки.
6	Змінити напрямок руху на протилежний (право на ліво, ліво на право).
7	Встановлюємо мінімальну швидкість динозавра.
8	Трохи стрибаємо від межі сцени.

СКРИПТ 3. ПЕРЕВІРЯЄМО КІЛЬКІСТЬ РОЗБИТИХ ЯЄЦЬ

- Якщо розіб'ються 11 яєць, гра закінчується. Цей скрипт весь час гри перевіряє, скільки вже розбито яєць.



1	Скрипт запускається на початку гри.
2	Нам треба почекати, поки не розіб'ється 11 яєць.
3	Говоримо, що гра закінчена.
4	Відтворюємо звук кінця гри. Ви можете використати будь-який звук з бібліотеки.
5-6	Трохи чекаємо і зупиняємо гру.

СКРИПТ 4. ЗБІЛЬШУЄМО ШВИДКІСТЬ РУХУ ДИНОЗАВРА

- Додайте динозавру скрипт, який завжди буде збільшувати його швидкість на три із затримкою в 0,3 секунди, але за умови, що ця швидкість не перевищує 10.

1-2	Скрипт запускається на початку та працює весь час гри.
3	Чекаємо, поки швидкість руху динозавра буде менша 10.
4	Чекаємо, щоб збільшення швидкості було поступовим.
5	Збільшуємо швидкість руху на 3.

СКРИПТ 5. ЗМІНЮЄМО РАХУНОК ГРИ

- Цей скрипт змінює рахунок гри, отримуючи повідомлення.

1	Скрипт запускається за допомогою повідомлення про зміну рахунка гри.
2	Змінюємо рахунок гри на одиницю.
3	Змінюємо образ динозавра на той, в якому відкритий рот.
4	Відтворюємо звук гарчання або будь-який інший.
5-6	Трохи чекаємо та повертаємо початковий образ спрайта.
7	Зупиняємо звуки, щоб динозавр більше не гарчав.

СКРИПТ 6. ОБРОБКА КЛАВІШ КЛАВІАТУРИ

- Тепер необхідно запрограмувати динозавра так, щоб гравець мав змогу керувати ним, змінюючи напрям руху. Також динозавр вміє стрибати, щоб мати змогу впіймати яйце у повітрі.
- У цьому скрипті ми використовуємо допоміжні блоки стрибка та повороту ліворуч і праворуч (скрипти 7-9).

1	Скрипт запускається на початку гри.
2	Під час гри нам потрібно перевіряти натискання клавіш і виконувати допоміжні блоки повороту та стрибка.
3,6	Повернути ліворуч, якщо натиснуто ЛІВОРУЧ або 1.
4,7	Повернути праворуч, якщо натиснуто ПРАВОРУЧ або 3.
5,8	Стрибнути, якщо натиснуто ВГОРУ або 2.

СКРИПТ 7. ДОПОМІЖНИЙ БЛОК ВЛІВО

- Додайте допоміжний скрипт, який буде очікувати натискання відповідної клавіші та розвертати спрайт вліво.
- Зауважте, що після розгортання швидкість має мінімальне значення, яке дорівнює 4.

1	Блок буде мати параметр, який нам скаже, чи взагалі потрібно змінювати напрямок руху.
2	Перевірка на значення параметру. Ми змінюємо напрямок руху, тільки коли він дорівнює ІСТИНА.
3	Нам треба повернути, тільки коли динозавр рухається у протилежному напрямку.
4	Встановлюємо мінімальну швидкість руху.
5	Повертаємо спрайт наліво.

СКРИПТ 8. ДОПОМІЖНИЙ БЛОК ВПРАВО

- Допоміжний скрипт, який відповідає за розворот направо.

1	Блок буде мати параметр, який нам скаже, чи взагалі потрібно змінювати напрямок руху.
2	Перевірка на значення параметра. Ми змінюємо напрямок руху, тільки коли він дорівнює ІСТИНА.
3	Нам треба повернути, тільки коли динозавр рухається у протилежному напрямку.
4	Встановлюємо мінімальну швидкість руху.

5	Повертаємо спрайт направо.

СКРИПТ 9. ДОПОМІЖНИЙ БЛОК СТРИБКА

- Тепер необхідно запрограмувати стрибок динозавра.

1	Блок стрибка буде мати параметр, який нам скаже, чи взагалі потрібно стрибати.
2	Перевірка на значення параметру, ми стрибаємо, тільки коли він дорівнює ІСТИНА.
3	Відтворюємо потрібний звук. Ви можете використати будь-який звук з бібліотеки.
4	Встановлюємо швидкість стрибка. Щоб стрибок був якнайбільш справжнім, одразу після підстрибування, швидкість руху вгору повинна бути максимальною.
5	Стрибок має два етапи - переміщення вгору та падіння.
6-7	Нам треба 20 разів змінити положення спрайта по у-координаті. Це робиться на двох етапах стрибка.
8	З часом швидкість руху буде зменшуватися до нуля (тобто динозавр досягнув найвищої точки), а потім поступово збільшуватиметься при падінні донизу.
9	Після виконання цього блоку пелікан починає падати.

КРОК 3: ПЕЛІКАН

- Перейдемо до програмування пелікана. Він повинен з’являтися угорі екрану та скидати яйця, для цього ми використаємо клонування спрайта яйця.

СКРИПТ 1. ВСТАНОВЛЮЄМО ПОЧАТКОВІ ЗНАЧЕННЯ

- Цей скрипт задає розмір та початковий образ, ховає спрайт.

1	Скрипт запускається на початку гри.
2	Нам треба трохи зменшити образ пелікана. Якщо у вас інший спрайт, вам треба змінити значення параметра цього блоку.
3	Встановлюємо початковий образ пелікана.
4	Ховаємо пелікана. Появленням спрайта буде керувати інший скрипт.

СКРИПТ 2. ПЕЛІКАН КИДАЄ ЯЙЦЕ

- Цей скрипт керує пеліканом, який впродовж гри викидає яйця, з'являючись у випадкових положеннях зверху сцени.

1	Скрипт запускається на початку гри.

2	Скрипт має постійний цикл, у якому пелікан то зникає, то з'являється на сцені та кидає яйця вниз.
3-4	Пелікан чекає випадкову кількість секунд і рухається у точку з випадковими координатами.
5-6	Пелікан повертається направо та з'являється на сцені.
7-10	Пелікан має невелику анімованість за допомогою блоків очікування та блоків повороту спрайта.
11-13	Наприкінці пелікан створює клон яйця та після невеликої затримки зникає зі сцени.

КРОК 4: ЯЙЦЕ

СКРИПТ 1. ВСТАНОВЛЮЄМО ПОЧАТКОВІ ЗНАЧЕННЯ

- Цей скрипт дуже простий, нам треба зробити початкові налаштування для яйця.

1	Скрипт запускається на початку гри.
2	Основний спрайт нам не потрібен, ми будемо використовувати тільки клони яйця.
3	Нам треба трохи зменшити розмір спрайта. Якщо ви користуєтесь замість яйця іншим спрайтом, вам треба змінити значення параметра цього блоку.
4	Змінюємо образ на початковий. Назва образу залежить від спрайта, яким ви користуєтесь замість яйця.

СКРИПТ 2. ПАДІННЯ ЯЙЦЯ

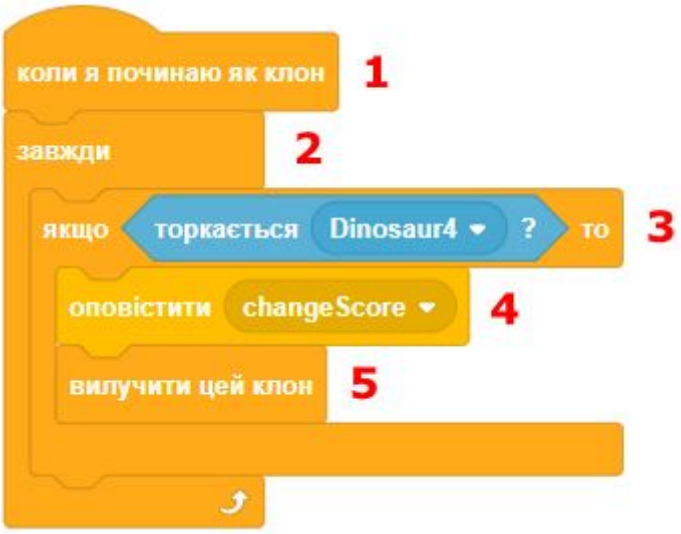
- Тепер необхідно запрограмувати спрайт яйця так, щоб його клони падали зверху. Після падіння яйця будуть недоступні динозавру та зникають зі сцени.

1	Скрипт запускається тільки для клонів яйця.
2-3	Переміщуємо клон яйця у місцезнаходження пелікана та показуємо його на сцені.
4	Відтворюємо потрібний звук. Ви можете використати будь-який звук з бібліотеки.
5-6	Цикл «повторити до», який зменшує y-координату клону яйця, поки це значення не стане меншим за -150
7-11	Коли герой не підхопив яйце, воно долетіло до землі. Змініть його образ на «розбите яйце» та після невеликої затримки вилучіть клон.

8	Збільшуємо кількість розбитих яєць.
9	Відтворюємо звук розбиття яйця. Ви можете використати будь-який звук з бібліотеки.
10	Трохи чекаємо.
11	Цей клон яйця нам більше не потрібен, вилучаємо його.

СКРИПТ 3. ПЕРЕВІРКА ТОРКАННЯ З ДИНОЗАВРОМ

- Цей скрипт весь час гри перевіряє, чи торкається клон яйця нашого динозавра. Якщо так, нам треба сповістити динозавра і вилучити цей клон яйця.



1	Скрипт запускається тільки для клонів яйця.
2	Ми весь час гри виконуємо потрібні перевірки.
3	Перевірка на торкання до динозавра.
4	Треба надіслати повідомлення про зміну рахунка гри.
5	Динозавр спіймав яйце, видаляємо цей клон.

КРОК 5: СЦЕНА

СКРИПТ 1. ФОНОВА МУЗИКА

- Цей скрипт керує відтворюванням музики, що грає весь час гри.

- Фонова музика не може звучати дуже голосно, тому ми зменшуємо її гучність до 5%.

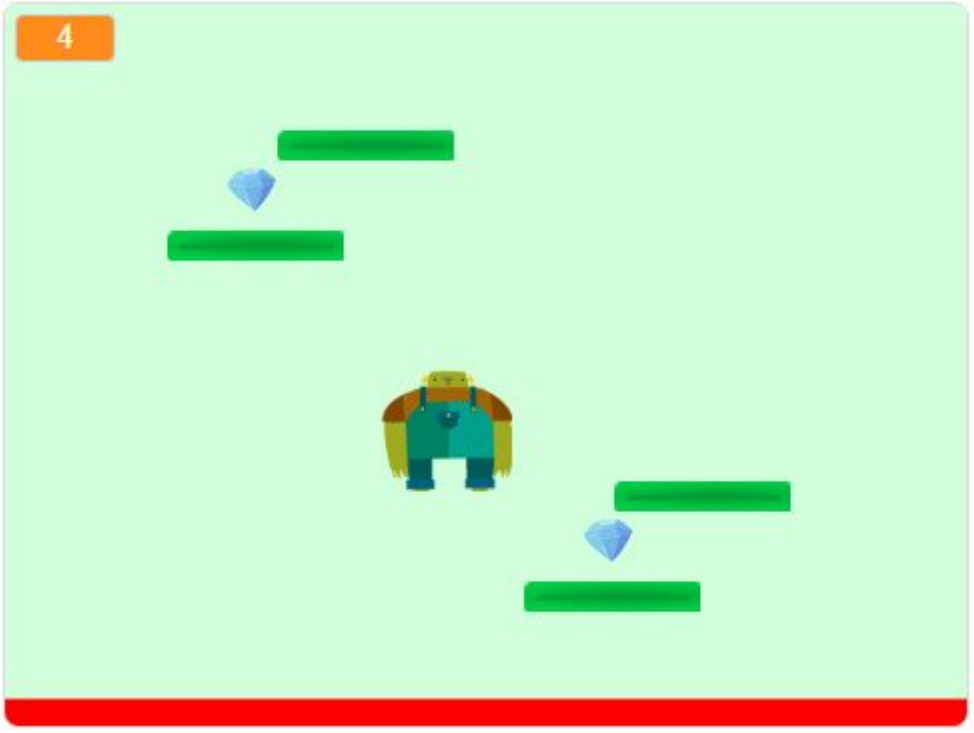
1	Скрипт запускається на початку гри.
2	Встановлюємо гучність відтворення фонової музики у 5%
3-4	Нам треба завжди відтворювати фонову музику. Запускаємо відтворення звуку до кінця, а потім робимо це знов і знов.

КРОК 6: ЗАВДАННЯ

- Допоміжні скрипти 7 и 8 динозавра дуже подібні один до одного. Переробіть ці скрипти, щоб у них був додатковий параметр, який буде відповідати за напрямок руху (-1 - ліворуч, 1 - праворуч), та зробіть все, що потрібно, у одному скрипті замість двох.
- Деякі яйця не потрібно ловити, вони будуть іншого кольору. Якщо динозавр їх зловить, рахунок гри зменшується. Також коли такі яйця розбиваються, це не враховується.
- Початкове тло з назвою гри та інструкцією.
- Тло кінця гри з написом Game Over.
- Що ще можно зробити?

УРОК 15. ВЕСЕЛІ ДІАМАНТИ

Ми створимо гру, у якій головний герой буде збирати діаманти, стрибаючи по платформах, що падають.

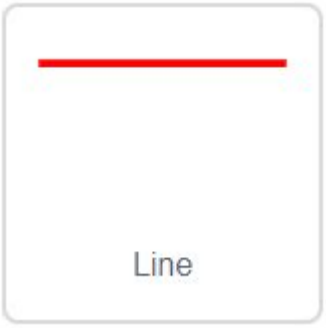
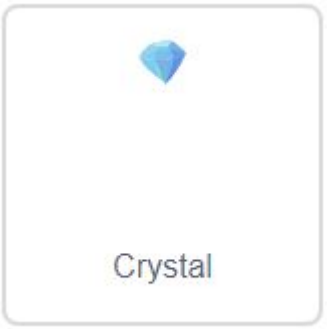


На цьому уроці ми будемо використовувати:

1. Змінні.
2. Клони.
3. Розгалуження.
4. Цикли.

КРОК 1: СПРАЙТИ ТА СЦЕНА

- Почніть новий проект і видаліть з нього спрайт кота.
- Додайте наступні спрайти з бібліотеки.

	
Це наш головний герой. Ви можете обрати інший спрайт, але в нього мають бути щонайменше два образи. Перший образ у звичайному стані. Другий образ у стані стрибка.	Це обмежувальна лінія, вона розташована знизу. Якщо наш головний герой торкається цієї лінії - гра закінчується. Інші спрайти також зникають зі сцени, якщо торкаються цієї лінії.
	
Це платформа, по ній буде стрибати наш головний герой.	Наш головний герой буде збирати ці діаманти.

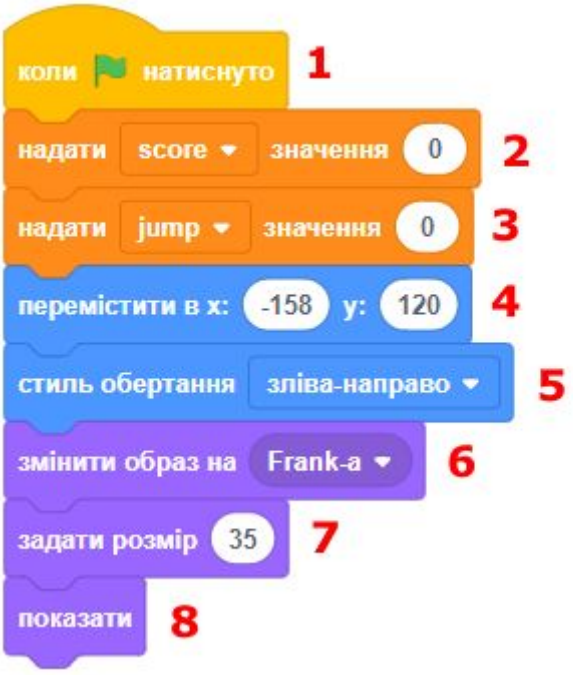
- Додайте нові змінні.

jump	Змінна стрибка головного героя. Вона дорівнює 0 в звичайному стані, та дорівнює 1 в стані стрибка.
position	Змінна, яка використовується для визначення x-координати під час створення клону діаманта.
score	Рахунок гри. Збільшується, якщо наш головний герой збирає діамант.

КРОК 2: ГОЛОВНИЙ ГЕРОЙ

СКРИПТ 1. ВСТАНОВЛЕННЯ ПОЧАТКОВИХ ЗНАЧЕНЬ

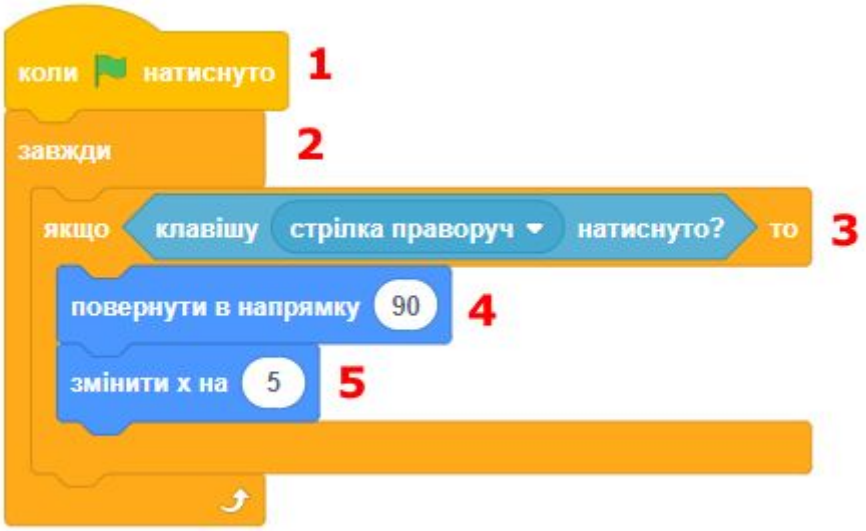
- Додайте скрипт до головного героя.



1	Скрипт запускається на початку гри.
2	Рахунок гри встановлюємо в нуль.
3	Змінну стрибка встановлюємо в нуль.
4	Переміщуємо спрайт у початкову позицію.
5	Встановлюємо потрібний стиль обертання.
6	Змінюємо на початковий образ.
7	Треба трохи зменшити наш спрайт.
8	Показуємо спрайт на сцені.

СКРИПТ 2. ПЕРЕМІЩЕННЯ ПРАВОРУЧ ТА ЛІВОРУЧ

- Додайте скрипт, за допомогою якого спрайт буде переміщатися праворуч при натисканні відповідної клавіші.

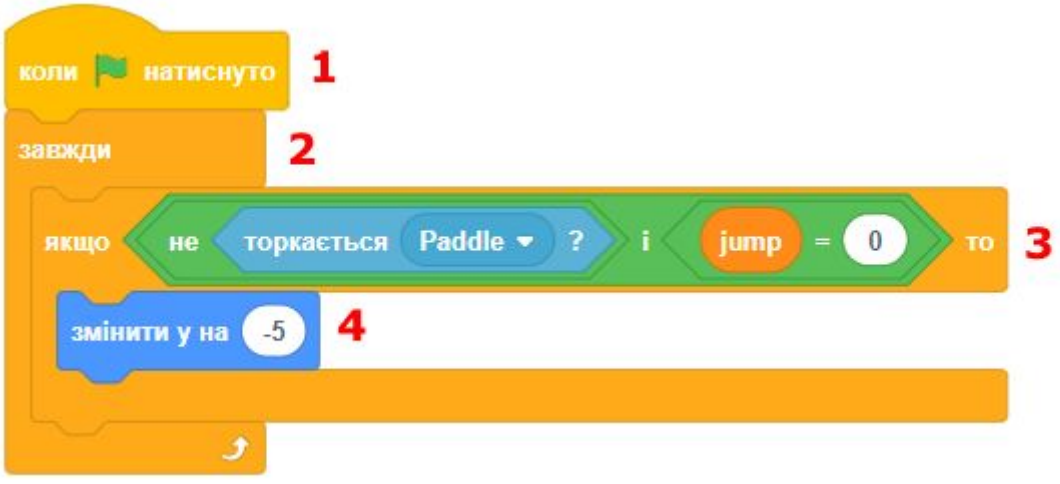


1	Скрипт запускається на початку гри.
2	Весь час гри ми виконуємо потрібну перевірку.
3	Перевіряємо, чи натиснута потрібна клавіша.
4	Повертаємо праворуч.
5	Змінюємо x-координату спрайта на 5 (спрайт переміщується праворуч).

- **Завдання.** Вам треба самостійно запрограмувати обробку натискання клавіші «стрілка ліворуч».

СКРИПТ 3. ПАДІННЯ ПІД ДІЄЮ СИЛИ ТЯЖІННЯ

- Додайте новий скрипт, який запрограмує силу тяжіння.



1	Скрипт запускається на початку гри.

2	Весь час гри ми виконуємо потрібну перевірку.
3	Головний герой буде падати вниз, якщо він не стрибає (змінна «jump» = 0) та не торкається платформи.
4	Падіння відбувається за допомогою зміни «положення по y» спрайта на -5.

СКРИПТ 4. СТРИБАЄМО ПО ПЛАТФОРМАХ

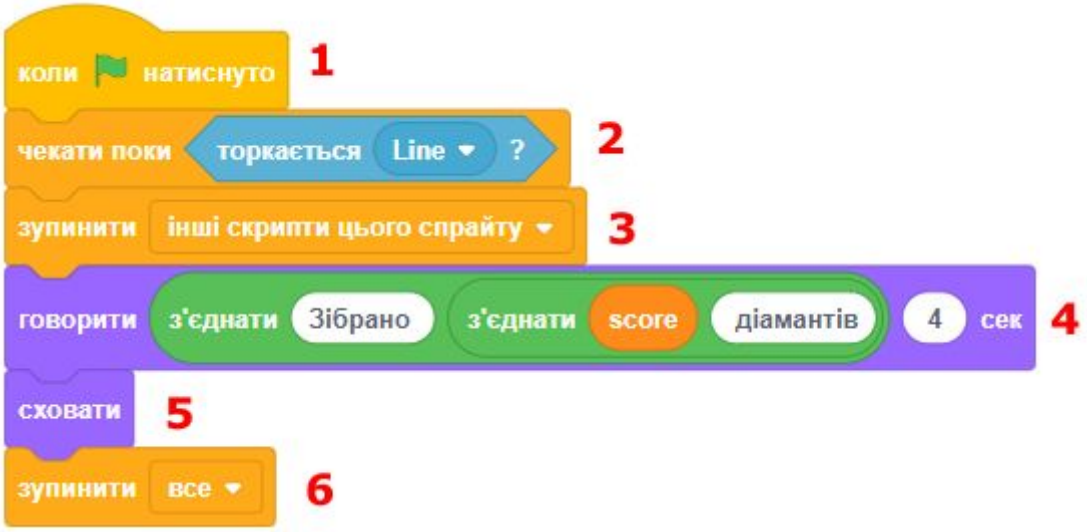
- Наш головний герой буде стрибати по платформах. Нам треба запрограмувати спрайт таким чином, щоб при дотику до платформи він підстрибував.

1	Скрипт запускається на початку гри.
2-3	Весь час гри ми перевіряємо, чи торкається головний герой платформи.
4	Встановлюємо змінну **jump** у значення 1. Це означає, що головний герой стрибає. Інші спрайти гри теж щось роблять, коли jump стає рівним 1.

5	Змінюємо образ героя у стані стрибка.
6-7	Нам треба плавно перемістити спрайт вгору, бо він стрибає. Нам не потрібно повертати його вниз, це буде зроблено скриптом № 3, який за це відповідає.
8	Повертаємо звичайний образ спрайта.
9	Встановлюємо змінну jump у значення 0. Стрибок головного героя закінчено.

СКРИПТ 5. ПАДІННЯ У ПРІРВУ

- Нам треба запрограмувати головного героя таким чином, щоб при дотику до лінії внизу гра закінчувалася.



1	Скрипт запускається на початку гри.
2	Ми чекаємо, поки головний герой не торкнеться лінії, яка розташована знизу.
3	Зупиняємо всі інші скрипти цього спрайта.
4	Спрайт доповідає, скільки діамантів він зібрав.
5-6	Ховаємо спрайт головного героя та зупиняємо гру.

КРОК 3: ПЛАТФОРМА

СКРИПТ 1. ДОПОМІЖНИЙ БЛОК ДЛЯ СТВОРЕННЯ ПЛАТФОРМИ

- Додайте свій власний блок, за допомогою якого ми будемо створювати клони платформи.

1	У блока є два параметри - **x** та **y**. Вони відповідають за координати клону платформи.
2	Основний спрайт платформи (він прихований) переміщується по новим координатам, щоб клони користувалися ними.
3	Змінна **position** встановлюється в значення параметру X. Це робиться тому, що клони діамантів будуть створюватися з x-координатою, отриманою зі змінної position.
4	Створюється клон платформи.
5	При створенні нової платформи треба запрограмувати можливу появу клону діаманта. Не всі платформи мають діаманти, а лише деякі з них.
6	Створення клону діаманта.

СКРИПТ 2. СТВОРЕННЯ ПЛАТФОРМ НА ПОЧАТКУ ГРИ

- Додайте початковий скрипт до платформи.

1	Скрипт запускається на початку гри.
2	Якщо потрібно, змінить розмір спрайта.
3	Приховайте основний спрайт платформи. Ми будемо бачити лише клони платформи.
4-7	Створіть чотири початкові клони платформи.

СКРИПТ 3. ПЕРЕМІЩЕННЯ ПЛАТФОРМ ВНИЗ

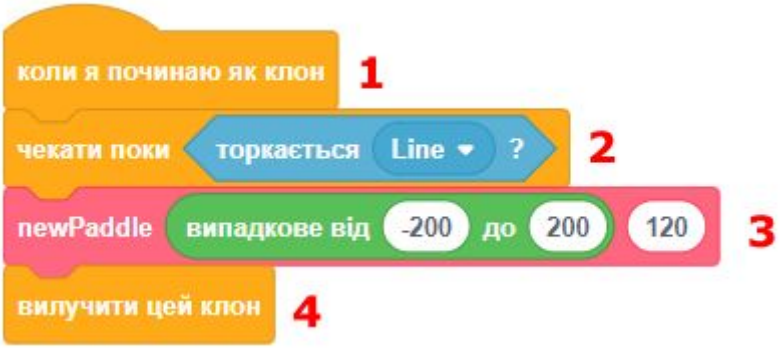
- Коли значення змінної **jump** дорівнює одиниці, наші клони повинні переміщатися донизу.

1	Скрипт запускається при створенні клону.

2	Клон треба показати на сцені.
3-5	Весь час гри ми виконуємо необхідну перевірку. Якщо герой у стані стрибка, треба перемістити платформу вниз.

СКРИПТ 4. ВИЛУЧЕННЯ НЕПОТРІБНИХ ПЛАТФОРМ

- Якщо клон платформи торкається нижньої межі екрана – він повинен зникнути, але перед цим – створити нову платформу у верхній частині екрана.



1	Скрипт запускається при створенні клону.
2	Ми чекаємо, поки цей клон платформи не торкнеться лінії, яка розташована знизу.
3	Створюємо нову платформу з випадковою x-координатою та y-координатою = 120 (зверху).
4	Вилучаємо клон платформи, він нам не потрібен.

КРОК 4: ДІАМАНТ

СКРИПТ 1. ВСТАНОВЛЕННЯ ПОЧАТКОВИХ ЗНАЧЕНЬ

- Додайте скрипт до діаманта.

1	Скрипт запускається на початку гри.

2	Потрібно, щоб діамант відображався над іншими спрайтами.
3	Треба сховати основний спрайт діаманта, ми будемо користуватися лише клонами.

СКРИПТ 2. ОСНОВНИЙ СКРИПТ КЛОНУ ДІАМАНТА

- Скрипт для клону діаманта.

1	Скрипт запускається при створенні клону діаманта.
2	Діаманти створюються на платформі. Змінна position містить x-координату останнього створеного клону платформи. Встановлюємо координати цього клону діаманта, щоб він з'явився на платформі.
3	Клон треба показати на сцені.
4-6	Весь час гри ми виконуємо потрібну перевірку. Якщо герой у стані стрибка, нам треба перемістити діамант вниз.

СКРИПТ 3. ВИЛУЧЕННЯ НЕПОТРІБНИХ ДІАМАНТІВ

- Скрипт для клону діаманту.

1	Скрипт запускається при створенні клону діаманта.
2	Ми чекаємо, доки цей клон діаманта не торкнеться лінії, яка розташована знизу.
3	Вилучаємо клон діаманта, він нам не потрібен.

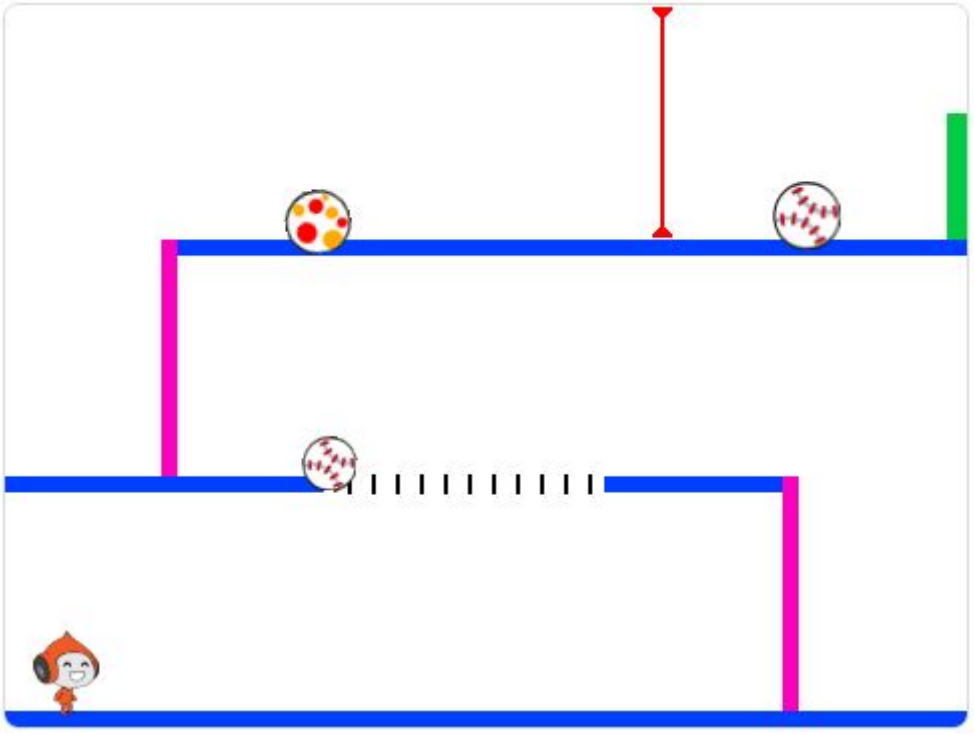
СКРИПТ 4. ЗБІЛЬШЕННЯ РАХУНКУ

- Скрипт для клону діаманта.

1	Скрипт запускається при створенні клону діаманта.
2	Чекаємо, поки цей клон діаманта не торкнеться героя.
3-4	Збільшуємо рахунок на одиницю та вилучаємо клон діаманта, він нам більше не потрібен. Гравець побачить, ніби головний герой зібрав діамант.

УРОК 16. ВЕРТИКАЛЬНИЙ ЛАБІРИНТ

У цьому проекті ви дізнаєтеся, як створити гру, в якій ви повинні ухилятися від рухомих м'ячів і дістатися до кінця рівня.

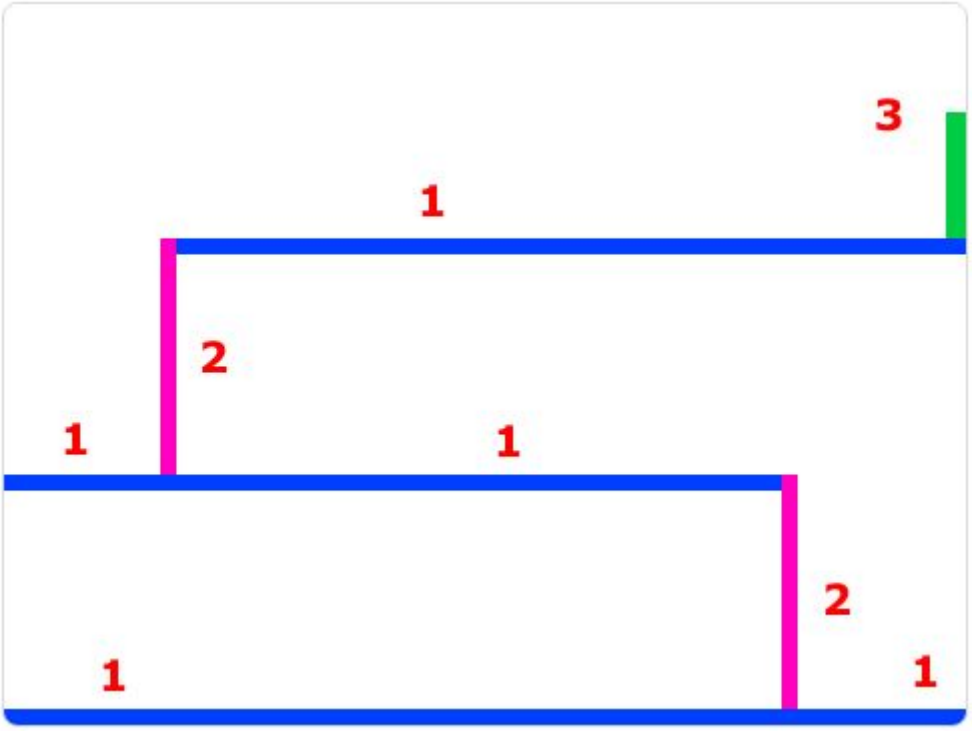


КРОК 1: РУХОМИЙ ПЕРСОНАЖ

Почнемо зі створення персонажа, який може рухатися вліво і вправо, а також підніматися вгору по стовпу.

- Створюємо новий проект, видаляємо звідти кота.
- Намалюйте власне тло сцени. Стовпи і платформи мають бути намальовані різними кольорами.

- Також переконайтеся, що на рівні є двері (або щось подібне), через які ваш персонаж повинен пройти, щоб закінчити гру. Ось як має виглядати тло сцени:

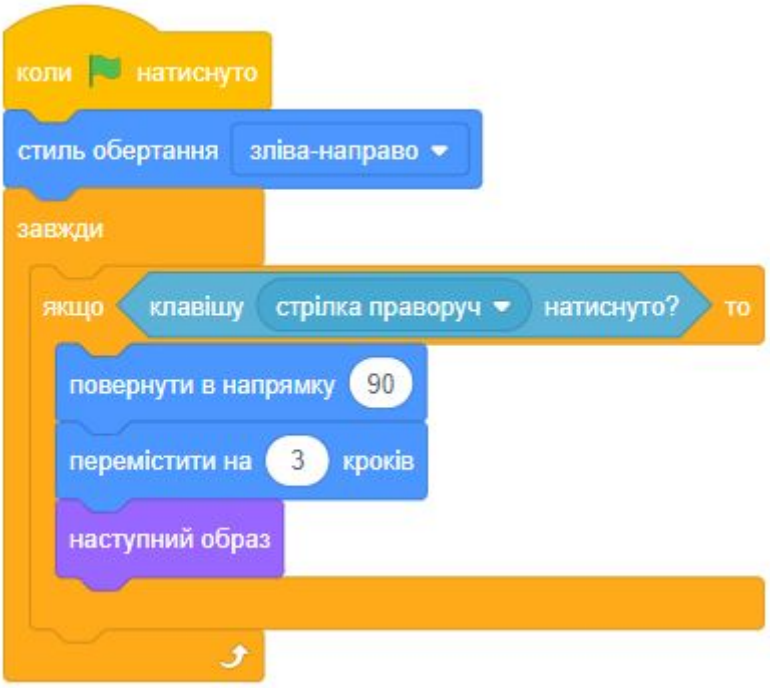


КОЛІР	ПОЯСНЕННЯ
1	Колір платформи. Наприклад, синій.
2	Колір стовпа. Наприклад, рожевий.
3	Колір дверей. Наприклад, зелений.

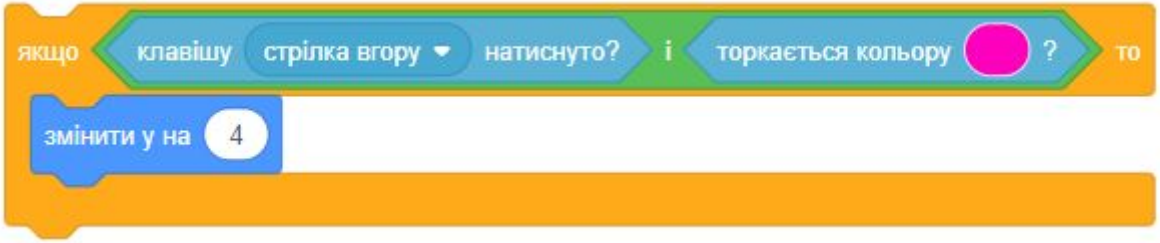
- Додайте новий спрайт, який буде вашим персонажем. Краще вибрати спрайт, у якого є кілька образів, завдяки чому можна створити видимість руху персонажа. Дайте йому ім'я **hero**.

- Зробимо переміщення персонажа при натисканні клавіш.

- Коли гравець натискає кнопку зі стрілкою вправо, ми хочемо, щоб персонаж повертався направо, переміщався на кілька кроків і міняв образ на наступний.



- Протестуйте поведінку вашого героя, натиснувши на зелений прапор, потім натиснувши клавішу зі стрілкою вправо. Чи правильно він рухається? Чи виглядає він так, ніби гуляє?
- **ЗАВДАННЯ.** Для того щоб ваш персонаж рухався вліво, вам необхідно додати ще один блок ЯКЩО в середину циклу ЗАВЖДИ. Аналогічно до попереднього новий доданий блок повинен бути складений так, щоб персонаж рухався вліво. Протестуйте ваш новий скрипт, щоб переконатися, що він працює правильно!
- Щоб піднятися на стовп, ваш персонаж повинен рухатися вгору, коли натиснута клавіша стрілка вгору, і при цьому він торкається кольору стовпа. Додайте цей код для персонажа в середину циклу ЗАВЖДИ:



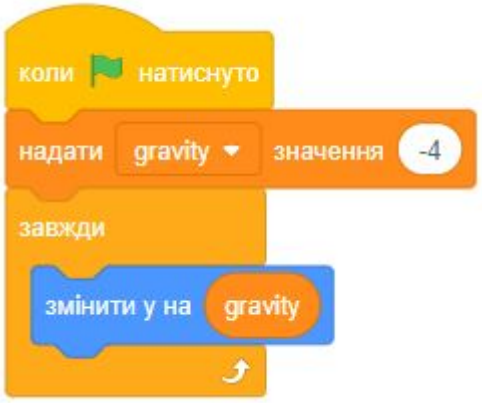
- У наведеному вище скрипті в блоці ТОРКАЄТЬСЯ КОЛЬОРУ ми використовуємо колір стовпа.

- Протестуйте ваш скрипт - чи може персонаж підніматися по стовпах і доходити до кінця рівня?

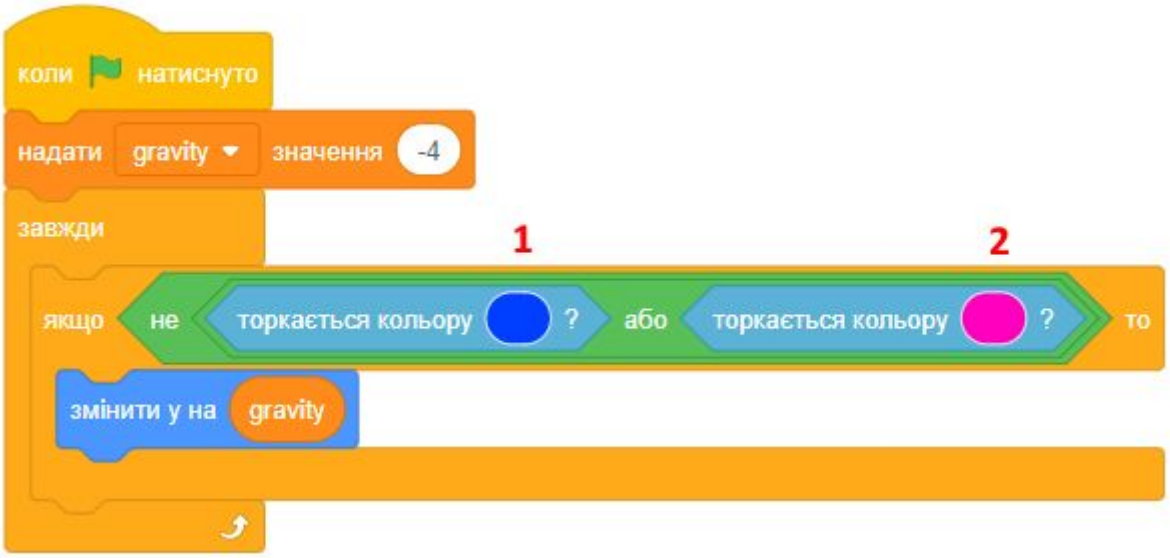
КРОК 2: ГРАВІТАЦІЯ І СТРИБКИ

Давайте зробимо так, щоб ваш персонаж рухався більш реалістично, додавши гравітацію і дозволивши йому стрибати.

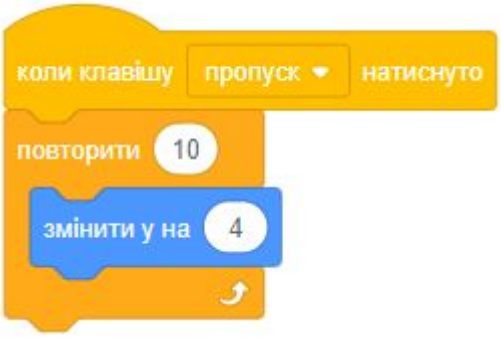
- Ви могли помітити, що персонаж може піти з платформи в повітря. Спробуйте йти від платформи і подивіться, що відбувається.
- Щоб виправити це, давайте додамо гравітацію до вашої гри. Додайте нову змінну і назвіть її **gravity** (гравітація). Сховайте цю змінну зі сцени.
- Додайте новий скрипт, який задає значення змінної **gravity** від'ємним числом, потім постійно повторюючи, змінює у-координату персонажа на значення в змінній.



- Клацніть на зелений прапор, а потім перетягніть персонажа в верхню частину сцени. Що відбувається? Чи працює гравітація?
- Завдяки гравітації ваш персонаж більше не буде переміщатися крізь платформу або стовп!
- Додайте в ваш скрипт блок ЯКЩО, щоб гравітація діяла, тільки коли персонаж знаходиться в повітрі. Тепер скрипт має виглядати так:



- У наведеному вище скрипті: (1) - колір горизонтальної платформи, (2) - колір стовпа.
- Протестуйте знову. Чи залишається персонаж на платформі або на стовпі? Чи може він перейти з краю платформи на платформу нижче?
- Давайте також зробимо так, щоб наш персонаж стрибав, коли гравець натискає на клавішу ПРОПУСК. Зробити це просто. Треба перемістити персонаж вгору кілька разів:



- Оскільки сила тяжіння (гравітація) постійно штовхає ваш персонаж вниз на 4 пікселя, вам потрібно вибрати в блоці число більше 4. Змінюйте це число, допоки ви не будете задоволені висотою, з якої ваш персонаж стрибає.
- Якщо ви протестуете цей скрипт, то побачите, що він працює, але рух не виглядає плавним.
- Щоб стрибки виглядали плавними, вам необхідно переміщати персонаж на маленькі відрізки, поки він не завершить стрибок.
- Для того щоб зробити це, треба створити змінну **jumpHeight** (висота стрибка). Сховайте її зі сцени.

- Видаліть скрипт, який ви склали для стрибків, і замініть його на:

- Цей скрипт переміщує наш персонаж вгору на 10 пікселів, потім на 9.5 пікселів, потім на 9 пікселів і так далі, допоки персонаж не виконає стрибок. Такий скрипт робить стрибок плавним.
- Міняйте початкове значення змінної в якій зберігається висота стрибка і перевіряйте результат, допоки ви не будете задоволені висотою, з якої ваш персонаж стрибає.

КРОК 3: УХИЛЕННЯ ВІД М'ЯЧІВ

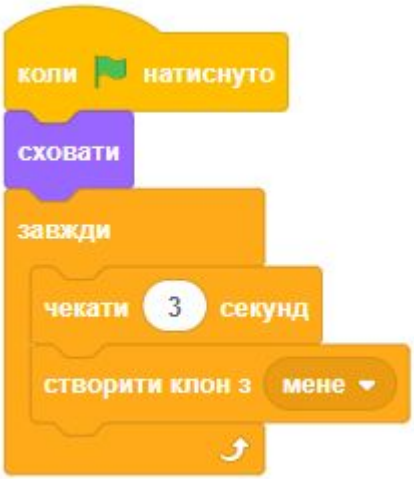
Тепер, коли у вас є персонаж, який може рухатися в різні боки, давайте додамо деякі м'ячі, від яких персонаж повинен буде ухилятися. Створіть новий спрайт м'яча. Виберить спрайт з бібліотеки:


Baseball


Basketball

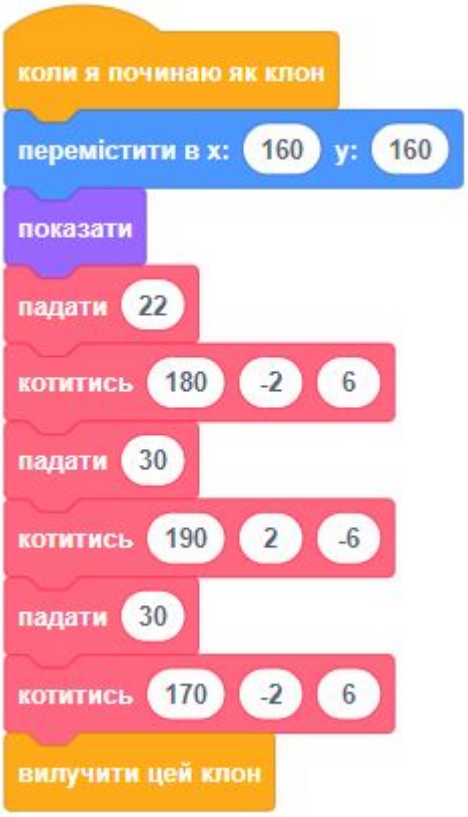

Beachball

- Змініть розмір м'яча щоб персонаж міг перестрибнути через нього. Спробуйте перестрибнути через м'яч, щоб перевірити, чи підходить розмір.
- Цей код створює клон м'яча кожні 3 секунди:

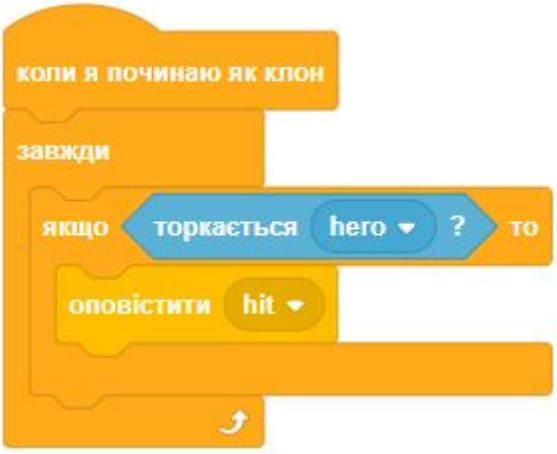


- М'яч може падати або котитися. Давайте передусім додамо допоміжні блоки, які роблять це.

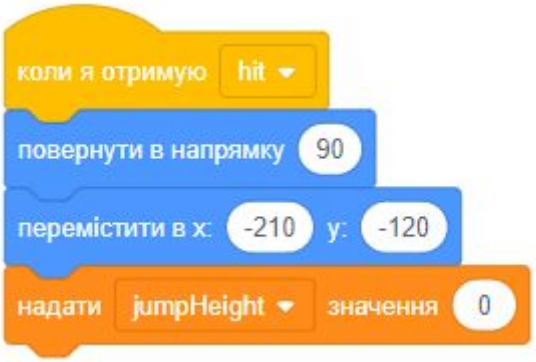
- Тепер нам потрібно запрограмувати поведінку клонів м'яча. Вони повинні з'являтися у верхній частині лабіринту.
- Кожен мяч рухається вздовж верхньої платформи наліво, падає вниз, рухається по середній платформі направо, знову падає вниз і рухається по нижній платформі наліво. В кінці клон м'яча зникає зі сцени.
- Додайте скрипт до м'яча:



- Протестуйте, як це працює, як рухаються м'ячі. Можливо, вам потрібно буде змінити параметри блоків падати і котитись у цьому скрипті, щоб м'ячі правильно рухалися у вашому лабіринті.
- Вам буде потрібен скрипт, який запрограмує поведінку персонажа, коли той отримає удар м'ячем!
- Додайте до **м'яча**:



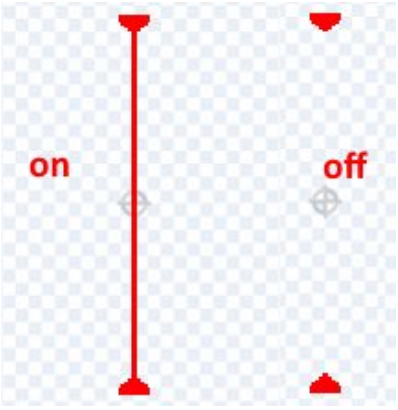
- Вам також необхідно додати скрипт до вашого персонажа, щоб при ударі він повертався в точку старту:



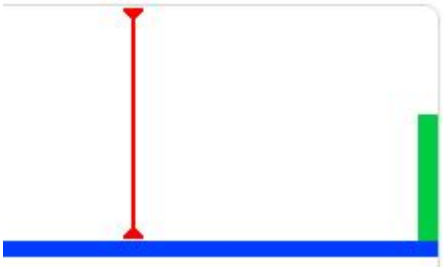
- Протестуйте поведінку персонажа, чи переміщується він в початкову позицію при ударі м'ячем.

КРОК 4: ЛАЗЕРИ

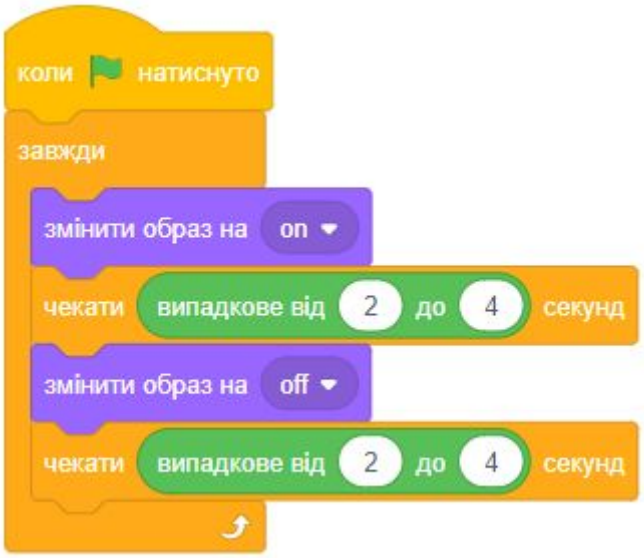
Давайте зробимо вашу гру більш складною за допомогою лазерів! Додайте новий спрайт. Дайте йому назву «laser». Він повинен мати два образи з назвами «on» і «off».



- Помістіть ваш новий лазер будь-де між двома платформами.



- Додайте скрипт для лазера, щоб він постійно змінював перший образ на другий. Якщо хочете, ви можете чекати випадковий проміжок часу між зміною образів.



- Додайте скрипт для спрайта лазера, щоб, коли лазер торкнувся персонажа, передавалося повідомлення «hit».

- Перевірте вашу гру, щоб переконатися, чи може головний персонаж пройти повз лазер. Змініть час роботи лазера, якщо лазер занадто легко або занадто важко пройти.

КРОК 5: ЗАВДАННЯ

ЗАВДАННЯ: ПОЛІПШЕННЯ СТРИБКІВ

Ваш персонаж здатний стрибати щоразу, коли натиснута кнопка пробіл, навіть якщо він уже в повітрі. Ви можете побачити це, просто утримуючи ПРОПУСК натиснутим.
Чи можете ви виправити це, щоб персонаж міг стрибати, тільки ЯКЩО він торкається платформи?

ЗАВДАННЯ: ВИПАДКОВІ М'ЯЧІ

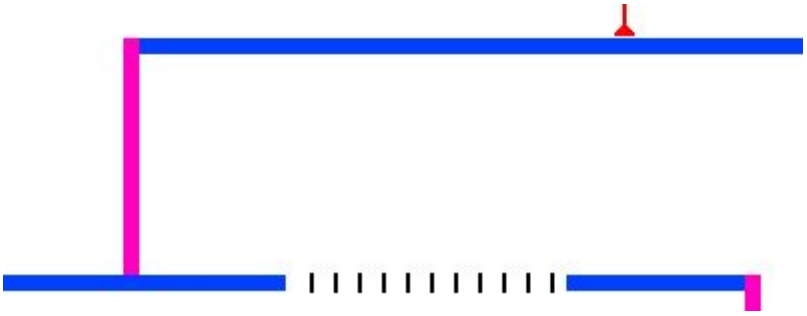
Всі м'ячі, від яких повинен ухилитися ваш персонаж, виглядають однаково і завжди з'являються кожні 3 секунди. Чи можете ви виправити це, щоб вони:

- не виглядали однаковими?
- з'являлися через якийсь випадковий проміжок часу?
- мали випадкові розміри?

ЗАВДАННЯ: БІЛЬШЕ ПЕРЕШКОД

Якщо ви думаєте, що гра все ж занадто легка, то можете додати більше перешкод. Ви можете додати все що завгодно, ось кілька ідей:

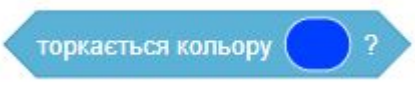
- Літаючі метелики-вбивці;
- Платформи, які з'являються і зникають;



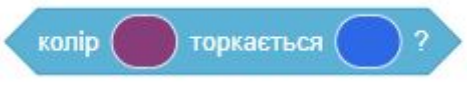
- Падаючі тенісні м'ячі, яких слід уникати.

ЗАВДАННЯ: ПОЛІПШЕННЯ ГРАВІТАЦІЇ

Є одна невелика помилка в грі: гравітація не тягне вниз ваш персонаж, якщо він торкається синьої платформи будь-якою своєю частиною, навіть головою! Ви можете побачити це після того, як персонаж підніметься вгору по стовпу (перемістіть його вліво). Ви можете виправити цю помилку? Щоб зробити це, вам треба поміняти колір брюк у спрайта на всіх образах і потім змінити код:



на:



В блоці КОЛІР..ТОРКАЄТЬСЯ перший параметр - це колір брюк, а другий - колір платформи.
Перевірте поліпшення, щоб переконатися, що все працює!

ЗАВДАННЯ: БІЛЬШЕ ЖИТТІВ

Чи можете ви дати гравцеві 3 життя, а не просто щоразу переміщати персонаж у початкову точку?

- Ваш гравець починає з 3-ма життями.
- Кожен раз, коли гравець отримує удар, він втрачає одне життя і повертається в початок.
- Якщо життів більше не залишилося, гра закінчується.

ЗАВДАННЯ: ЗАВЕРШЕННЯ РІВНЯ

Спробуйте додати більше коду для персонажа, щоб цей код описував його поведінку, коли персонаж дістався до дверей.
Ви навіть можете створити ще кілька фонів і переходити на наступні рівні після того, як ваш персонаж досягне дверей на одному з рівнів.

ЗАВДАННЯ: КЛЮЧ ВІД ДВЕРЕЙ

Щоб відкрити двері і перейти на інший рівень, потрібно знайти ключ. Додайте спрайт ключа і запрограмуйте нову поведінку дверей.

ЗАВДАННЯ: ГРЕМЛІНИ

По платформах можуть ходити маленькі істоти, яких ви повинні уникати. Додайте гремліна на кожен поверх лабіринту, використовуйте клони для цього.

ЗАВДАННЯ: КНОПКИ НА СТІНАХ

На кожному поверсі лабіринту додайте кнопку на стіні. Гравцю потрібно натиснути кожну кнопку, щоб було можливо відчинити двері. Кнопка натискається за допомогою стрибка.

УРОК 17. СТВОРИТИ ВЛАСНУ ГРУ

ПИТАННЯ ДЛЯ РОЗДУМІВ

- **Опис гри.** У вас має бути опис гри, сюжет та опис управління.
- **Управління в грі.** Вам потрібно обдумати те, як гравець гратиме в вашу гру. Ми робили різні типи управління: за допомогою мишки (коли персонаж слідує за мишкою), після натискання на клавіші, коли ви кудись натискаєте мишкою та інші.
- **Мета гри. Як в неї виграти.** Вам потрібно добре продумати, навіщо гравцеві грати в вашу гру, яка її мета і як гравець може перемогти у вашій грі.
- **Багаторазова гра.** Гравцю має бути цікаво грати в вашу гру більше одного разу. Зазвичай роблять або великі ігри, де багато рівнів, щоб можна було в них довго грати. Або створюють невеликі ігри, але в них кожен раз цікаво грати.
- **Привітання та пояснення гравцеві, як грати.** На початку гри треба привітати гравця і ввести його в гру, пояснити, що йому потрібно спочатку робити або розповісти сюжет/історію гри. Також можливі підказки, коли гравець намагається грати вперше.
- **Скільки рівнів у грі.** Чи буде у вас багато рівнів або один, що буде постійно змінюватися?
- **Гра на 1 чи 2 гравців або більше.** Скільки гравців може грати в вашу гру? Вони грають одночасно або по черзі?
- **Спрайти та фони.** Потрібно буде знайти в інтернеті або намалювати спрайти і фони для вашої гри.

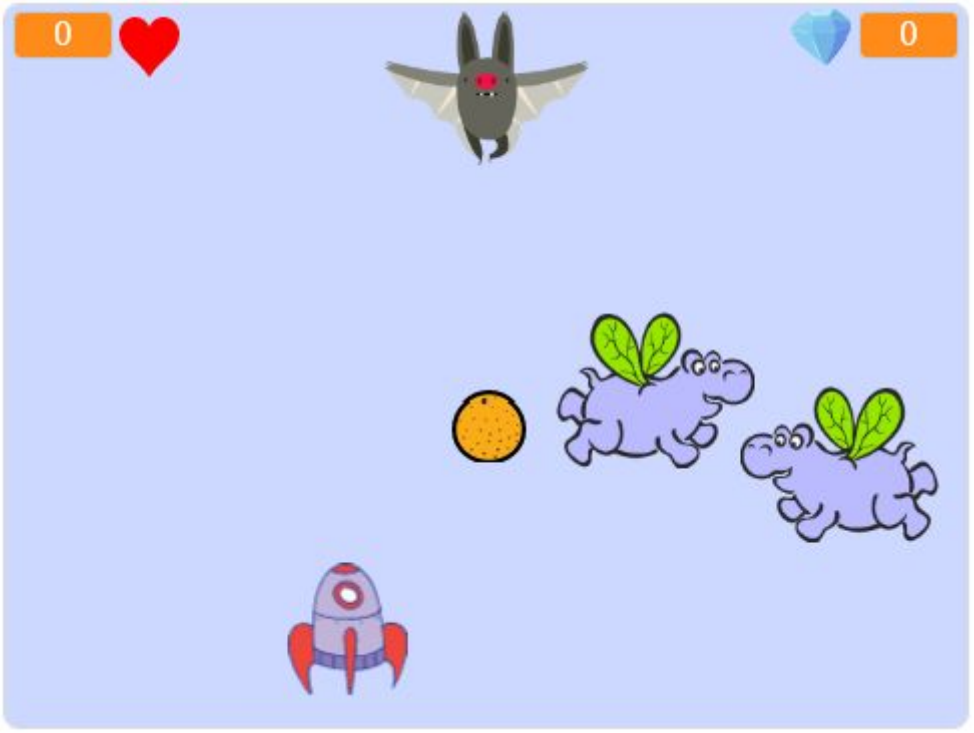
Простіше шукати англійською мовою. Додавайте в кінці слово: png. Так буде легше знайти картинки.

ЖАНРИ ІГОР

- **Платформери.** У таких іграх зазвичай є кілька рівнів. У персонажа є здоров'я, він збирає якісь монетки/кристали протягом декількох рівнів. Також є вороги, від яких потрібно ухилятися або стрибати зверху. Зазвичай є мета, чому головний герой проходить через всі ці рівні.
- **Гонки.** Спочатку в такій грі вибирається машина на основі різних характеристик, потім - траса. Можна їздити з іншим гравцем або з противником-комп'ютером. Також це може бути будь-який вид транспорту, який підходить для гонок.
- **Військові (бійки).** Є можливість вибору декількох персонажів, які борються один з одним. Є кілька раундів, і за кількістю виграних раундів визначається переможець. Є здоров'я, прийоми, супер прийоми.
- **Квести (Quest, Hidden Objects).** У таких іграх дуже важливий сюжет/історія. Також потрібно придумати фони та предмети, які потрібно шукати. Додати логічні завдання, загадки.
- **Аркади.** Аркади - це по суті простенькі ігри. В принципі вони бувають абсолютно різними.
- **Карткові ігри.** Гравці грають один з одним наборами карт. У кожної карти є характеристики (сила, здоров'я, захист).
  Карти викладаються на стіл, і відбувається якась агресивна дія. Зазвичай є обмеження на кількість карт, які можна викласти.
- **Покрокові стратегії.** Тут персонажі можуть подорожувати по світу, щось досліджувати, але коли справа доходить до бою, то гра переходить у покроковий режим. Зазвичай у кожного гравця є набір персонажів, якими вони можуть управляти. У кожного такого персонажа є характеристики (здоров'я, сила атаки, магія, захист). І вони можуть по черзі атакувати один одного.

УРОК 18. КОСМІЧНІ ВІЙНИ КЛОНІВ

У цьому проекті ви дізнаєтеся, як створити гру, в якій вам доведеться рятувати Землю від космічних монстрів.

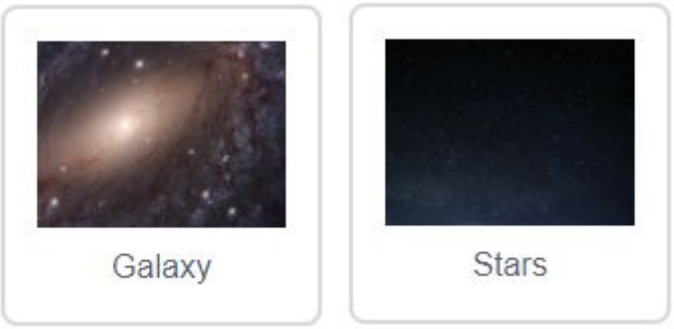


На цьому уроці ми будемо використовувати:

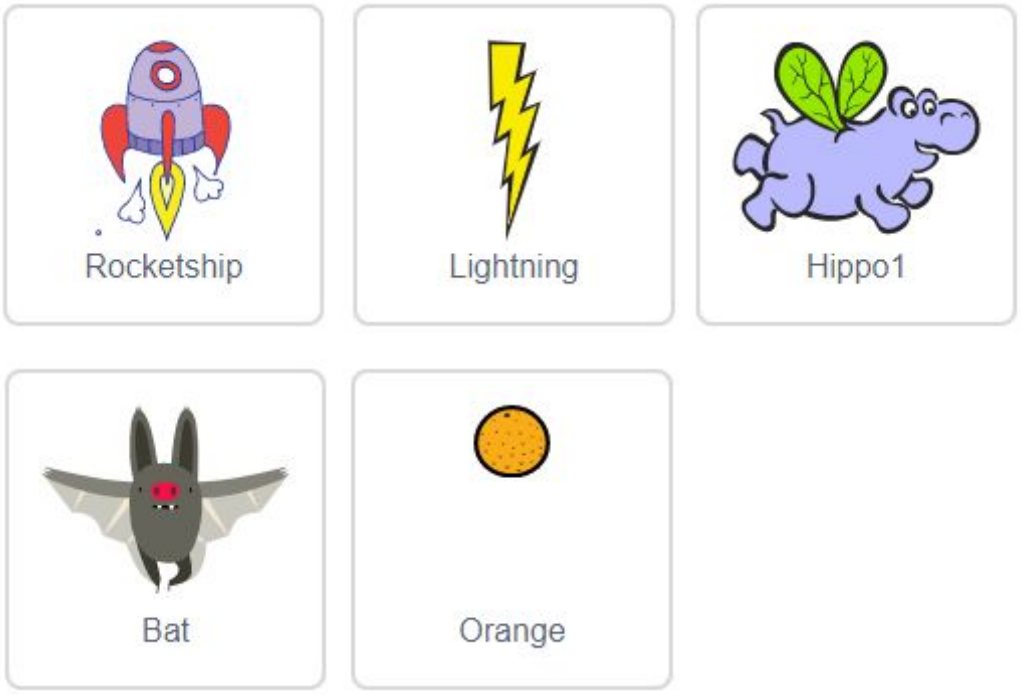
1. Змінні.
2. Повідомлення.
3. Клони.

КРОК 1: СПРАЙТИ ТА СЦЕНА

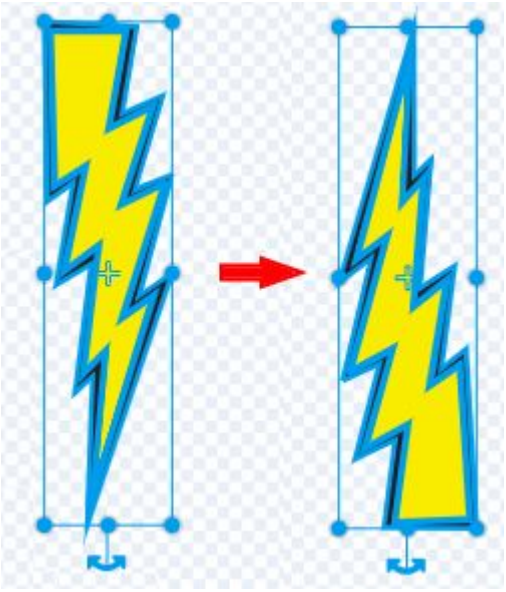
- Створіть новий проект і видаліть спрайт кота.
- Додайте в свій проект тло зоряного неба. Наприклад:



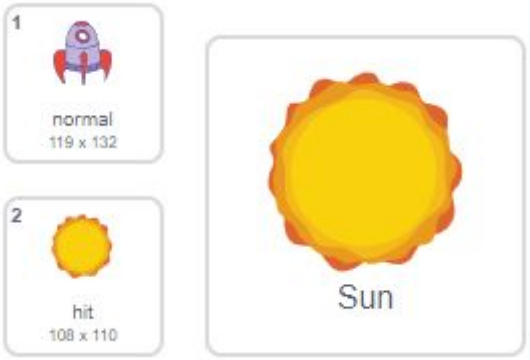
- Додайте з бібліотеки спрайти космічного корабля, блискавки, бегемота, кажана, апельсина.



- Треба зменшити розмір космічного корабля і перемістити його в нижню частини сцени. Він має бути по центру.
- Кажан буде постійно переміщатися по верхній частині сцени із лівого кутка в правий. Розмістить його там.
- Апельсинами будуть стріляти кажани. Сховайте основний спрайт апельсина, нам будуть потрібні лише його клони.
- Натисніть на образ спрайта блискавки і переверніть його.



- У космічного корабля повинно бути два образи: **normal** (нормальний) і **hit** (вдарений). Вдарений образ космічного корабля можна зробити, імпортувавши зображення Sun (сонце) з бібліотеки.



- Намалюйте спрайт під назвою **gameOver** (кінець гри), використовуючи текстовий інструмент.

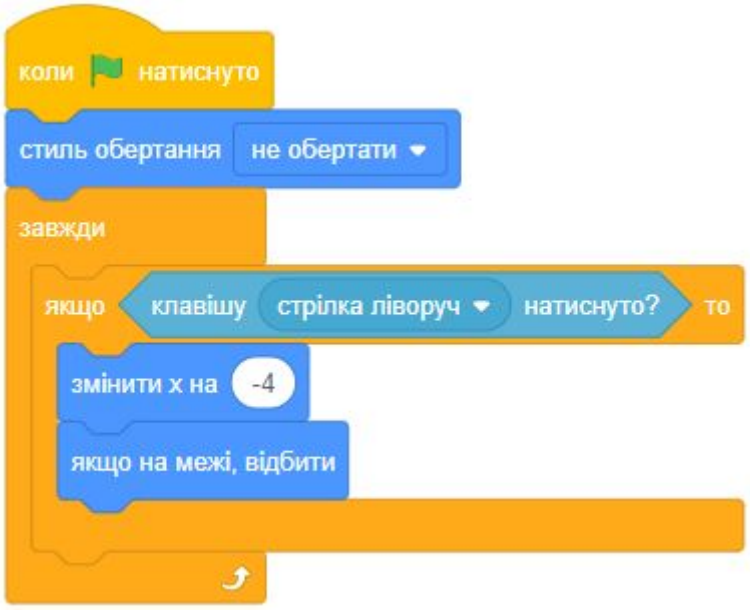
- Вам також може допомогти сайт https://cooltext.com/

- Замість запропонованих спрайтів, ви можете додати інші.

КРОК 2: КОСМІЧНИЙ КОРАБЕЛЬ

Давайте створимо космічний корабель для захисту Землі!

- Додайте скрипт, щоб перемістити **космічний корабель** вліво при натисканні клавіші зі стрілкою вліво.



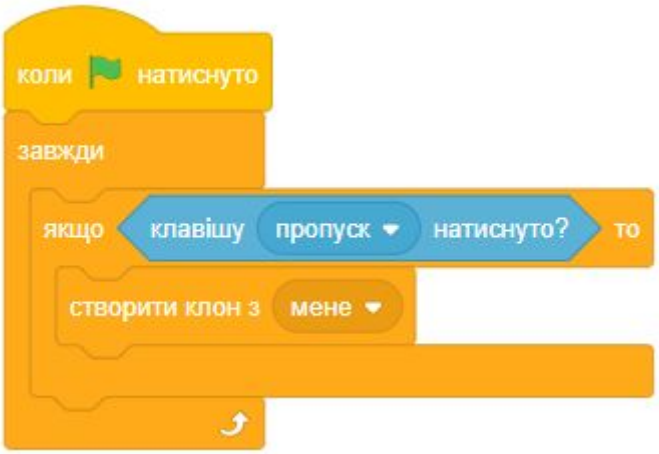
- Самостійно додайте скрипт для переміщення космічного корабля вправо при натисканні клавіші зі стрілкою вправо.
- Протестуйте проект, щоб дізнатися, чи можете ви керувати космічним кораблем за допомогою клавіш зі стрілками.

КРОК 3: БЛИСКАВКА

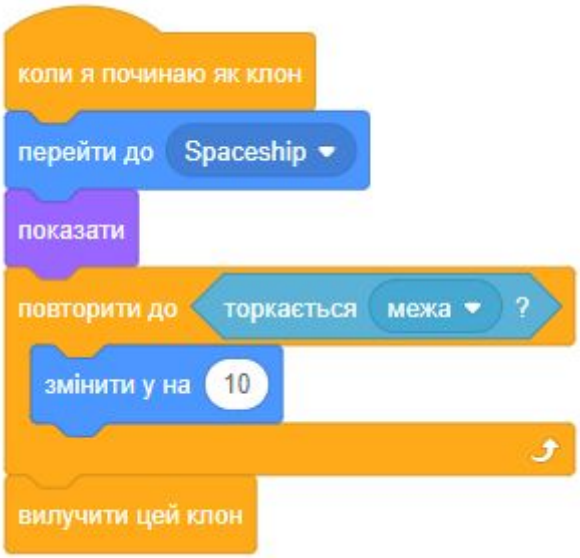
Давайте дамо космічному кораблю можливість стріляти блискавками!

- Коли гра запущена, блискавку потрібно приховати, поки космічний корабель не запустить свої лазерні гармати. Додайте скрипт до **блискавки**.

- Додайте до **блискавки** скрипт, щоб створювати новий клон блискавки при кожному натисканні клавіші ПРОПУСК.



- Кожен раз, коли створюється новий клон, він повинен з'являтися з того ж місця, що і космічний корабель, а потім підніматися по сцені, поки не торкнеться краю. Додайте скрипт до **блискавки**:

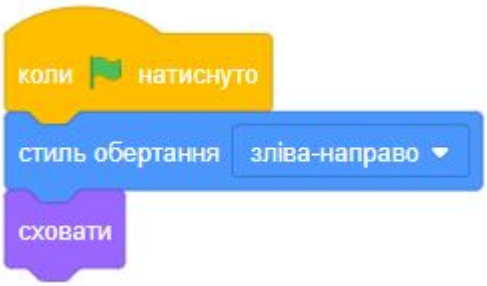


- Примітка: ми рухаємо новий клон на космічний корабель, поки він ще прихований, перш ніж показувати його.
- Перевірте, як все працює, натиснувши ПРОПУСК.

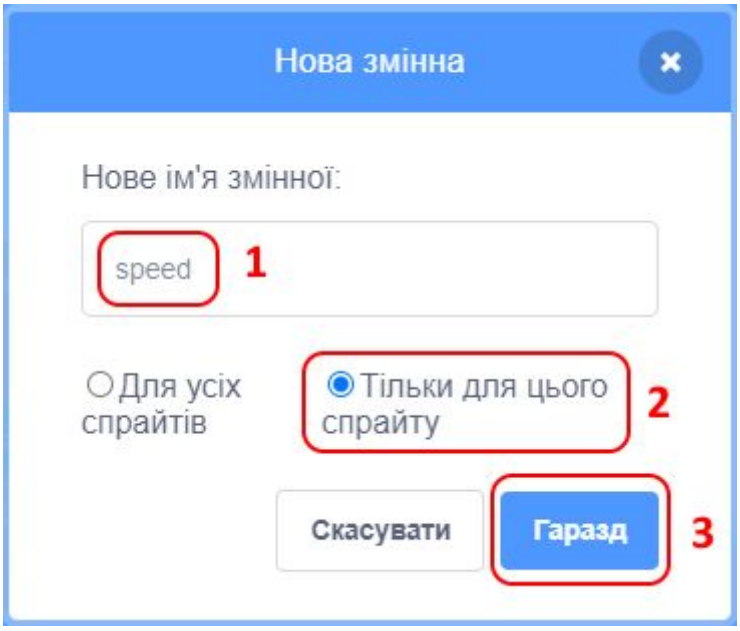
КРОК 4: ПОЛЬОТИ БЕГЕМОТІВ

Давайте додамо безліч літаючих бегемотів, які намагаються знищити космічний корабель.

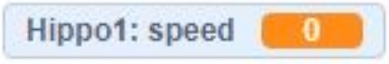
- Додайте скрипт до **бегемоту**, щоб приховати його при запуску гри. Також цей скрипт встановлює стиль обертання **зліва-направо**.



- Створіть нову змінну з ім'ям **speed** (швидкість), яка призначена тільки для спрайта бегемота.



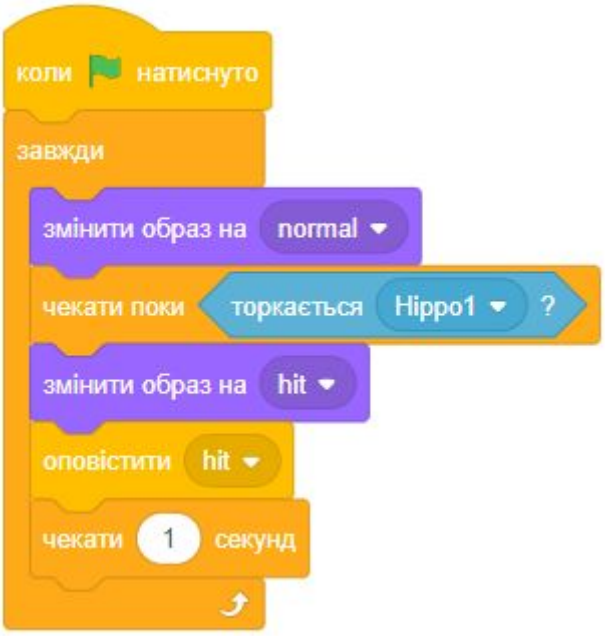
- Ви дізнаєтеся, чи правильно зробили це, тому що поруч зі змінною на сцені буде ім'я спрайта.



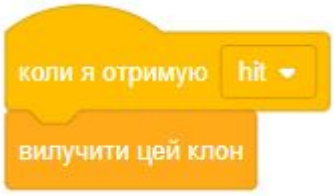
- Наступний скрипт буде створювати нового бегемота кожні 2-4 секунди. Додайте скрипт до **бегемота**.

- Коли кожен клон бегемота стартує, змусьте його переміщатися по сцені (з випадковою швидкістю), поки його не вдарить блискавка. Додайте скрипт у спрайт **бегемота**.

- Перевірте скрипт бегемота. Ви повинні бачити, як кожні кілька секунд з'являється новий клон бегемота, кожен з яких рухається зі своєю швидкістю.
- Перевірте лазер. Якщо ви вдарили ним по бегемоту, він зникне?
- Коли бегемот торкається космічного корабля, нам потрібно змусити космічний корабель вибухнути!
- Додайте скрипт у **космічний корабель**, щоб він змінював свій образ при зіткненні з літаючим бегемотом.



- Ви помітили, що в наведеному вище скрипті надсилається повідомлення hit (потрапляння)?
- Ви можете використовувати це повідомлення, щоб бегемоти зникали при влученні в космічний корабель.
- Додайте скрипт до **бегемота**:

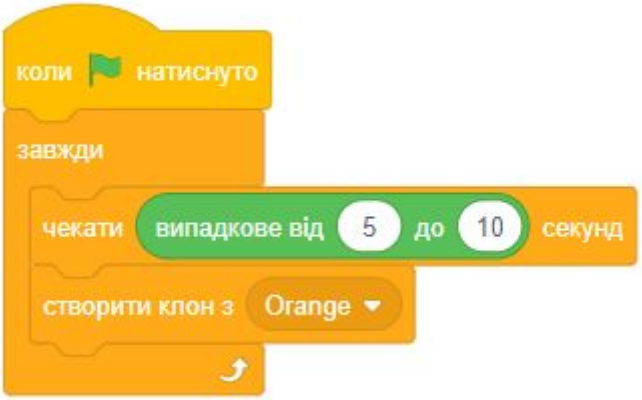


- Перевірте скрипти, запустивши гру і зіткнувшись з бегемотом.

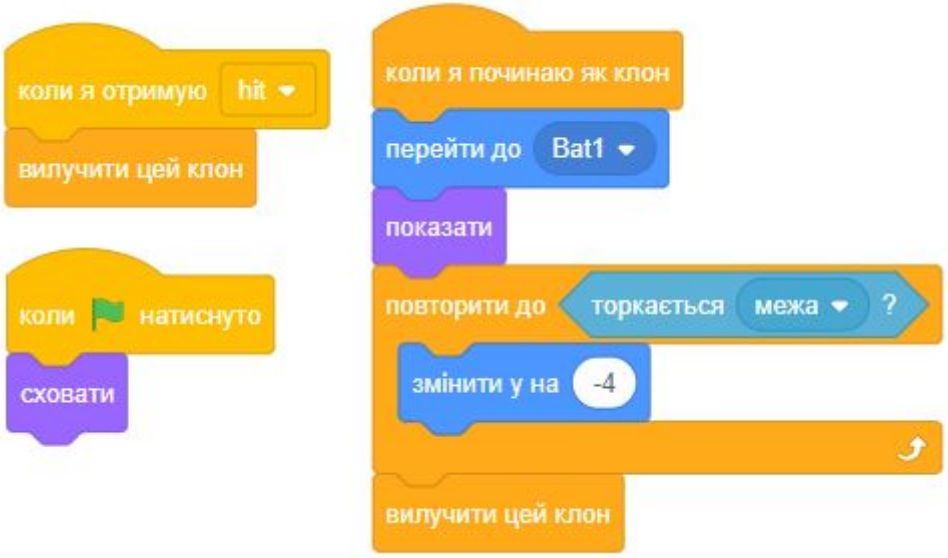
КРОК 5: ФРУКТОВІ КАЖАНИ

Давайте зробимо кажана, який кидає апельсини в корабель.

- Зробіть так, щоб спрайт кажана постійно літав по верхній частині сцени з лівого краю до правого і назад.
- У кажана є безліч образів. Залиште лише перші два образи, видаливши інші.
- Використовуйте блок переходу до наступного образа, щоб змусити кажана махати крилами під час руху.
- Додайте скрипт до **кажана,** щоб він створював новий клон апельсина кожні кілька секунд.



- Натисніть на спрайт **апельсина** і додайте скрипти, щоб кожен клон апельсина рухався по сцені від кажана до корабля.



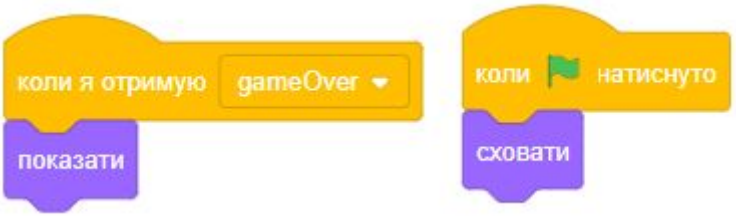
- В скрипті **космічного корабля** вам потрібно змінити блок ЧЕКАТИ ПОКИ, щоб обробляти дотик бегемота або апельсина:

- Перевірте гру. Що станеться, якщо вас вдарить апельсин?

КРОК 6: ГРУ ЗАКІНЧЕНО

- Створіть нову змінну з ім'ям **life** (життя). Космічний корабель повинен починати з трьома життями і втрачати їх при зіткненні з противником. Ваша гра також повинна зупинитися, коли у вас закінчаться життя.
- На **сцені** передайте **gameOver** подію відразу перед кінцем гри.

- Додайте скрипт у спрайт **gameOver,** щоб він показувався в кінці гри:



- Перевірте гру. Скільки очок ви можете набрати?
- Чи можете ви придумати способи поліпшити гру, якщо вона занадто проста чи занадто складна?

КРОК 7: ЗАВДАННЯ

ЗАВДАННЯ: ВИПРАВИТИ ПОМИЛКУ БЛИСКАВКИ

Що станеться, якщо тримати натиснутою клавішу пропуск? Ви можете використати блок очікування, щоб виправити цю помилку?

ЗАВДАННЯ: ІНДИКАТОР ЖИТТІВ ТА РАХУНКУ

Чи можете ви додати в свою гру **life** (життя), **score** (рахунок) і **highScore** (кращий рахунок)?
У цьому вам допоможуть попередні ігрові проекти.

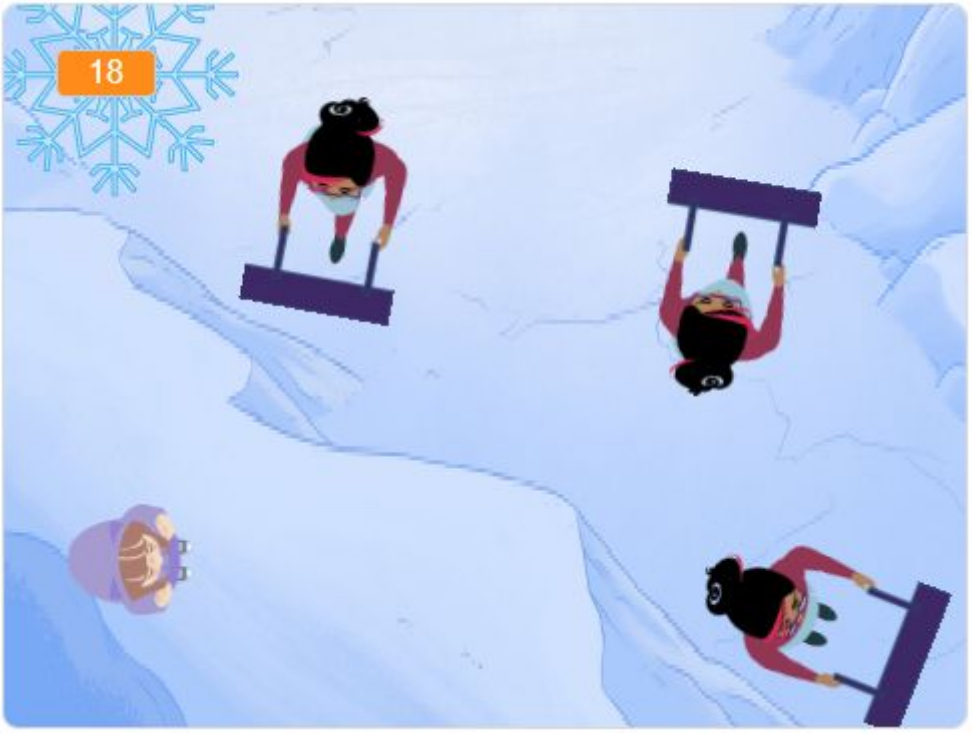
ЗАВДАННЯ: ПОЛІПШЕННЯ ГРИ

Які поліпшення ви можете додати? Кілька ідей:

- Додайте аптечки, які можна зібрати, щоб отримати додаткові життя для космічного корабля.
- Додайте метеорити, яких космічний корабель повинен уникати.
- Зробіть так, щоб з'явилися нові вороги, коли рахунок буде 100.
- Створіть стартове тло з назвою гри та інструкцією.
- Що ще можна зробити?

УРОК 19. СНІГУРКА ПРОТИ ДВІРНИКІВ

Зима. Падає сніг. Двірники намагаються прибрати весь сніг. Снігуроньці це не подобається. Чи для того вони з Дідом Морозом його розсипали, щоб його хтось прибирав. Снігуронька вирішила завадити двірникам, заморожуючи їх.



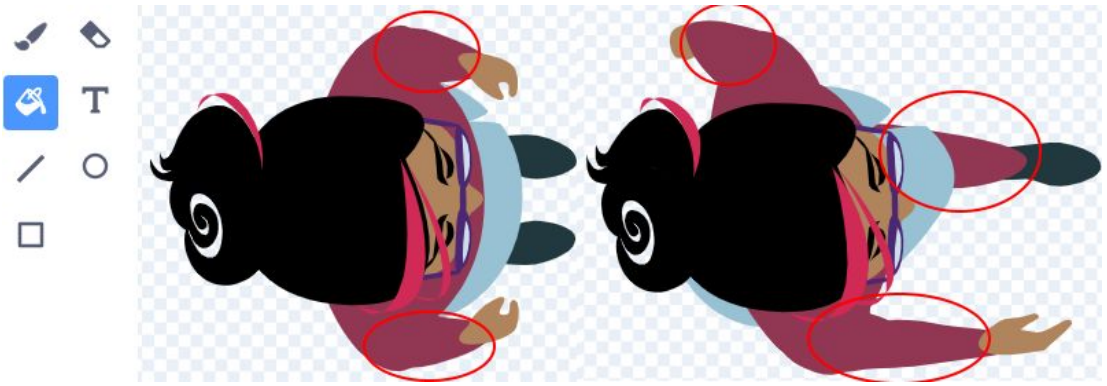
- Заморожені двірники не прибирають сніг.
- Магія снігуроньки діє тільки п'ять секунд.
- Керуючи стрілками, треба спіймати двірників і заморозити їх, щоб завадити прибирати сніг.
- Доведіть кількість снігу до максимально можливої за 60 секунд!

КРОК 1: СПРАЙТИ ТА СЦЕНА

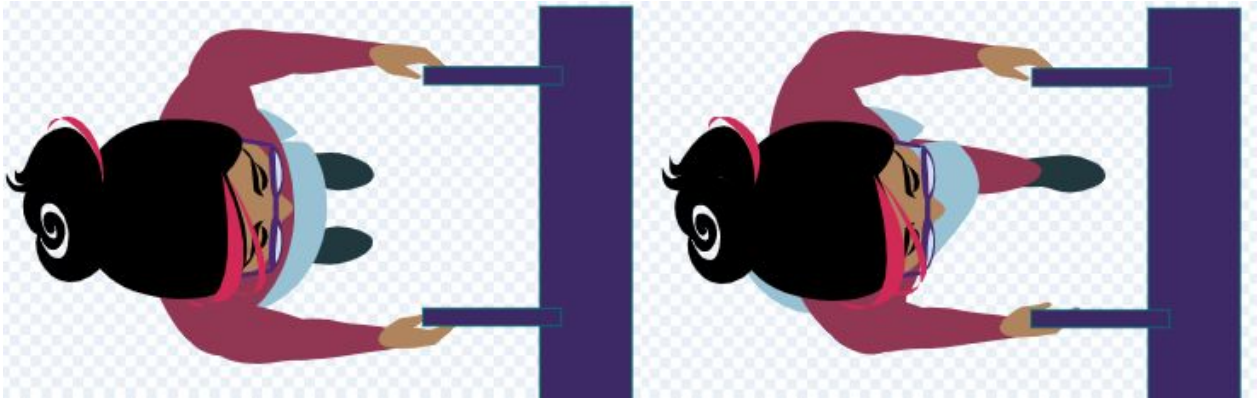
- Створюємо новий проект, видаляємо звідти кота.
- Додайте спрайт снігуроньки. Це може бути будь-який спрайт із бібліотеки. Я обрав спрайт Tatiana, він має образи для анімації ходьби та намальований, як нам потрібно (від зверху). Додатково ви можете відредагувати його, щоб ця дівчинка була більш схожа на снігуроньку. Але це може бути снігурка, яка не хоче, щоб її впізнали, тому вона й має не зовсім звичний вигляд.

- Додайте спрайт двірника. Це також може бути будь-який спрайт з бібліотеки. Я обрав цей:

- Двірника потрібно домалювати, одягнувши в теплий одяг. Я одягнув усі образи спрайта, сховавши руки і ноги:

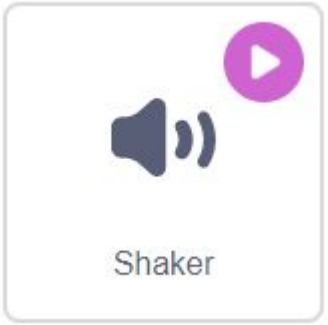


- Двірнику треба намалювати лопату на всіх образах. Крім цього треба змінити положення рук, щоб у них була лопата.



- Намалюйте **тло сцени**. На білому кольорі намалюйте кілька невеликих заметів. Вони будуть декоративними, не будуть змінюватися і заважати двірникам переміщатися по сцені.
  А можете намалювати щось цікавіше, наприклад:

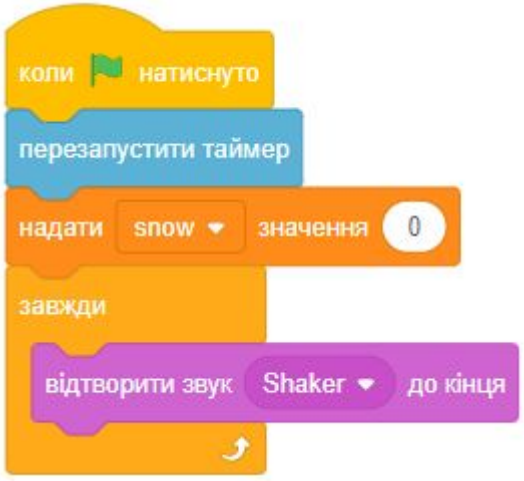
- Додайте звук Shaker (або інший) до сцени.



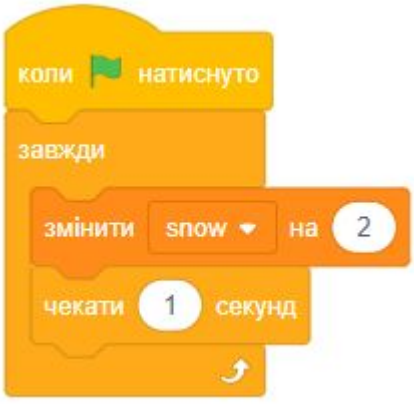
КРОК 2: СКРИПТИ ДЛЯ СЦЕНИ

- Створіть змінну **snow** (сніг). Вона буде показувати, скільки снігу нападало у дворі. Завдання гравця - збільшити її значення.
- Розмістить змінну snow у лівому верхньому кутку сцени. Додайте спрайт сніжинки поруч зі змінною, щоб гравцю було зрозуміло.

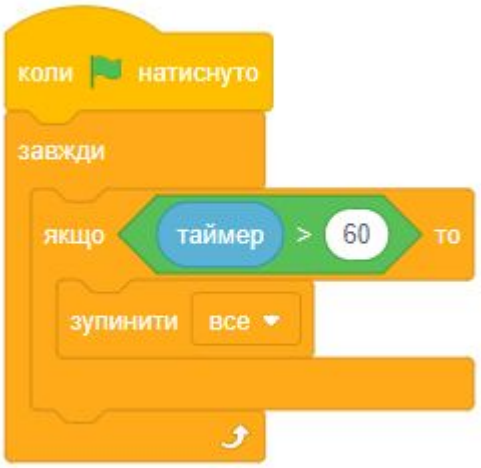
- Перший скрипт **сцени**. Обнуляємо таймер і змінну **snow** (сніг). Весь час гри звучить музика.



- Другий скрипт **сцени**. Падає сніг, щосекунди значення змінної **snow** збільшується на 2.



- Третій скрипт **сцени** через хвилину зупинить гру.

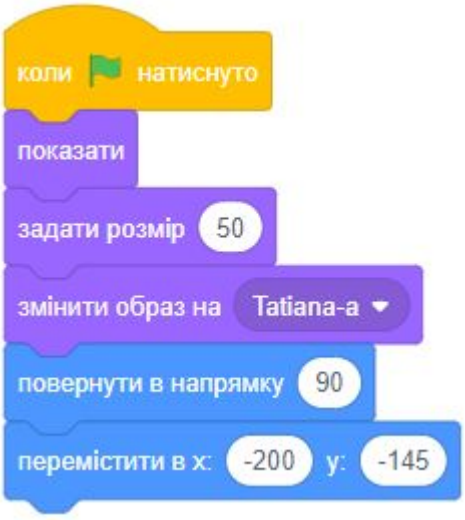


КРОК 3: СНІГУРОНЬКА

- У снігуроньки буде скрипт, який управляє переміщенням по сцені. Ми анімуємо спрайт, його образи змінюються, коли він рухається по сцені.

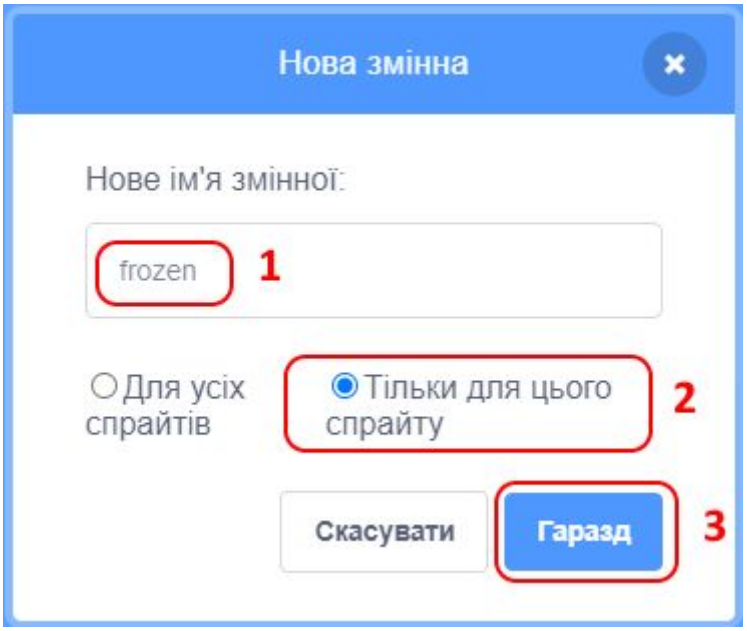
- Вам треба самостійно зробити скрипт для повороту ліворуч.
- Також додайте скрипт до **снігуроньки**, який запускається на початку гри та виконує певні дії.

- Скрипт встановлює початкове положення спрайта на сцені, його образ та напрямок руху. Також розмір спрайта зменшується, щоб було цікавіше грати.

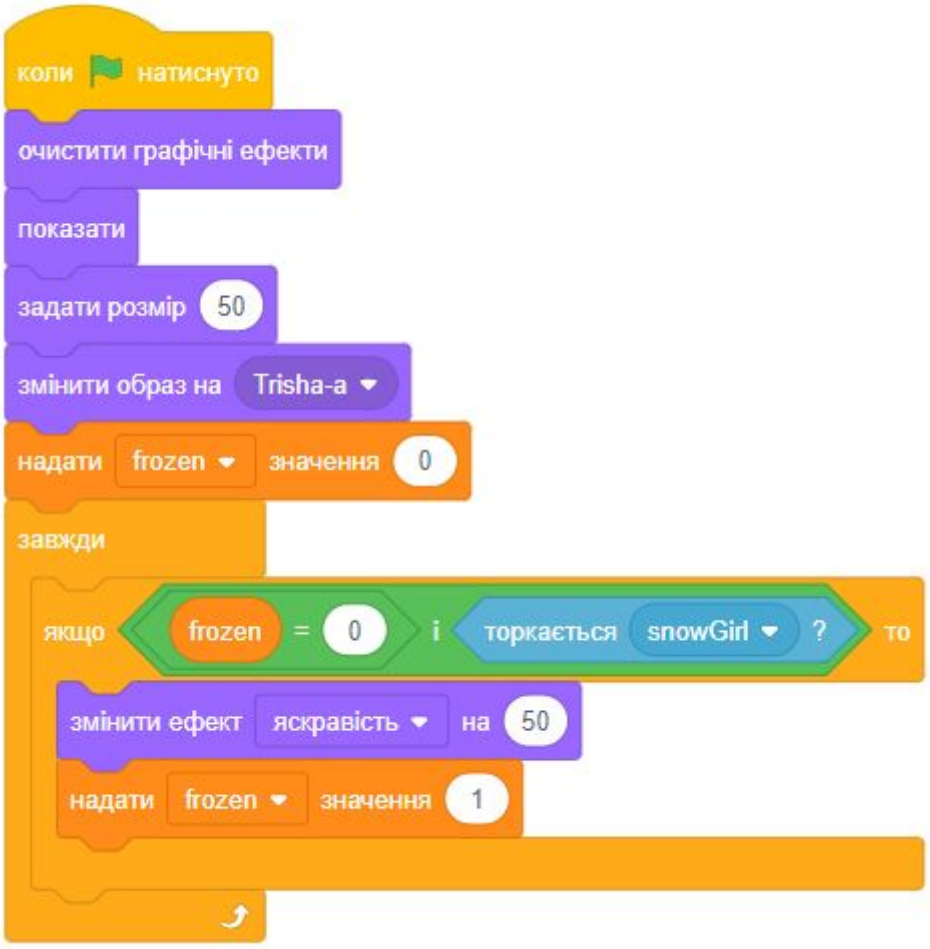


КРОК 4: ДВІРНИК

- Перш ніж зробити перший скрипт двірника, треба створити для нього локальну змінну **frozen** (заморожений).
  Ця змінна буде приймати значення 1 або 0. Якщо вона має значення 1, то двірник заморожений, і не працює. Якщо вона дорівнює 0, то двірник не замерз і продовжує розчищати сніг.
- Змінна обов'язково повинна бути локальною, бо вона відповідає за стан тільки одного двірника, якому належить.

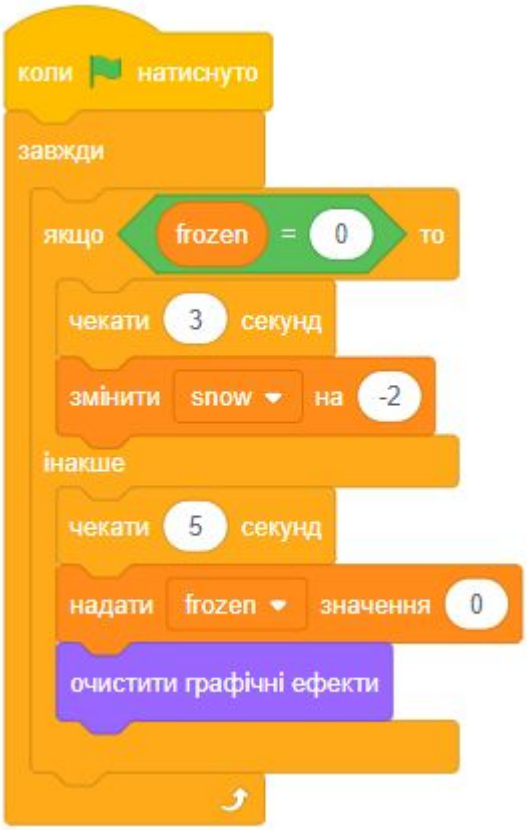


- Перший скрипт **двірника.** Що він робить?
  - На початку двірник одягає звичайний робочий костюм.
  - Локальна змінна frozen встановлюється в значення 0 (двірник не заморожений).
  - Потім постійно перевіряється виконання подвійної умови - чи не заморожений двірник торкається снігуроньки?
  - Якщо дві умови одночасно виконуються, то змінна frozen тимчасово встановлюється в значення 1, і двірник одягає заморожений костюм (за допомогою ЗМІНИТИ ЕФЕКТ “ЯСКРАВІСТЬ” на 50).

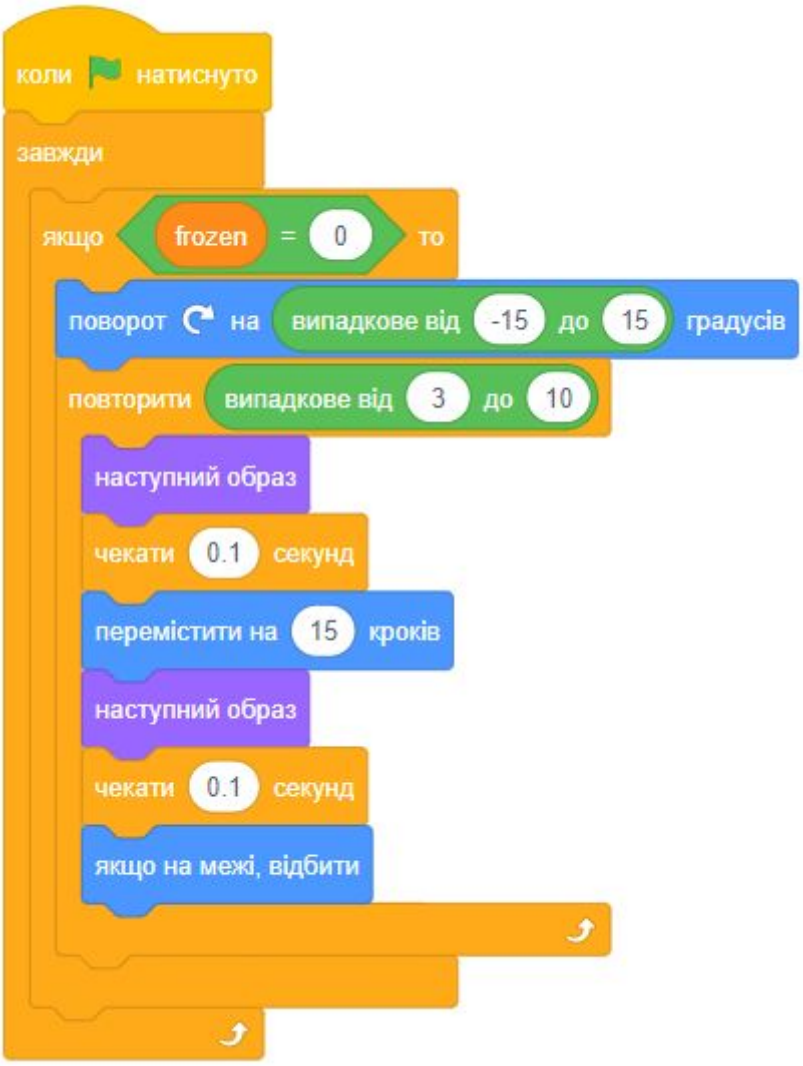


- Другий скрипт **двірника.**
  - Завжди перевіряє одну умову - заморожений двірник чи ні.
  - Якщо двірник не заморожений, він прибирає сніг - це зменшує значення змінної snow на "-2". Прибирання триває три секунди.
  - Якщо ж двірник знаходиться в замороженому стані, то він не прибирає сніг, а чекає п'ять секунд і відтає.

- Переодягання в основний образ робимо за допомогою блока ОЧИСТИТИ ГРАФІЧНІ ЕФЕКТИ.



- Математика в нашій грі така: кожен двірник прибирає за три секунди 2 одиниці снігу. Три двірника за три секунди прибирають 6 одиниць снігу. За цей же час скрипт сцени збільшить значення змінної snow на 6. Таким чином, значення змінної snow збільшиться на 6 і зменшиться на 6, тобто залишиться без змін. Проте якщо снігуронька заморозить деяких двірників, то вони не прибирають сніг, і змінна **snow** збільшиться.
- Третій скрипт **двірника** дає можливість йому пересуватися по сцені у випадкове положення. Звісно, якщо він не заморожений.
- Коли двірник рухається, він анімується за допомогою зміни його образів. Двірники рухаються в межах сцени, тому ми додали відштовхування від меж.



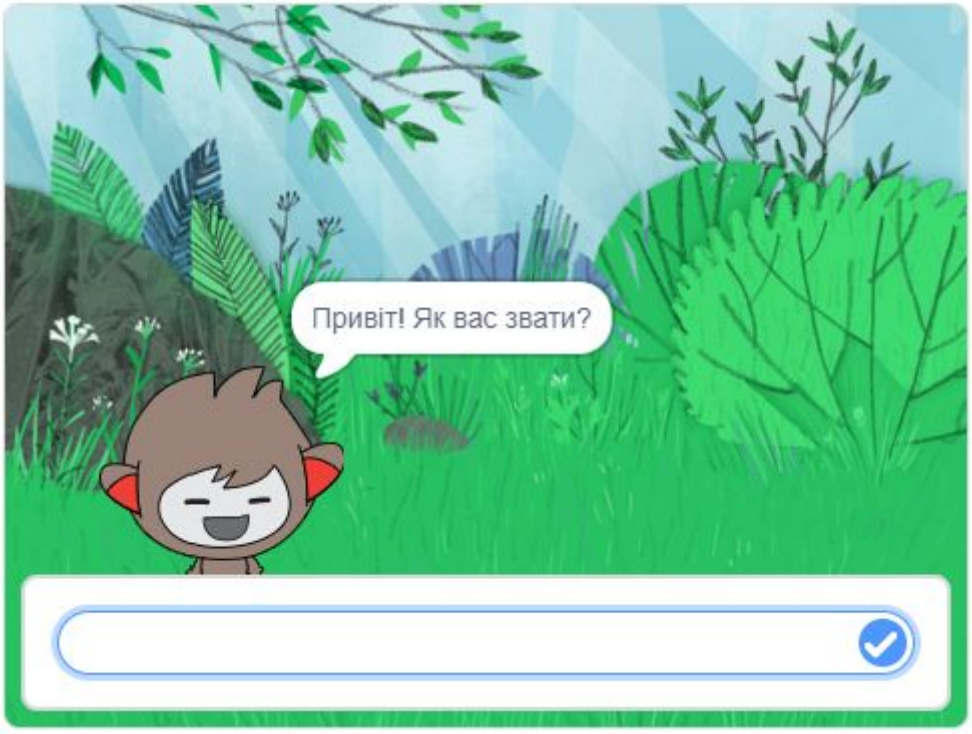
- Дублюйте спрайт двірника два рази. З'являться ще два нових спрайта двірників. Скрипти всіх трьох двірників будуть однаковими, у кожного з них буде своя локальна змінна frozen.
- Гра готова! Протестуйте, що все працює як треба.

КРОК 5: ЗАВДАННЯ

- Створіть таку ж гру з використанням клонів.
- Зробіть початковий екран з привітанням, назвою гри та інструкцією користувача.
- Запрограмуйте обчислення найкращого рахунку, як ми робили в попередніх проектах.
- Додайте інші удосконалення.

УРОК 20. ЧАТБОТ З ПИТАННЯМИ

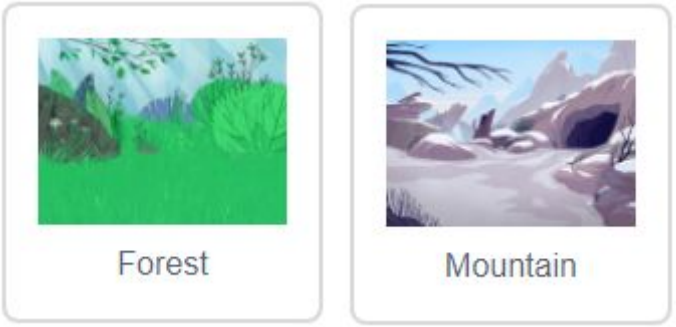
У цьому проекті ми збираємося навчитися програмувати свого власного мовця робота!



КРОК 1: СПРАЙТИ ТА СЦЕНА

- Почніть новий проект і видаліть з нього спрайт кота.
- Виберіть з бібліотеки спрайт, який більше вам підходить, і додайте його в свій проект. Наприклад:

- Виберіть з бібліотеки тло, яке підходить для вашого чатбота. Наприклад:

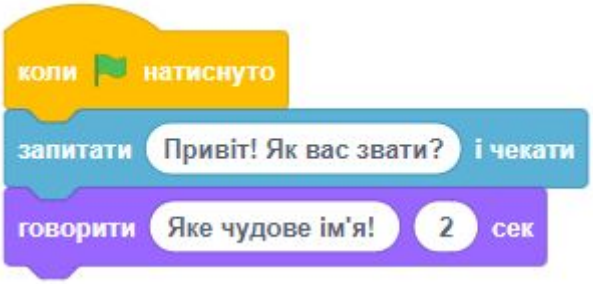


- Також додайте ще одне тло - місце куди спрайт буде подорожувати. Це може бути що завгодно. Наприклад:

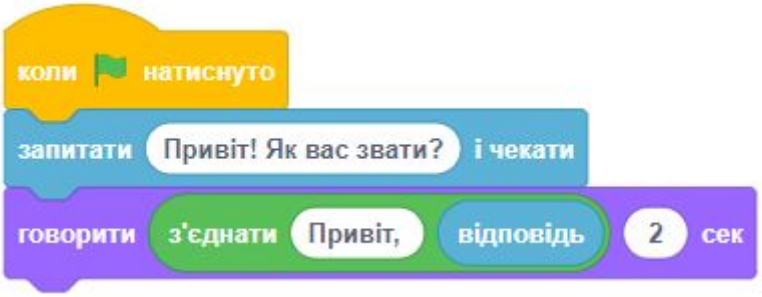
КРОК 2: ГОВОРИТЬ ЧАТБОТ

Давайте запрограмуємо чатбот так, щоб він міг з вами поспілкуватися.

- Натисніть на персонаж чатбот і додайте такий скрипт:



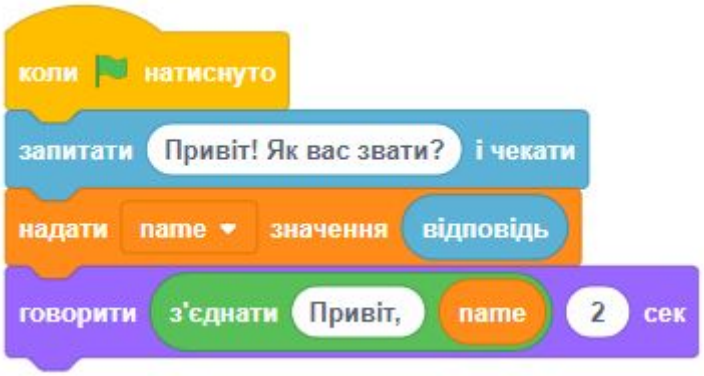
- Запустіть гру. Після того як чатбот запитає ваше ім'я, введіть його в поле, що з'явилося в нижній частині сцени. Ваш чатбот буде відповідати: **Яке чудове ім'я!**
- Ви можете створити його розумнішим, зробивши так, щоб він використовував дані з відповідей. Змініть скрипт у чатбота, як показано нижче:



- Для того щоб створити останній блок, вам необхідно взяти команду З'ЄДНАТИ (категорія ОПЕРАТОРИ) і перетягнути її в ГОВОРИТИ. Потім ви можете замінити перший текст на ПРИВІТ і перетягнути блок ВІДПОВІДЬ з команд категорії ДАТЧИКИ замість другого тексту.

- Протестуйте вашу гру. Вона працює так, як ви очікували? Чи можете ви виправити помилки, які виявили? Підказка: Ви можете спробувати додати подекуди пробіл!
- Інформація, яку ви пишете, буде збережена в спеціальній змінній **відповідь**. Перейдіть до групи команд СЕНСОРИ і позначте блок **відповідь** так, щоб поруч з ним з'явилася галочка. Поточне значення, яке зберігається у **відповідь,** має з'явитися в верхньому лівому кутку сцени.
- Ви можете запам'ятати ім'я користувача за допомогою змінної.
- Створіть нову змінну і назвіть її **name** (ім'я).

Після того як ви її створили, переконайтеся, що ваш скрипт виглядає так:



- Якщо ви знову протестуете свою програму, то виявите, що відповідь зберігається в змінній **name** і відображається у верхньому лівому кутку сцени. В змінній **name** будуть ті самі дані, що і в змінній **відповідь**. Якщо ви не хочете бачити змінну на сцені, просто натисніть на галочку поруч з її ім'ям.

КРОК 3: УХВАЛЕННЯ РІШЕНЬ

- Давайте зробимо так, щоб чатбот міг відповісти на запитання користувача «**так**» чи «**ні**». Зверніть увагу, що тепер ім'я збережено в змінній, і ви можете використовувати його.

- Щоб протестувати, вам треба перевірити цей код двічі - перший раз введіть відповідь «**ні**», вдруге - відповідь «**так**». Ви повинні отримати відповідь від чатбота, тільки якщо ви відповіли «**так**».
- Не дуже добре, що чатбот не дає відповіді, якщо користувач відповідає «**ні**». Ви можете виправити це, змінивши блок ЯКЩО на блок ЯКЩО..ІНАКШЕ, щоб код виглядав так:

- Якщо ви перевірите ваш код, то побачите, що тепер ви отримуєте відповідь і коли відповідаєте «**так**», і коли відповідаєте «**ні**».
- Чатбот повинен відповісти «**Радий це чути!**», коли ви відповідаєте «**так**», або «**О, ні!**», якщо відповісти щось інше.
- Ви можете використовувати блок ЯКЩО..ІНАКШЕ не тільки для того, щоб чатбот говорив. Наприклад, ви можете змінити образ чатбота в залежності від відповіді.
- У чатбота має бути багато образів. Якщо ні, ви можете додати ще образів! Ви можете використовувати ці образи, щоб урізноманітнити відповідь чатбота, додавши зміни в ваш скрипт:

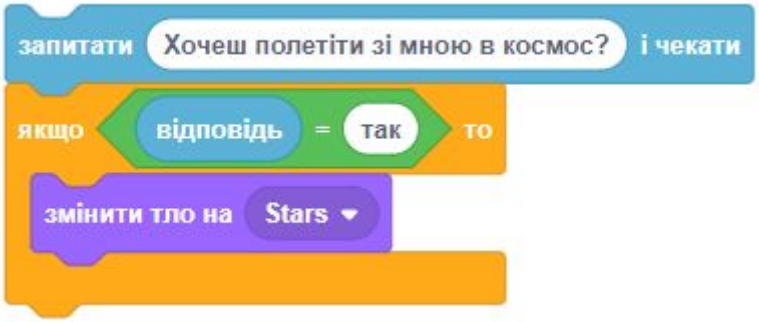
- У кожного спрайта свої образи, тому назви образів у блоках ЗМІНИТИ ОБРАЗ залежать від спрайта, яким ви користуєтесь.
- Протестуйте вашу гру. Ви повинні побачити, що обличчя вашого чатбота змінюється в залежності від відповіді.

КРОК 4: ЗМІНА МІСЦЯ РОЗТАШУВАННЯ

Тепер ви можете запрограмувати ваш чатбот, так щоб він змінював своє місце розташування, подорожуючи у космос або до інших планет.

- Переконаємося, що на початку роботи програми чатбот знаходиться не в космосі. Додайте скрипт до чатботу:

- Тепер додайте цей код знизу головного скрипта вашого чатбота:



- Перевірити вашу програму і дайте відповідь «**так**», коли чатбот запитає, чи хочете ви полетіти у космос. Ви повинні побачити, що чатбот поміняв своє місце розташування. Чи поміняв чатбот своє місцезнаходження, якщо ви відповіли «**ні**»? Що станеться, якщо ви надрукуєте щось інше?
- Ви також можете додати код всередині блока ЯКЩО, так щоб чатбот пострибав 4 рази вгору і вниз:

- Перевірте скрипт. Чи стрибає чатбот вгору і вниз?

КРОК 5: ЗАВДАННЯ

ЗАВДАННЯ: БІЛЬШЕ ПИТАНЬ

Запрограмуйте чатбот, щоб він ставив більше запитань.
Чи можете ви зберегти відповіді в нових змінних?

ЗАВДАННЯ: БІЛЬШЕ РІШЕНЬ

Запрограмуйте чатбот так, щоб він ставив більше запитань з відповідями «**так**» чи «**ні**». Чи можете ви зробити так, щоб чатбот реагував на відповідь?

ЗАВДАННЯ: ЗРОБИТИ СВІЙ ВЛАСНИЙ ЧАТБОТ

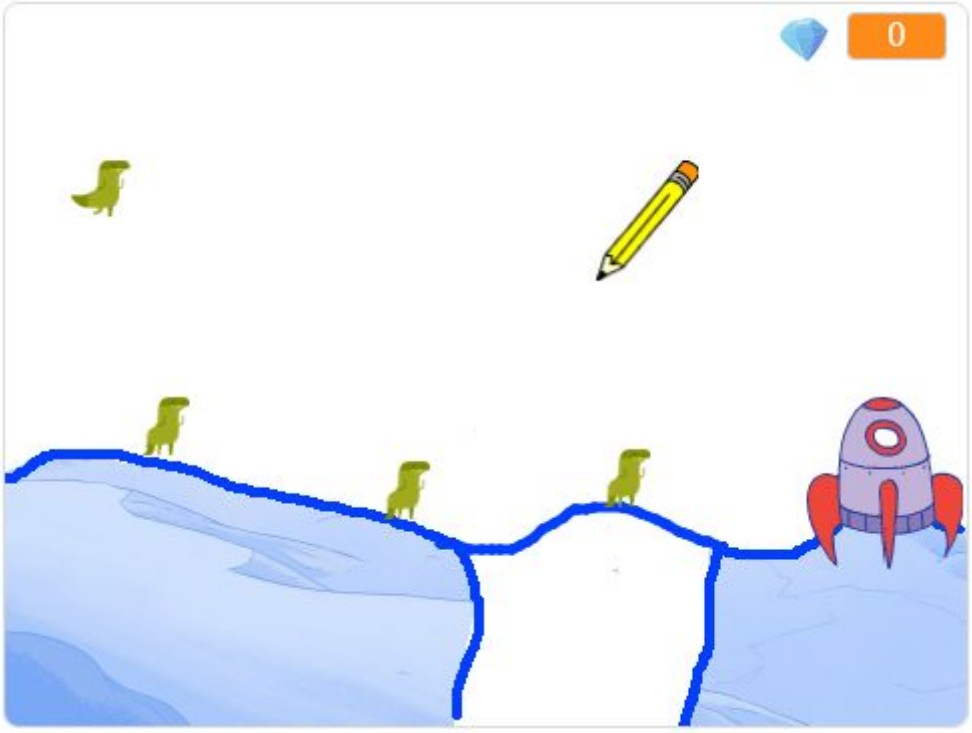
Використовуйте ваші нові знання, щоб запрограмувати інтерактивного чатбота самостійно.

1. Створити проект з новим чатботом.
2. Бот повинен назвати своє ім'я, дізнатися ваше ім'я та вік.
3. Бот повинен міняти образ при відповіді.
4. Бот повинен запропонувати заспівати разом із ним (і співати 5 секунд, якщо відповідь «**так**»).
5. Бот повинен запропонувати станцювати (і танцювати 5 секунд, якщо відповідь «**так**»). Підказка: для танцю можна використовувати зміну положення спрайта по осі X.
6. Бот повинен запропонувати стати на голову (і стояти на голові 5 секунд, якщо відповідь «**так**»).

Як тільки ви закінчите створення свого чатбота, дайте вашим друзям поспілкуватися з ним! Чи сподобався їм ваш персонаж? Чи виявили вони будь-які проблеми?

УРОК 21. ОЛІВЕЦЬ РЯТУЄ ДИНОЗАВРІВ

У цьому проекті ми створимо гру, в якій треба проводити динозаврів у безпечне місце та не давати їм потрапити в прірву!



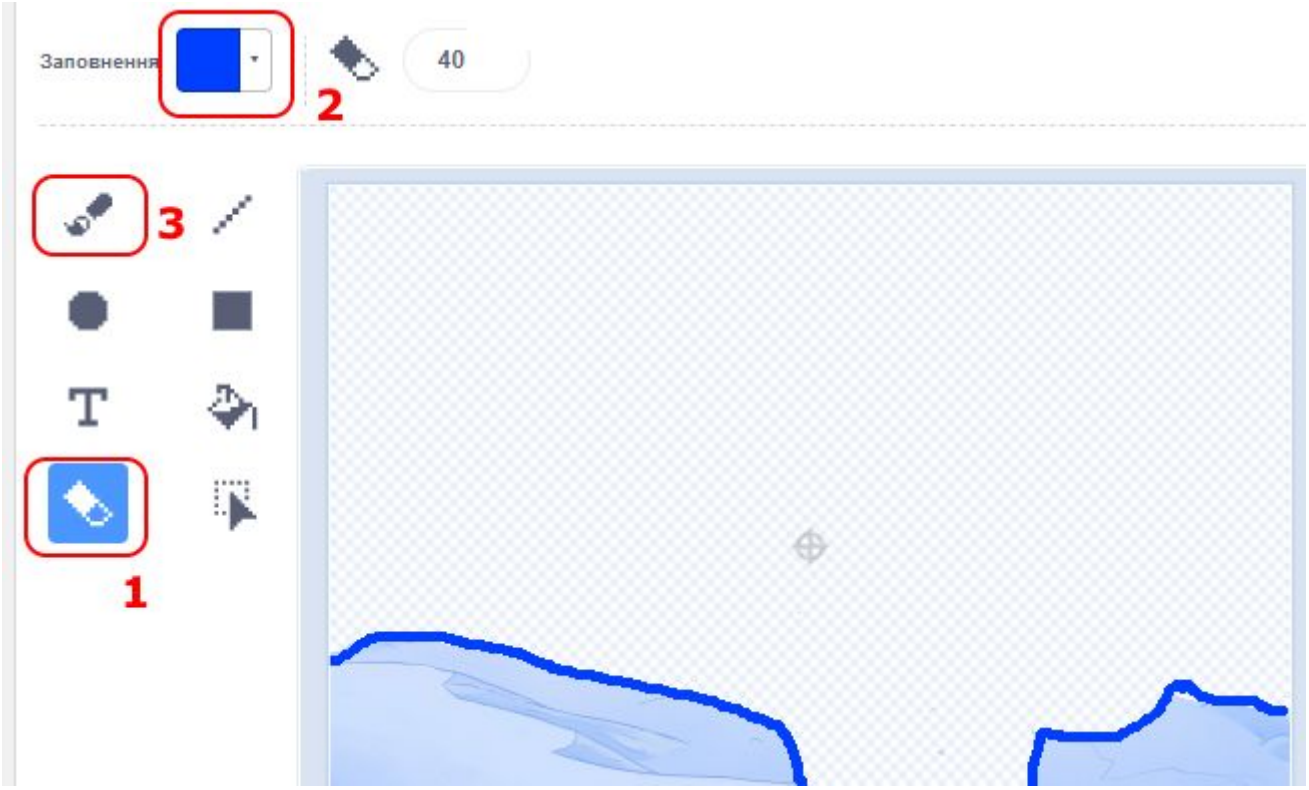
На цьому уроці ми будемо використовувати:

1. Олівець.
2. Клони.

КРОК 1: СПРАЙТИ ТА СЦЕНА

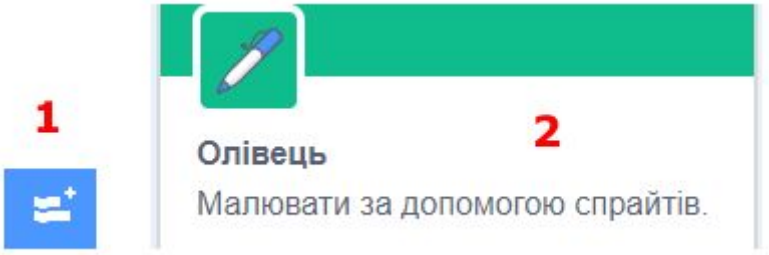
- Додайте до вашої гри такі спрайти:

- У динозавра нам потрібні тільки перші два образи, щоб зробити анімацію руху, інші треба видалити.
- У космічного корабля треба встановити активний образ, де його двигун не працює.
- Встановіть тло сцени, яке вам подобається. Потім відредагуйте тло сцени за допомогою гумки, щоб залишити тільки нижню частину малюнку. Також намалюйте синю лінію, по якій будуть мандрувати динозаври.

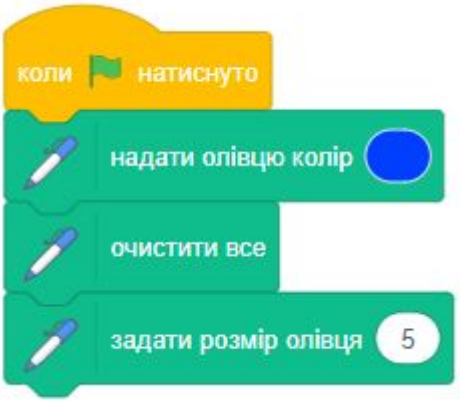


КРОК 2: МАЛЮВАННЯ ЛІНІЙ

- У новому Scratch категорія блоків управління олівцем захована в розширеннях. Для того щоб її додати, необхідно натиснути кнопку додавання розширення та вибрати розширення ОЛІВЕЦЬ.

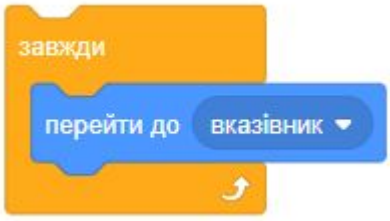


- Клацніть на **спрайт олівця** та додайте скрипт, щоб встановити синій колір того ж відтінку, що й перешкоди на сцені.

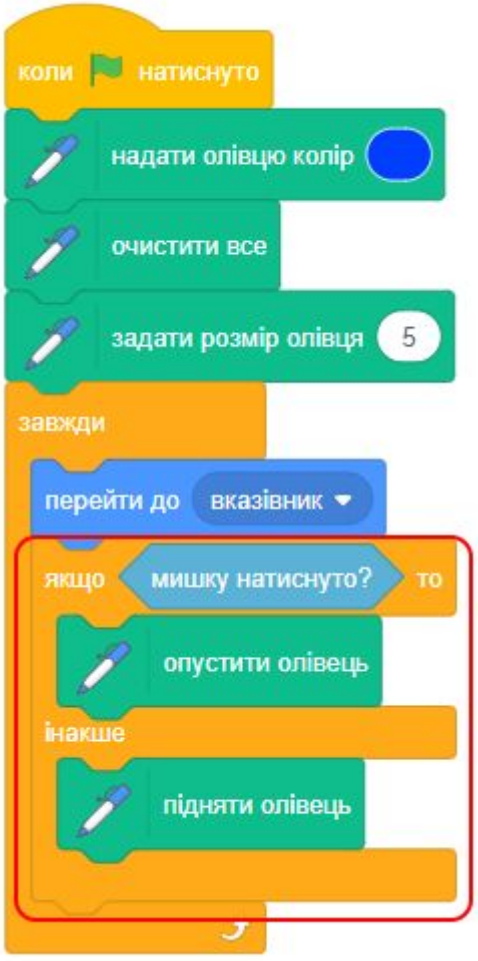


- Щоб обрати колір, натисніть на параметр кольору в блоці НАДАТИ ОЛІВЦЮ КОЛІР та виберіть піпетку, а потім натисніть нею на потрібний колір на сцені.

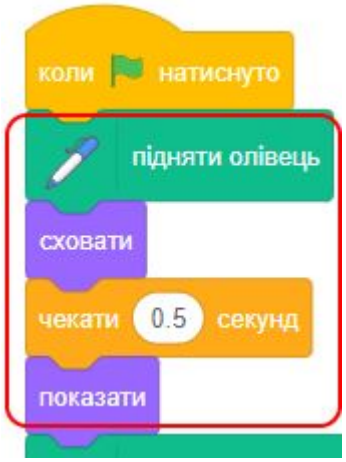
- Додайте ще деякі блоки, щоб спрайт рухався за вказівником мишки. Протестуйте гру для перевірки правильності коду.



- Додайте блоки, щоб спрайт міг намалювати лінії на сцені, якщо натиснута кнопка мишки. Ось як має виглядати ваш скрипт:



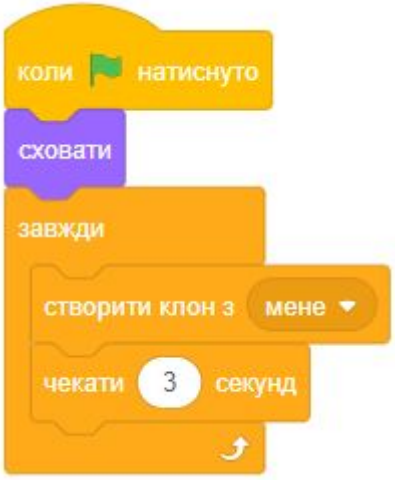
- Протестуйте свій скрипт. У вас має бути можливість клікнути й перетягнути мишку для малювання лінії на сцені.
- Напевно, ви іноді бачите, що якась синя крапка завжди з’являється у верхньому правому куті сцени. Коли ви натискаєте зелений прапорець, щоб почати гру, ви робите це кнопкою мишки, і олівець теж негайно починає малювати. Тому потрібно додати деякі блоки до вашого скрипта, якщо це трапляється.



КРОК 3: КЛОНУВАННЯ ДИНОЗАВРІВ

Вам потрібен нескінченний потік динозаврів, яких гравець повинен скеровувати на шлях до космічного корабля.

- Клацніть **спрайт динозавра** і додайте скрипт, щоб сховати основний спрайт, а також створювати його клон кожні 3 секунди.



- Якщо ви зараз запустите гру, на сцені нічого не відбуватиметься. Щоб перевірити, чи створюються нові клони кожні три секунди, зробіть так, щоб вони з’являлися в небі та падали вниз.
- Додайте скрипт до **спрайта динозавра**, щоб коли він починав як клон, він мав показатися і падати, доки не торкнеться синьої підлоги, яка зображена на сцені. Ось як має виглядати скрипт:

- Після натискання на зелений прапорець ви маєте побачити, як новий динозавр падає з верхньої частини сцени кожні три секунди. Кожен динозавр повинен приземлитися на велику купу динозаврів на підлозі внизу.

КРОК 4: ДИНОЗАВРИ, ЩО РУХАЮТЬСЯ

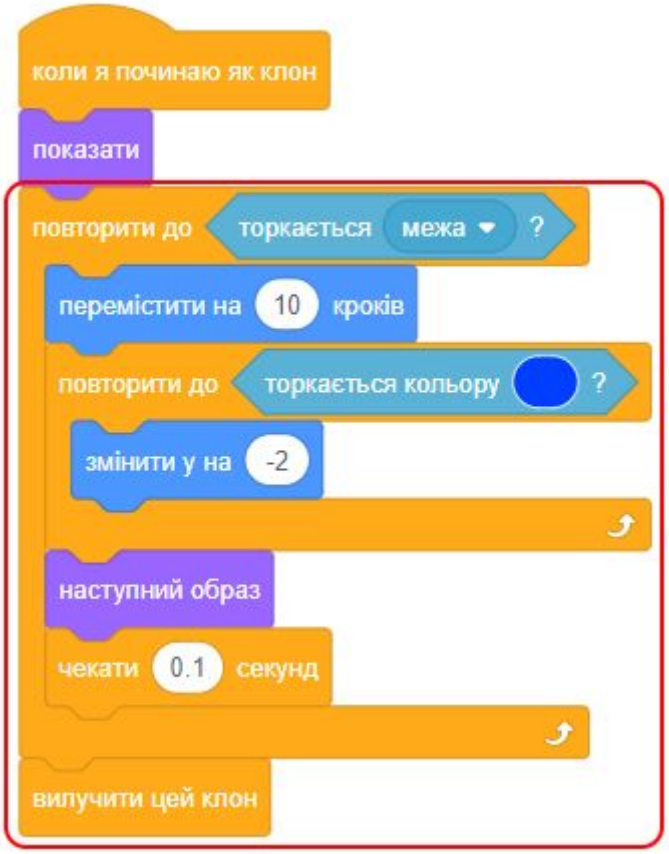
Як тільки динозавр досягає підлоги, він повинен крокувати праворуч.

- Додайте нові блоки до секції КОЛИ Я ПОЧИНАЮ ЯК КЛОН, щоб змусити спрайт динозавра переміщатися на 10 кроків і перемикатися між двома своїми образами кожні 0.1 секунди. Так буде здаватися, що динозавр йде, переставляючи ніжки.

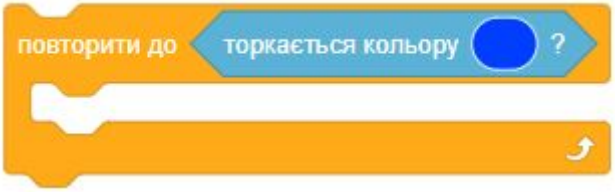
- Ось як має виглядати ваш скрипт:

- Натисніть зелений прапорець і перевірте, чи динозаври тепер переміщаються по синій платформі внизу.
- Якщо намалювати міст через прірву, щоб динозаври могли дістатися до правої сторони сцени, ви можете бачити, що вони в кінці-кінців застрягають біля правого краю сцени.

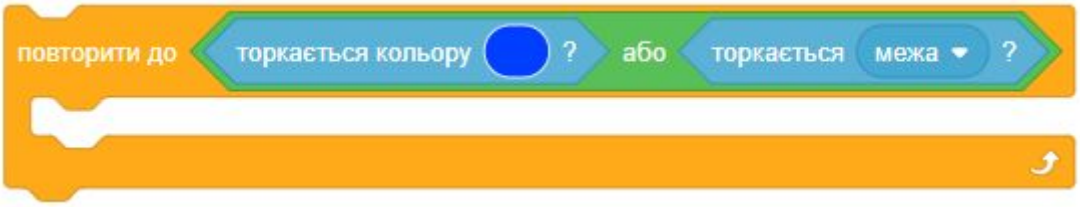
- Видаліть цикл ЗАВЖДИ і замість нього додайте інший цикл, щоб змусити динозаврів ходити лише до того, як вони опиняться біля краю. Коли динозавр досягає краю сцени, він повинен зникнути.



- Натисніть зелений прапорець і перевірте, чи зникають динозаври, коли досягають краю сцени.
- Ви можете помітити, що коли динозаври потрапляють у прірву, вони не зникають, а застрягають внизу. Це тому, що вони намагаються продовжувати падати вниз.
- Ця частина скрипта каже динозавру продовжувати падати, доки він не торкнеться синього кольору:



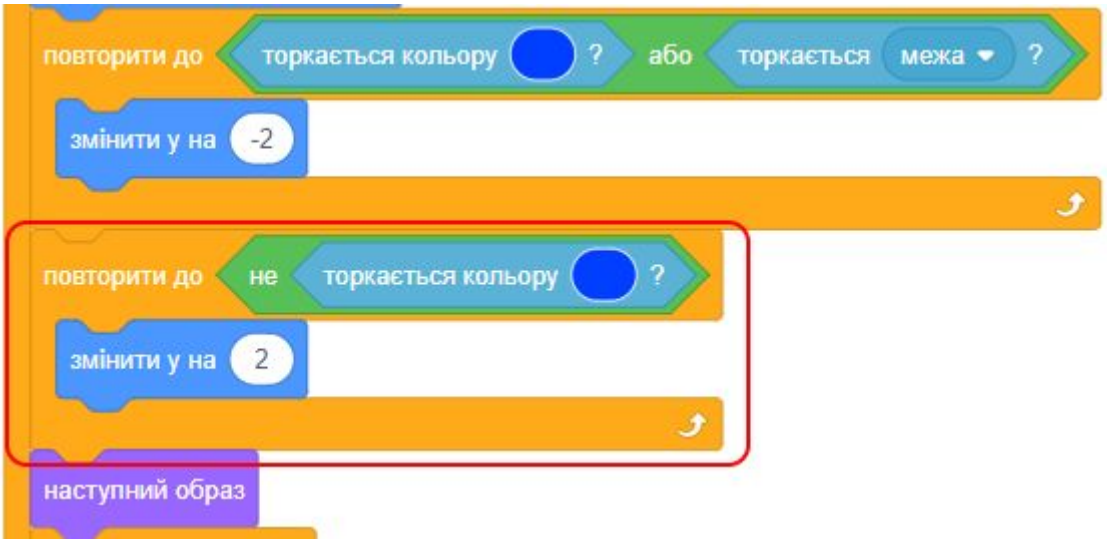
- Однак у прірві динозавр не може досягти блакитного кольору, тому застрягає там назавжди.
- Додайте більше блоків до цього циклу, щоб він повторювався, поки спрайт динозавра не торкнеться синього або не торкнеться межі. Таким чином, спрайт зупиняє падіння, коли динозавр досягає краю сцени.



КРОК 5: ПЕРЕМІЩЕННЯ ПО ПОВЕРХНІ

- Ви можете помітити, що якщо намалювати низький міст між двома платформами або висхідну лінію, то динозаври будуть йти крізь платформу, а не по ній!

- У скрипт для **спрайта динозавра** додайте ще один цикл перед блоком НАСТУПНИЙ ОБРАЗ. Цього разу цикл повинен сказати динозавру рухатися вгору на 2 кроки, доки він не торкнеться синього кольору. Ось де потрібно додати нові блоки:

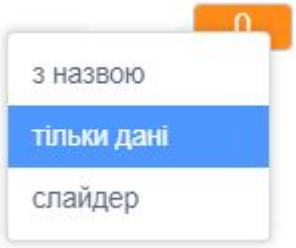


- Натисніть на зелений прапорець і спробуйте намалювати висхідну лінію. Перевірте, чи ваш динозавр рухається по цій лінії.

КРОК 6: ШЛЯХ ДО БЕЗПЕЧНОГО МІСЦЯ

Мета гри - скерувати динозаврів у безпечне місце, створюючи шлях, щоб вони могли дістатися до космічного корабля.

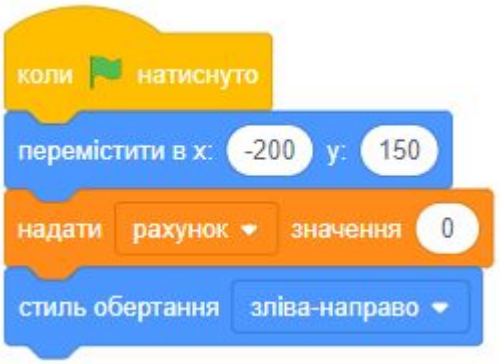
- Створіть змінну для балів, щоб відстежувати, скільки динозаврів доходить до космічного корабля.
- Додайте нову змінну з назвою **score** (рахунок). Треба змінити її відображення на “тільки дані”.



- Розмістить змінну у верхньому правому кутку сцени поряд з якимось спрайтом, який проінформує гравця про то, що це рахунок гри.

- Додайте скрипт до **спрайта динозавра**, щоб збільшувати рахунок на 1 кожного разу, коли динозавр досягне космічного корабля. Також додайте блоки для того, щоб спрайт відтворював якийсь звук і зникав після того, як дістанеться корабля.
- Ось як має виглядати ваш скрипт:

- Також встановіть **score** рівним 0 на початку гри. Інші блоки цього скрипта встановлюють початкову точку появи динозаврів та стиль їх обертання.

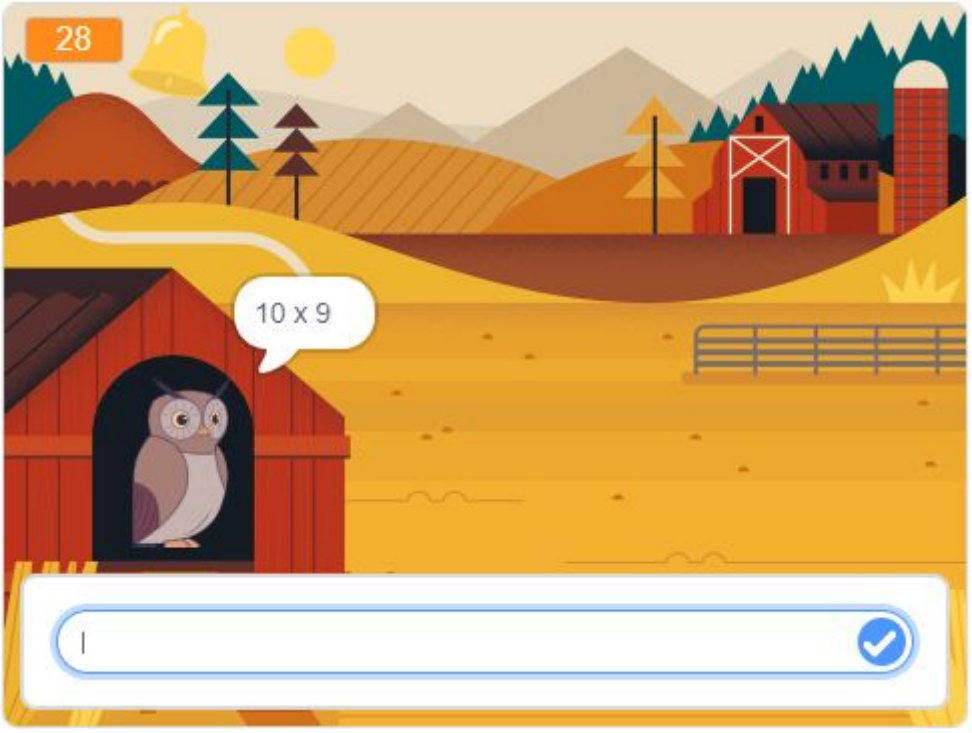


ЗАВДАННЯ: БІЛЬШЕ ПЕРЕШКОД

- Додайте більше фонів з різними дизайнами платформ.
- Змінюйте розташування дверей в залежності від фону.
- Додайте нові спрайти як рухомі перешкоди. Можливо, це могли би бути собаки або обертові шипи, яких потрібно уникати?
- Дозвольте гравцеві використовувати обмежену кількість «чорнил» для малювання ліній у грі. Ви можете відслідковувати, скільки «чорнил» витрачається, створивши змінну таймера, яка буде зберігати тривалість натискання мишки.
- Створіть спрайт, щоб показувати гравцю залишок чорнил.
- Якщо по намальованому мосту пройде 5 динозаврів, міст мусить зникати. Скористайтеся блоком ОЛІВЕЦЬ: ОЧИСТИТИ ВСЕ.
- Якщо ви зібрали 50 динозаврів, космічний корабель злітає, а замість нього прилітає інший корабель (з іншим образом). Додайте анімацію цієї події.
- Рахуйте скільки динозаврів падає в прірву та показуйте цю змінну на сцені. Програвайте звук, коли динозавр падає у прірву.
- Інколи замість динозаврів будуть з'являтися інші істоти (наприклад, динозаври більшого розміру), які не повинні потрапити на корабель. Якщо вони опинилися там, корабель зламається, а гра зупиниться.
- Зробіть ще інструмент ГУМКА додатково до олівця. Ця гумка може стерти намальований вами міст. Клавіша ПРОПУСК переключає між олівцем та гумкою.

УРОК 22. РОЗВИВАЮЧА ГРА ДЛЯ МОЗКУ

У цьому проекті ви дізнаєтеся, як створити вікторину з відліком часу, у якій потрібно дати якнайбільше правильних відповідей за 30 секунд.

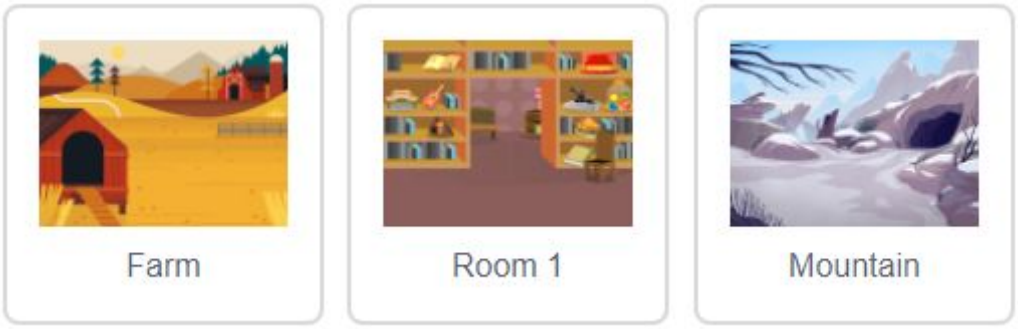


На цьому уроці ми будемо використовувати:

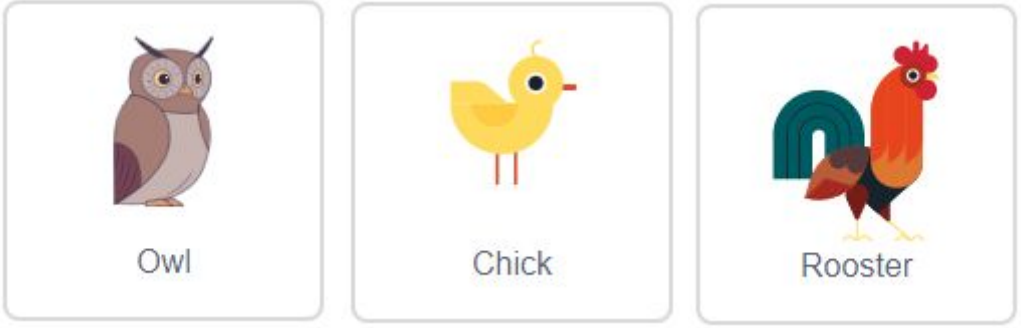
1. Змінні.
2. Повідомлення.

КРОК 1: СПРАЙТИ ТА СЦЕНА

- Почніть новий проект і видаліть з нього спрайт кота.
- Оберить тло для вашої гри. Наприклад:



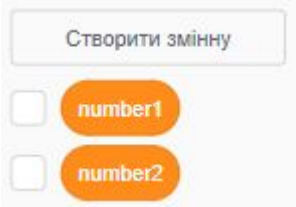
- Додайте головний спрайт вашої гри. Наприклад:



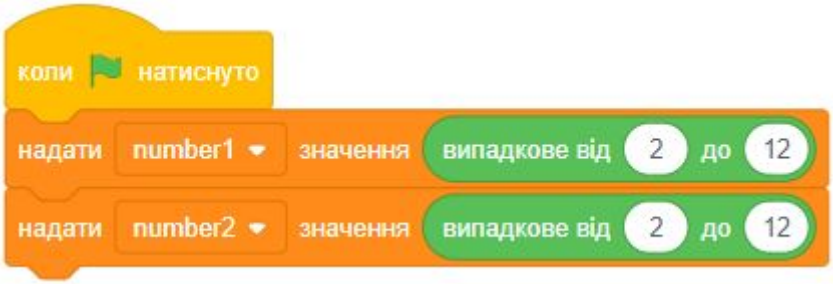
КРОК 2: СТВОРЕННЯ ЗАПИТАНЬ

Розпочнемо зі створення випадкових запитань до гравців.

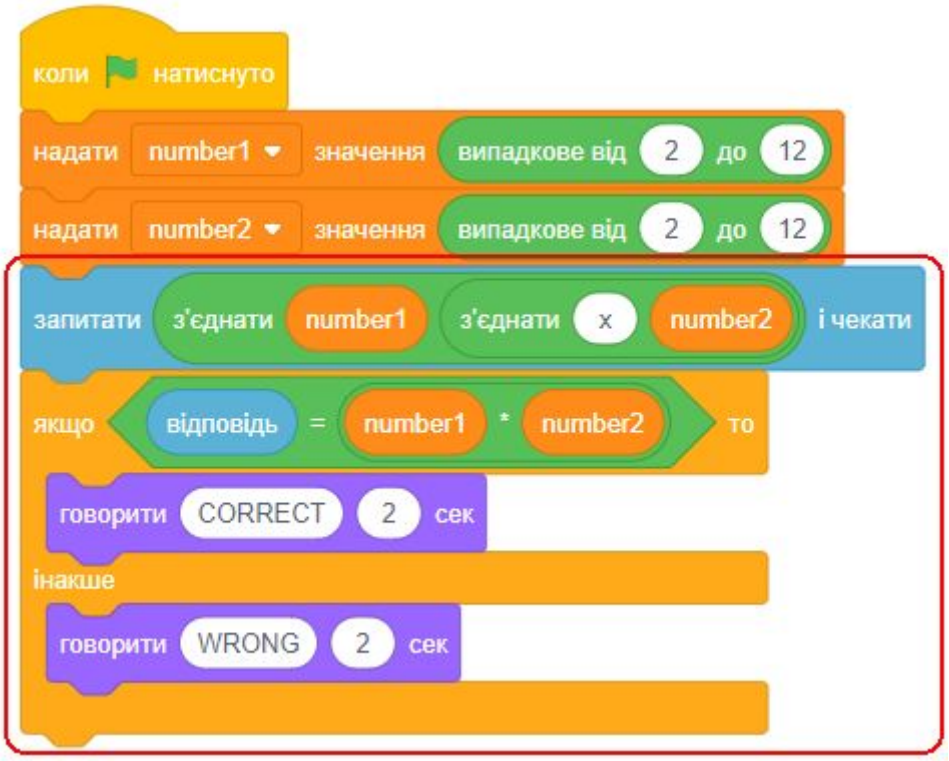
- Створіть дві нові змінні **number1** і **number2**. Ці змінні збережуть два числа, що будуть множитися.



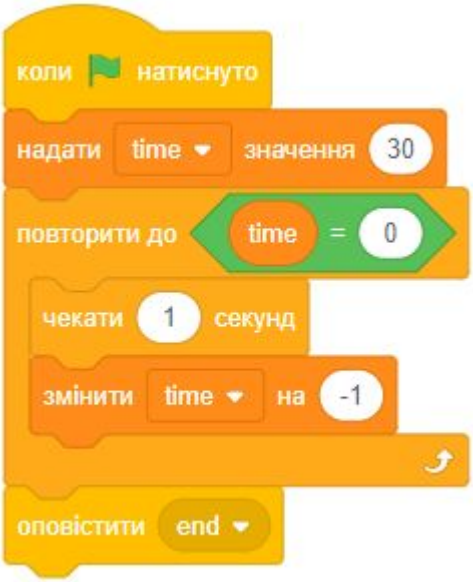
- Додайте код до вашого спрайта, надавши змінним випадкове значення від 2 до 12.



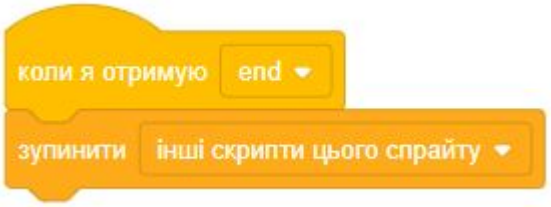
- Ви можете попросити гравця дати відповідь на запитання і повідомити його, чи була вона правильною (correct) або неправильною (wrong).



- Повністю перевірте свій проект: на одне питання дайте правильну відповідь, а на інше - неправильну.
- Додайте команду ЗАВЖДИ біля цього коду, щоб гравець міг відповісти на багато запитань.
- Створіть таймер зворотнього відліку на **сцені** за допомогою нової змінної на ім'я **time** (час). Відлік таймера повинен починатися з 30 секунд і відраховувати до 0 секунд. Скрипт виглядає так:



- Ми додали блок ОПОВІСТИТИ в кінець коду таймера, щоб скрипт відправляв повідомлення «**end**» (кінець), коли час дійде до 0. Повідомлення схоже на оголошення через гучномовець: його можуть почути всі ваші спрайти.
- Виберить спрайт свого персонажа і додайте код, щоб спрайт зупиняв інші скрипти, коли він отримує повідомлення «end».



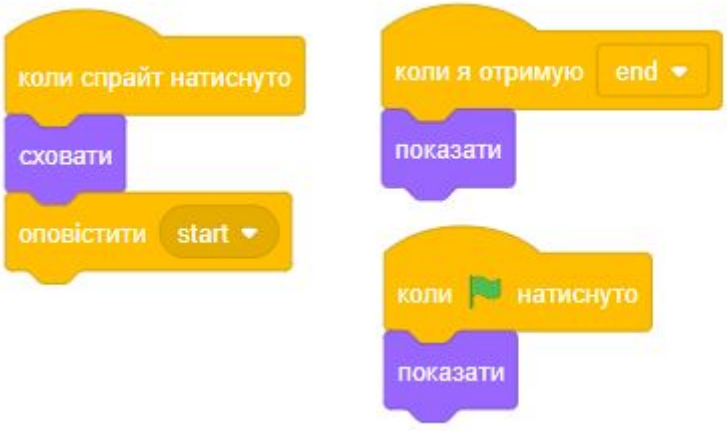
- Перевірте свій проект знову - потрібно, щоб можна було відповідати на запитання, поки не закінчиться час.

КРОК 3: БАГАТОРАЗОВІ ІГРИ

Додамо кнопку «Грати», щоб можна було відкривати гру багато разів.

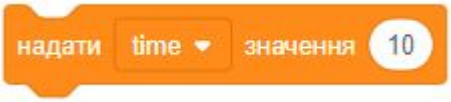
- Створіть спрайт нової кнопки «Грати», за допомогою якої гравець розпочинатиме нову гру. Ви можете намалювати її або відредагувати спрайт із бібліотеки.

- Додайте цей скрипт до нової кнопки.



- Новий код включає в себе ще один блок ОПОВІСТИТИ, який відправляє повідомлення **start** (старт).
- Новий код робить так, щоб спрайт кнопки «Грати» з'являвся, коли гравець натискає на зелений прапор. Коли гравець натискає на спрайт кнопки, він ховається, а потім передає повідомлення, на яке можуть реагувати інші спрайти.
- В даний момент спрайт персонажа починає ставити питання, коли гравець натискає на зелений прапор. Змінить код гри, щоб спрайт персонажа почав задавати питання, коли він отримає повідомлення **start**.
- Виберіть спрайт персонажа і у вкладці код замініть блок КОЛИ НАТИСНУТО ЗЕЛЕНИЙ ПРАПОР на блок КОЛИ Я ОТРИМАЮ «start».

- Натисніть на зелений прапор, а потім на нову кнопку «Грати», щоб перевірити, чи працює вона. Гра не повинна запускатися, поки ви не натиснете на цю кнопку.
- Чи бачите ви, що таймер запускається після натискання на зелений прапор, а не коли починається гра?
- Чи можете ви змінити код таймера так, щоб він запускався, коли гравець натиснув на кнопку?
- Перевірте кнопку «Грати», зігравши пару ігор. Кнопка повинна відображатися в кінці кожної гри. Щоб перевірити гру швидше, можете змінити значення змінної time так, щоб кожна гра тривала всього кілька секунд.



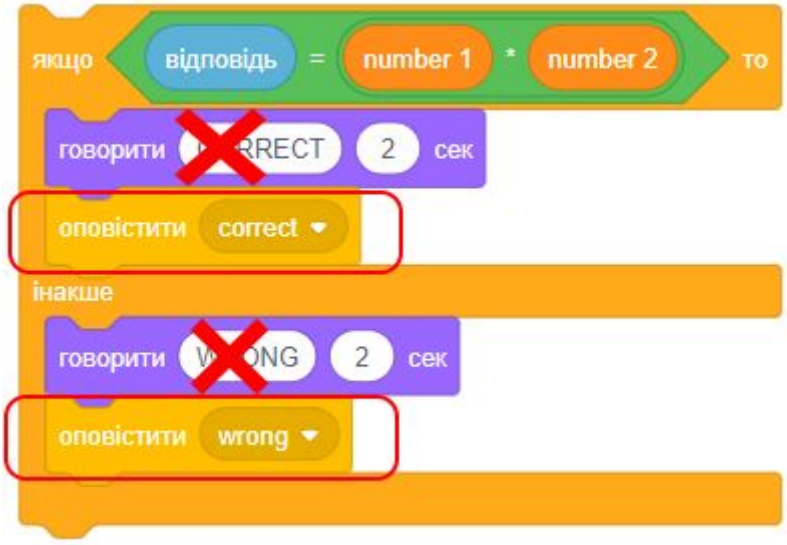
- Кнопка може трохи змінюватися, коли на неї наводять вказівник мишки. Для цього ми використаємо блок ВСТАНОВИТИ ЕФЕКТ.

КРОК 4: ДОДАЙТЕ ГРАФІКУ

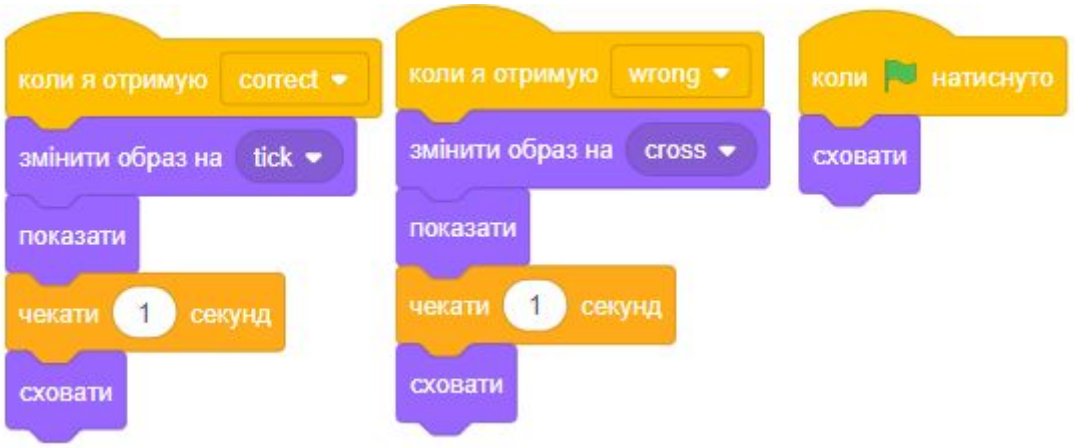
Спрайт персонажа занадто просто реагує на відповіді. Додайте графіку, щоб гравець розумів, правильна відповідь чи ні.

- Створить новий спрайт з ім'ям **result** (результат) і дайте йому образи **tick** (галочка) і **cross** (хрестик). Ці образи є у бібліотеці.


- Змінить код спрайта свого персонажа так, щоб він не говорив що-небудь гравцеві, а передавав повідомлення **correct** (правильно) або **wrong** (помилка).

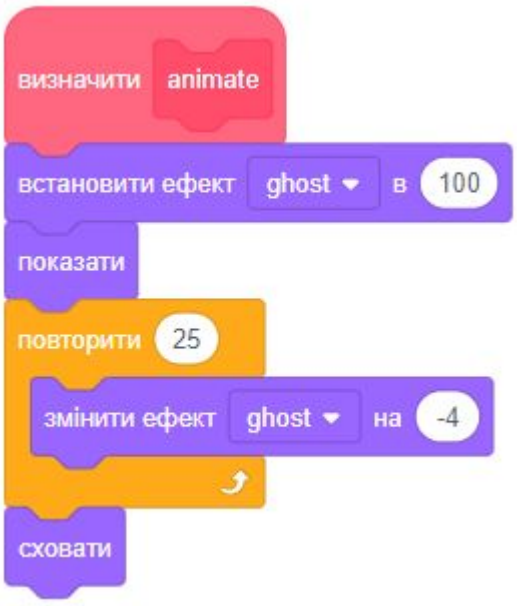


- Тепер ви можете використовувати ці повідомлення, щоб показати образи «галочка» або «хрестик». Додайте код до спрайта **result**:



- Перевірте гру знову. Ви повинні бачити галочку, коли відповідаєте на питання правильно, і хрестик, коли відповідаєте неправильно!
- Бачите, що код для правильної відповіді і для помилки майже однаковий? Таким чином, щоб код було легше змінити, створимо призначений для користувача блок. Використаємо цей блок:

- Завдяки призначеному для користувача блоку **animate** (анімувати) тепер потрібно лише внести одну зміну в код, якщо хочете показувати образи спрайта довше або коротше.
- **Завдання.** Змінить код так, щоб образи «галочка» або «хрестик» відображалися протягом 2 секунд.
- Замість того щоб показувати і приховувати образи «галочка» або «хрестик», ви можете змінити блок **animate**, щоб образи з’являлись та зникали поступово.



- **Завдання.** Чи можна поліпшити анімацію графіки «галочка» або «хрестик»? Ви можете додати код до цього скрипта, щоб образи зникали не миттєво, а поступово?

КРОК 5: ЗАВДАННЯ

ЗАВДАННЯ 1: ЗМІНА ОБРАЗІВ

Чи можете ви змінити образи об’єктів так, щоб вони залежали від правильності відповідей гравця?

ЗАВДАННЯ 2: СТВОРЕННЯ РАХУНКУ

Чи можете ви створити рахунок для гри? Ви можете додавати бал за кожну правильну відповідь. Якщо ж любите побешкетувати, то можна скинути бали гравця аж до 0, якщо він відповість неправильно!

ЗАВДАННЯ 3: ЗВУК І МУЗИКА

Чи можете ви додати звуки і музику в гру? Наприклад, можна:

- Відтворити якийсь звук, коли гравець дає правильну відповідь. Відтворити інший звук, коли відповідь гравця є неправильною.
- Додати цокання, коли таймер відраховує час.
- Додати звук, коли час гравця закінчується.
- У грі може також постійно звучати фонова музика.

ЗАВДАННЯ 4: СТВОРИТИ СТАРТОВИЙ ЕКРАН

Чи можете ви додати ще одне тло, яке стане стартовим екраном гри?

- Ви можете використовувати ці блоки, щоб перемикатися між стартовим і основним тлом гри:

- Щоб показати або приховати персонажа, коли в грі змінюється тло, ви можете використовувати блоки ПОКАЗАТИ і СХОВАТИ.
- Щоб показати або приховати змінні (час і рахунок), коли гра перемикає тло, ви можете використовувати блоки ПОКАЗАТИ ЗМІННУ і СХОВАТИ ЗМІННУ.

ЗАВДАННЯ 5: ГОНКА ДО 10 ОЧОК

Чи можете ви змінити гру так, щоб гравець замість відповідей на якомога більше запитань за 30 секунд, якнайшвидше відповідав на 10 запитань.

Щоб зробити це, необхідно лише змінити код таймера. Ви вже здогадались, які блоки повинні відрізнятися?

ЗАВДАННЯ 6: ВІКНО З ІНСТРУКЦІЯМИ

Чи можете ви додати вікно з інструкціями, де гравцеві пояснювалися би правила гри? Для цього вам знадобиться кнопка «Інструкції» і ще одне тло сцени. Також треба буде додати кнопку «Назад», яка дозволить гравцеві повернутися до початкового екрана.

ЗАВДАННЯ 7: ІНШІ АРИФМЕТИЧНІ ДІЇ

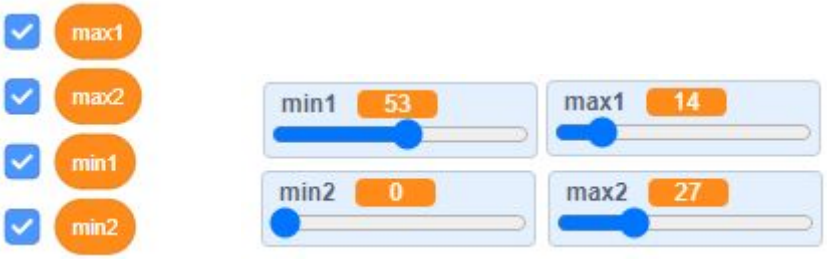
Чи можете ви запитувати гравця, щоб він не тільки множив числа, а й виконував інші арифметичні операції.

Скористайтеся блоком ВИПАДКОВЕ, щоб обрати арифметичну дію:

КОД ОПЕРАЦІЇ	БЛОК
1	
2	
3	
4	

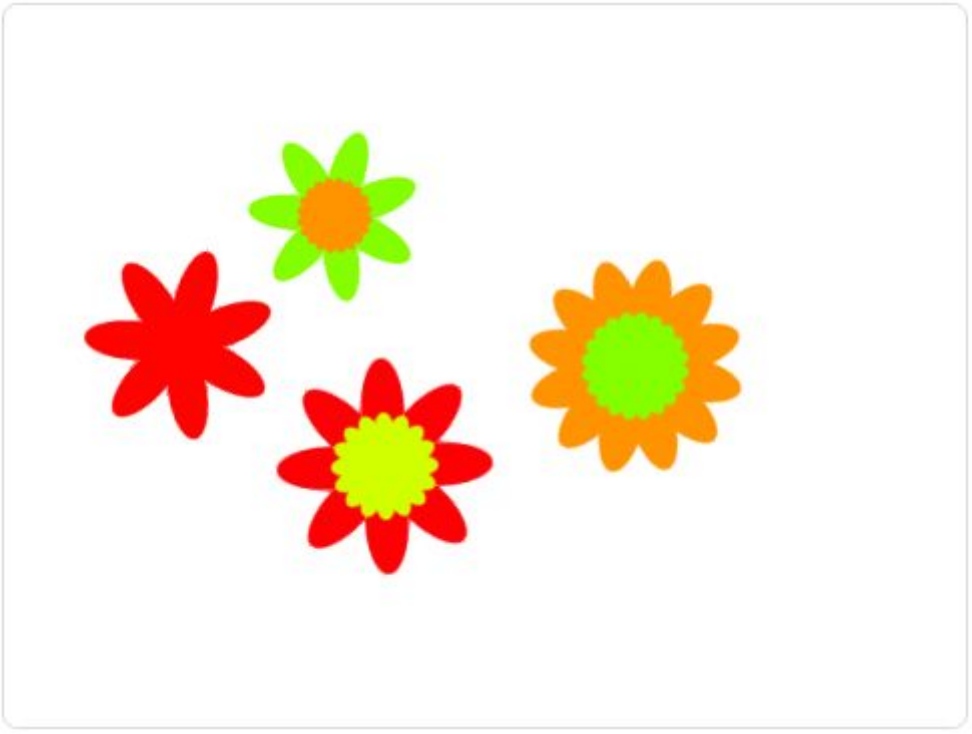
ЗАВДАННЯ 8: МІНІМУМ ТА МАКСИМУМ

Чи можете ви додати чотири нові змінні, відобразити їх на сцені як слайдери? Їх треба використати як мінімальне і максимальне значення у блоках ВИПАДКОВЕ..ВІД..ДО там, де надаються значення змінним **number1** і **number2**.



УРОК 23. ГЕНЕРАТОР КВІТІВ

У цьому проекті ви дізнаєтеся, як створити генератор квітів. Ви створите сотні квітів різних розмірів, форм і кольорів. Ці зображення квітів ви зможете експортувати і використовувати як шпалери на телефоні чи комп’ютері або як тло для інших проектів.

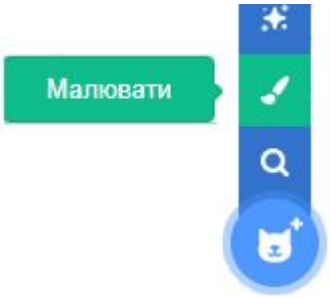


На цьому уроці ми будемо використовувати:

1. Цикли.
2. Створення власних блоків.

КРОК 1: СПРАЙТИ ТА СЦЕНА

- Почніть новий проект і видаліть з нього спрайт кота.
- Додайте розширення ОЛІВЕЦЬ до вашого проекту.
- Додайте головний спрайт вашого проекту. Це буде спрайт у формі квіткової пелюстки. Для його створення використайте вбудований інструмент для малювання.



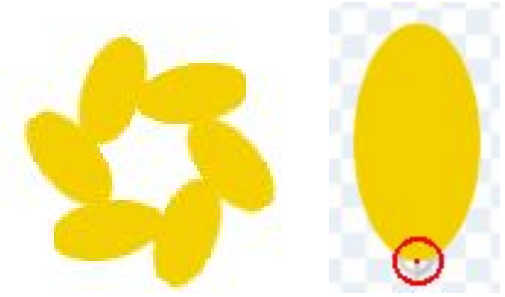
- Використовуйте інструмент КОЛО, щоб намалювати контур пелюстки, заповнений помаранчевим кольором. Далі за допомогою програмування ми додамо більше кольорів.

КРОК 2: МАЛЮЄМО КВІТКУ

- Додайте такий скрипт до спрайта квітки, щоб зробити її штамп із шістьма рівномірно розподіленими пелюстками, коли зелений прапорець натиснутий.

- Може виявитися, що пелюстки квітки розміщені дивним чином. Це тому, що спрайт обертається навколо свого центру.

Перемістить пелюстку квітки, щоб її низ був у центрі спрайта. Це легше зробити, якщо змінити масштаб.



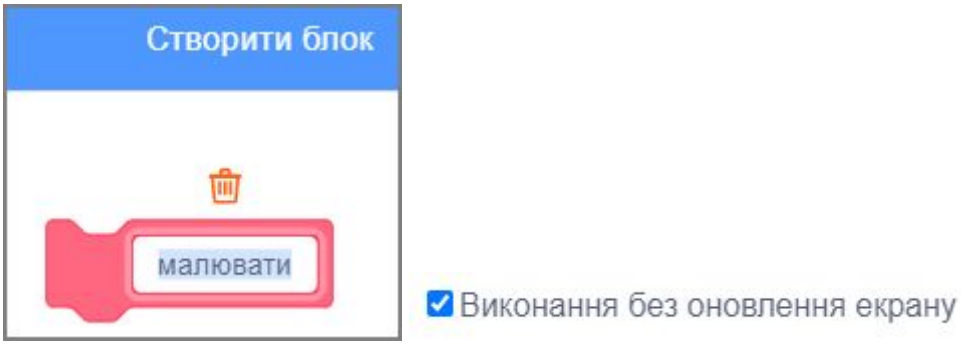
- Перед повторним запуском скрипта вам треба видалити всі штампи спрайтів зі сцени. Тому додайте блок ОЧИСТИТИ ВСЕ з категорії ОЛІВЕЦЬ на початку вашого скрипта.

- Запустіть скрипт знову, щоб переконатися, що квіткові пелюстки тепер прямі. Якщо ні, то змінюйте розміщення пелюстки, поки її низ не опиниться в центрі спрайта.

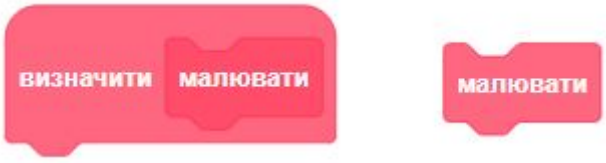
КРОК 3: ВЛАСНИЙ БЛОК ДЛЯ МАЛЮВАННЯ КВІТІВ

Що робити, якщо ви хочете намалювати багато квітів? Замість того, щоб робити багато копій якогось коду, ви створите свій власний блок і будете його використовувати кожного разу, коли захочете намалювати квітку.

- Клацніть на МОЇ БЛОКИ, а далі на кнопку СТВОРИТИ БЛОК, щоб створити власний блок із назвою «**малювати**». Під час створення клацніть на пункт «Виконання без оновлення екрана», щоб прискорити малювання квітів.



- Тепер в розділі МОЇ БЛОКИ з’явився блок «**малювати**», а в області скриптів - місце для його визначення.



- Перенесіть свій код малювання квітки з-під блока КОЛИ ЗЕЛЕНИЙ ПРАПОР НАТИСНУТО під визначення нового блока МАЛЮВАТИ. Ваш скрипт повинен виглядати так:

- Клацніть на зелений прапор, щоб протестувати скрипт, і перевірте, чи бачите ви квітку.
- Тепер змінить скрипт, щоб перемістити спрайт, а потім намалюйте ще одну квітку:

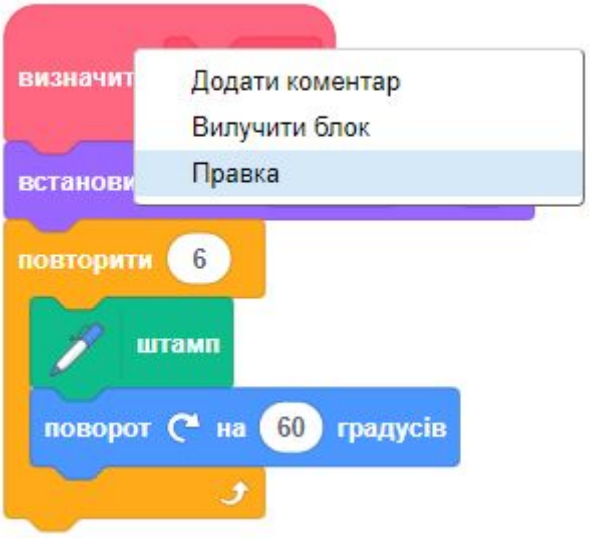
КРОК 4: НАЛАШТУЙТЕ СВОЇ КВІТИ

Наразі всі намальовані вами квіти абсолютно однакові. Далі вам треба додати деякі параметри до блока малювання квітки, щоб можна було створювати квіти різних кольорів, розмірів і з різною кількістю пелюсток.

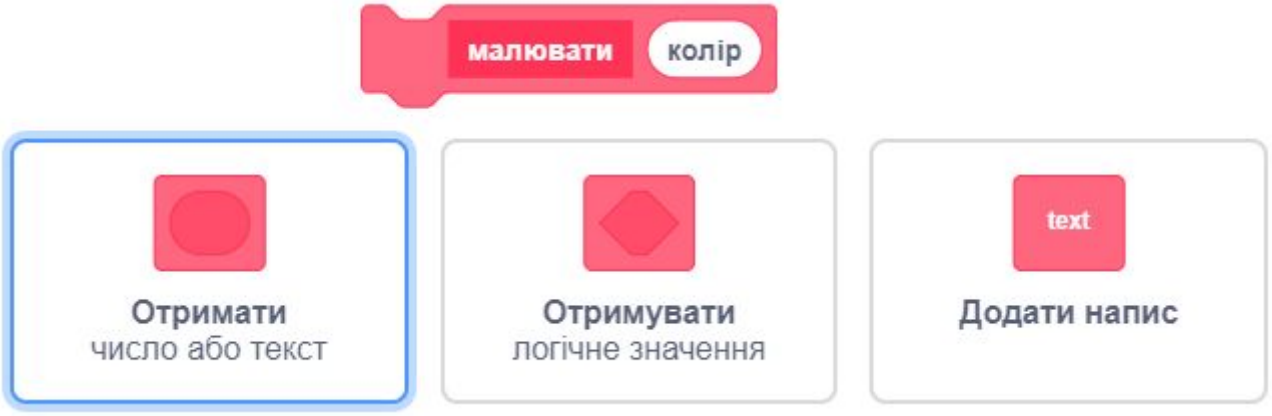
- Ми будемо використовувати блок ВСТАНОВИТИ ЕФЕКТ КОЛІР з категорії ВИГЛЯД, щоб змінювати колір спрайтів. Змініть визначення блоку «**малювати**», щоб змінювати колір квітки. Запустіть скрипт, щоб побачити квіти іншого кольору.

- Блок ВСТАНОВИТИ ЕФЕКТ КОЛІР змінює колір на основі початкового кольору спрайта, тому, якщо ваш спрайт спочатку не був помаранчевим, ви побачите інший результат.

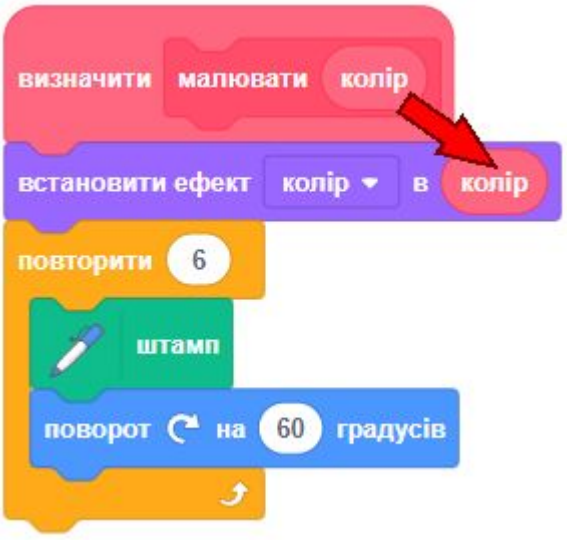
- Експериментуйте з різними числами від 0 до 199 в блоці ВСТАНОВИТИ ЕФЕКТ КОЛІР, щоб побачити, які результати ви можете отримати.
- Зараз усі квіти мають однаковий колір. Щоб малювати квіти різних кольорів, нам треба додати **параметр** до блока малювання квітки. Клацніть правою кнопкою мишки на визначення блока малювання квітки і виберіть ПРАВКА.



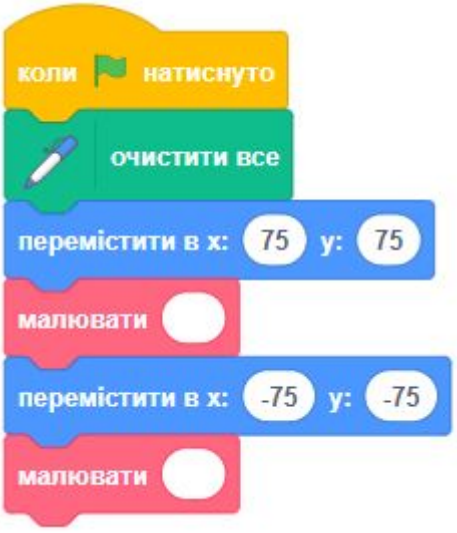
- Тепер додайте числовий параметр з назвою «**колір**».



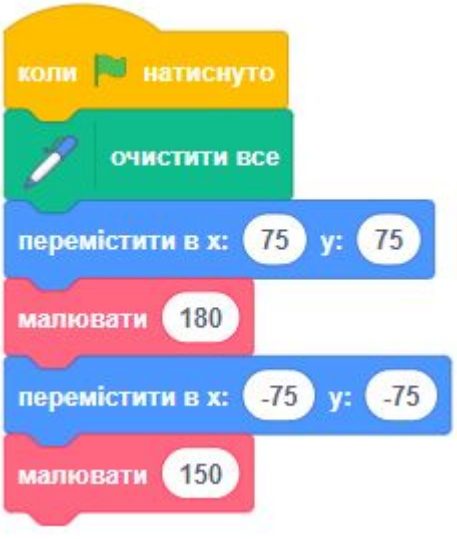
- Цей параметр з’явиться у визначенні блока малювання квітки, і ви зможете перетягувати його в будь-яке місце для використання. Перетягніть параметр «**колір**» у блок ВСТАНОВИТИ ЕФЕКТ КОЛІР. Ваш скрипт повинен виглядати так:



- Зверніть увагу, що тепер ваші блоки малювання квітки мають новий параметр без числового значення:



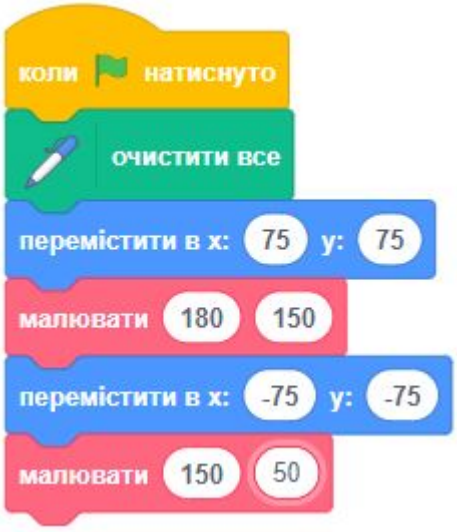
- Змінить числа в блоках малювання квітки, щоб дві квітки були різного кольору. Ви можете вибрати будь-які числа між 0 та 200:



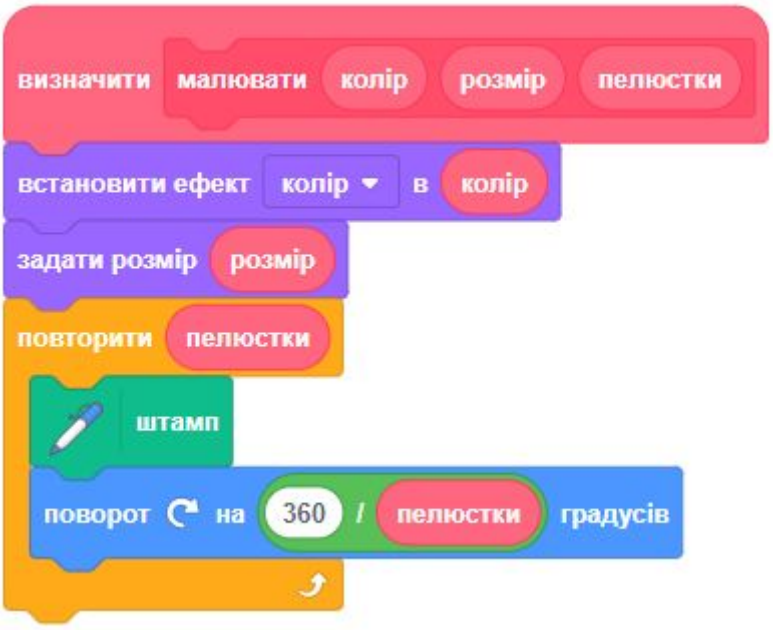
- Тепер додамо ще один параметр, щоб задавати розмір квітки, а отже блок малювання квітки буде виглядати так:

- За допомогою вищенаведеного блоку ви можете створювати квіти різних розмірів. Клацніть правою кнопкою мишки на визначення блока малювання квітки, далі виберіть ПРАВКА і додайте числовий параметр із назвою «**розмір**».
- Змініть скрипт визначення блока малювання квітки, щоб він виглядав таким чином:

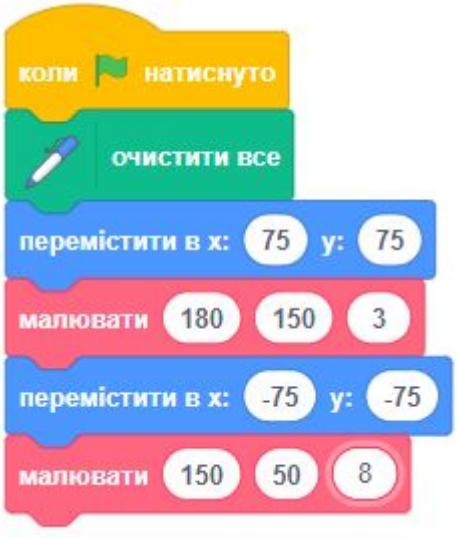
- У скрипті КОЛИ ЗЕЛЕНИЙ ПРАПОР НАТИСТНУТО змініть друге число в обох блоках малювання квітки, щоб обидві квітки стали різного розміру.



- Було би добре змінювати кількість пелюсток квітів. Додайте ще один параметр, щоб ви могли малювати такі квіти.
- Змініть блок визначення малювання квітки, додавши новий параметр із назвою «**пелюстки**». Ваш скрипт стане таким:



- У скрипті КОЛИ ЗЕЛЕНИЙ ПРАПОР НАТИСНУТО змініть третє число в обох блоках малювання квітки, щоб квіти мали різну кількість пелюсток.



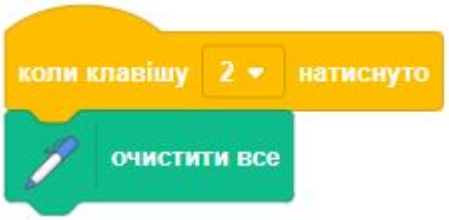
- Далі змініть свій скрипт, щоб ви могли малювати різні квіти, натискаючи клавішу 1. Перенесіть свій код для малювання квітів з-під блока КОЛИ ЗЕЛЕНИЙ ПРАПОР НАТИСНУТО під блок КОЛИ КЛАВІШУ 1 НАТИСНУТО.

- **Завдання.** Додайте ще кілька блоків малювання квітки до своєї програми, щоб намалювати по всій сцені квіти різного кольору, розміру та з різною кількістю пелюсток.

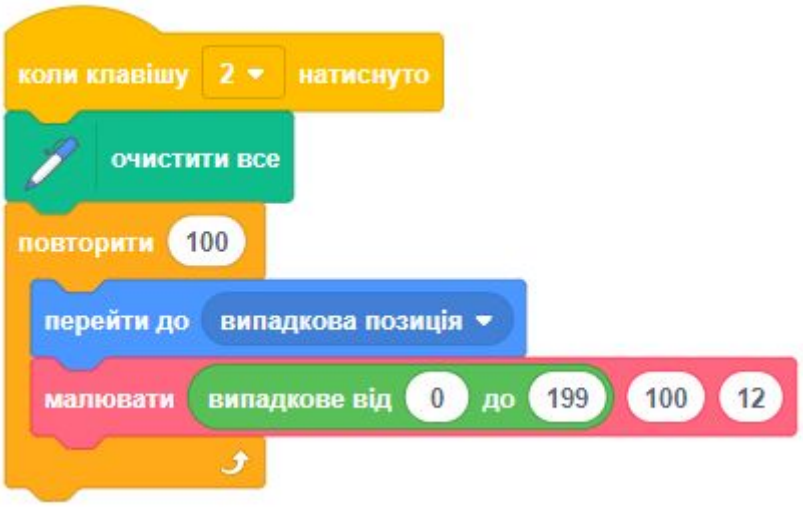
КРОК 5: ГЕНЕРАТОР ВИПАДКОВИХ КВІТІВ

Тепер ми використаємо блок малювання квітки, щоб після натиснення клавіші 2 створити сотні випадкових квітів по всій сцені.

- Додайте новий блок з розділу ПОДІЇ до свого спрайта, щоб КОЛИ КЛАВІШУ 2 НАТИСНУТО, екран був очищений.



- Додайте до цього скрипта блок ПОВТОРИТИ, щоб перейти до випадкової позиції 100 разів. Використайте блок малювання квітки, щоб створити квітку, яка має випадковий колір від 0 до 199. Ваш скрипт повинен виглядати так:

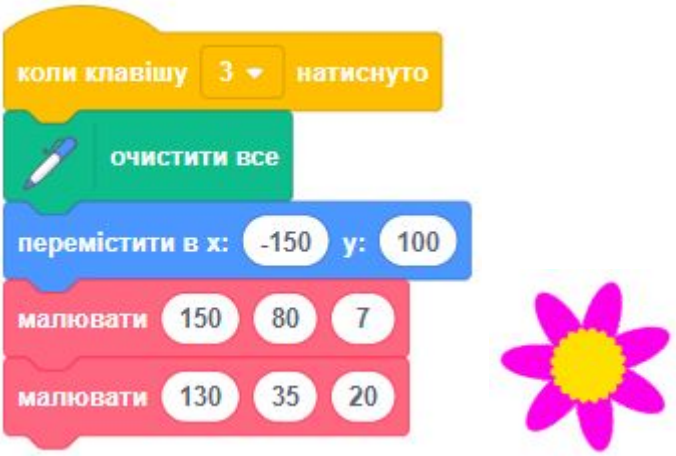


- Цей скрипт створює сотню квітів різного кольору, але з однаковим розміром і кількістю пелюсток.
- **Завдання.** Чи можете ви змінити скрипт, щоб розмір квітів та кількість пелюсток теж були випадковими? Додайте блок отримання випадкових чисел до другого (розмір від 50 до 150) і третього параметрів (кількість пелюсток від 4 до 12) блока малювання квітів.

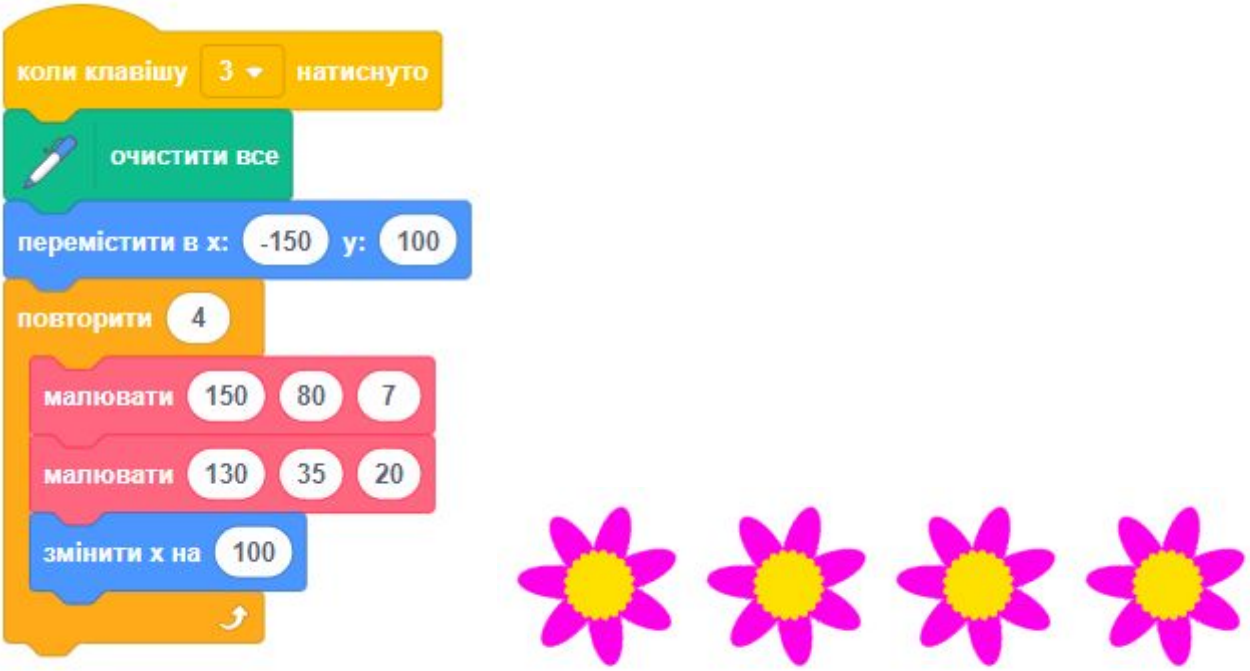
КРОК 6: КВІТКОВІ ВІЗЕРУНКИ

Ви також можете використовувати блок малювання квітки, щоб створювати гарні квіткові візерунки.

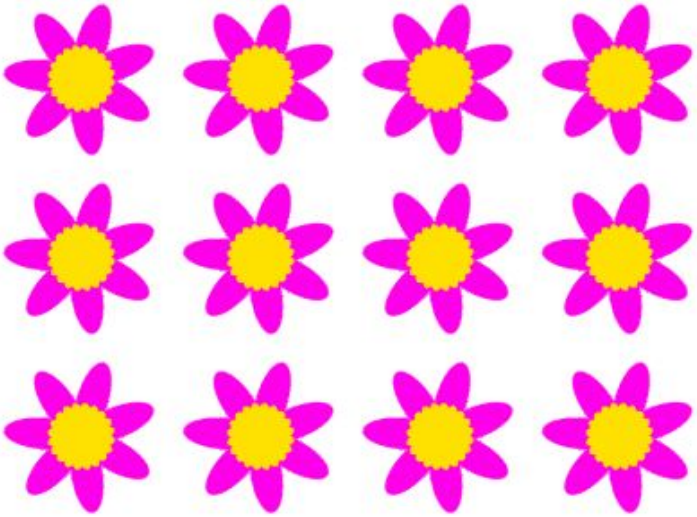
- Перед тим як ви будете створювати власний візерунок, треба очистити сцену від усіх квітів, що на ній залишилися.
- Намалюйте квітку або комбінацію квітів, яка вам подобається. Ось приклад:



- Тепер намалюйте рядок квітів зверху сцени. Ось зразок скрипта, в якому вам потрібно змінити значення параметрів малювання, щоб він більше підходив для вашої квітки:

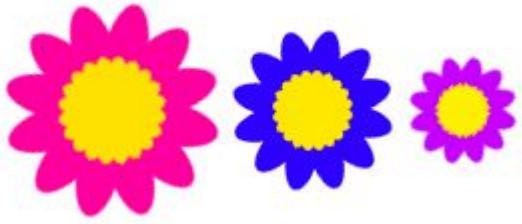


- Додайте ще один цикл, щоб створити більше рядків із квітів. Цей приклад додає цикл повторити 3, щоб намалювати три рядки.

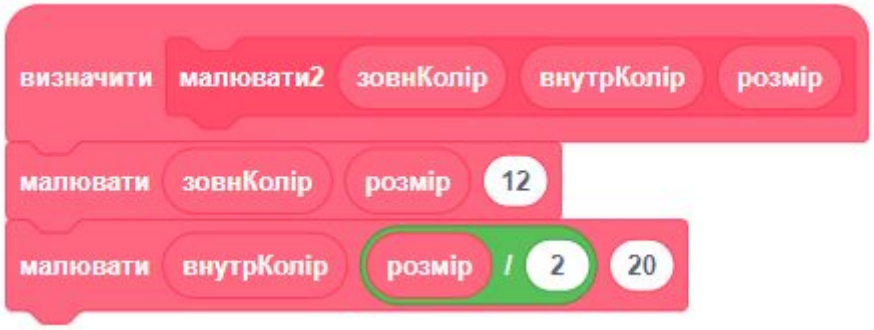


КРОК 7: ПОДВІЙНА КВІТКА

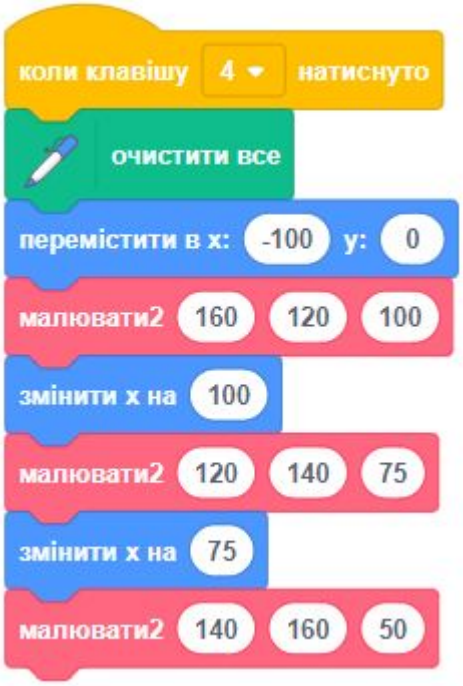
Ці квіти мають однакову кількість зовнішніх і внутрішніх пелюсток, а розмір внутрішньої квітки є певною частиною від розміру зовнішньої:



- Визначимо блок «**малювати2**» з параметрами: зовнішній колір, внутрішній колір і розмір:



- Новий блок може малювати велику кількість квітів такого виду:



- **Завдання.** Створіть новий власний блок з необхідними параметрами, які можна змінювати для створення бажаних квітів. Потім використайте цей блок для малювання квітів!

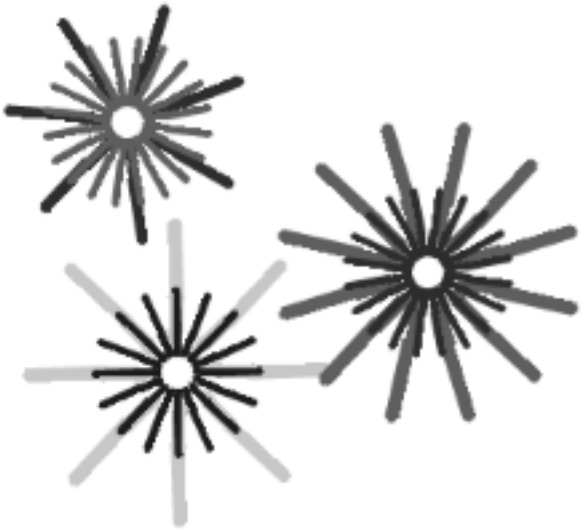
КРОК 8: ЗАВДАННЯ

ЗАВДАННЯ: СТВОРІТЬ ВЛАСНИЙ КВІТКОВИЙ МАЛЮНОК

Чи можете ви використати блок малювання квітки декілька разів, щоб зобразити багато квітів і створити цікавий малюнок? Розташування різних квітів в одному й тому ж місці створює цікавий ефект.

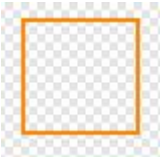
- Створіть малюнок, який вам подобається. Ось приклад:

- Вам не обов’язково використовувати пелюстки в формі еліпсів. Наприклад, ви можете намалювати товсті прямі лінії, щоб створити візерунок у формі феєрверків:



- Пелюсткою для візерунка у формі феєрверків є звичайна лінія:

- Додайте нові образи пелюсток, щоб побачити, які квіти ви намалювали. Спробуйте використати форму спрайта без заливки, наприклад, квадрат, і подивіться, що вийде.

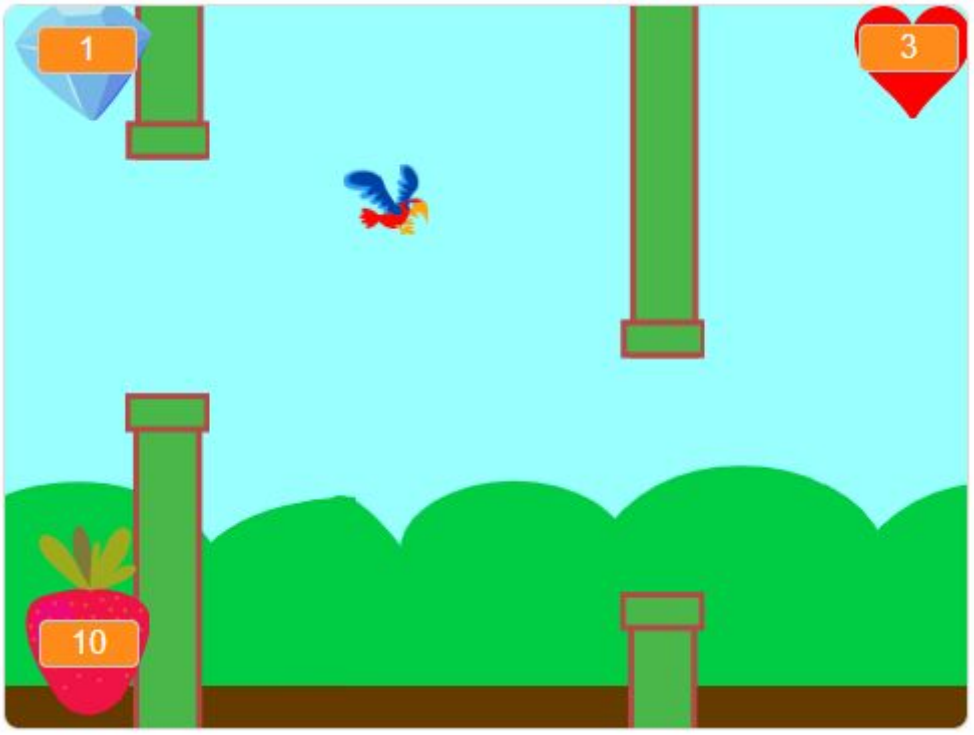


ЗАВДАННЯ: ЗБЕРЕЖІТЬ СВОЇ ЗОБРАЖЕННЯ

Коли ви створите зображення, яке вам подобається, можете його зберегти, щоб використовувати в інших проектах.
Коли на сцені є зображення, яке вам подобається, клацніть на ньому правою кнопкою мишки і виберіть ЗБЕРЕГТИ ЗОБРАЖЕННЯ ЯК.
Також ви можете зберегти зображення сцени в форматі PNG.
ПРИМІТКА: на різних комп’ютерах та браузерах пункт меню для збереження зображення може мати іншу назву.

УРОК 24. ПАПУГА ЛЕТИТЬ НА ПІВДЕНЬ

У цьому проекті ви дізнаєтеся, як створити гру, в якій гравець керує папугою, що пролітає крізь труби, щоб дістатися теплих країв.



Гравець натискає на клавішу ПРОПУСК, щоб папуга махав крилами, і отримує один діамант за кожну трубу, через яку йому вдасться провести папугу.

На цьому уроці ми будемо використовувати:

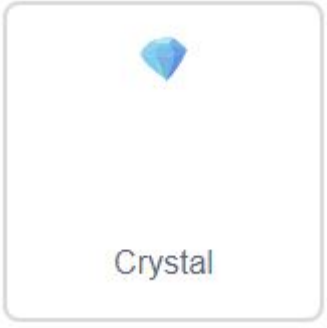
1. Цикли.	3. Повідомлення
2. Клони	4. Змінні

КРОК 1: СПРАЙТИ ТА СЦЕНА

- Почніть новий проект і видаліть з нього спрайт кота.
- Додайте змінні до вашої гри.

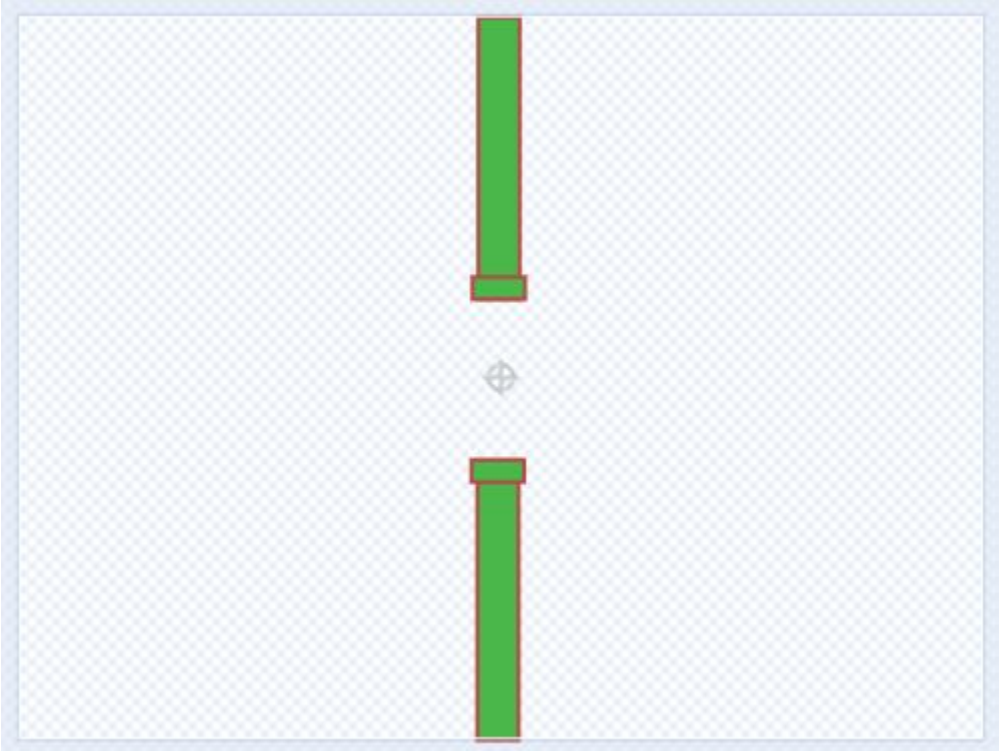
score	Рахунок гри. Збільшується, якщо наш папуга пролітає крізь труби. Показуйте цю змінну на сцені без назви, розмістить її у правому верхньому кутку.
life	Життя папуги. На початку гри життя дорівнює трьом. Кожне зіткнення віднімає одне життя. Коли кількість життів дорівнює нулю - гра закінчується.
food	Показує, чи папуга зголоднів. На початку гри дорівнює 10. Через деякий проміжок часу число зменшується. Щоб збільшити цю змінну, папуга повинен спіймати якусь їжу в повітрі. Якщо ця змінна стає рівною нулю - кінець гри.
bestScore	Найкращий рахунок гри. Наприкінці гри рахунок записується сюди, якщо він більший за поточне значення цієї змінної.

- Додайте такі спрайти з бібліотеки.

	Це наш головний герой. Ви можете обрати інший спрайт, але це має бути літаюча істота. Також у нього мають бути щонайменше два образи, бо ми будемо робити анімацію польоту.
	Цей спрайт потрібно розмістити у лівому верхньому кутку сцени поряд зі змінною рахунка гри. 

	Цей спрайт потрібно розмістити у правому верхньому кутку сцени поряд зі змінною кількості життів папуги. Вам треба відредагувати спрайт, щоб усі його частини були червоними. Змінить розмір спрайта, щоб він був таким же, як у діаманта. 
	Цей спрайт буде інформувати про кількість їжі у папуги, розмістить його поряд зі змінною food на сцені у лівому нижньому кутку. Також треба зменшити спрайт, як ви зробили із попереднім. 
	Додайте два пустих спрайта. Назвіть їх image1 та image2. За допомогою цих двох спрайтів ми будемо анімувати тло сцени.

- **Намалюйте спрайт Pipes** (труби). Цей спрайт повинен являти собою пару труб з отвором посередині. Перемістивши спрайт вгору або вниз, ви зможете змінити місце розташування отвору.
- На цьому малюнку показано приклад того, як труби можуть бути розташовані. Частини спрайта за межами сцени зазвичай приховані, ви бачите їх тільки тоді, коли переміщається спрайт.



- Ви не можете намалювати спрайт таким же великим, як мають бути труби, але ви зможете потім збільшити розмір спрайта на сцені за допомогою блока ЗАДАТИ РОЗМІР.
- Додайте тло з бібліотеки.

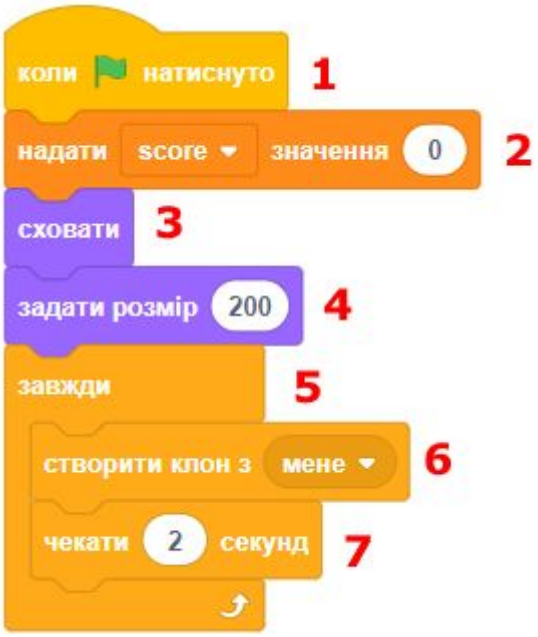
- Перетягніть образ тла у спрайти image1 та image2, щоб він додався до їх образів. Потім треба встановити тло у білий колір.

КРОК 2: ТРУБИ

Ми зробимо так, щоб труби рухалися по сцені, це створить смугу перешкод для папуги.

СКРИПТ 1. СТВОРЕННЯ КЛОНІВ ТА ПОЧАТКОВІ НАЛАШТУВАННЯ

- Цей скрипт постійно створює нові перешкоди для нашого папуги. Також встановлює потрібні початкові значення для труб.



1	Скрипт запускається на початку гри.
2	Встановлюємо рахунок гри в нуль.
3	Нам треба сховати основний спрайт. Ми будемо використовувати тільки його клони.
4	Задаємо потрібний розмір труб, щоб вони перекривали всю сцену. Налаштуйте параметри цього блоку в залежності від розміру ваших труб.
5-7	Під час гри нам потрібно створювати нові клони труби, чекаючи 2 секунди. Клон труби створюється на правій стороні сцени, рухається вліво та потім зникає з лівої сторони сцени.

СКРИПТ 2. ЗБІЛЬШИТИ РАХУНОК ГРИ

- Після того як папуга пролетів крізь трубу, нам потрібно збільшити рахунок нашої гри.

1	Скрипт запускається при створенні клону.
2	Ми чекаємо, поки папуга не пролетить крізь трубу. Він пролетів, коли його x-координата більша за x-координату цього клона труби.
3	Збільшуємо рахунок гри.
4	Відтворюємо потрібний звук. Ви можете використати будь-який звук з бібліотеки.

СКРИПТ 3. НАЛАШТУВАННЯ ТА ПЕРЕМІЩЕННЯ КЛОНУ

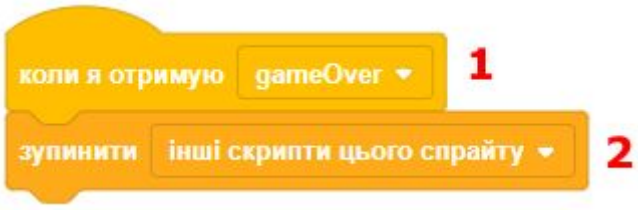
- Цей скрипт встановлює потрібні початкові налаштування для нового клону труби та переміщує його по сцені.

1	Скрипт запускається при створенні клону.
2	Нам треба показати клон труби на сцені, тому що основний спрайт завжди схований.

3	Переміщуємо в початкове положення з правої сторони сцени. У-координата вибирається випадково. Перемістивши спрайт вгору або вниз, ми можемо змінити місце отвору.
4	Переміщуємо трубу вліво до меж сцени. Змініть 4 на інше число, щоб збільшити або зменшити швидкість руху труб.
5	Цей клон труби нам більше не потрібен.

СКРИПТ 4. В КІНЦІ ГРИ

- Цей скрипт зупиняє пересування труб по сцені в кінці гри.



1	Скрипт запускається в кінці гри. Повідомлення gameOver надсилає спрайт папуги.
2	Нам треба зупинити всі скрипти, щоб труби не рухалися.

КРОК 3: ПАПУГА

Тепер запрограмуємо поведінку папуги, щоб він падав на сцені. Потім ми додаємо код, щоб папуга злітав при натисканні клавіші ПРОПУСК.

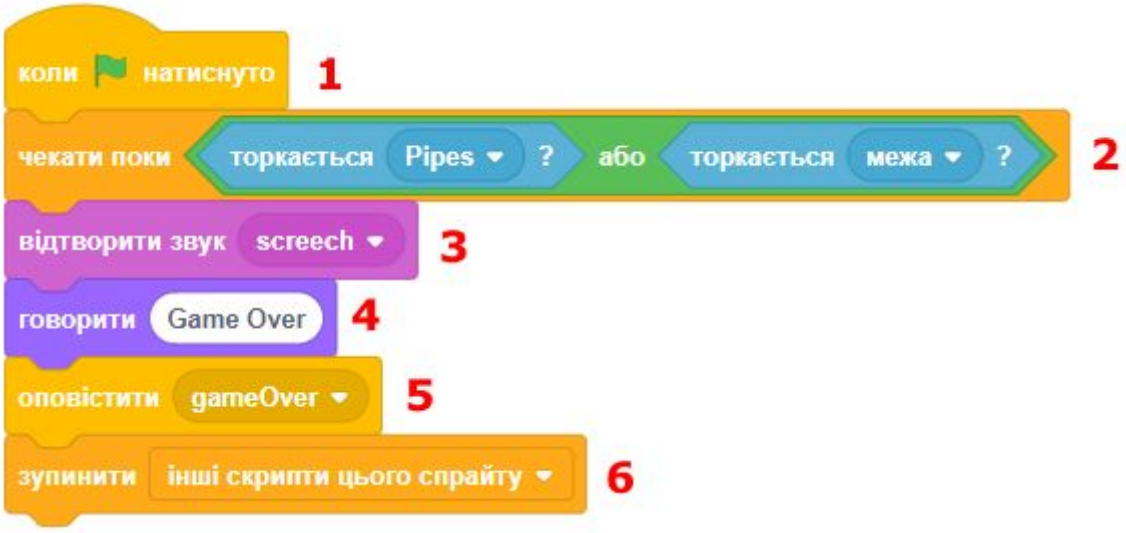
СКРИПТ 1. ПОЧАТКОВІ НАЛАШТУВАННЯ ТА СИЛА ТЯЖІННЯ

- Цей скрипт встановлює початкове положення та розмір папуги. Також ми змушуємо спрайт завжди падати під дією сили тяжіння.

1	Скрипт запускається на початку гри.
2	Зменшуємо нашого головного героя. Якщо у вас інший спрайт, а не папуга, змініть число 25 на інше.
3	Нам треба перемістити папугу в початкове положення.
4-5	Ці блоки програмують дію сили тяжіння. Наш папуга завжди падає вниз, якщо не махає крилами.

СКРИПТ 2. ПЕРЕВІРКА НА КІНЕЦЬ ГРИ

- Цей скрипт перевіряє, чи треба закінчити гру. Гра закінчується, коли папуга торкається труб або меж екрану.



1	Скрипт запускається на початку гри.
2	Чекаємо, поки папуга не торкнеться труби або меж екрану.
3	Відтворюємо звук кінця гри. Ви можете використати будь-який звук з бібліотеки.
4	Папуга говорить про кінець гри.
5	Нам треба сповістити всім іншим спрайтам про кінець гри, щоб вони виконали потрібні дії.
6	Зупиняємо всі скрипти папуги, щоб він перестав рухатися.

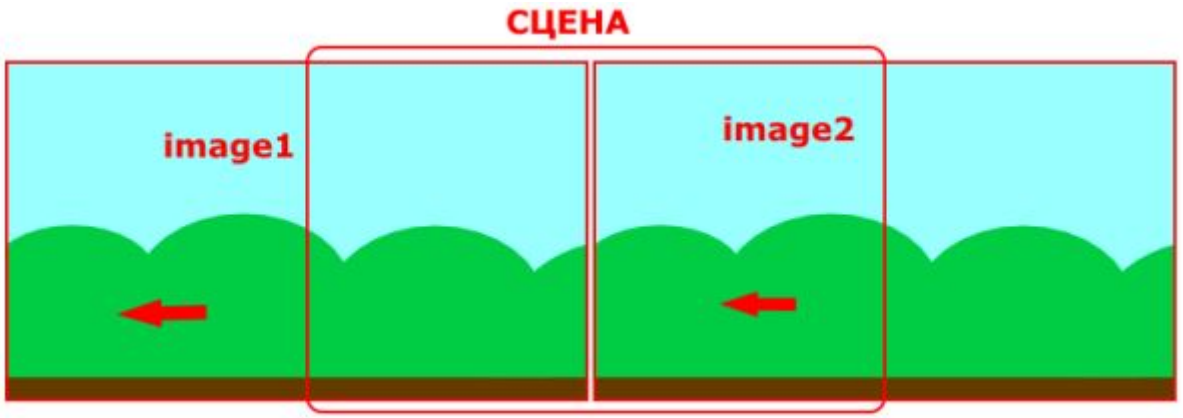
СКРИПТ 3. ПАПУГА ЛЕТИТЬ ВГОРУ

- Цей скрипт дає змогу головному герою летіти вгору.

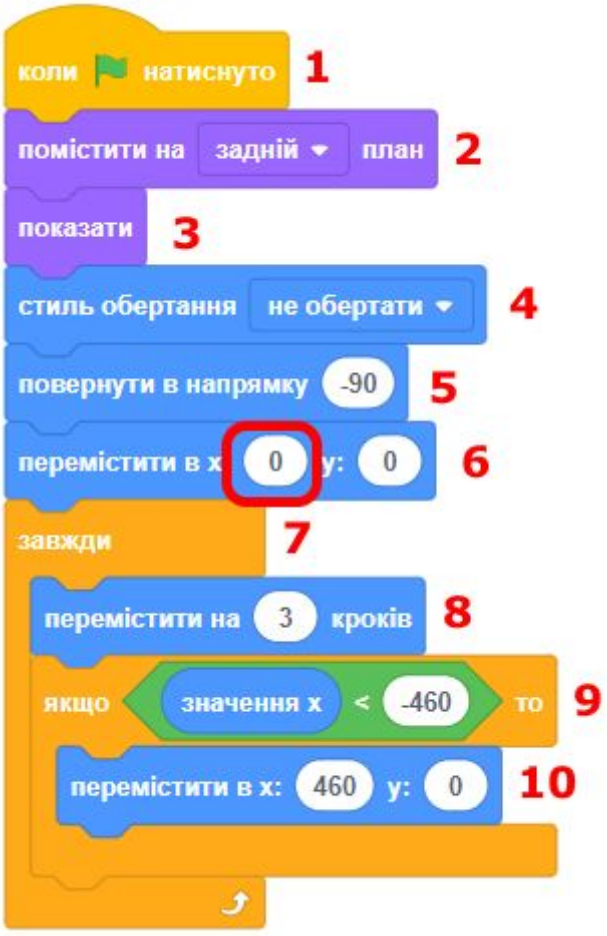
1	Скрипт запускається, коли натиснуто клавішу ПРОПУСК.
2	Змінюємо образ спрайта на потрібний, що більше підходить до стану польоту. Якщо у вас інший спрайт, а не папуга, вам треба змінити назву образу у параметрі цього блоку.
3-4	Ми 10 разів переміщуємо папугу вгору на 6 кроків.
5	Змінюємо образ на початковий.
6-7	Переміщуємо папугу ще трохи вгору.

КРОК 4: АНІМАЦІЯ ТЛА СЦЕНИ

- Ми маємо два спрайта (image1 та image2) для анімації тла сцени. Вони постійно рухаються з правої сторони сцени вліво. Коли ці спрайти зникають зі сцени, вони знову переміщуються вправо, і все починається спочатку.



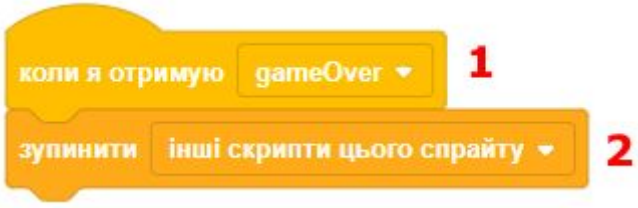
- У image1 та image2 буде однаковий скрипт, але різне початкове положення (блок №6). У image2 там буде (460,0) замість (0,0).
- Вам треба трохи відредагувати образи спрайтів, щоб між ними не було білої лінії. Також потрібно трохи відредагувати зелені кущі так, щоб вони збіглися на обох малюнках.



1	Скрипт запускається на початку гри.
2	Спрайти мають бути позаду всіх інших спрайтів.
3	Про всяк випадок нам треба показати спрайт.

4	Нам не треба обертати спрайти.
5	Встановлюємо напрямок руху спрайта.
6	Встановлюємо початкове положення. У другого спрайта це буде інше значення - (460,0).
7-8	Спрайти постійно рухаються вліво, а нам здається, що папуга летить.
9-10	Якщо спрайт зникає зліва, переміщуємо його вправо за межу сцени.

- Цей скрипт зупиняє анімацію тла сцени.



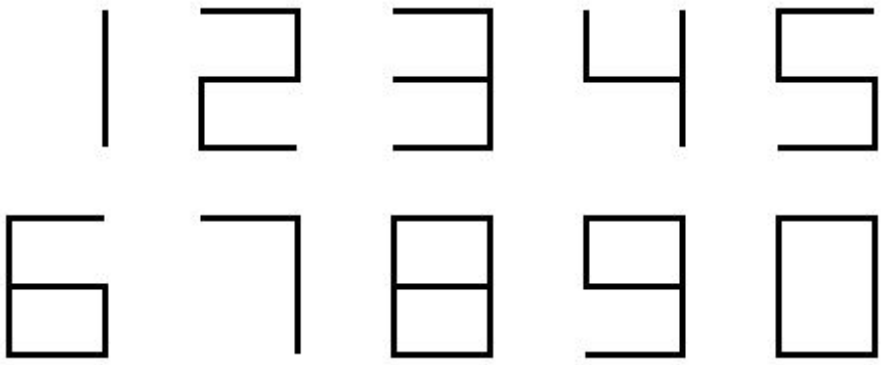
1	Скрипт запускається в кінці гри.
2	Нам треба зупинити всі скрипти, щоб припинити анімацію

КРОК 5: ЗАВДАННЯ

- Гра дуже складна або легка для вас? Скільки способів змінити складність ви можете знайти? Налаштовуйте гру, поки ви не будете задоволені її складністю!
- Чи можете ви додати кращий результат у гру, щоб крім відстеження рахунку у поточному раунді, він відстежував найвищий результат, якого гравець коли-небудь досягав?
- Додайте початкове тло і кнопку початку гри.
- Додайте кінцеве тло з красивим написом Game Over.
- Запрограмуйте так, щоб гра закінчувалася, тільки коли папуга втратить свої три життя.
- Додайте появу фруктів у випадкових місцях, щоб папуга міг поїсти і поповнити змінну food. Якщо папуга не їсть, то через деякий час гра повинна зупинитись.
- Коли папуга торкається труби, змінюйте образ папуги.
- Додайте мисливців, які будуть ходити і полювати на папугу. Якщо куля потрапляє у папугу, його життя зменшується на 1.

УРОК 25. МАЛЮВАННЯ ЧИСЕЛ

У цьому проекті ви дізнаєтеся, як малювати числа великого розміру на сцені. Це дуже корисно, якщо протягом гри вам потрібно показати гравцю його рахунок або інші змінні. Також ми створимо невеликий текстовий редактор, щоб можна було переміщувати курсор по сцені та вводити числа. Знизу сцени ми будемо показувати час, як це робиться у електронному годиннику.

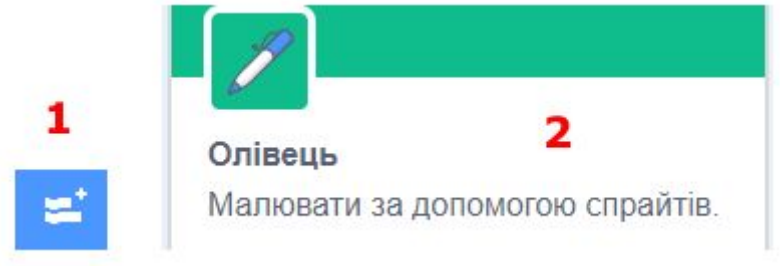


На цьому уроці ми будемо використовувати:

1. Цикли.
2. Розгалуження.
3. Створення власних блоків.
4. Змінні та списки.
5. Повідомлення.

КРОК 1: СПРАЙТИ ТА СЦЕНА

- Почніть новий проект і видаліть з нього спрайт кота.
- У новому Scratch категорія блоків управління олівцем захована в розширеннях. Для того щоб її додати, необхідно натиснути кнопку додавання розширення та вибрати розширення ОЛІВЕЦЬ.



- Додайте змінні та список до вашої гри.

НАЗВА	**ПОЯСНЕННЯ**
charHeight	Ця змінна відповідає за висоту цифри.
charWidth	Ця змінна задає ширину цифри.
charPos	Поточна позиція курсору по горизонталі. Задає номер стовпчика, де буде друкуватися цифра.
linePos	Поточна позиція курсору по вертикалі. Задає номер рядка, де буде друкуватися цифра.
positions	Цей **список** використовується для тимчасового зберігання позицій курсору.

- Додайте такі спрайти.

	Цей спрайт буде інформувати користувача про поточне положення курсору. Там, де блимає цей спрайт, будуть з'являтися цифри, які ви друкуєте.

	Додайте два таких спрайта. Перший буде показувати горизонтальне положення курсору, а другий - вертикальне.
	Додайте пустий спрайт та назвіть його **Painter**. Він буде малювати цифри на сцені. Перший образ цього спрайта має бути абсолютно порожнім, назвіть його **default**. Другий образ має бути прямокутником білого кольору з розміром, що дорівнює розміру цифр, які будуть друкуватися на сцені. Назвіть цей образ **eraser**.

КРОК 2: КЕРУВАННЯ КУРСОРОМ

- Місце, в якому буде друкуватися наступна цифра, вказується за допомогою змінних **charPos** (горизонтальна позиція, номер стовпчика) та **linePos** (вертикальна позиція, номер рядка).

- Верхній лівий кут сцени має координати: charPos=0, linePos=0.
- Сцена може вмістити обмежену кількість цифр - 4 рядки та 13 стовпчиків. Але це залежить від розміру ваших цифр, тому ви легко можете це змінити.

- Нам потрібні деякі скрипти, щоб можна було вказати місце, де буде друкуватися наступна цифра, тому запрограмуємо їх. Усі скрипти робляться для спрайта **Painter**.

СКРИПТ 1. ПЕРЕВІРКА ПОТОЧНОГО НОМЕРА РЯДКА

Цей допоміжний блок перевіряє, чи поточна позиція курсору по вертикалі (рядок, в якому блимає зелена лінія та буде друкуватися наступна введена цифра) є така, як нам треба (в межах від мінімального до максимального значення).

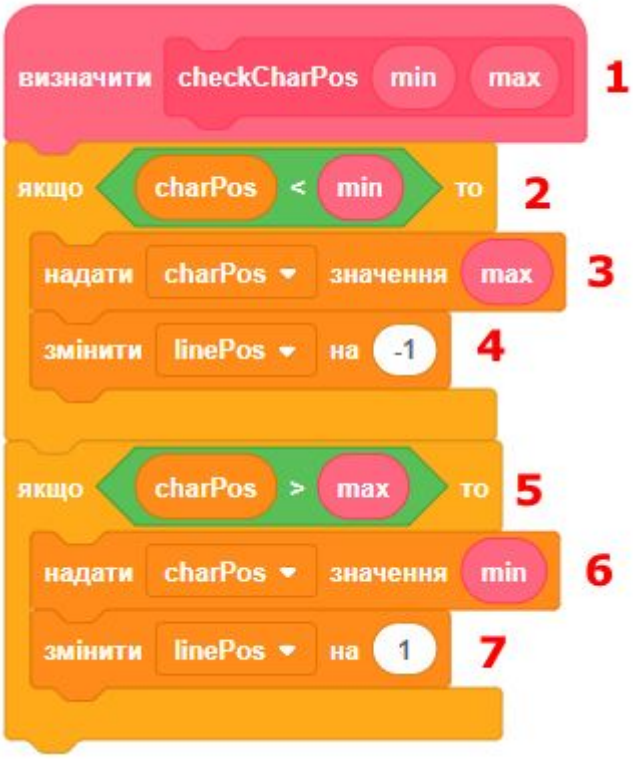
1	Блок має параметри, які задають можливі мінімальне (min) та максимальне (max) значення для змінної **linePos**.
2-3	Якщо поточний рядок менший за мінімальне значення, то встановити його в максимальне значення.
4-5	Якщо поточний рядок більший за максимальне значення, то встановити його в мінімальне значення.

СКРИПТ 2. ПЕРЕВІРКА ПОТОЧНОГО НОМЕРА СТОВПЧИКА

Цей допоміжний блок перевіряє, чи поточна позиція курсору по горізонталі така, як нам треба (в межах від мінімального до максимального значення).

Поточна позиція - це місце, де блимає зелена лінія та буде друкуватися наступна введена цифра.

1	Блок має параметри, які задають можливі мінімальне (min) та максимальне (max) значення для змінної **charPos**.
2-4	Якщо поточний номер стовпчика менший за мінімальне значення, то встановити його в максимальне значення та зменшити поточний номер рядка.
5-7	Якщо поточний номер стовпчика більший за максимальне значення, то встановити його в мінімальне значення та збільшити поточний номер рядка.



СКРИПТ 3. ВСТАНОВИТИ ПОЗИЦІЮ КУРСОРУ

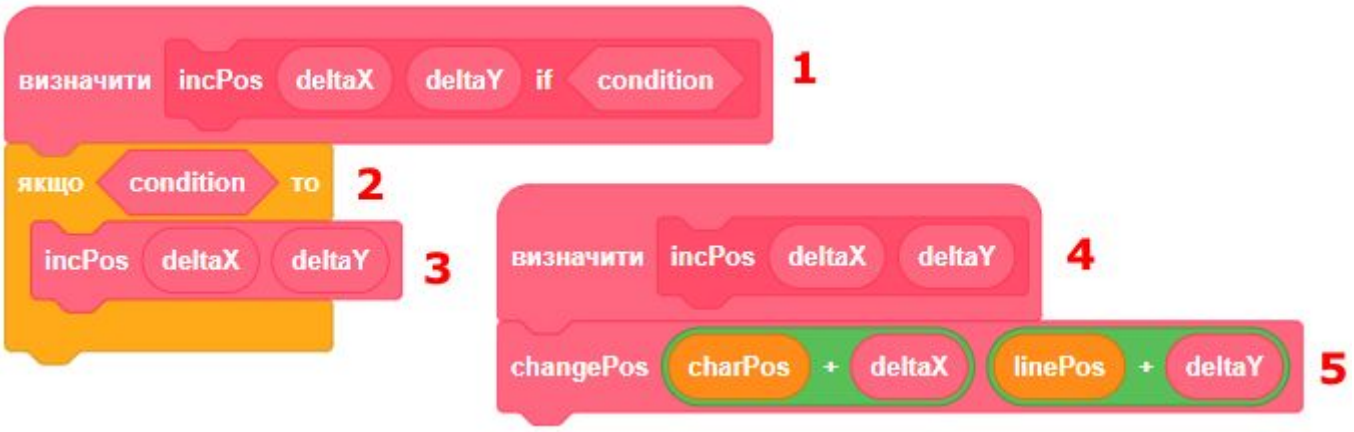
Цей скрипт встановлює поточну позицію курсору.

1	Визначення допоміжного блоку **changePos**, який встановлює нову позицію курсору. Блок має два параметри, які задають нове поточне положення курсору. Параметр **char** - номер стовпчика, **line** - номер рядка.
2-3	Надаємо нові значення змінним **charPos** та **linePos**.

	Ці змінні зберігають позицію курсору.
4-5	За допомогою блоків перевірки максимального та мінімального значень вказуємо можливу кількість стовпчиків та рядків. Ці значення залежать від розміру цифр.
6-7	Задаємо нове положення спрайта, керуючись поточними номерами рядка і стовпчика, розміром цифр та початковою точкою (-194, 119) на сцені.
8	Треба повідомити всі інші спрайти, що положення курсору змінилося, щоб вони відреагували.

СКРИПТ 4. ЗМІНИТИ ПОЗИЦІЮ КУРСОРУ

Ці два блоки змінюють поточну позицію курсору, додаючи до неї значення своїх параметрів.



1, 4	Блоки мають параметри, що задають числа, на які треба змінити поточну позицію курсору. Параметр **deltaX** задає число, на яке треба змінити номер стовпчика. Параметр **deltaY** задає число, на яке треба змінити номер рядка. Параметр **condition** вказує, чи потрібно щось змінювати.
2-3	Якщо параметр **condition** дорівнює ПРАВДА, то нам треба викликати блок для зміни позиції курсору.
5	Встановлюємо нове положення курсору, додаючи до старого положення значення параметрів блока.

СКРИПТ 5. ПОЧАТКОВІ НАЛАШТУВАННЯ

Цей скрипт виконується на початку роботи програми. Він встановлює початкові значення змінних та налаштовує олівець.

1	Скрипт запускається на початку гри.
2	Олівець буде малювати цифри та літери зазначеним кольором.
3	Товщина ліній, якими будуть малюватися цифри.
4-5	Піднімаємо олівець та змінюємо образ спрайта на прозорий.
6-7	Задаємо висоту та ширину цифри.
8	Встановлюємо початкову позицію курсору. Це буде лівий верхній кут сцени.
9	Нам треба очистити сцену від попередніх малюнків.

10	Повертаємо спрайт праворуч.

КРОК 3: МАЛЮВАННЯ ЧИСЕЛ

СКРИПТ 1. ДОПОМІЖНІ БЛОКИ МАЛЮВАННЯ ЧИСЕЛ

Ці два блока будуть використовуватись для малювання цифр на сцені.

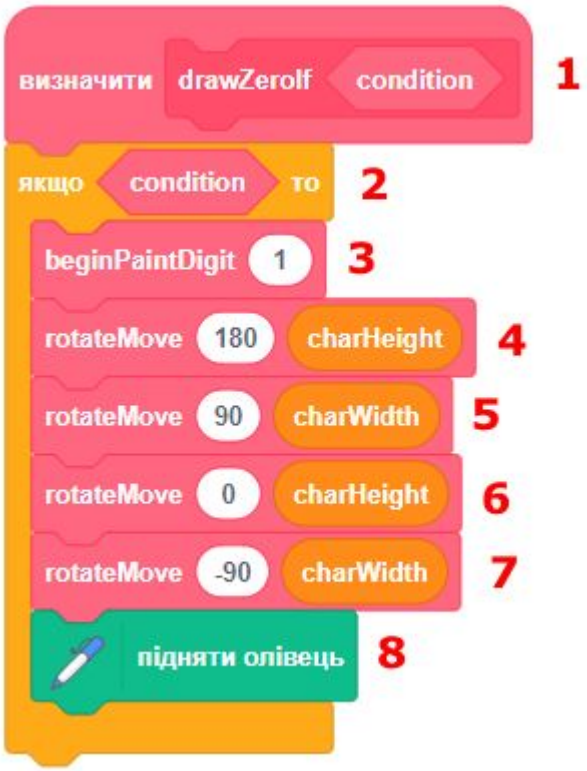
1	Визначення допоміжного блока **beginPaintDigit**, який встановлює початкову позицію для малювання та опускає олівець. Параметр **isLeft** вказує, чи початкове положення по горизонталі буде ліворуч (**isLeft**=1) або в центрі.
2	Встановлюємо початкове положення для малювання по вертикалі. Це робимо за допомогою переміщення спрайта вгору на половину його висоти.
3-4	Якщо треба (**isLeft**=1), переміщуємо початкове положення для малювання ліворуч на половину ширини спрайта.

5	Перед початком малювання нам треба опустити олівець.
6	Визначення допоміжного блоку **rotateMove**, який повертає спрайт у заданому напрямку (параметр **degree**) та переміщує на зазначену кількість кроків (параметр **steps**).
7-8	Повертає спрайт і переміщує його на потрібну відстань.

СКРИПТ 2. МАЛЮВАННЯ НУЛЯ

Цей скрипт малює на сцені цифру нуль.

1	Визначення блока малювання нуля. Параметр **condition** вказує, чи взагалі потрібно щось малювати.
2	Якщо **condition** дорівнює ПРАВДА, то ми будемо малювати.
3	Перед тим як малювати, нам треба встановити спрайт у початкове положення, викликавши блок **beginPaintDigit**.
4-7	Блоки малювання нуля. Дивиться на зображення нуля поряд зі скриптом для пояснення.
8	Малювання закінчено, нам треба підняти олівець.



СКРИПТ 3. МАЛЮВАННЯ ОДИНИЦІ

Цей скрипт малює на сцені цифру один.

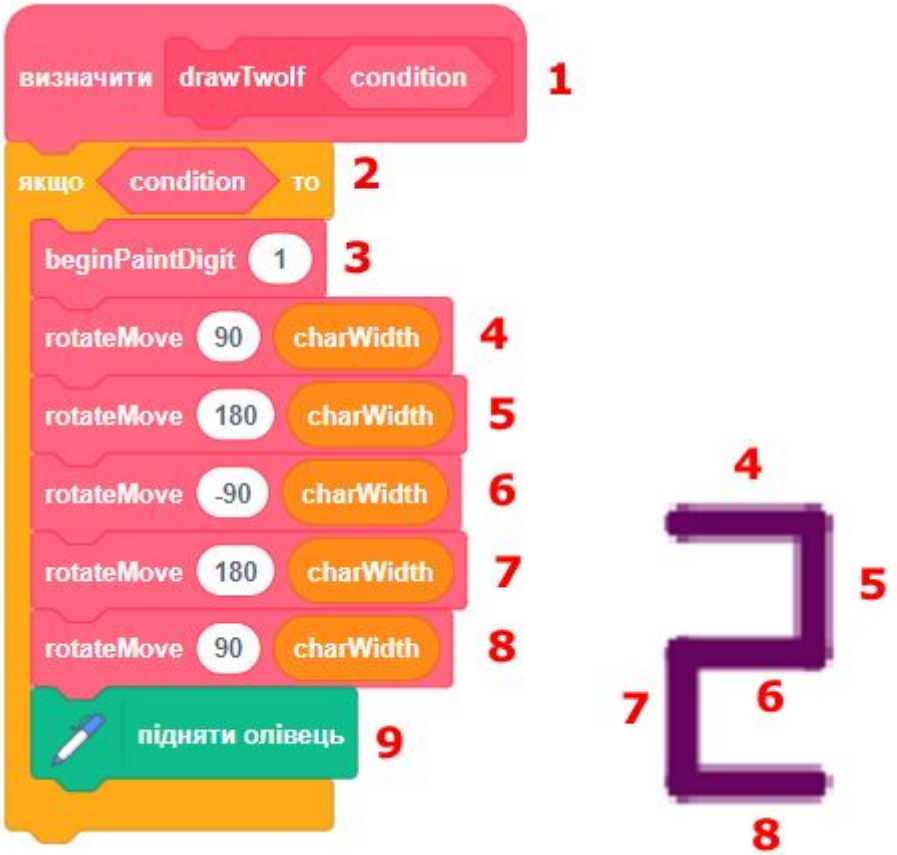
1	Визначення блока малювання одиниці. Параметр **condition** вказує, чи взагалі потрібно щось малювати.
2	Якщо **condition** дорівнює ПРАВДА, то ми будемо малювати.
3	Перед тим як малювати, нам треба встановити спрайт у початкове положення, викликавши блок **beginPaintDigit**.

4	Одиниця - це вертикальна лінія в центрі спрайта.
5	Малювання закінчено, нам треба підняти олівець.

СКРИПТ 4. МАЛЮВАННЯ ДВІЙКИ

Цей скрипт малює на сцені цифру два.

1	Визначення блока малювання двійки.
2	Параметр **condition** вказує, чи взагалі потрібно щось малювати. Якщо він дорівнює ПРАВДА, то ми будемо малювати.
3	Перед тим як малювати, нам треба встановити спрайт у початкове положення, викликавши блок **beginPaintDigit**.
4-8	Блоки малювання двійки. Дивиться на зображення двійки поряд зі скриптом для пояснення.
9	Малювання закінчено, нам треба підняти олівець.



СКРИПТ 5. МАЛЮВАННЯ ДВОКРАПКИ

Цей скрипт малює на сцені двокрапку. Він використовується, коли ми друкуємо поточний час, для розділення чисел годин, хвилин і секунд.

1	Визначення блока малювання двокрапки. Параметр **condition** вказує, чи взагалі потрібно щось малювати.
2	Якщо **condition** дорівнює ПРАВДА, то ми будемо малювати.
3	Переключити образ спрайта на той, де намальована двокрапка.
4	Проштампувати образ спрайта на сцені.
5	Повернути початковий образ спрайта.

Щоб цей блок працював як треба, нам потрібно додати новий образ до спрайта **Painter**, в якому буде намальована двокрапка.
Назвіть цей образ **dots**.

СКРИПТ 6. ЗБЕРЕГТИ ТА ВІДНОВИТИ ПОЗИЦІЮ КУРСОРУ

Ці два скрипта дозволяють зберегти (блок **pushPos**) та відновити (блок **popPos**) позицію курсору.

Вони використовують список **positions** для збереження даних.

1	Блок не має додаткових параметрів.
2	Зберегти змінну **charPos**. Для цього нам треба додати її до кінця списку **positions**.
3	Зберегти змінну **linePos**. Це робиться таким же чином.

1	Блок не має додаткових параметрів.
2-3	Відновити значення змінної **linePos** та видалити останній елемент зі списку, в якому воно зберігалося.
4-5	Відновити значення змінної **charPos** та видалити останній елемент зі списку, в якому воно зберігалося.
6	Перемістити курсор у нове значення змінних charPos та linePos.

СКРИПТ 7. МАЛЮВАННЯ СИМВОЛУ

Цей блок малює всі відомі нам символи на сцені. В залежності від параметру **char** він викликає інші блоки малювання.

1	Визначення блока малювання символу. Параметр **char** вказує, якій символ потрібно малювати.
2	Зберегти поточну позицію курсору.
3-5	Змінити образ спрайта на білий прямокутник, залишити його штамп на сцені та змінити образ знову на прозорий.
6-9	Викликаємо блоки малювання відомих нам символів. Якщо будете програмувати нові блоки малювання, їх треба додавати в це місце у скрипті, після 9 рядка.
10	На всякий випадок повертаємо в початковому напрямку.

11	Відновити позицію курсору, яка була збережена в 2 рядку.
12	Переміщуємо курсор ліворуч на 1 позицію.

КРОК 4: МАЛЮВАННЯ ЧАСУ

Тепер давайте запрограмуємо малювання поточного часу в спеціальному місці на сцені.

СКРИПТ 1. МАЛЮВАННЯ ДВОЗНАЧНОГО ЧИСЛА

Цей скрипт вміє малювати числа від 0 до 99.

1	Визначення блока малювання двозначного числа. Параметр **value** вказує, яке число потрібно малювати.
2	Викликаємо блок малювання для лівої цифри числа. Для отримання значення лівої цифри ми використовуємо блоки з категорії ОПЕРАТОРИ.
3	Викликаємо блок малювання для правої цифри числа.

СКРИПТ 2. МАЛЮВАННЯ ЧАСУ

Цей скрипт вміє малювати поточний час у вказаному місці на сцені.

1	Визначення блока малювання поточного часу. Параметри вказують, де потрібно малювати (номери стовпчика та рядка).
2	Зберегти поточну позицію курсору.
3	Встановити курсор у місце, де нам потрібно малювати час.
4	Блок малювання годин.
5	Малювати двокрапку між годинами та хвилинами.
6	Блок малювання хвилин.
7	Малювати двокрапку між хвилинами та секундами.
8	Блок малювання секунд.
9	Відновити позицію курсору, яка була збережена в 2 рядку.

КРОК 5: ПОКАЖЧИКИ КУРСОРУ ТА ОБРОБКА КЛАВІШ

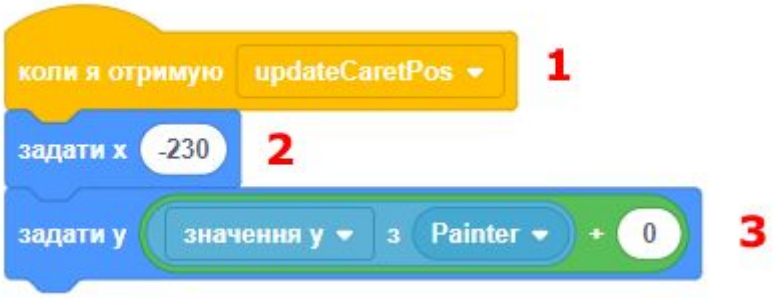
У нас є три індикатори, які показують поточну позицію курсору. Кожен з них отримує повідомлення про зміну поточної позиції та переміщується у нове місце.

СКРИПТ 1. ПОКАЖЧИК ПОТОЧНОГО РЯДКА

Цей спрайт (червона стрілка під назвою **Arrow1**) відображається на сцені ліворуч та вказує на поточний рядок, в якому буде намальована наступна цифра.

Нам потрібно оновити положення спрайта **Arrow1**, коли ми отримуємо повідомлення про зміну поточного рядка.

1	Цей скрипт починає працювати, коли отримує повідомлення про зміну поточної позиції курсору.
2	X-координата не змінюється, вона завжди біля лівої межі сцени.
3	Ми просто беремо y-координату нашого малювальника.



СКРИПТ 2. ПОКАЖЧИК ПОТОЧНОГО СТОВПЧИКА

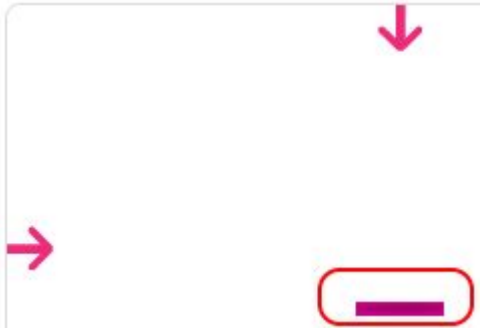
Цей спрайт (червона стрілка під назвою **Arrow2**) відображається зверху сцени та вказує на поточний стовпчик, в якому буде намальована наступна цифра.

Нам потрібно оновити положення спрайта **Arrow2**, коли ми отримуємо повідомлення про зміну поточного стовпчика.

1	Цей скрипт починає працювати коли отримує повідомлення про зміну поточної позиції курсору.
2	Ми просто беремо x-координату нашого малювальника.
3	У-координата ніколи не змінюється, вона завжди біля верхньої межі сцени.

СКРИПТ 3. ПОКАЖЧИК КУРСОРУ

Цей спрайт (маленький прямокутник під назвою **Paddle**) відображається точнісінько там, де буде малюватися наступна цифра.

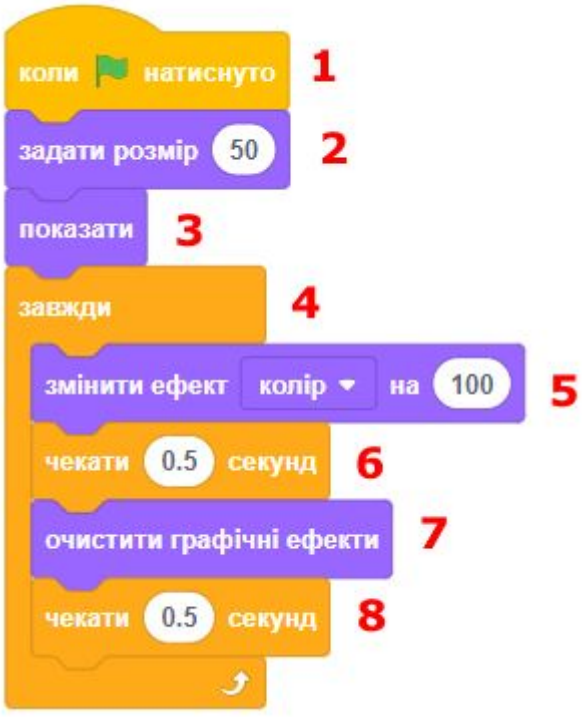


Нам потрібно оновити положення спрайта **Paddle**, коли ми отримуємо повідомлення про зміну позиції курсору.

1	Цей скрипт починає працювати, коли отримує повідомлення про зміну поточного положення курсору.
2-3	Ми беремо x-координату та y-координату малювальника.
4	Нам треба додатково перемістити спрайт трохи вниз.

СКРИПТ 4. АНІМАЦІЯ КУРСОРУ

Цей скрипт анімує поточну позицію курсору за допомогою зміни його кольору. Також він робить початкові налаштування спрайта курсору. Додайте цей скрипт до спрайта **Paddle**.



1	Цей скрипт запускається при старті програми.
2	Нам треба зменшити розмір спрайта вдвічі.

3	На всякий випадок показуємо спрайт.
4	Спрайт буде постійно блимати, тому нам весь час треба виконувати анімацію.
5-6	Змінюємо ефект колір, на сцені спрайт буде тимчасово відображатися трохи іншого кольору. Після цього ми чекаємо пів секунди.
7-8	Відміняємо графічні ефекти та знову трохи чекаємо.

СКРИПТ 5. ОБРОБКА КЛАВІШ КЛАВІАТУРИ

Цей скрипт чекає натиснення потрібних клавіш та викликає допоміжні блоки, які їх обробляють.

1	Цей скрипт запускається при старті нашої програми.
2	Весь час ми виконуємо потрібні перевірки за допомогою циклу.

3	Намалювати нуль, якщо натиснута клавіша 0.
4	Намалювати одиницю, якщо натиснута клавіша 1.
5	Намалювати двійку, якщо натиснута клавіша 0.
6	Вставити порожній символ, якщо натиснута клавіша ПРОПУСК.
7-8	Перемістити курсор праворуч або ліворуч, якщо відповідна клавіша натиснута.
9	Перемістити курсор вниз, якщо відповідна клавіша натиснута.
10	Перемістити курсор вгору, якщо відповідна клавіша натиснута. Після цього блока ви можете додати перевірки натискання інших клавіш клавіатури.
11	Треба почекати, щоб не обробляти клавіши надто швидко. Змініть параметри цього блока, якщо клавіатура працює дуже швидко або дуже повільно.
12	Викликаємо малювання поточного часу в лівому нижньому кутку сцени.

КРОК 7: ЗАВДАННЯ

ЗАВДАННЯ 1. МАЛЮВАННЯ ІНШИХ ЦИФР

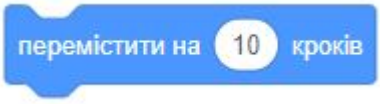
- Мабуть, ви помітили, що ми запрограмували малювання тільки трьох цифр - нуля, одиниці та двійки. Вашим завданням буде додати скрипти малювання інших цифр (від 3 до 9).
- Подивіться, як зроблено скрипт 3:4 (малювання двійки), та запрограмуйте інші блоки малювання таким же чином.
- Також вам треба додати виклик нових блоків малювання до скрипта 3:7 (малювання символа) після 9 рядка.
- Протестуйте програму, щоб вона працювала для всіх цифр.

ЗАВДАННЯ 2. КНОПКИ ВИБОРУ КОЛЬОРУ ЦИФР

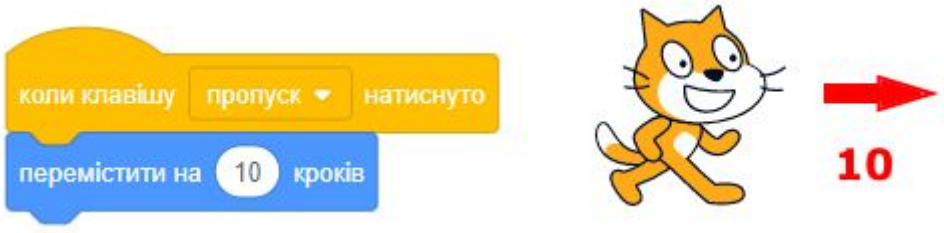
- Додайте справа на сцені пять кнопок для вибору кольору при малюванні цифр, як ми робили це у 8 уроці, коли програмували редактор малюнків. Протестуйте, що все працює.

ДОВІДНИК: РУХ

РУХ: ПЕРЕМІСТИТИ НА

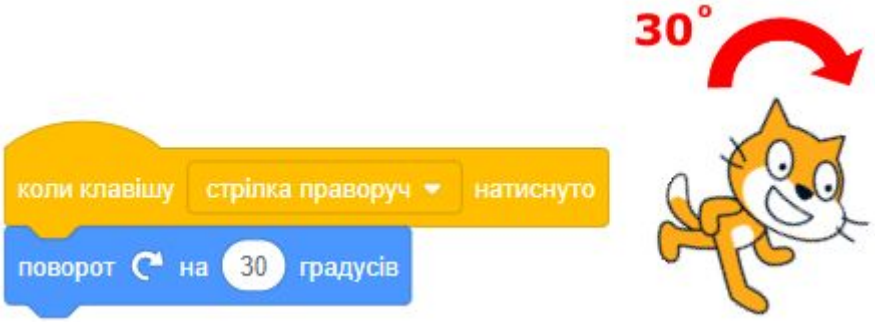


- Перемістити спрайт на певну кількість кроків. Спрайт буде рухатися в поточному напрямку.
- Крок є дуже малою відстанню. Сцена містить 480 кроків по горизонталі і 360 кроків по вертикалі.
- Введіть число, яке визначає, як далеко треба перемістити спрайт.
- Якщо ввести від'ємне число (наприклад, -10), спрайт буде рухатися в протилежному напрямку.
- У цьому прикладі кіт йде вперед на 10 кроків, коли натискається клавіша ПРОПУСК:

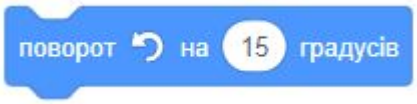


РУХ: ПОВОРОТ ПРАВОРУЧ

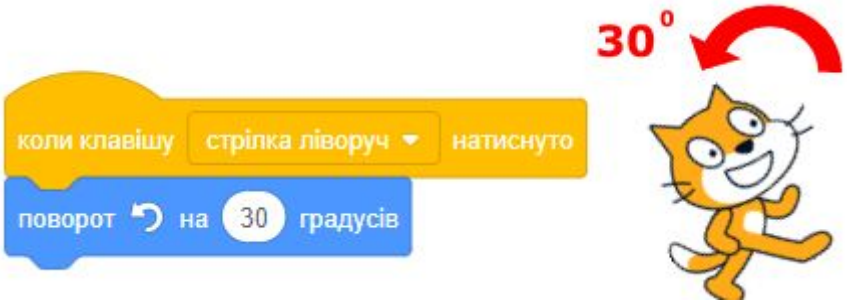
Поворот праворуч. Вкажіть число градусів, на яке повинен повернутися спрайт. Якщо вказати від'ємне число (наприклад, -30), спрайт повернеться в протилежному напрямку (наліво).



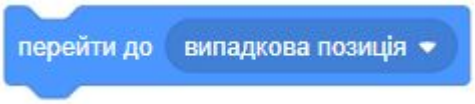
РУХ: ПОВОРОТ ЛІВОРУЧ



Поворот ліворуч. Введіть число градусів кута, на який слід повернути спрайт. Якщо вказати від'ємне число (наприклад, -30), спрайт буде повертатися в протилежному напрямку (направо).



РУХ: ПЕРЕЙТИ ДО



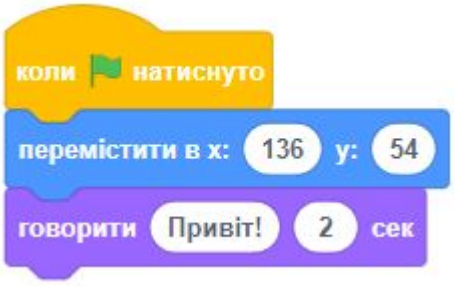
Перемістити спрайт на позицію вказівника мишки, у випадкову позицію або в позицію іншого спрайта. При виконанні цього блока, координати спрайта змінюються.

Нову позицію для спрайта ви можете вибрати з меню, яке випадає коли натиснути на трикутник. У списку ви побачите всі спрайти проекту, а також **випадкову позицію** та **вказівник мишки**.

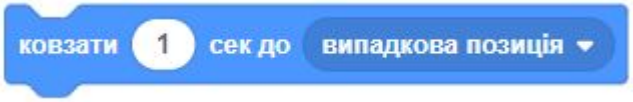
РУХ: ПЕРЕМІСТИТИ В X,У

Перемістити спрайт на задану позицію. Введіть X та У, щоб вказати точку, в яку треба миттєво перенести спрайт на сцені.

У прикладі при натисканні на зелений прапор спрайт переміщається на позицію (136, 54) і говорить слово «Привіт!» протягом 2 секунд.



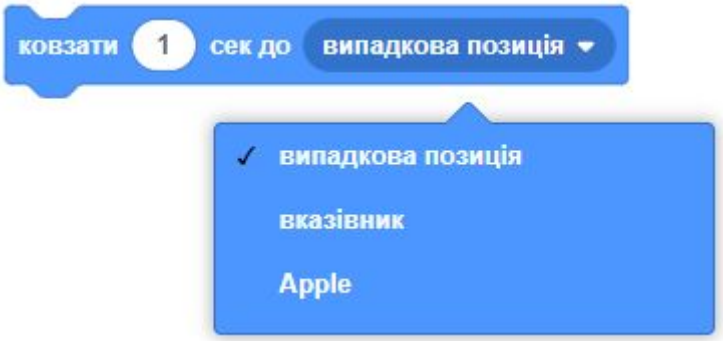
РУХ: КОВЗАТИ ДО



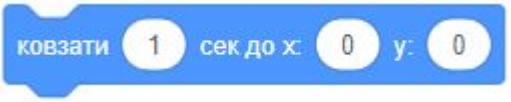
Ковзання протягом певного часу до іншого спрайта, випадкової позиції або вказівника мишки. Спрайт починає рух, ковзаючи з однієї точки в іншу.

Використовуйте ковзання, щоб плавно перемістити спрайт на іншу позицію на сцені. Якщо часу мало, а відстань велика - спрайт буде пересуватися ривками.

Клацніть на трикутник, щоб обрати, куди спрайт буде ковзати:



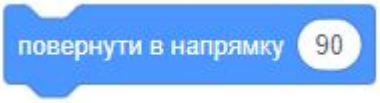
РУХ: КОВЗАТИ ДО X,У



Ковзання протягом певного часу до позиції (x, y).
Спрайт починає рух, ковзаючи з однієї точки в іншу.
Використовуйте ковзання, щоб плавно перемістити спрайт на іншу позицію. Якщо час малий, а відстань велика - спрайт буде пересуватися дуже швидко. Приклад:

У цьому прикладі спрайт постійно ковзає по сцені, обираючи випадкові позиції. Час ковзання задається випадковим числом секунд.

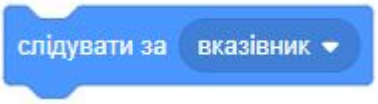
РУХ: ПОВЕРНУТИ В НАПРЯМКУ



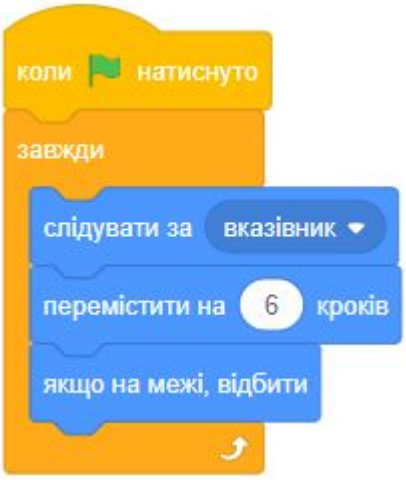
Задати напрямок спрайта в градусах.
Клацніть мишкою всередині білого кола, щоб вибрати потрібний напрямок за допомогою діалогу.

Також ви можете ввести число від 0 до 360 за допомогою клавіатури.
Напрямок: **90** - направо, **-90** - наліво, **0** - вгору, **180** - вниз.

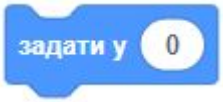
РУХ: СЛІДУВАТИ ЗА



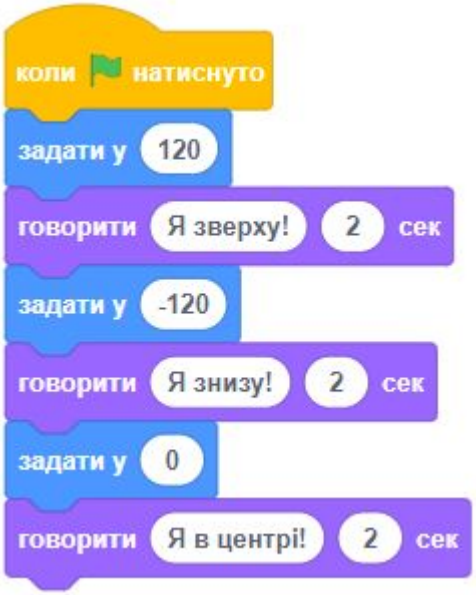
Повертає спрайт до вказівника мишки або до іншого спрайта. Після виконання цього блока напрямок спрайта змінюється.
Потрібний напрямок спрайта ви можете вибрати з меню, натиснувши на трикутник. Меню включає в себе всі інші спрайти проекту, крім обраного, а також вказівник мишки.



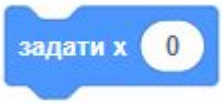
РУХ: ЗАДАТИ У



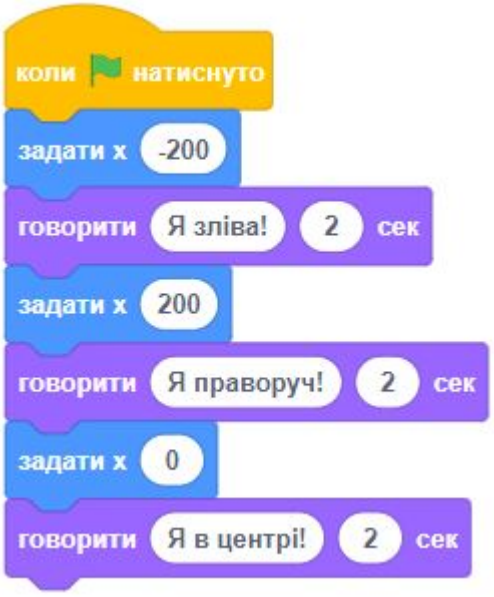
Встановлює у-координату спрайта (вгору-вниз, вертикальна позиція).



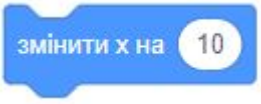
РУХ: ЗАДАТИ X



Встановлює x-координату спрайта на сцені (горизонтальна позиція). Якщо координата X має значення від -240 до 240, спрайт знаходиться на сцені. При інших значеннях він знаходиться поза сценою.

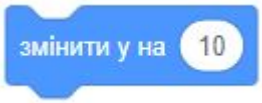


РУХ: ЗМІНИТИ X НА



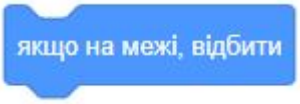
Змінити x-координату спрайта на зазначену величину.
Значення параметру блока додається до поточного значення координати x. При позитивному значенні спрайт зміститься праворуч. При від'ємному значенні спрайт переміститься ліворуч. Приклад:

РУХ: ЗМІНИТИ У НА



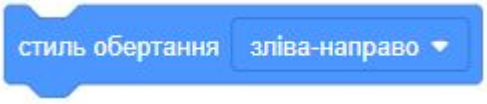
Змінити у-координату спрайта на зазначену величину.
Значення параметру блока додається до поточного значення координати у. При позитивному значенні спрайт переміститься вгору. При від'ємному значенні спрайт зміститься вниз. Приклад:

РУХ: ЯКЩО НА МЕЖІ, ВІДБИТИ



Відскочити, доторкнувшись до краю сцени. Спрайт відскочить під кутом, якщо торкнеться бічних, верхнього або нижнього країв сцени. Коли спрайт торкається краю сцени (координати перевищують розмір сцени), то програма відраховує від поточного напряму 180 градусів (повертає спрайт у зворотному напрямку).
Наприклад, 90 змінюється на -90, 0 змінюється на -180, -135 змінюється на 45.

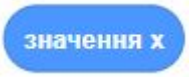
РУХ: СТИЛЬ ОБЕРТАННЯ



Задає стиль обертання спрайта, коли змінюється напрямок його руху. Натисніть на трикутник, щоб змінити стиль обертання.

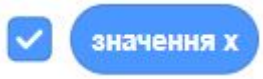
Виберіть **ЗЛІВА-НАПРАВО**, щоб спрайт обертався тільки по горизонталі (ліворуч та праворуч).
Виберіть **НАВКОЛО**, щоб спрайт обертався в усіх напрямках.
Виберіть **НЕ ОБЕРТАТИ**, щоб спрайт мав тільки один напрямок.

РУХ: ЗНАЧЕННЯ X

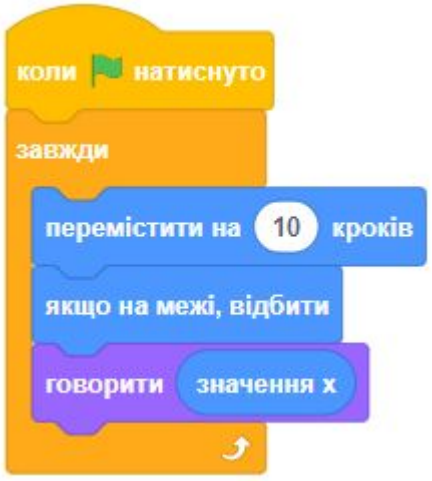


Повертає x-координату спрайта.
X-координата задає положення спрайта на горизонтальній лінії.
Якщо X має значення від -240 до 240, спрайт розташований на сцені.
При інших значеннях він знаходиться поза сценою.
Центр екрана має координату: X=0.
Від'ємні координати (позначені з мінусом) знаходяться ліворуч від центру екрана. Позитивні - праворуч.

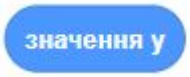
Щоб показати x-координату спрайта на сцені, встановіть галочку поруч з блоком на палітрі блоків.



У цьому прикладі спрайт постійно переміщується по сцені і говорить свою X-координату.



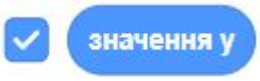
РУХ: ЗНАЧЕННЯ У



Повертає у-координату спрайта.

У-координата задає положення спрайта на вертикальній лінії.
Якщо у-координата від -180 до 180, спрайт розташований на сцені.
При інших значеннях він знаходиться поза сценою.
Центр екрана має координату: Y=0.
Від'ємні координати (позначені з мінусом) знаходяться внизу від центру екрана. Позитивні - зверху.

Щоб показати у-координату спрайта на сцені, встановіть галочку поруч з блоком на палітрі блоків.



У цьому прикладі спрайт постійно перевіряє, чи його у-координата менше за -150. Якщо так, гра закінчується.

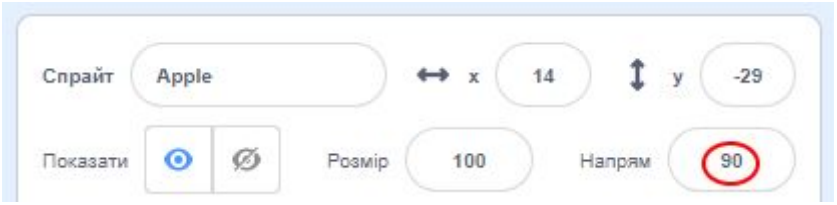
РУХ: НАПРЯМ

Повідомляє поточний напрямок спрайта (в який бік повернуто спрайт). Напрямок спрайта вимірюють у градусах (від 0 до 360).

Якщо спрайт повернутий вгору, це напрямок 0.
Якщо спрайт повернутий праворуч, його напрямок дорівнює 90.
Якщо спрайт повернутий вниз, його напрямок дорівнює 180.
Отже: **90** - направо, **-90** - наліво, **0** - вгору, **180** - вниз.

Щоб напрямок відображався на сцені, встановіть відповідну позначку в панелі блоків:

Поточний напрямок спрайта можна подивитись, а також змінити в інформаційному вікні спрайта. Натисніть мишкою, щоб обрати напрям та стиль обертання у додатковому вікні.



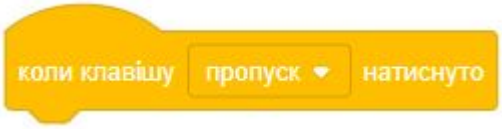
ДОВІДНИК: ПОДІЇ

ПОДІЇ: КОЛИ НАТИСНУТО ЗЕЛЕНИЙ ПРАПОР

Виконує скрипт, коли користувач клацає по зеленому прапору.

Це блок початку виконання програми. Приєднайте знизу до цього блока скрипт, який повинен виконуватися при старті проекту. В проекті може бути багато стартових блоків, всі вони будуть виконуватися одночасно.

ПОДІЇ: КЛАВІШУ НАТИСНУТО

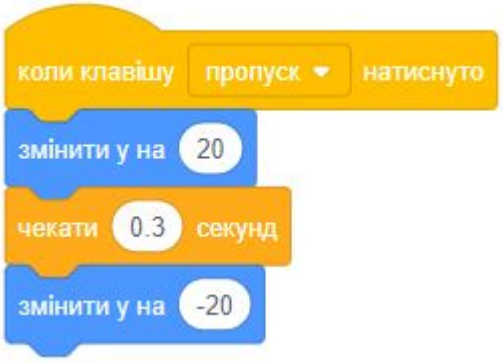


Запускає приєднаний скрипт, коли натиснута клавіша на клавіатурі. Натисніть на трикутник, щоб вибрати потрібну клавішу.

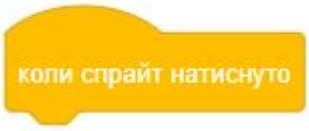
Якщо вибираєте англійську літеру, переконайтеся, щоб для клавіатури була встановлена англійська мова.

Переключити мову в Windows можна за допомогою одночасного натискання Alt та Shift (в залежності від налаштування це може бути також Ctrl та Shift).

У цьому прикладі кіт підстрибує, коли натиснути ПРОПУСК.

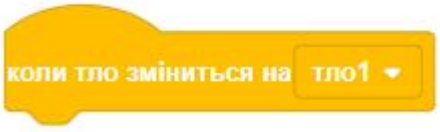


ПОДІЇ: КОЛИ СПРАЙТ НАТИСНУТО



Запускає приєднаний скрипт, коли на спрайт натиснуто. Скрипт виконується, якщо лівою клавішею мишки клацнути на спрайті.

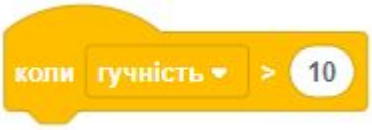
ПОДІЇ: КОЛИ ТЛО ЗМІНЮЄТЬСЯ НА



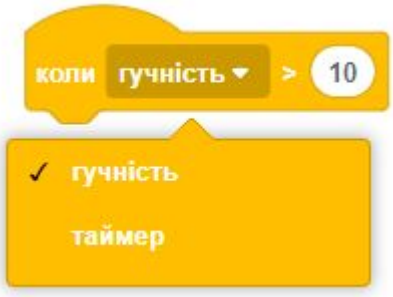
Виконує скрипт, якщо тло сцени переключиться на вказане.

У цьому прикладі при натисканні на клавішу ПРОПУСК змінюється тло сцени. Ми додали два тла (Пустеля і Ферма). Кожен раз при зміні тла спрайт повідомляє, де він знаходиться.

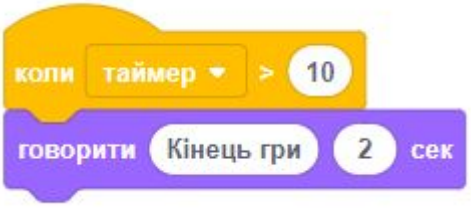
ПОДІЇ: КОЛИ ЗМІННА БІЛЬШЕ



Запускає приєднаний скрипт, коли обрана змінна (гучність, таймер) більше, ніж вказане значення.
Ви можете вибрати змінну в меню, що випадає.



У цьому прикладі ми відстежуємо за допомогою таймера 10 секунд з початку гри.



ДОВІДНИК: КЕРУВАННЯ

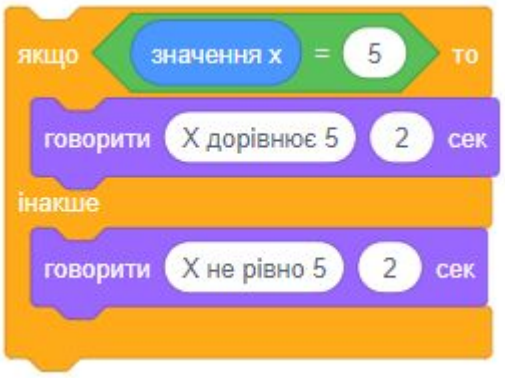
РОЗГАЛУЖЕННЯ

КЕРУВАННЯ: ЯКЩО...ТО

Якщо умова є ІСТИНА (true), виконуються блоки всередині.
Блок ЯКЩО...ТО майже найважливіший у програмуванні. Слова ЯКЩО й ТО по'єднують умову та наслідок. Якщо логічна умова істинна, то буде виконуватися наслідок. Задайте умову в порожньому шестикутнику, а наслідок - всередині блока. Приклад:

КЕРУВАННЯ: ЯКЩО...ТО...ІНАКШЕ

Якщо умова є ІСТИНА (true), виконуються блоки всередині частини ЯКЩО.
Якщо умова є БРЕХНЯ (false), виконуються блоки всередині частини ІНАКШЕ.
Задайте умову в порожньому шестикутнику.
Приклад:

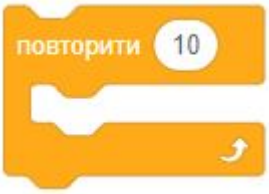


ЦИКЛИ

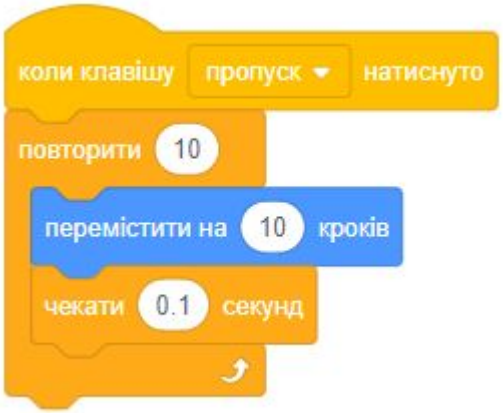
КЕРУВАННЯ: ПОВТОРИТИ ДО

Повторює виконання блоків всередині, поки умова не буде дорівнювати ІСТИНА (true).
Перевіряє стан умови. Коли умова не витримується, виконуються блоки всередині, і знову стан перевіряється. Якщо умова витримується, комп'ютер продовжує виконувати наступні блоки.

КЕРУВАННЯ: ПОВТОРИТИ

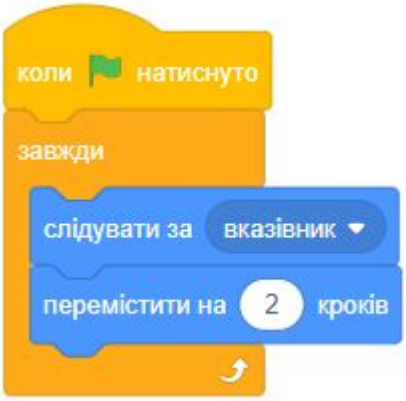


Запускає блоки, які знаходяться всередині, певну кількість разів.



КЕРУВАННЯ: ЗАВЖДИ

Блок ЗАВЖДИ використовують у програмуванні для повторення команд і називають циклом. Він повторює блоки, що вставлені в нього, один за одним, поки програма не буде зупинена. Приклад:

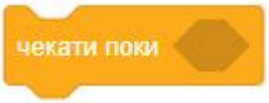


ІНШІ БЛОКИ

КЕРУВАННЯ: ЧЕКАТИ...СЕКУНД

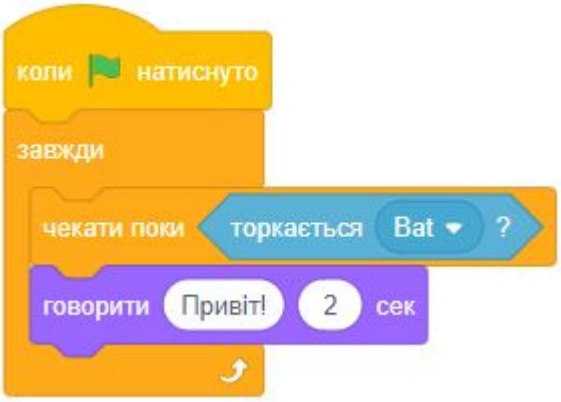
Чекає певну кількість секунд, а потім виконує наступний блок. Якщо вам потрібно чекати менше ніж 1 секунду, вказуйте час очікування, використовуючи крапку. Наприклад: 0.3 або 0.05.

КЕРУВАННЯ: ЧЕКАТИ ПОКИ

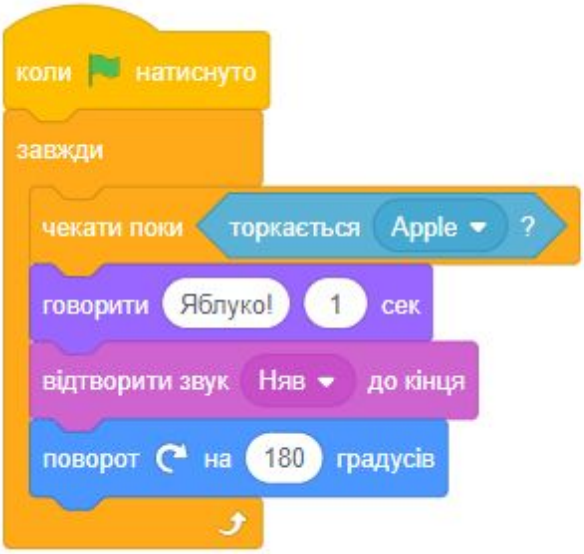


Чекає, поки умова не буде дорівнювати ІСТИНА (true), потім виконує наступні блоки у скрипті.

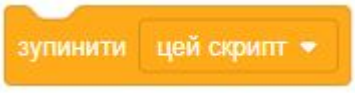
У цьому прикладі спрайт завжди чекає дотику до іншого спрайта та говорить привітання, коли це трапляється.



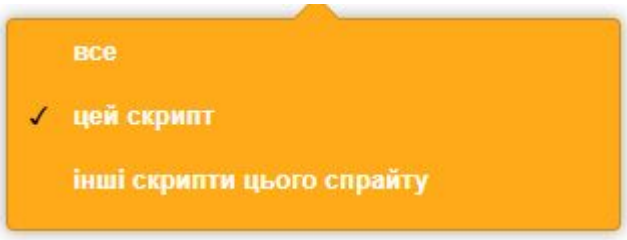
У цьому прикладі кіт пересувається випадковим чином по сцені та відтворює звук, коли торкається яблука.



КЕРУВАННЯ: ЗУПИНИТИ СКРИПТИ



Зупиняє скрипти для спрайтів. Клацніть на трикутник, щоб вибрати, які саме скрипти зупиняти.



Якщо вибрано «все», цей блок зупиняє всі скрипти гри, як і червона кнопка СТОП у верхній частині екрана.

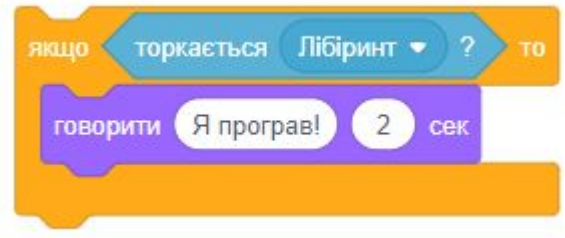
ДОВІДНИК: ДАТЧИКИ

ДАТЧИКИ СПРАЙТІВ

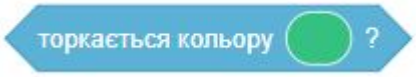
ДАТЧИКИ: ТОРКАЄТЬСЯ?

Результатом є ІСТИНА (true), якщо цей спрайт торкається іншого спрайта, країв екрана або вказівника мишки.

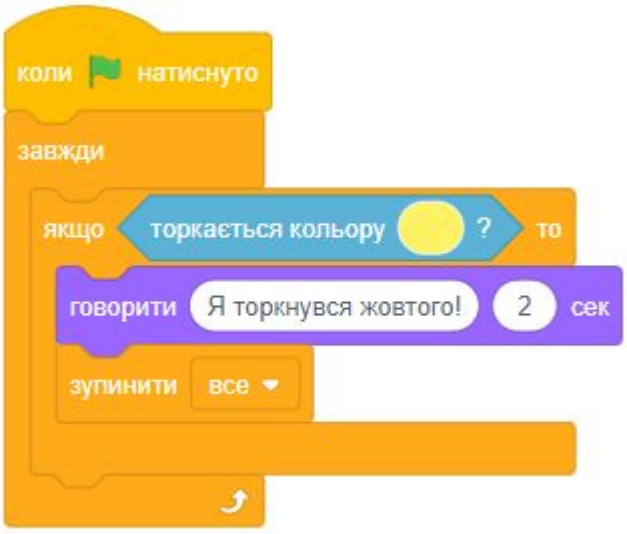
Ви можете натиснути на трикутник і вибрати, яким саме чином будете використовувати цей блок. Використовуйте цей блок, коли ви хочете відстежувати торкання спрайтів у блоці ЯКЩО або інших. Приклад:



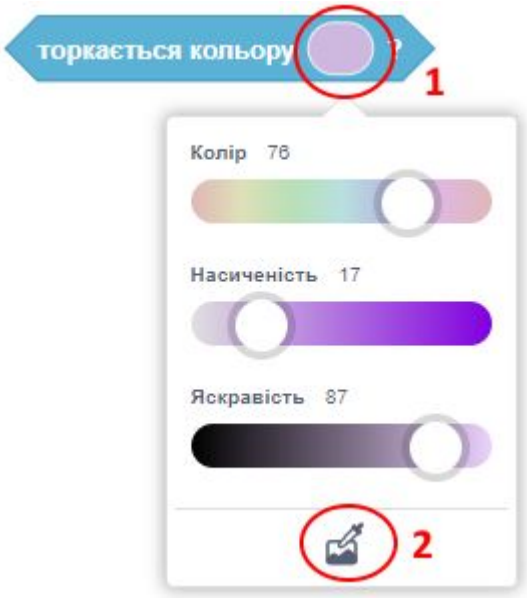
ДАТЧИКИ: ТОРКАЄТЬСЯ КОЛЬОРУ?



Результатом є ІСТИНА (true), якщо спрайт торкається певного кольору. В цьому прикладі гра зупиняється, коли спрайт торкається жовтого кольору.



Можна обрати колір за допомогою додаткового вікна, яке з’являється, коли клацнути мишкою на параметрі.

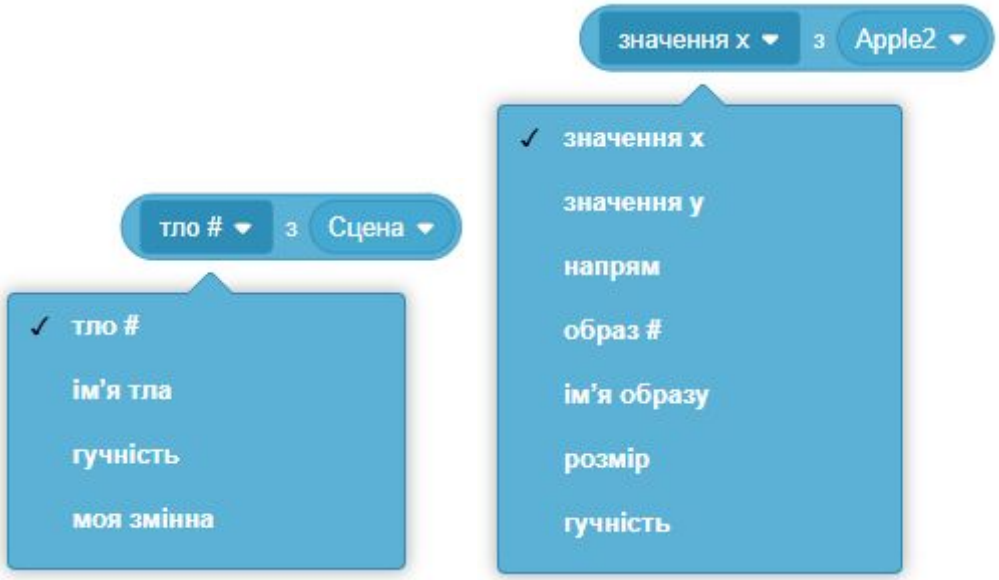


ДАТЧИКИ: ЗНАЧЕННЯ ЗМІННИХ СПРАЙТА АБО СЦЕНИ

Повідомляє змінні інших спрайтів і сцени. Натисніть перший трикутник, щоб вибрати необхідну змінну в меню, що розкривається.

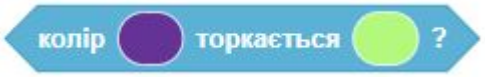
Сцена та спрайт мають різні набори змінних.

Натисніть другий трикутник, щоб обрати спрайт або сцену зі списку.



Використовуйте цей блок всередині інших блоків, якщо вони мають параметри із закругленими краями.

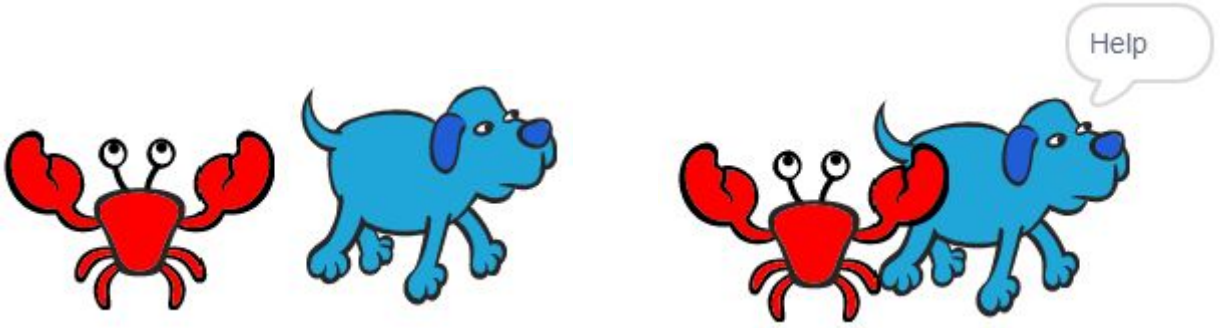
ДАТЧИКИ: КОЛІР ТОРКАЄТЬСЯ КОЛЬОРУ?



Результатом є ІСТИНА (true), коли колір якоїсь точки активного образа спрайта (перший параметр) торкається кольору якоїсь точки на тлі сцени або на іншому спрайті (другий параметр).

Можливо обрати колір за допомогою додаткового вікна, яке з’являється, коли клацнути мишкою на параметрах блока.

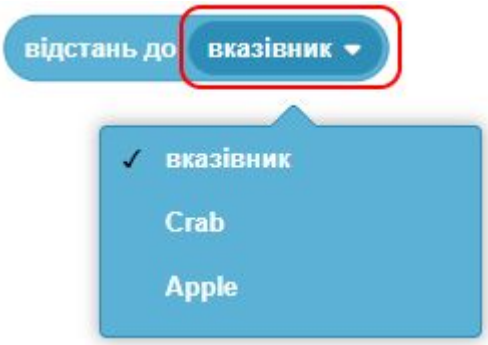
У цьому прикладі песик кличе на допомогу, коли його шкіри (блакитний колір, перший параметр) торкається краб (червоний колір, другий параметр).



ДАТЧИКИ: ВІДСТАНЬ ДО

Результатом є відстань від цього спрайта до іншого спрайта або до вказівника мишки.

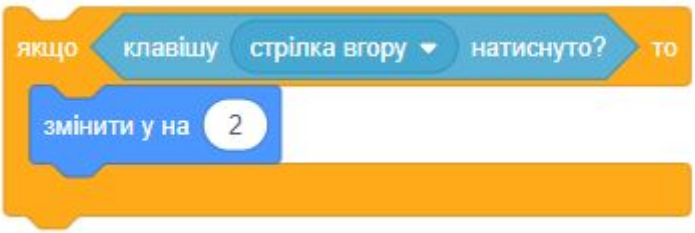
Натисніть на трикутник, щоб вибрати, до якого об'єкта обчислювати відстань. У випадаючому списку з'являться вказівник мишки та всі спрайти проекту, крім поточного:



ДАТЧИКИ МИШКИ ТА КЛАВІАТУРИ

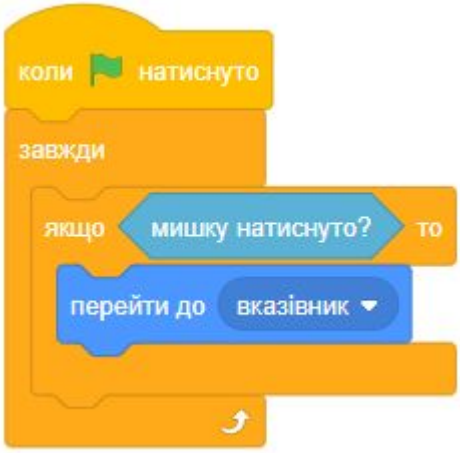
ДАТЧИКИ: КЛАВІШУ НАТИСНУТО?

Результатом є ІСТИНА (true), якщо натиснута певна клавіша клавіатури. Натисніть трикутник, щоб вибрати зі списку, яку саме клавішу відстежувати. Використовуйте цей блок, коли ви хочете відстежувати натискання клавіш в блоці ЯКЩО або інших. Приклад:



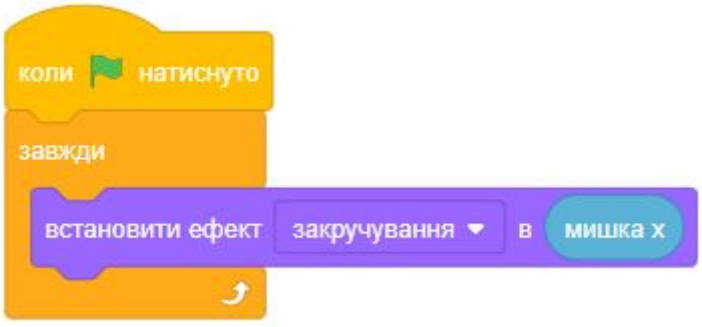
ДАТЧИКИ: МИШКУ НАТИСНУТО?

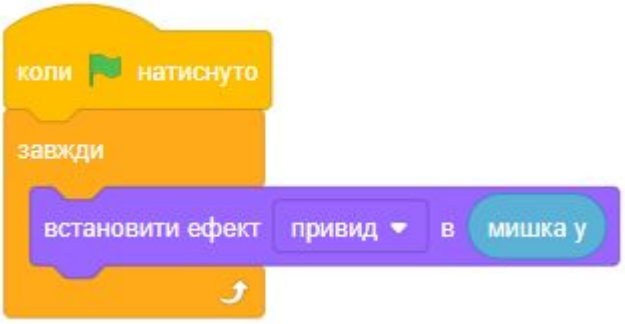
Результатом є ІСТИНА (true), якщо ліва кнопка мишки натиснута в будь-якому місці сцени. Приклад:



ДАТЧИКИ: МИШКА X ТА МИШКА У

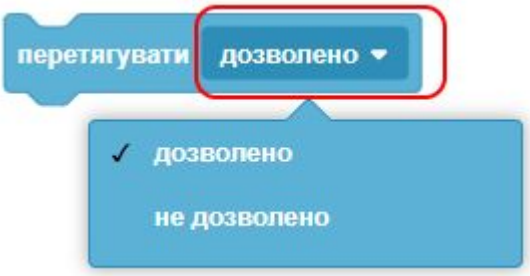
Повідомляють координати **X** (позицію по горизонталі) та **Y** (позицію по вертікалі) покажчика мишки.





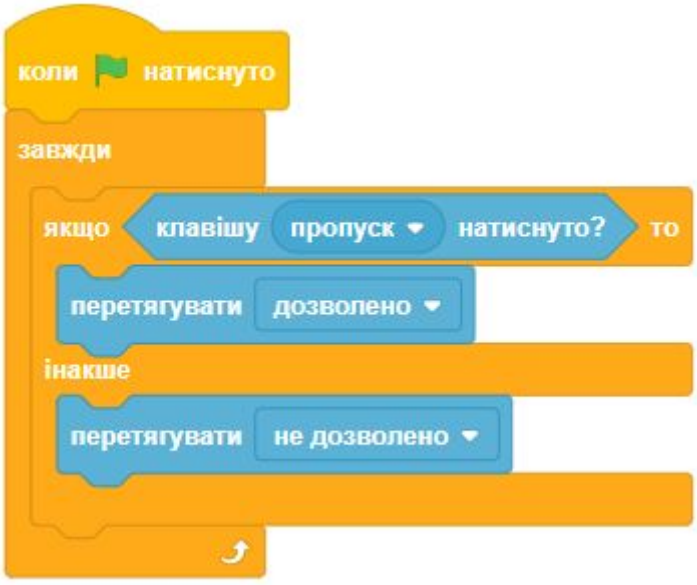
ДАТЧИКИ: ПЕРЕТЯГУВАТИ ДОЗВОЛЕНО/НІ

Встановлює, чи можна перетягувати спрайт по сцені за допомогою мишки. Цей блок працює тільки в повноекранному режимі.



У режимі редагування перетягування спрайтів завжди дозволено. Щоб перейти до повноекранного режиму, натисніть кнопку

У цьому прикладі перетягування спрайта дозволено, тільки коли клавіша ПРОПУСК натиснута:



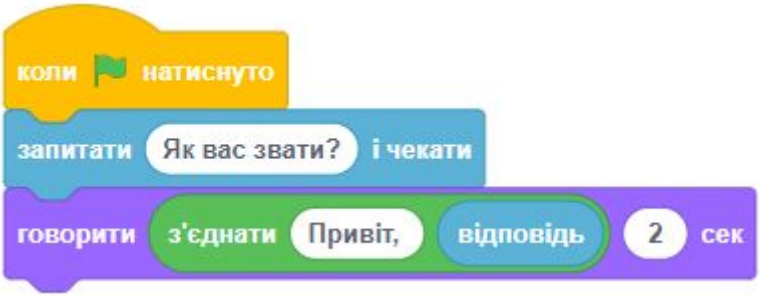
ДАТЧИКИ ЗАПИТАННЯ КОРИСТУВАЧА

ДАТЧИКИ: ЗАПИТАТИ І ЧЕКАТИ

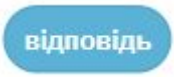
Задає питання на екрані та дозволяє ввести з клавіатури відповідь, яка потрапляє в змінну:

Питання з'являється в хмарі на екрані поряд зі спрайтом. Програма очікує, коли користувач набирає відповідь, поки не натиснута клавіша «Enter» або кнопка галочки.

Приклад:



ДАТЧИКИ: ВІДПОВІДЬ



Результат введення з клавіатури при останньому використанні блока:

Змінна **відповідь** доступна для всіх спрайтів. Якщо вам потрібно використовувати результат запитання, збережіть її значення в змінній:

Для перегляду значення відповіді на сцені натисніть галочку поруч з блоком відповіді. Приклад:

ДАТЧИКИ ЧАСУ

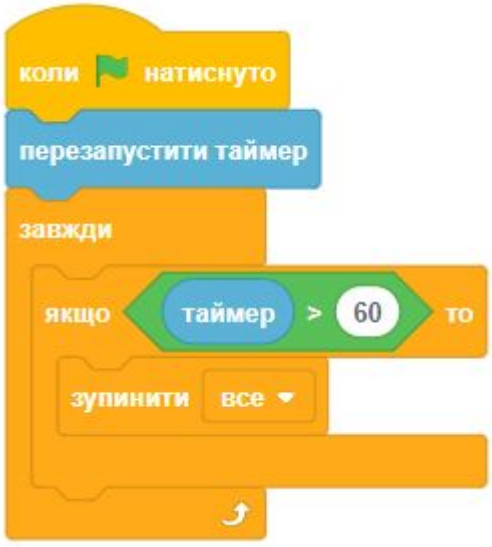
ДАТЧИКИ: ТАЙМЕР

Результатом є значення таймера в секундах.

Щоб побачити значення таймера на сцені, натисніть галочку поруч з блоком. Таймер завжди працює. Щоб встановити таймер в нуль, використовуйте ПЕРЕЗАПУСТИТИ ТАЙМЕР.

Якщо вам потрібно точно визначити час, користуйтеся операторами «>» та «<», а не «=». Час вимірюється дуже швидко, і буває складно точно визначити момент, коли таймер відрахував, наприклад, рівно 30 секунд. Проте легко визначити, коли минуло більше 30 секунд.

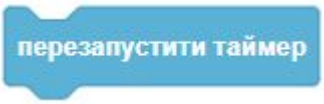
У цьому прикладі гра триває одну хвилину, а потім зупиняється:



ДАТЧИКИ: ДНІВ ПІСЛЯ 2000

Повідомляє кількість днів від 1 січня 2000 року.

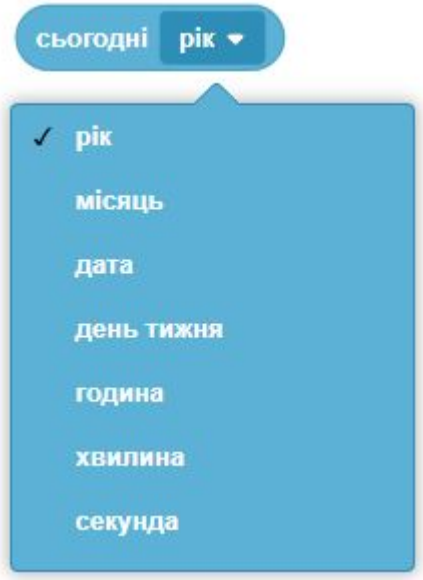
ДАТЧИКИ: ПЕРЕЗАПУСТИТИ ТАЙМЕР



Встановити таймер в нуль. Таймер завжди працює. Після виконання цієї команди він почне свій відлік заново. Приклад:

ДАТЧИКИ: СЬОГОДНІ

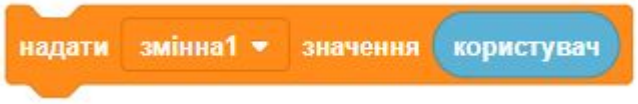
Повідомляє елемент поточного часу. Ви можете відкрити меню, щоб вказати, який елемент вам потрібен.



ІНШІ ДАТЧИКИ

ДАТЧИКИ: КОРИСТУВАЧ

Повідомляє, який користувач відкрив проект. Якщо ви хочете зберегти поточне ім'я користувача, використовуйте змінну:



ДАТЧИКИ: ГУЧНІСТЬ

Результатом є гучність (від 1 до 100) встановленого в комп'ютері мікрофона. Щоб побачити рівень гучності на сцені, поставте галочку біля блока гучності.

Ваш комп'ютер повинен мати мікрофон, щоб цей блок працював. Також потрібно дозволити браузеру користуватися мікрофоном, коли він запитає.

ДОВІДНИК: ОПЕРАТОРИ

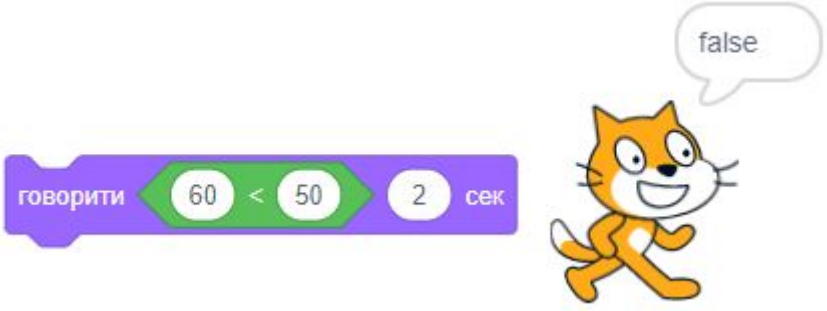
БЛОКИ ПОРІВНЯННЯ

- Блоки порівняння ви можете знайти в категорії ОПЕРАТОРИ, їх називають операторами порівняння, тому що вони порівнюють два числа між собою. За допомогою блоків порівняння можливо дізнатися: число більше, менше чи дорівнює іншому числу.
- Ці блоки можна вставляти в блоки ЯКЩО...ТО, щоб порівнювати числа та виконувати дії в залежності від результатів порівняння.

ОПЕРАТОРИ: МЕНШЕ

- Результатом є ІСТИНА (true), якщо перше значення є меншим ніж друге. В іншому випадку результатом є БРЕХНЯ (false).
- Приклад: Число 40 менше ніж 50, результатом є ІСТИНА (true).

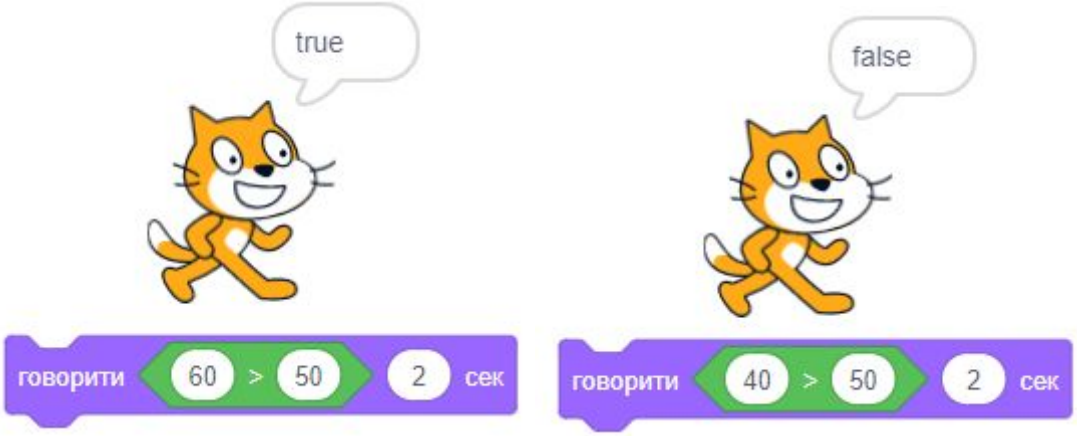
- Приклад: Число 60 менше ніж 50, результатом є БРЕХНЯ (false).



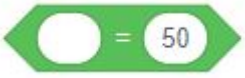
- Приклад: Змінній **рахунок** встановлюється значення 50.
  Тепер вона не є меншою ніж 50, результатом є БРЕХНЯ (false).

ОПЕРАТОРИ: БІЛЬШЕ

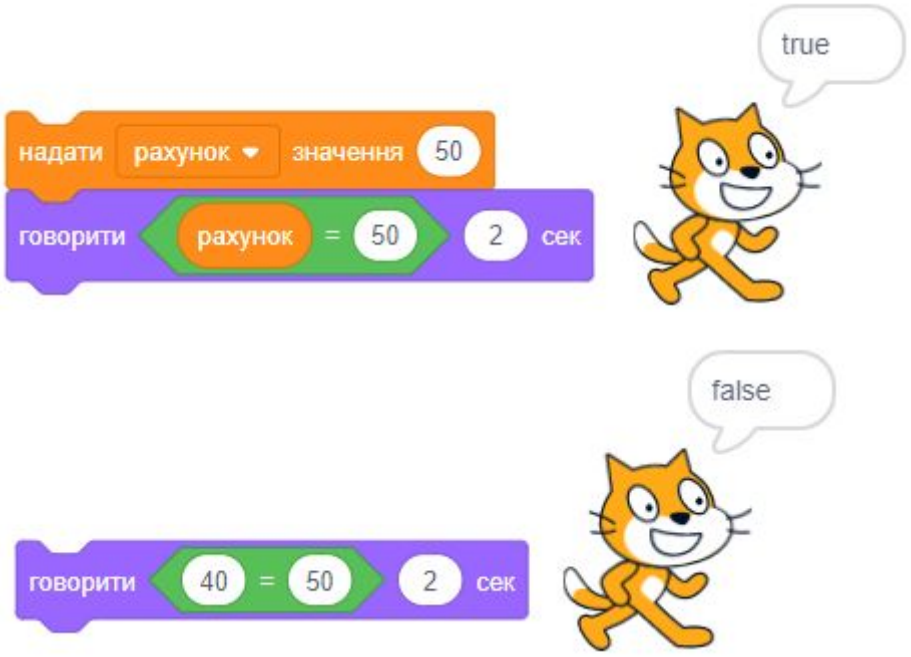
Результатом є ІСТИНА (true), якщо перша величина більша ніж друга.
Результатом є БРЕХНЯ (false) в іншому випадку.



ОПЕРАТОРИ: ДОРІВНЮЄ



Результатом є ІСТИНА (true), якщо дві величини рівні.
Результатом є БРЕХНЯ (false), якщо дві величини не рівні.



ЛОГІЧНІ ОПЕРАТОРИ

ОПЕРАТОРИ: І

Результатом є ІСТИНА (true), якщо обидва вирази є ІСТИНА.
Результатом є БРЕХНЯ (false), якщо хоча б один з виразів є БРЕХНЯ.

X	Y	
істина	істина	істина
істина	брехня	брехня
брехня	істина	брехня
брехня	брехня	брехня

Приклад: «Якщо батьки придбають мені кішку **І** собаку, я буду щасливий». Оператор «І» робить істинним твердження «буду щасливий» тільки за умови, якщо батьки придбають кішку і собаку (обох тварин одночасно).

ОПЕРАТОРИ: АБО

Результатом є ІСТИНА (true), якщо хоча б один з виразів є ІСТИНА.
Результатом є БРЕХНЯ (false), якщо обидва вирази є БРЕХНЯ.

X	Y	X або Y
істина	істина	істина
істина	брехня	істина
брехня	істина	істина
брехня	брехня	брехня

Приклад: «Якщо батьки придбають мені кішку **АБО** собаку, то я буду щасливий». Оператор АБО робить істинним твердження «буду щасливий» тільки за умови, якщо батьки придбають кішку АБО собаку (лише одну тварину).

ОПЕРАТОРИ: НЕ

Результатом є ІСТИНА (true), якщо вираз є БРЕХНЯ.
Результатом є БРЕХНЯ (false), якщо вираз є ІСТИНА.

X	не X
істина	брехня
брехня	істина

Приклад: «Ми **НЕ** купимо собаку». Оператор «НЕ» свідчить про заперечення: фраза є істиною, тільки якщо ми НЕ купимо собаку.

АРИФМЕТИЧНІ ОПЕРАТОРИ

- Комп’ютер виконує різні обчислення дуже швидко. Ви можете додати до своїх проектів обчислення, у яких будуть використовуватися чотири арифметичні дії. Такі блоки з категорії ОПЕРАТОРИ називають **АРИФМЕТИЧНІ ОПЕРАТОРИ**.
- За допомогою математичних операторів складають (+), віднімають (-), перемножують (*) і ділять (/) числа. Використовуючи їх, ви миттєво отримаєте відповідь.
- Ви можете вставляти математичні оператори до інших блоків, щоб виконувати складні обчислення.

ОПЕРАТОРИ: ДОДАВАННЯ Складає два числа.	
ОПЕРАТОРИ: ВІДНІМАННЯ Віднімає друге число з першого.	

ОПЕРАТОРИ: МНОЖЕННЯ

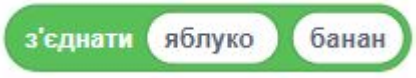
Перемножує два числа.

ОПЕРАТОРИ: ДІЛЕННЯ

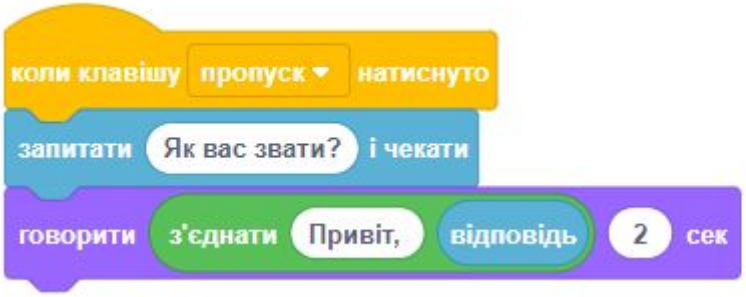
Ділить перше число на друге.

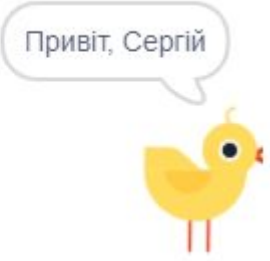
БЛОКИ ДЛЯ РЯДКІВ

ОПЕРАТОРИ: З'ЄДНАТИ



Сполучає рядки і повертає результат. Приклад:



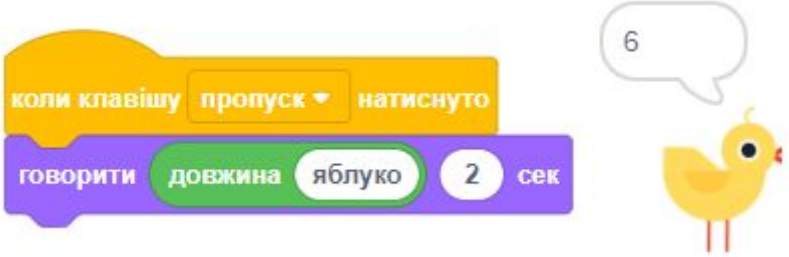


ОПЕРАТОРИ: СИМВОЛ РЯДКА

Повідомляє символ з вказаної позиції в рядку. В цьому прикладі спрайт говорить слово «яблуко» по буквах.

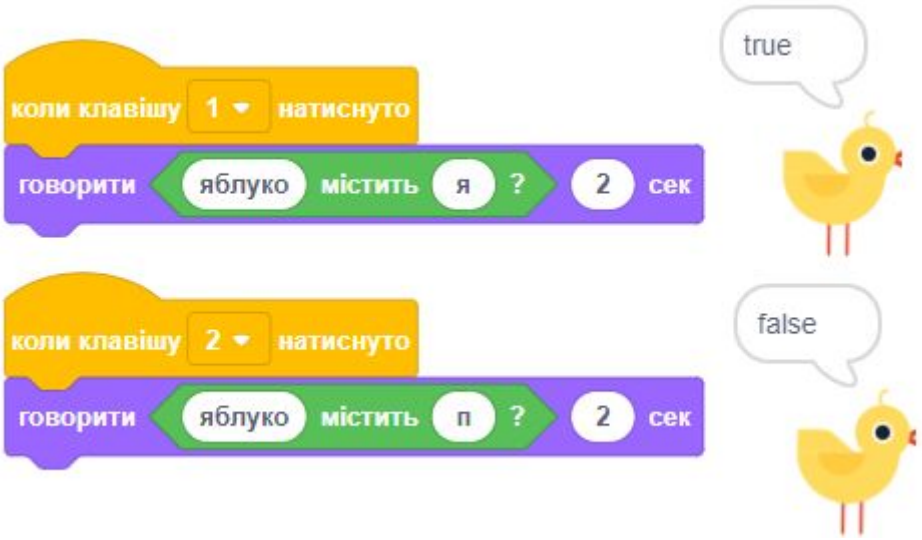
ОПЕРАТОРИ: ДОВЖИНА РЯДКА

Результатом є довжина (кількість символів) у рядку. Приклад:



ОПЕРАТОРИ: РЯДОК МІСТИТЬ

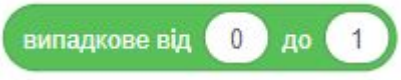
Результатом є ІСТИНА (true), якщо рядок (перший параметр) містить підрядок (другий параметр). Результатом є БРЕХНЯ (false) в іншому випадку. Приклад:



ОБЧИСЛЕННЯ ФУНКЦІЙ

ОПЕРАТОРИ: ВИПАДКОВЕ ВІД...ДО

- Вибирає випадкове число в межах визначеного діапазону чисел.
- Назвіть будь-яке випадкове число, що спало вам на думку. Це дуже легко, чи не так? Однак комп'ютер не може це зробити з такою ж легкістю, як ви.
- Блок ВИПАДКОВЕ ВІД дозволяє комп'ютеру обрати число самотужки, але він не вміє приймати такі рішення. Як же тоді працює цей блок? Незважаючи на те, що блок має таку назву, обране число не є випадковим.
- Результат виконання цього блока отримують за допомогою складних обчислень, що лише імітують випадковість.

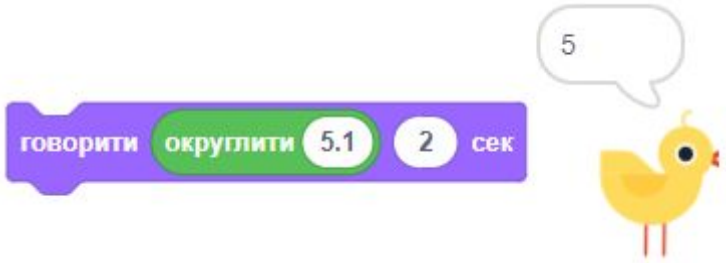
ПРИКЛАД	МОЖЛИВИЙ РЕЗУЛЬТАТ
	{0, 1, 2, 3, ... , 9, 10}
	{0, 1}

випадкове від 0 до 1.0	{0, 0.1, 0.17, 0.357, ... , 1.0}
випадкове від -2 до 2	{−2, −1, 0, 1, 2}
випадкове від 1 до 100 / 100	{0, 0.01, 0.13, 0.43, 0.5, ... , 1.0}
10 * випадкове від 0 до 10	{0, 10, 20, 30, ... , 90, 100}

ОПЕРАТОРИ: ОКРУГЛИТИ

округлити

Результатом є значення найближчого цілого числа.
Параметр команди округляється до найближчого цілого числа.



ОПЕРАТОРИ: ОСТАЧА

остача на

Результатом є залишок від ділення першого числа на друге число. Наприклад, (143 остача 6) становить 5 (143 розділити на 6 становить 23, і залишок дорівнює 5).

ОПЕРАТОРИ: ABS

«abs» - це абревіатура від «absolute value» (абсолютне значення). Абсолютним значенням числа є його відстань від 0. Іншим способом опису абсолютного значення є те, що воно робить число позитивним - якщо воно від'ємне, воно стає позитивним, а якщо воно є позитивним, то залишається позитивним. Іншим способом записати абсолютне значення є |число|. abs (-3) - це теж саме, що й |-3|.

Наприклад: abs(-3) = +3. abs(+4) = +4.

ОПЕРАТОРИ: МЕНШЕ ЦІЛЕ

Округлює число до найбільшого цілого числа, яке менше або дорівнює числу. Наприклад, floor (1,73) = 1 і floor (-2,74) = -3.

ОПЕРАТОРИ: БІЛЬШЕ ЦІЛЕ

Округлює число до найменшого цілого числа, що перевищує чи дорівнює числу. Наприклад, ceiling (3,14) = 4, а ceiling (7,68) = 8.

ОПЕРАТОРИ: SQRT

«sqrt» - це абревіатура від «square root» (квадратний корінь).

Квадратним коренем із невід'ємного числа **a** називається таке невід'ємне число **b**, квадрат якого дорівнює **a**.

Позначають так	$\sqrt{a} = b$
Цей запис означає	$b^2 = a,\ b \geq 0$
Наприклад	$\sqrt{16} = 4$, оскільки $4^2 = 16$, 4>0

Отже, квадратний корінь з 4 дорівнює 2. Аналогічно, квадратний корінь з 2 дорівнює приблизно 1,414213562373095, оскільки число (1,414213562373095 × 1,414213562373095) близьке до 2.

Скретч не підтримує уявних чисел, які є квадратними коренями від'ємних чисел. Спроба знайти квадратний корінь з від'ємного числа створить результат рівний NaN (не число).

ОПЕРАТОРИ: SIN, COS, TAN

- Функції **sin, cos** і **tan** називаються «тригонометричними функціями». Результатом цих функцій є співвідношення між сторонами прямокутного трикутника.
- Значення, введене в блок, - це кут (у градусах), який є одним із кутів у трикутнику.
- «**cos**» - це абревіатура від «cosine» (косинус). Косинусом гострого кута A прямокутного трикутника називається **відношення прилеглого катета (b) до гіпотенузи (c).**
- «**sin**» - це абревіатура від «sine» (синус). Синусом гострого кута A називається **відношення протилежного катета (a) до гіпотенузи (c)**.

- «**tan**» - це абревіатура від «tangent» (тангенс). Тангенсом гострого кута A називається **відношення протилежного катета (a) до прилеглого катета (b)**. У нас в математиці прийнято писати **tg** замість **tan**.

$$\cos A = \frac{b}{c}$$

$$\sin A = \frac{a}{c}$$

$$\tan A = \frac{a}{b}$$

ОПЕРАТОРИ: ASIN, ACOS, ATAN

- Функції asin, acos і atan - це «зворотні тригонометричні функції».
- Тоді як sin, tan і cos знаходять співвідношення з кутів, asin, acos і atan знаходять кути у градусах з відношень.
- Параметри asin та acos змінюються від -1 до 1.
- Функція atan приймає будь-яке число (включаючи Infinity).
- Діапазон результатів коливається від -90 до 90 градусів.
- «**asin**» - це абревіатура для «arcsine». Отримавши відношення (у десятковому вигляді) довжини протилежної сторони (a) та гіпотенузи (c) прямокутного трикутника, вона визначає кут.
- «**acos**» - це абревіатура для «arccosine». Отримавши відношення (у десятковому вигляді) довжини сусідньої сторони (b) та гіпотенузи (c) прямокутного трикутника, вона знаходить кут.
- «**atan**» - це абревіатура «arctangent» (арктангенс). Отримавши відношення (у десятковому вигляді) довжини протилежної сторони (a) та сусідньої сторони (b) прямокутного трикутника, вона знаходить кут.

ОПЕРАТОРИ: E^, 10^

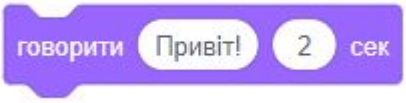
- «e^», «10^» - «експоненціальні функції» або «функції ступеня».
- «e» - це абревіатура для «числа Ейлера», що становить близько 2,718. За допомогою функції «e^» число e множиться на себе стільки разів, скільки задано в параметрі. Наприклад, якщо параметр дорівнює 3 (три), відповідь буде e^3 (це e × e × e), що становить приблизно 20,086.
- Функція «10^» перемножує число 10 стільки разів, скільки задано в параметрі. Наприклад, якщо параметр дорівнює 6 (шість), відповідь буде 10^6 (це 10×10×10×10×10×10), що дорівнює 1000000.

ОПЕРАТОРИ: LN, LOG

- ln і log - це «логарифмічні функції». Вони роблять прямо протилежне тому, що роблять експоненційні функції.
- **ln** - «натуральний логарифм». Він визначає, скільки разів число «e» потрібно було б помножити на себе, щоб отримати значення множника. Наприклад, якщо значення дорівнювало 148,4, відповідь була б приблизно 5, оскільки e^5 (e × e × e × e × e) приблизно 148,4.
- **log** - це скорочення від "логарифм". Функція визначає, скільки разів 10 потрібно помножити на себе, щоб отримати значення множника. Наприклад, якщо значення дорівнює 100, відповідь буде 2, оскільки 10 × 10 дорівнює 100.

ДОВІДНИК: ВИГЛЯД

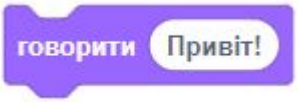
ВИГЛЯД: ГОВОРИТИ..СЕК



Показати повідомлення в хмарі над спрайтом протягом певного числа секунд. Ви можете ввести довільне повідомлення.

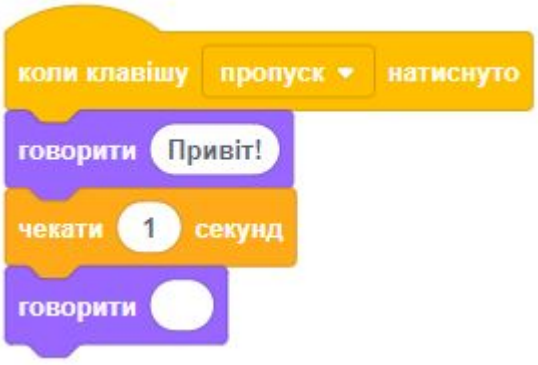
Число секунд вказує, протягом якого часу буде видна хмарка з повідомленням. Виконання скрипта на цей час зупиняється.

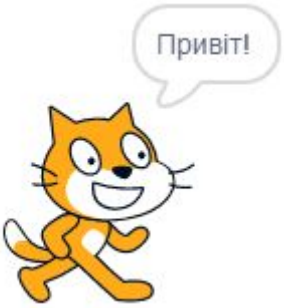
ВИГЛЯД: ГОВОРИТИ



Показати повідомлення в хмарі над спрайтом. Наберіть будь-який текст в параметрі блока.

Щоб сховати показаний текст, використовуйте цей блок з порожнім параметром. Приклад:

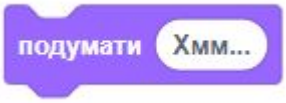




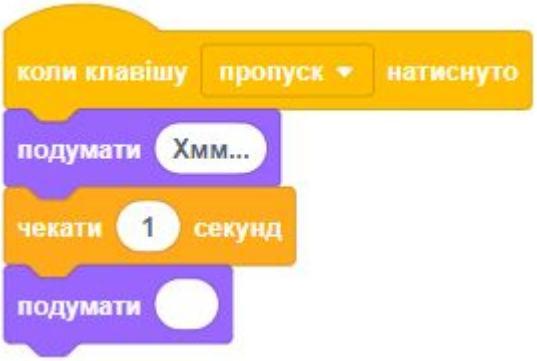
ВИГЛЯД: ПОДУМАТИ СЕК

Показати думку в хмарі протягом певного числа секунд.
Число секунд вказує, протягом якого часу буде видно хмару з думкою. Подальше виконання скрипта на цей час припиняється.

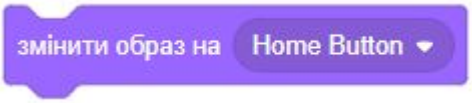
ВИГЛЯД: ПОДУМАТИ



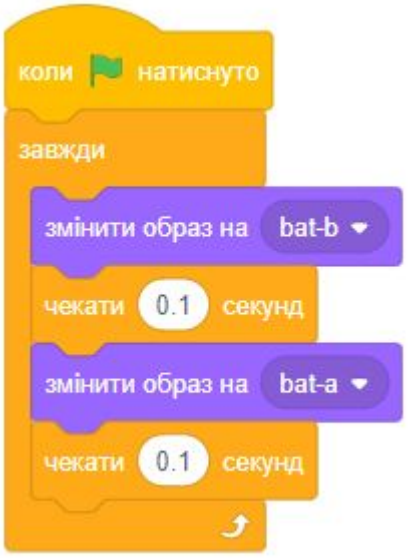
Показує повідомлення в хмарі думки. Наберіть довільний текст у параметрі блока, заданий текст з'явиться в хмарі. Щоб видалити хмару з текстом, використовуйте цей блок з порожнім параметром.



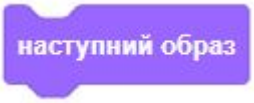
ВИГЛЯД: ЗМІНИТИ ОБРАЗ НА



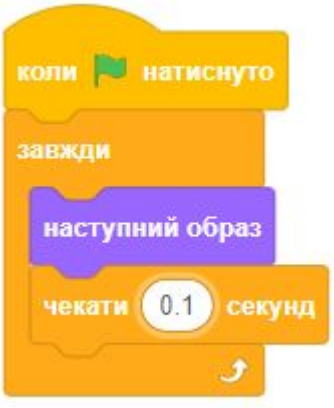
Перемикає образ для зміни зовнішнього вигляду спрайта.
Виберіть ім'я образа у меню, натиснувши на трикутник. Щоб побачити або редагувати образи, клацніть на вкладці ОБРАЗИ. Приклад:



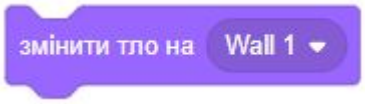
ВИГЛЯД: НАСТУПНИЙ ОБРАЗ



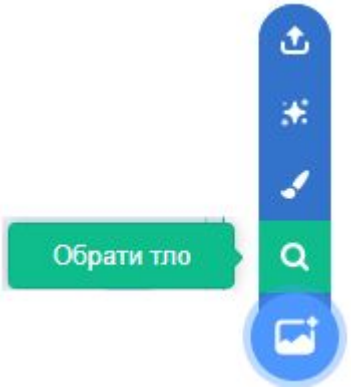
Активує наступний образ із списку образів спрайта.
Щоб побачити або редагувати образи спрайта, клацніть на вкладці «Образи». Ви можете змінити послідовність цих образів перетягуванням. Якщо поточний образ є останнім, цей блок вибирає перший (верхній) образ. Приклад:



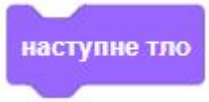
ВИГЛЯД: ЗМІНИТИ ТЛО НА



Перемикає на обране тло сцени. Тло - це картинка, яка буде показуватися на сцені позаду всіх спрайтів.
Натисніть на трикутник, щоб вибрати бажане тло сцени.
Перед використанням цього блока, необхідно додати потрібні фонові картинки на сцену. Це можна зробити, натиснувши на спеціальну кнопку в правому нижньому кутку вікна програми.

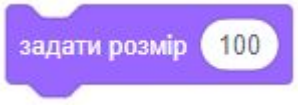


ВИГЛЯД: НАСТУПНЕ ТЛО



Перемикання на наступне тло сцени. Якщо поточне тло є останнім, цей блок вибирає перший (верхній) фон.

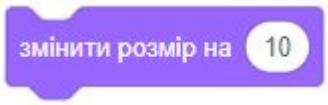
ВИГЛЯД: ЗАДАТИ РОЗМІР



Встановлює розмір спрайта в процентах від оригінального розміру образа спрайта. Є межі того, наскільки можна зменшити або збільшити спрайт. Експериментуйте з різними числами.

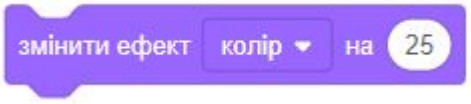
Значення 100 - встановити оригінальний розмір спрайта.
Значення 50 - зменшити спрайт у два рази.
Значення 200 - збільшити спрайт у два рази.

ВИГЛЯД: ЗМІНИТИ РОЗМІР

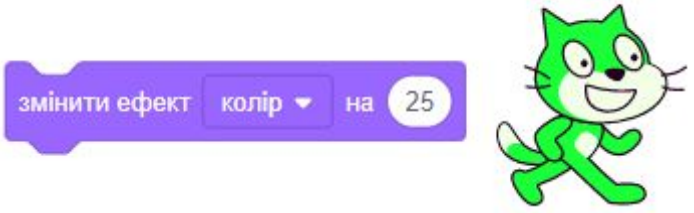
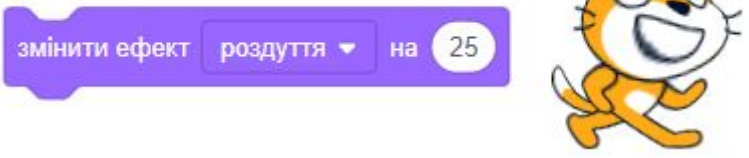
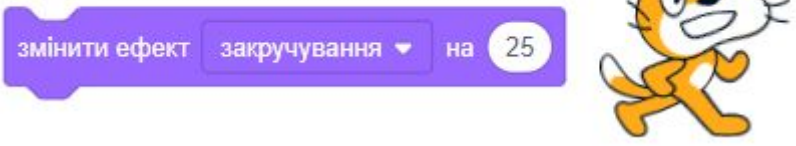


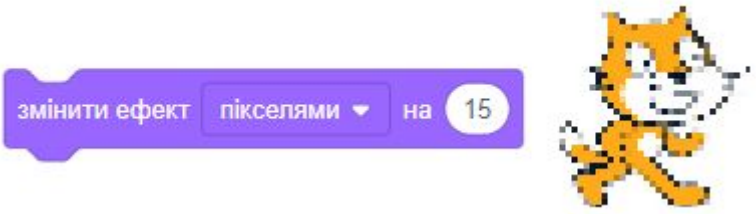
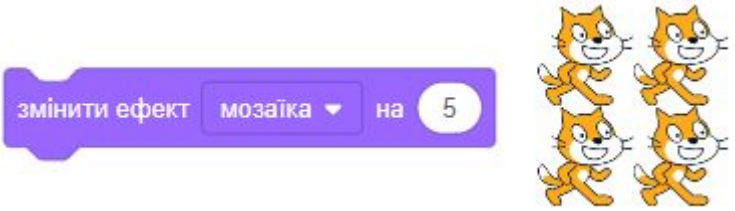
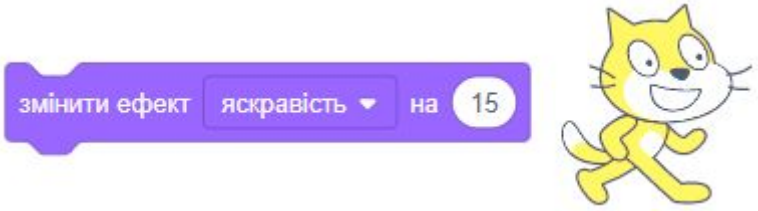
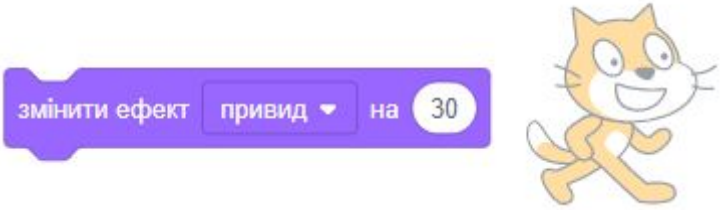
Змінити розмір спрайта на зазначену величину. Негативні величини зменшують спрайт, позитивні - збільшують. Величини обчислюються у процентах від орігінального розміру образа спрайта.

ВИГЛЯД: ЗМІНИТИ ЕФЕКТ НА

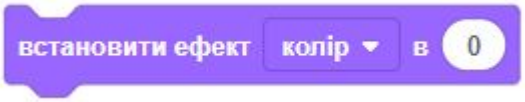


Змінити графічний ефект для спрайта на задане число. Клацніть на трикутнику, щоб вибрати потрібний ефект з меню. Задавайте числа в межах від -100 до 100. А для деяких ефектів - від 0 до 100.

ЕФЕКТ	ПРИКЛАД
колір	
роздуття	
закручування	

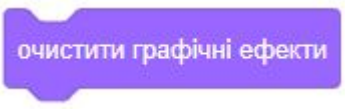
пікселями	
мозаїка	
яскравість	
привид	

ВИГЛЯД: ВСТАНОВИТИ ЕФЕКТ



Встановити графічний ефект для спрайта за допомогою обраного числа. Клацніть на трикутнику, щоб вибрати ефект з меню. Список ефектів з поясненнями дивіться в блоці ЗМІНИТИ ЕФЕКТ.

ВИГЛЯД: ОЧИСТИТИ ГРАФІЧНІ ЕФЕКТИ

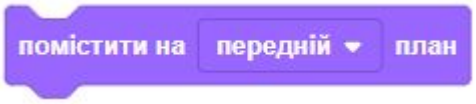


Очистити всі графічні ефекти для цього спрайта, встановлені для нього за допомогою блоків ЗМІНИТИ ЕФЕКТ та інших. Додайте цей блок на початку гри, якщо потім змінюєте графічні ефекти.

ВИГЛЯД: СХОВАТИ АБО ПОКАЗАТИ

Перший блок змушує спрайт сховатися зі сцени. Другий блок змушує спрайт з'явитися на сцені.

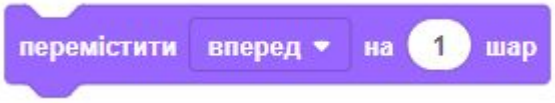
ВИГЛЯД: ПОМІСТИТИ НА...ПЛАН



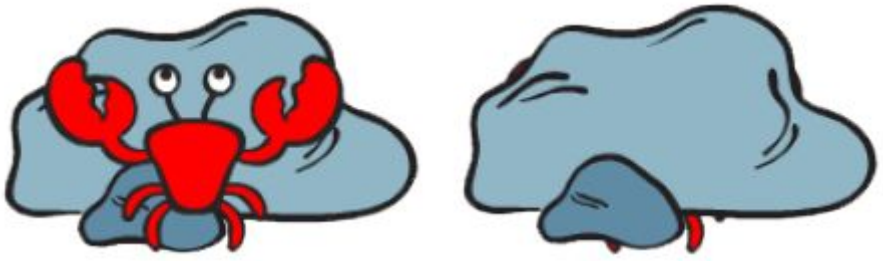
Поміщає спрайт перед усіма іншими спрайтами (якщо вибрано «передній» план). Поміщає спрайт за всіма іншими спрайтами (якщо вибрано «задній» план). Приклад:

На першому малюнку м'ячик знаходиться на передньому плані.
На другому малюнку м'ячик розміщується на задньому плані.

ВИГЛЯД: ПЕРЕМІСТИТИ ВПЕРЕД/НАЗАД НА..ШАР



Переміщає спрайт вперед або назад на певну кількість шарів.
Клацніть на трикутнику, щоб вибрати, як переміщати спрайт.



ВИГЛЯД: ОБРАЗ НОМЕР/НАЗВА

Результатом є номер або ім'я поточного образа спрайта.
Кожен образ спрайта має своє ім'я і номер.

ВИГЛЯД: ТЛО НОМЕР/НАЗВА

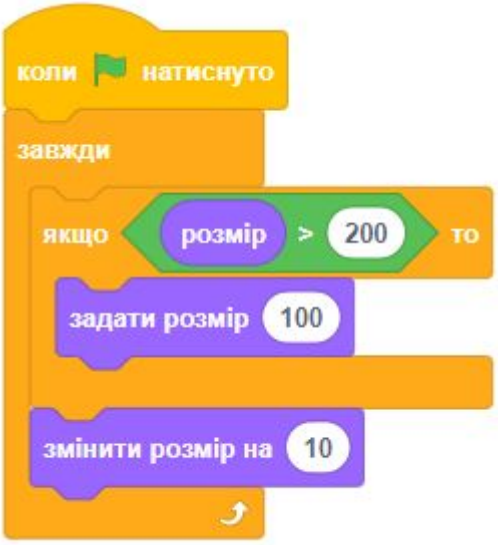
Повідомляє поточне ім'я або номер тла сцени.
Кожне тло сцени має власний номер, а також ім'я.

ВИГЛЯД: РОЗМІР

Повідомляє розмір спрайта як % від його початкового розміру. Для перегляду значення розміру на сцені встановіть галочку поруч з блоком в палітрі блоків:

Розмір буде показаний у верхньому лівому кутку сцени:

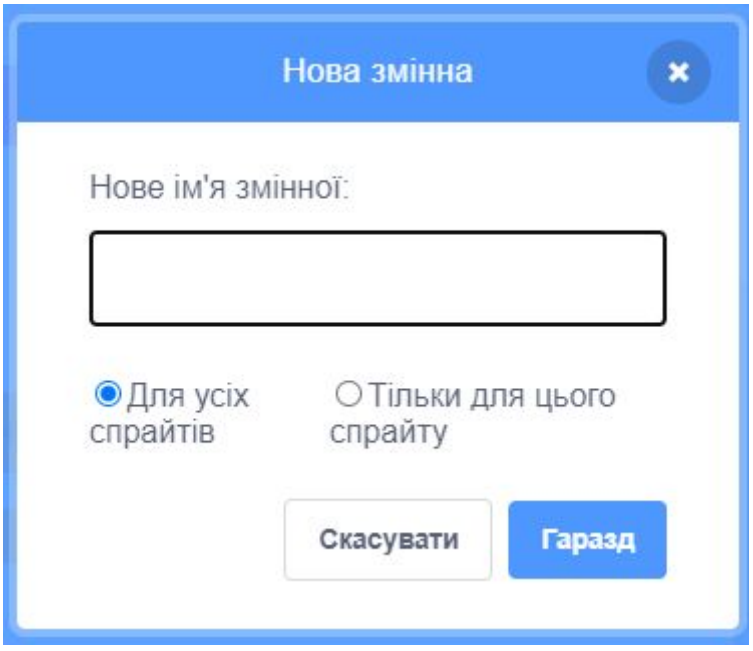
Приклад:



ДОВІДНИК: ЗМІННІ

- **ЗМІННА** - це область пам'яті комп'ютера, яка має назву і зберігає в собі будь-які дані.
- В комп'ютерних іграх багато інформації зберігають у змінних. Це може бути здоров'я персонажа, швидкість пересування, набутий досвід, кількість життів, місце розташування на екрані та інше.
- Якщо вас запитають, скільки вам років, ви відразу дасте відповідь. Це дуже легко тому, що ви запам’ятали частину інформації (змінну) під назвою “**мій вік**”. У цій змінній у вигляді числа містяться роки, що минули від дня вашого народження.
- Змінна, як виходить з її назви, може змінюватися. Наприклад, щороку ваш вік збільшується, але завжди залишається вашим віком. Комп'ютер знає команди для надання нових значень змінних, зміни та отримання поточних значень змінних.
- Змінна буває проста (зберігає одне значення) і більш складна - список (зберігає більше одного значення).
- Змінні не встановлюють початкове значення самостійно. Ви ж не хочете, щоб почавши нову гру, значення рахунку було з попередньої гри? Щоб цього не сталося, треба вказувати початкове значення для кожної змінної на початку гри.
- Не всі змінні - числа. Є ще логічні змінні (мають тільки два значення: ІСТИНА або БРЕХНЯ), рядкові змінні (містять слова, речення, літери) та багато інших типів змінних.
- Спрайти і сцена також є змінними, але дуже складними.
- Перш за все змінну необхідно створити. Це можна зробити у категорії блоків ЗМІННІ. Для цього натисніть на кнопку:

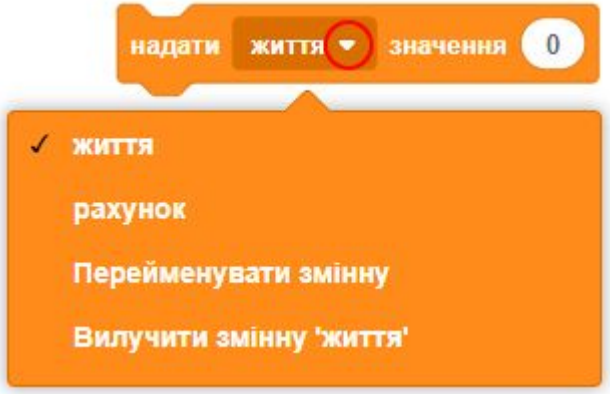
- На екрані з'явиться вікно створення нової змінної. У цьому вікні потрібно ввести ім'я змінної, а також встановити правила користування змінною (чи можуть всі спрайти використовувати цю змінну або вона створюється тільки для цього спрайта).



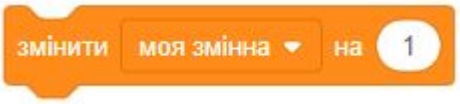
- Після створення змінної з'являються блоки для роботи з нею.

ЗМІННІ: НАДАТИ ЗНАЧЕННЯ

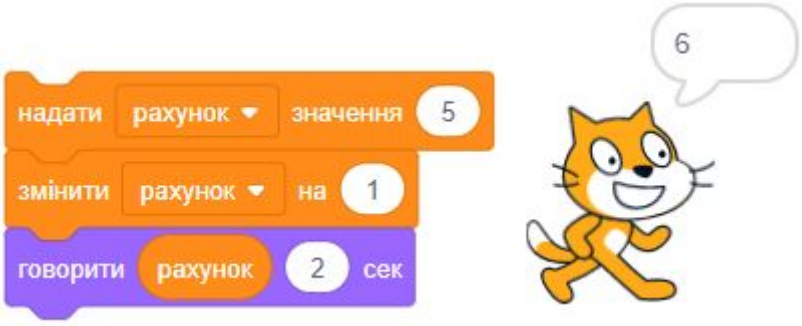
Надає змінній нове вказане значення, не зважаючи на її поточний вміст. Натисніть на трикутник, щоб вибрати змінну, якій буде надано нове значення. У цьому вікні також можна перейменувати або вилучити змінну.



ЗМІННІ: ЗМІНИТИ НА



Збільшити або зменшити змінну на вказану величину. Цей блок використовують для оновлення значення змінної на певну кількість одиниць відносно її поточного значення. Якщо у вас є більше однієї змінної, натисніть на трикутник та використайте меню, щоб вибрати ім'я змінної. Приклад:



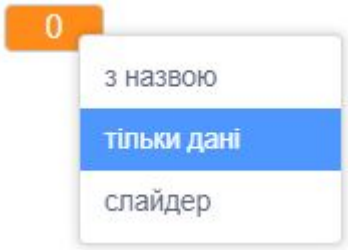
Щоб зменшити змінну, вкажіть від'ємне число (з мінусом).

ЗМІННІ: ПОКАЗАТИ/СХОВАТИ ЗМІННУ

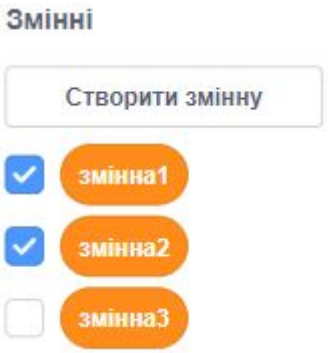
Перший блок показує вікно змінної на сцені. Другий блок ховає його.

Вікно змінної може виглядати так:

Режим відображення можна вибрати за допомогою меню, яке випадає, коли клацнути вікно змінної правою клавішею мишки.



Щоб відобразити вікно змінної на сцені також можна клацнути на галочку поряд з її назвою у списку створених змінних.



ІМЕНА ЗМІННИХ

- Найпопулярніший варіант – починати писати з малої літери, але робити великою першу літеру кожного наступного слова.

довжинаСторін	відсотковаСтавка	першаЛітераРечення

- У Scratch імена змінних можна починати із цифр та не заборонено використовувати в них пропуски, але в інших мовах програмування це не дозволяється.
- Краще не користуватися незвичними іменами для змінних. Змінну ви можете називати, як заманеться, але краще використовувати зрозумілі іншим імена.
- Змінні, імена яких складаються з однієї літери, краще не використовувати. А через дуже довгі імена змінних скрипти можуть втратити зрозумілість.

- Імена змінних чутливі до регістру. Це різні змінні: довжина, ДОВЖИНА, довЖИНА. Не створюйте в межах одного проекту змінні з іменами, що відрізняються лише регістром.

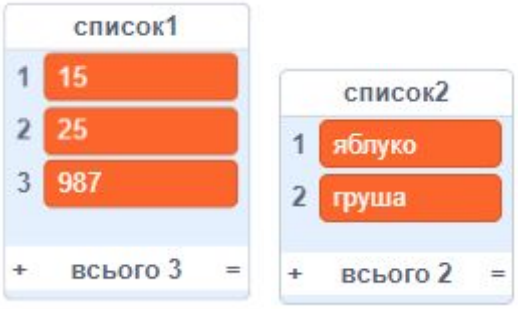
ОБЛАСТЬ ВИДИМОСТІ ЗМІННИХ

- Видимість змінної визначає, які спрайти можуть змінювати значення змінної та де змінна буде зберігатися.
- Ви можете обрати область видимості, коли створюєте змінну.
- Змінні з видимістю «**Для всіх спрайтів**» надають доступ до своїх даних і дозвіл на редагування будь-якому спрайту. Ці змінні називають ГЛОБАЛЬНИМИ ЗМІННИМИ. Вони є зручним способом для передачі даних між спрайтами та для зберігання інформації, яка стосується всього проекту.
- Обираючи «**Тільки для цього спрайта**», ми створюємо змінну, значення якій може надавати лише спрайт, до якого вона належить. Інші спрайти можуть тільки зчитати її значення.
- Змінні, створені з областю «Тільки для цього спрайта», мають локальну видимість, їх називають ЛОКАЛЬНИМИ ЗМІННИМИ.
- Різні спрайти можуть використовувати однакове ім’я для локальних змінних, і жодних проблем із цим не буде.
- Кожен клон спрайта буде мати свою персональну копію змінної, якщо створити її з областю видимості «Тільки для цього спрайта».
- **ПРИКЛАД 1.** У кожного із спрайтів вашої гри може бути локальна змінна «швидкість», яка визначатиме швидкість руху. Кожен спрайт може змінювати свою «швидкість» незалежно один від одного. Тобто, якщо ви задаєте «швидкість» першого спрайта в 10, а «швидкість» другого спрайта в 20, то другий спрайт буде рухатися швидше за перший.
- **ПРИКЛАД 2.** В грі є кнопки, за допомогою яких можна вибрати рівень складності. Ви можете створити глобальну змінну та назвати її «рівеньСкладності». Кожна кнопка-спрайт може задавати цій змінній інше число, коли на неї натиснути. Отже, ви зможете дізнатися, який рівень складності обрав гравець, користуючись цією змінною.

ДОВІДНИК: СПИСКИ

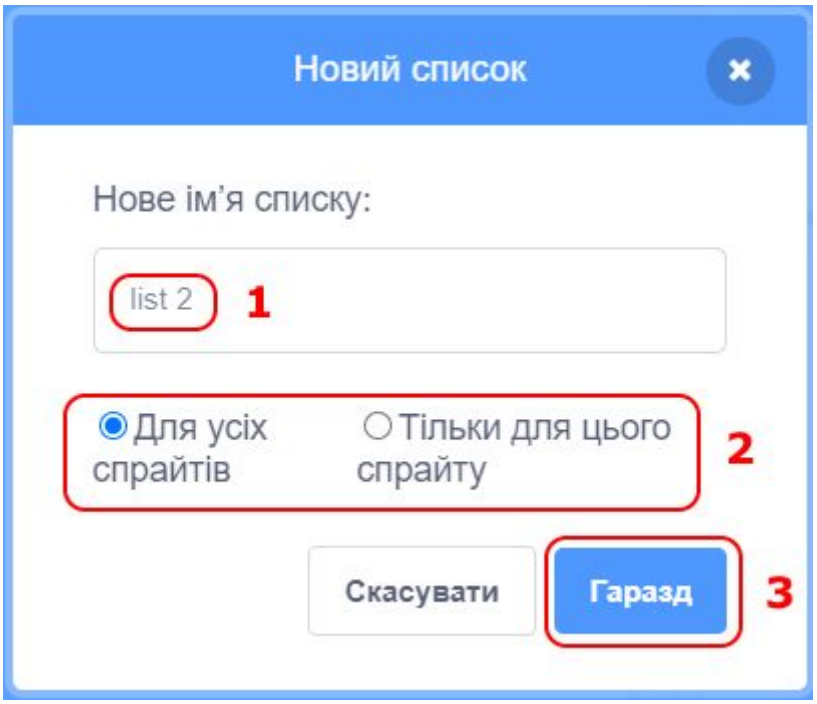
ЯК СТВОРИТИ СПИСОК

- Список – це сховище для інформації зі спільними характеристиками. Список дуже схожий на звичайні змінні, але містить не одне значення, як вони, а багато елементів одночасно. Ви можете створювати списки чисел, літер, слів, фраз та інші.

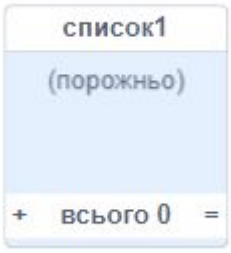


- Щоб створити новий список, виберіть категорію ЗМІННІ та натисніть кнопку СТВОРИТИ СПИСОК.

- У вікні створення нового списку введіть його назву та оберіть інші параметри. Потім натисніть кнопку ГАРАЗД.

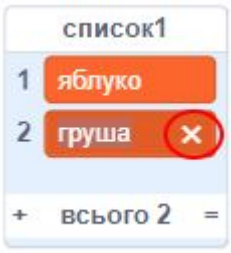


- Створений список завжди порожній.

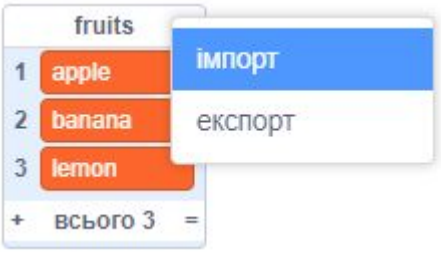


- У списках ви можете додавати, видаляти та змінювати всі елементи або лише деякі з них.
- Щоб додати новий елемент до списку, натисніть кнопку (+) у вікні списку, що показується на сцені. Доданий елемент буде порожній, тому ви повинні ввести його значення.

- Щоб видалити непотрібний елемент зі списку, натисніть на кнопку X біля елемента.



- Список можна імпортувати (завантажити) з файлу, а також експортувати (зберегти) в файл. Для цього натисніть на моніторі списку праву кнопку мишки і виберіть відповідний пункт з меню:



- Під час імпорта підтримуються різні формати файлів:

  csv: Значення, розділені комою.

  tsv: Значення, розділені клавішею TAB.

  txt: Звичайний текстовий файл.

- Нам краще всього використовувати текстовий файл, який можна створити в будь-якому текстовому редакторі.

  Важливо: Кожен елемент списку повинен бути в окремому рядку.

- Є декілька блоків, які працюють зі списками. Після створення нового списку, ці блоки з'являються в категорії ЗМІННІ.

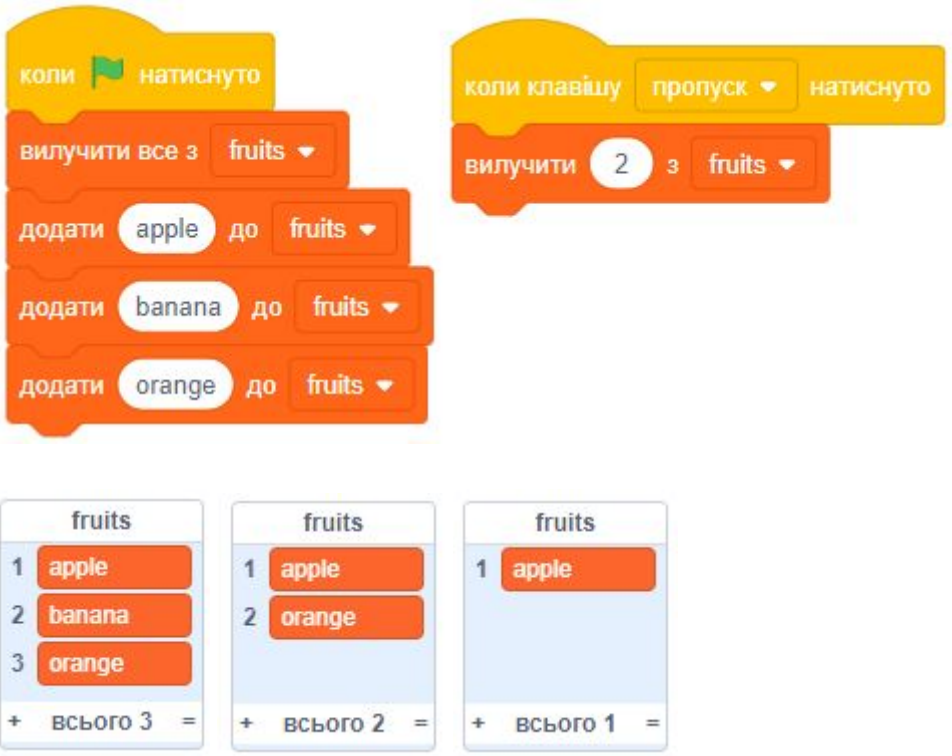
ЗМІННІ: ДОДАТИ ЕЛЕМЕНТ ДО СПИСКУ

Додає вказаний елемент в кінець списку. Елемент може бути будь-якого типа (число, рядок або інший). Приклад:

ЗМІННІ: ВИЛУЧИТИ ЗІ СПИСКУ

Вилучає елемент зі списку. Вкажіть номер елемента, який треба вилучити, в якості параметра блока. Вилучення елемента зменшує довжину списку на одиницю. Приклад:

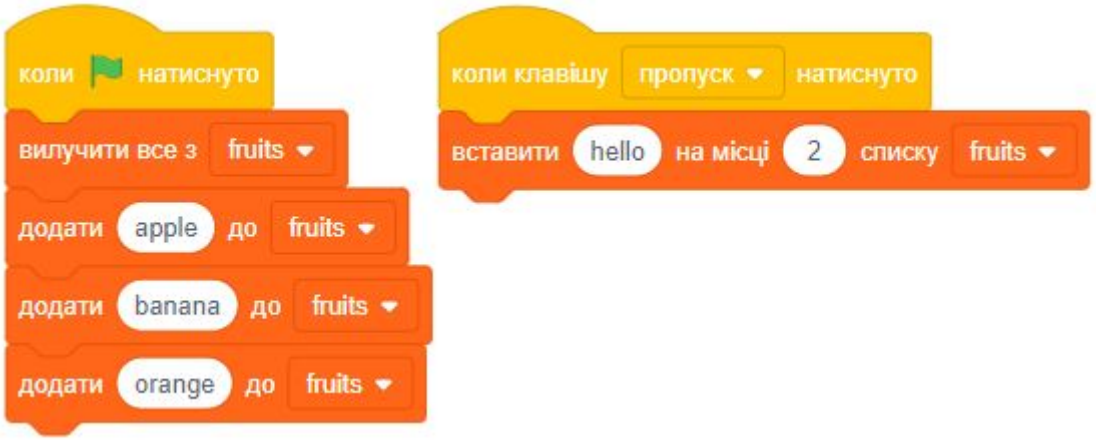
Після запуску скрипта в списку є три елементи (малюнок 1). Після першого натискання клавіші ПРОПУСК залишається два елементи (малюнок 2). Після натискання клавіші ПРОПУСК вдруге, залишається один елемент (малюнок 3). Всі подальші натиснення клавіші ПРОПУСК нічого не роблять, оскільки в списку більше немає другого елементу.



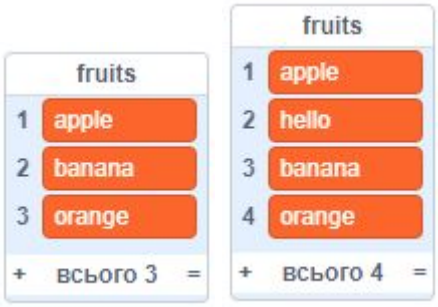
ЗМІННІ: ВСТАВИТИ В СПИСОК

Вставляє елемент у визначену позицію в списку.

Після виконання цього блока, довжина списку збільшиться на одиницю. Приклад:



Після запуску скрипта в списку є три елементи (малюнок 1). Після натискання клавіші ПРОПУСК, вставляється ще один елемент на другу позицію (малюнок 2).

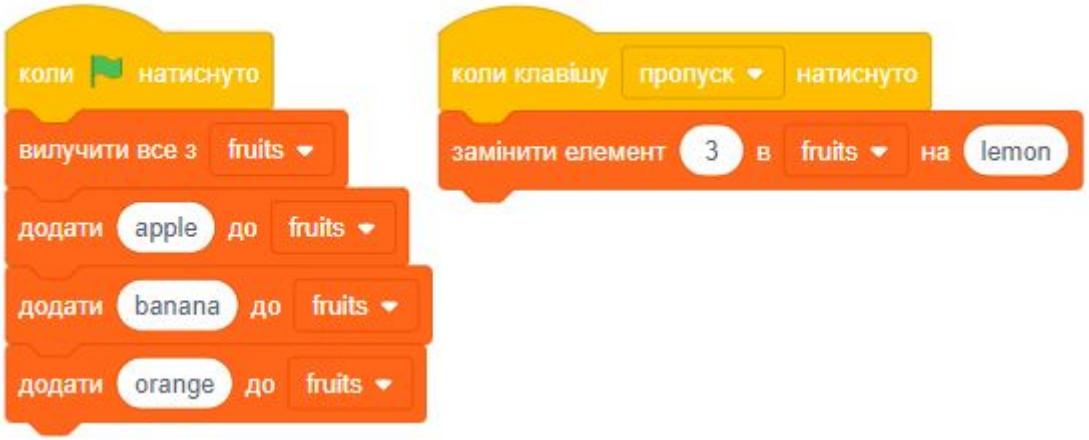


ЗМІННІ: ВИЛУЧИТИ ВСЕ ЗІ СПИСКУ

Цей блок вилучає всі елементи зі списку, роблячи його порожнім (з нульовою довжиною).

ЗМІННІ: ЗАМІНИТИ ЕЛЕМЕНТ В СПИСКУ

Замінює елемент в списку вказаним значенням. Довжина списку після виконання цього блока не змінюється. Ви можете вказати номер елемента для заміни. Приклад:

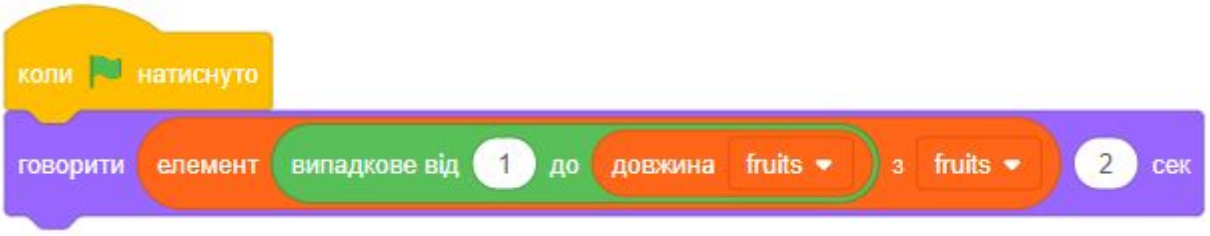


Після запуску скрипта в списку є три елементи (малюнок 1). А після натискання клавіші ПРОПУСК третій елемент змінюється на нове значення (малюнок 2).

ЗМІННІ: ОТРИМАТИ ЕЛЕМЕНТ ЗІ СПИСКУ

Результатом є елемент із зазначеної позиції в списку.

Ви можете вставляти цей блок в інші, щоб передавати туди значення потрібного елемента зі списку. Приклад:

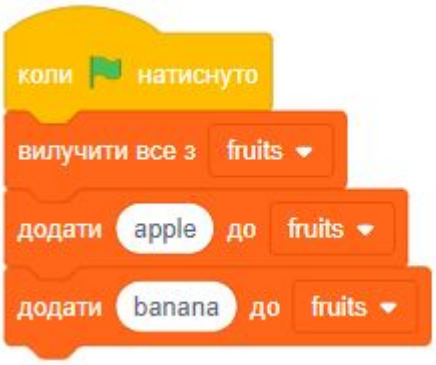


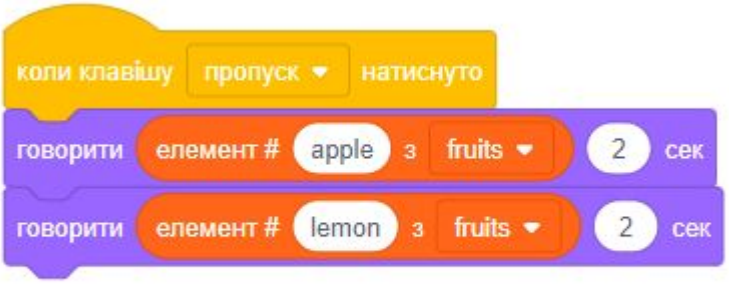
Цей скрипт програмує спрайт говорити випадковий елемент зі списку.

ЗМІННІ: ШУКАТИ ЕЛЕМЕНТ В СПИСКУ

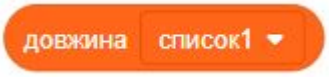
Шукає в списку елемент із вказаним значенням, а потім повертає його номер. Якщо такий елемент не знайдено, то повертається 0 (нуль).

Шукати можна не тільки рядки, а й числа. Приклад:





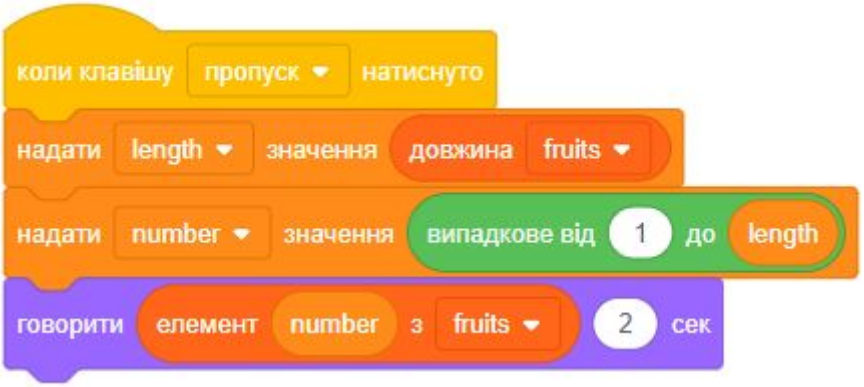
ЗМІННІ: ДОВЖИНА СПИСКУ



Результатом є кількість елементів у списку. Ви можете вставляти цей блок в інші блоки, щоб передавати туди кількість елементів списку.

Довжина списку також показується внизу монітора списку.

Приклад: Спрайт говорить значення випадкового елементу зі списку.



ЗМІННІ: СПИСОК МІСТИТЬ?

Результатом виконання є ІСТИНА (true), якщо список містить вказане значення. В іншому випадку повертається результат БРЕХНЯ (false). Елемент списку повинен точно відповідати вказаному значенню, щоб отримати ІСТИНА.

В прикладі спочатку додається три елемента. При натисканні клавіші ПРОПУСК спрайт перевіряє, чи є в нього яблуко. Спочатку він буде говорити, що в нього є яблуко (I have apple). Після натискання клавіші **b**, яка вилучає всі елементи, він скаже, що яблука немає (I have no apple).

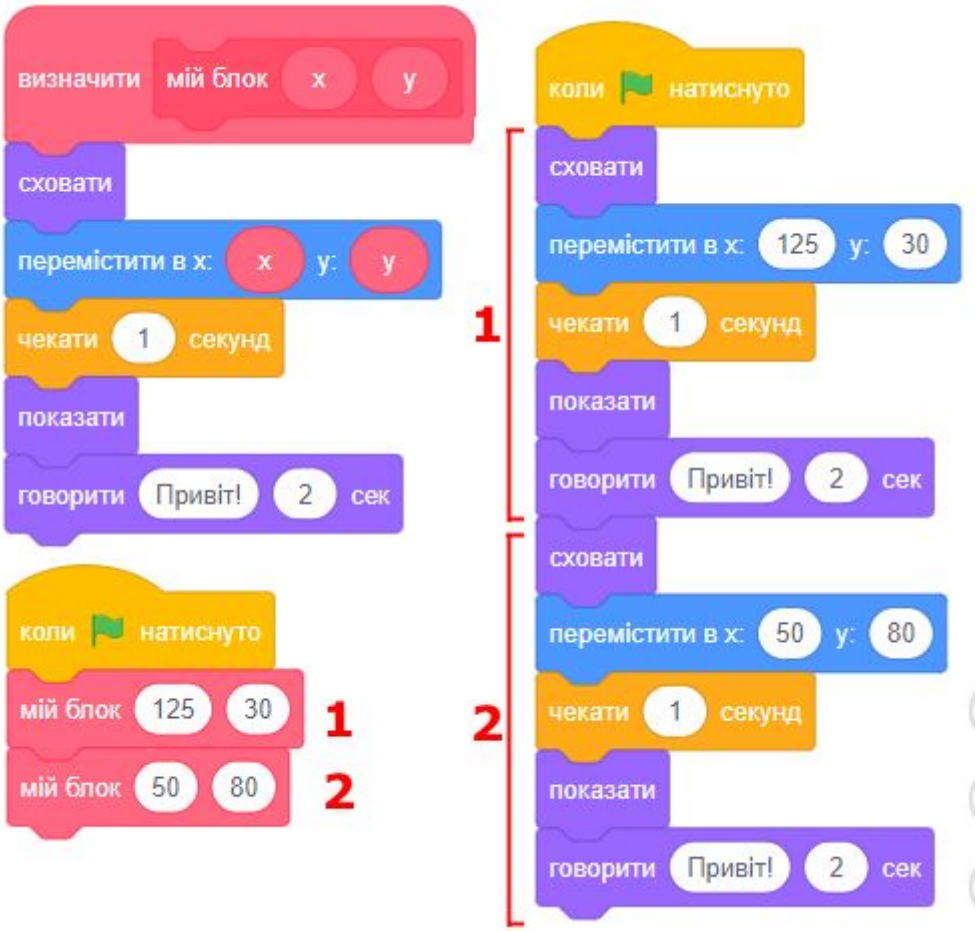
ЗМІННІ: ПОКАЗАТИ/СХОВАТИ СПИСОК

- Перший блок показує вікно списку на сцені. Другий блок ховає його. Вікно списку виглядає так:

- Натисніть на маленький «+», щоб додати новий елемент.
- Перетягніть мишкою правий нижній кут вікна списку (там де «=»), щоб змінити його розмір.
- Вікно списка можна перетягувати по сцені за допомогою мишки (тримати ліву кнопку мишки натиснутою та тягнути).

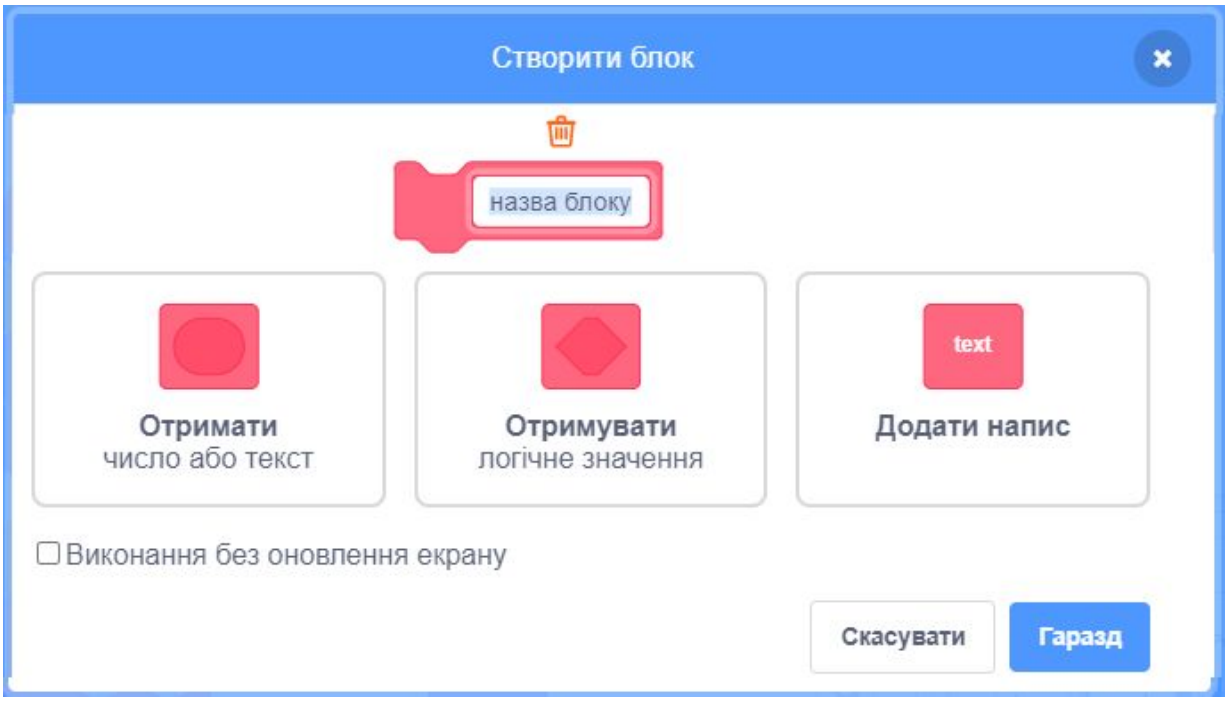
ДОВІДНИК: МОЇ БЛОКИ

- Спочатку категорія МОЇ БЛОКИ порожня. У ній ви можете створити власні блоки та користуватися ними в скриптах, як ви робите це з іншими блоками. Навіщо вам створювати нові блоки? Це може зекономити ваш час та робить меншими ваші скрипти!
- Наприклад, спрайт повинен виконати декілька разів якусь певну послідовність блоків. Щоб кожного разу не додавати в скрипт ці блоки знов і знов, об’єднайте їх в один новий власний блок. Використовуйте створений блок щоразу, коли знадобиться виконати послідовність дій. Приклад:

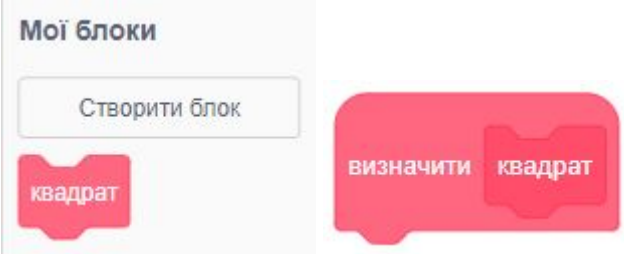


- Виберіть спрайт, до якого треба додати власний блок. Створений блок буде доступний лише цьому спрайту. Ви не зможете використовувати його для інших спрайтів.
- Натисніть кнопку СТВОРИТИ БЛОК в категорії МОЇ БЛОКИ:

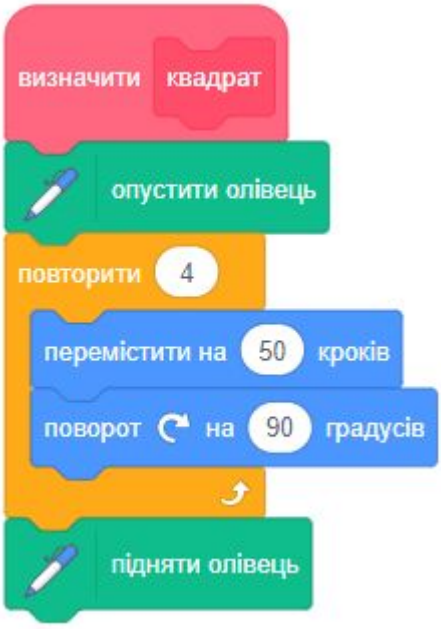
- Перед вами з'явиться вікно створення нового блока. У ньому треба ввести назву блока, а також вказати його параметри.

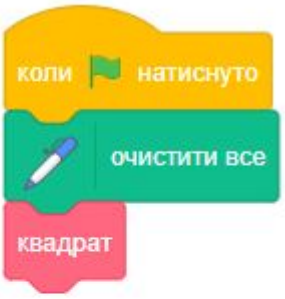


- Давайте створимо простий блок без параметрів та назвемо його «**квадрат**». Після того як ми введемо назву нового блоку і натиснемо ГАРАЗД, створений блок з'явиться на палітрі блоків. Також в області скриптів з'явиться команда визначення вашого нового блоку.

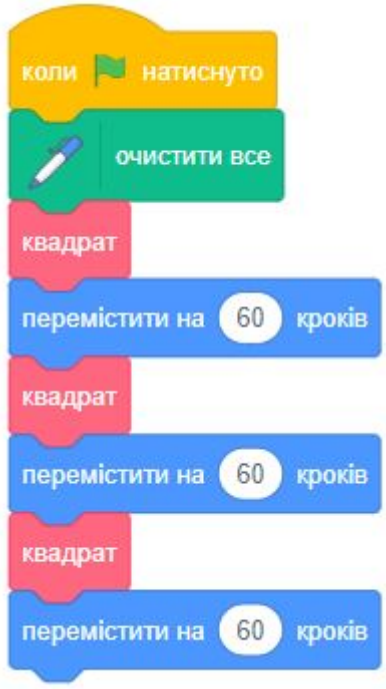


- Давайте додамо туди скрипт для малювання квадрата та використаємо новий блок:

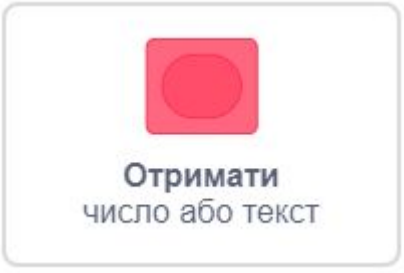




- Ми можемо використовувати блок «квадрат» багаторазово в різних місцях нашої програми. Наприклад так:

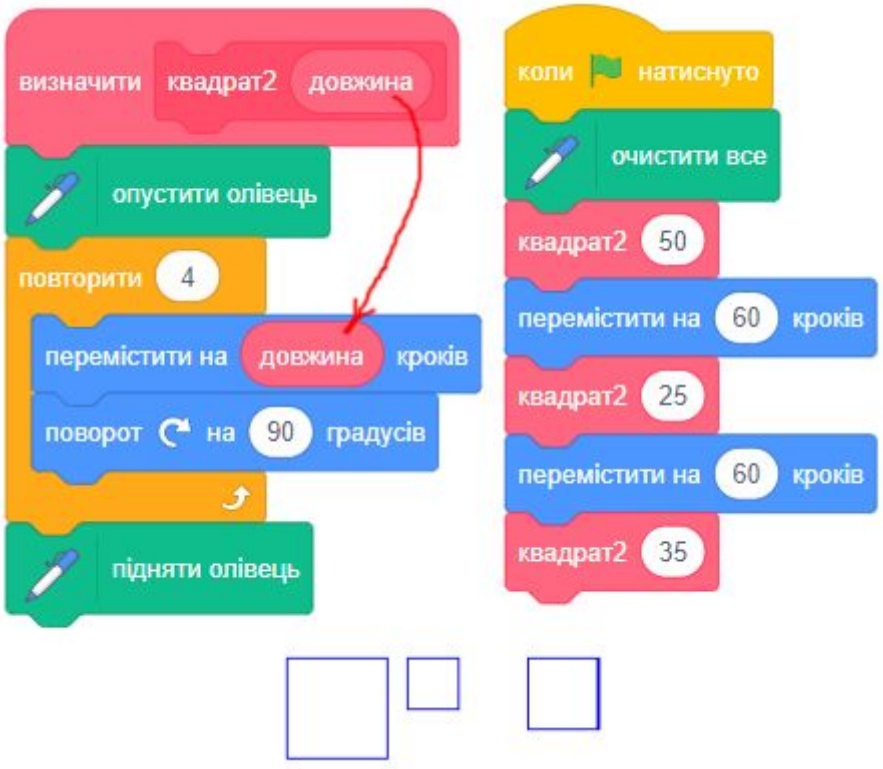


- Додайте новий блок «**квадрат2**» для малювання квадратів різного розміру. Для цього під час створення блоку натисніть на:



- При цьому ваш блок буде виглядати так:

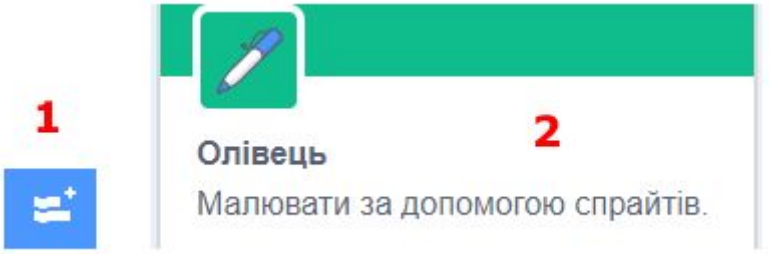
- Відразу ж введіть назву параметра «**довжина**». Сміттєва корзина над назвою параметра дозволяє видалити його.
- Якщо потрібно два параметри (або більше), вам треба натиснути кнопку додавання параметра ще раз.
- Блок визначення команди з параметрами показаний нижче. Ви можете перетягнути за допомогою мишки назву параметра в місце, де його потрібно використовувати.



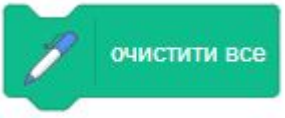
- Тепер ми можемо малювати квадрати різного розміру.

ДОВІДНИК: ОЛІВЕЦЬ

У новому Scratch категорія блоків управління олівцем захована в розширеннях. Для того щоб її додати, необхідно натиснути кнопку додавання розширення та вибрати розширення ОЛІВЕЦЬ.



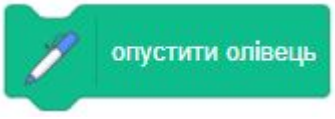
ОЛІВЕЦЬ: ОЧИСТИТИ ВСЕ



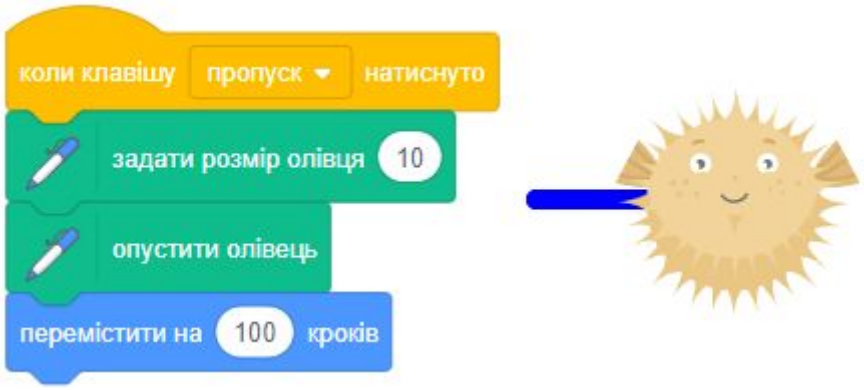
Очистити сцену від слідів олівця і штампів.

Примітка: сліди олівця і штампи не є частиною фону сцени, і будуть знищені при його зміні.

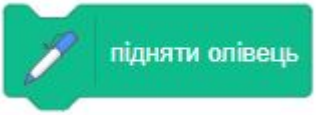
ОЛІВЕЦЬ: ОПУСТИТИ ОЛІВЕЦЬ



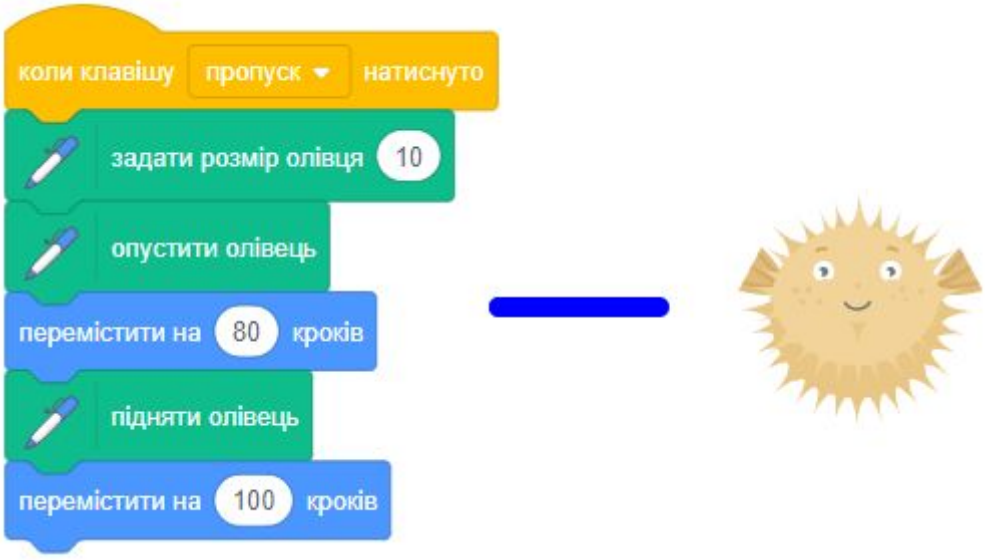
Опускає вниз олівець спрайта і дозволяє малювати при переміщенні. Після виконання цього блоку спрайт буде залишати слід при русі.



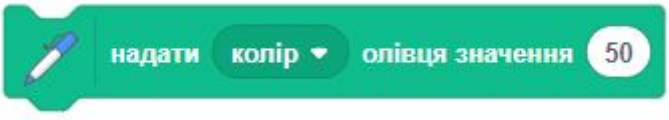
ОЛІВЕЦЬ: ПІДНЯТИ ОЛІВЕЦЬ



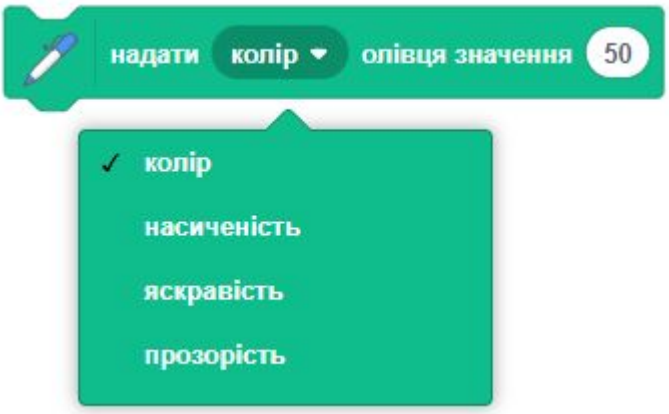
Піднімає вгору олівець спрайта, тому він не буде малювати при переміщенні. Приклад:



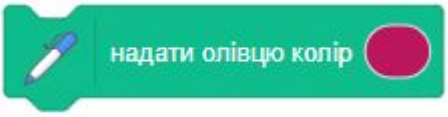
ОЛІВЕЦЬ: НАДАТИ КОЛІР ОЛІВЦЯ ЗНАЧЕННЯ



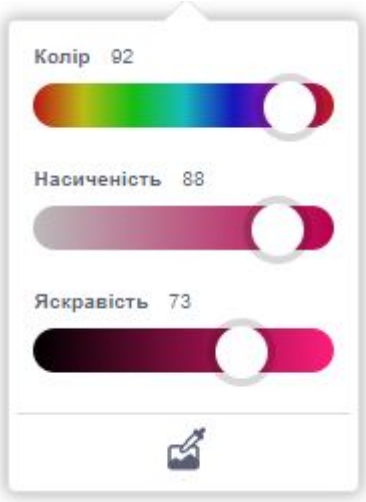
Встановити колір олівця у відповідності з наведеним значенням. Ви можете встановити колір олівця за допомогою змінної, або задати число від 0 до 100. Також можна встановити конкретні атрибути кольору олівця, вибравши їх зі списку:



ОЛІВЕЦЬ: НАДАТИ ОЛІВЦЮ КОЛІР

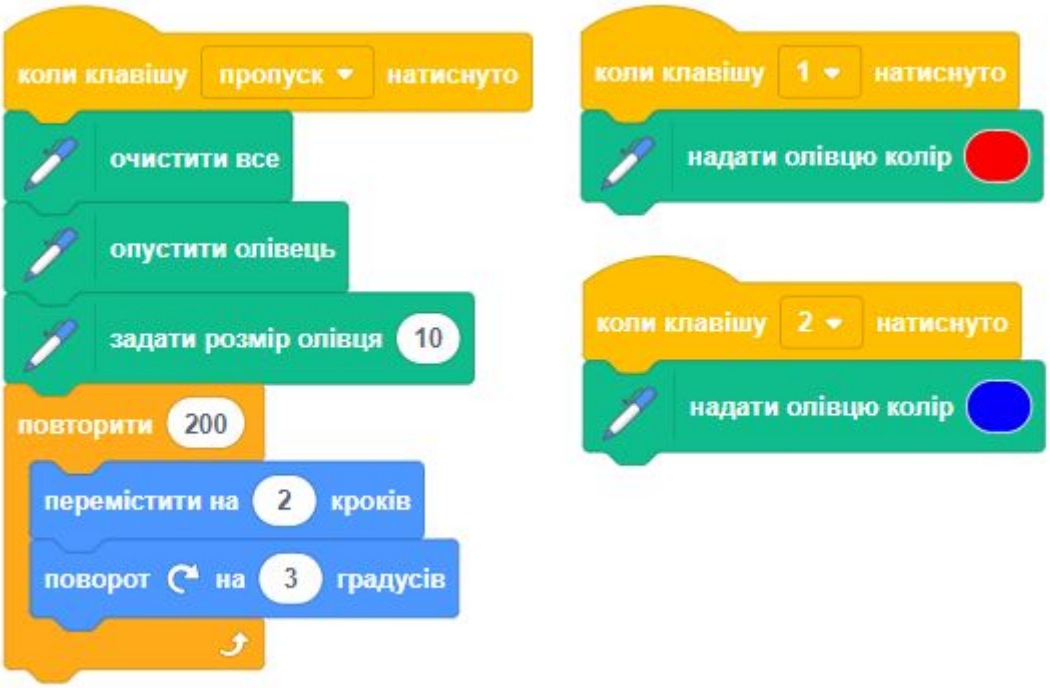


Встановити колір олівця, яким спрайт малює після виклику блока ОПУСТИТИ ОЛІВЕЦЬ. Натисніть на трикутник, щоб вибрати колір за допомогою спеціального вікна.

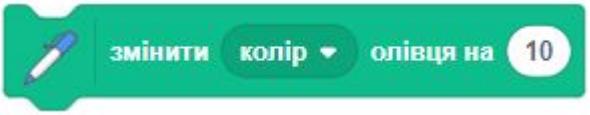


У вікні вибору кольору ви також можете скористатися піпеткою, натиснувши на її зображення.

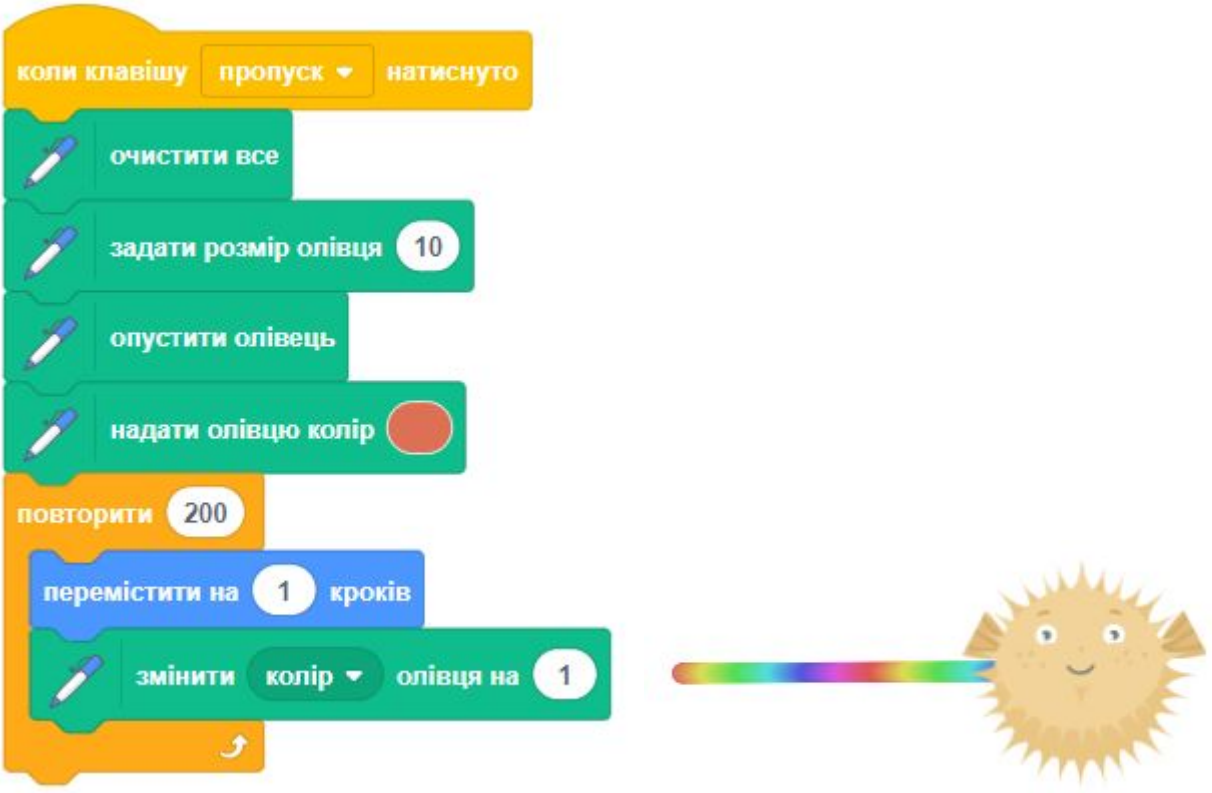
Далі використовуйте піпетку, щоб натиснути на потрібний вам колір на сцені. Бажаний колір буде обраний.



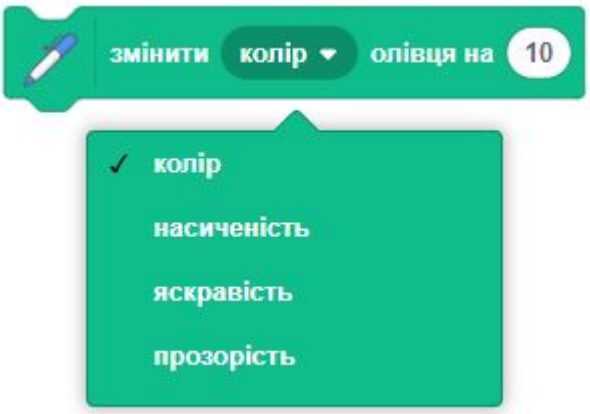
ОЛІВЕЦЬ: ЗМІНИТИ КОЛІР ОЛІВЦЯ



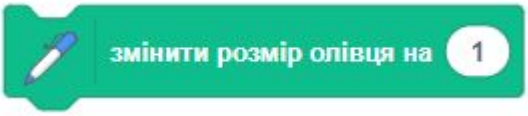
Змінює колір олівця на зазначену величину.



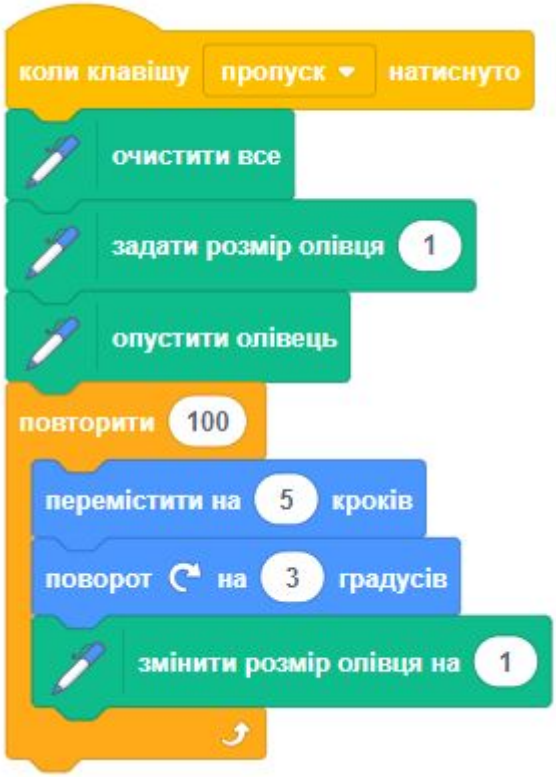
Можна змінити інші атрибути кольору олівця, вибравши їх зі списку:

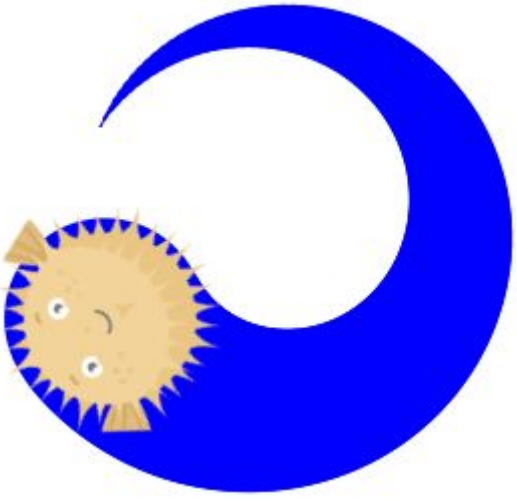


ОЛІВЕЦЬ: ЗМІНИТИ РОЗМІР ОЛІВЦЯ

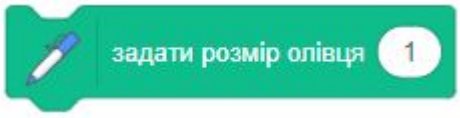


Змінити товщину олівця. Позитивні значення збільшують розмір пера, від'ємні - зменшують. Приклад:

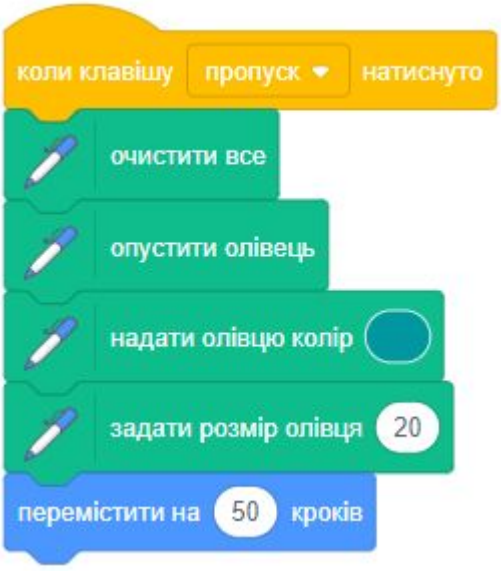




ОЛІВЕЦЬ: ЗАДАТИ РОЗМІР ОЛІВЦЯ

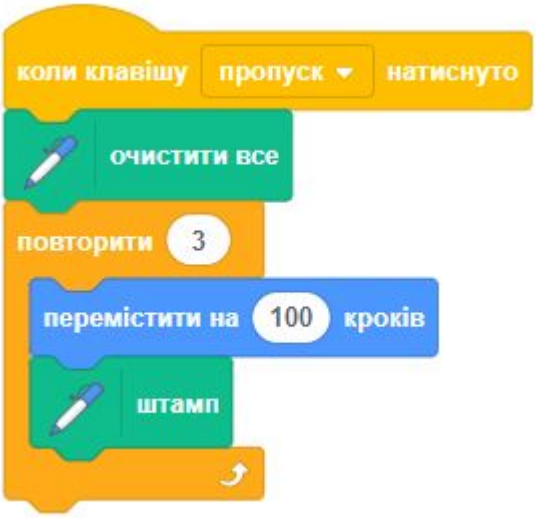


Задає товщину олівця, яким спрайт малює на сцені після виклику блоку ОПУСТИТИ ОЛІВЕЦЬ. Приклад:



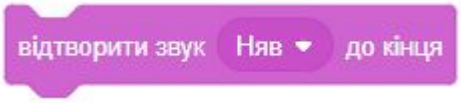
ОЛІВЕЦЬ: ШТАМП

Друкує зображення спрайта (штамп) на сцені. Штампом є тимчасове зображення, яке спрайт залишає на сцені. Штамп не може переміщатися і не може мати скриптів. Приклад:



ДОВІДНИК: ЗВУК

ЗВУК: ВІДТВОРИТИ ЗВУК ДО КІНЦЯ

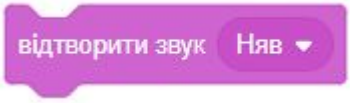


Відтворює звук і **чекає**, поки він не закінчиться. Виконання наступних блоків скрипта припиняється.

Натисніть на трикутник, щоб вибрати звук для програвання.
Можна обирати тільки звуки, які були додані до проекту.

Для додавання звуків виберіть вкладку ЗВУКИ.

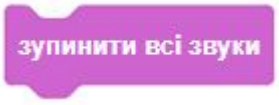
ЗВУК: ВІДТВОРИТИ ЗВУК



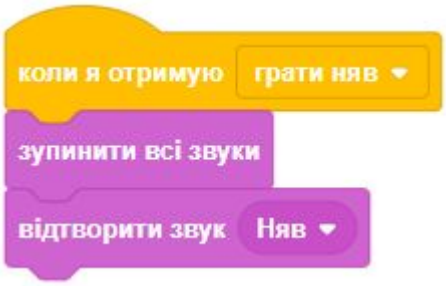
Програти звук. Цей блок починає відтворення звуку і переходить до виконання наступного блока без очікування.

Натисніть на трикутник, щоб вибрати звук для програвання. Ви можете вибрати тільки вже доданий до спрайта звук. Щоб додати звук, виберіть вкладку ЗВУКИ.

ЗВУК: ЗУПИНИТИ ВСІ ЗВУКИ

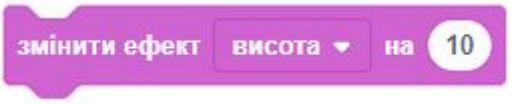


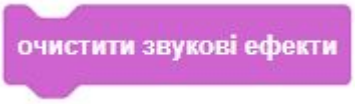
Зупинити відтворення всіх звуків. Приклад:



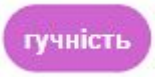
ЗВУК: ЗВУКОВІ ЕФЕКТИ

Ви можете встановити, змінити або прибрати звукові ефекти. Натисніть на трикутник, щоб вибрати, який саме звуковий ефект встановити або змінити (висота або панорама).





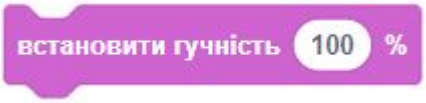
ЗВУК: ГУЧНІСТЬ



Повідомляє гучність звука спрайта. Гучність змінюється від 0 до 100, за замовчуванням гучність 100. Ви можете встановити рівень гучності звука окремо для кожного спрайта.

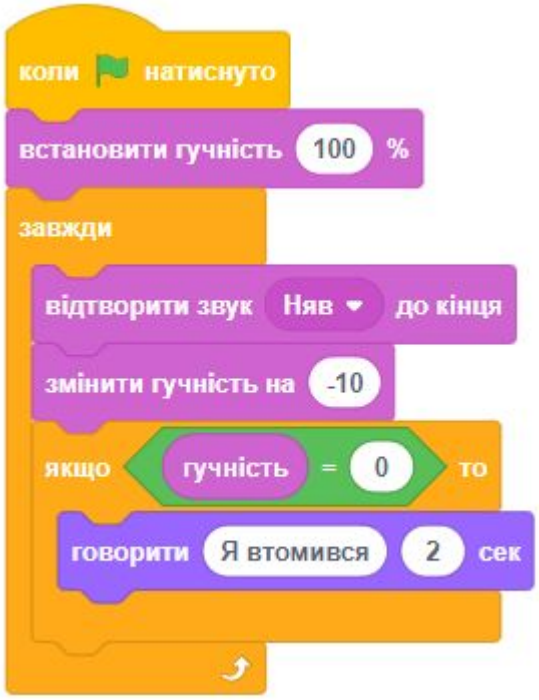
Для відтворення двох звуків одночасно з різною гучністю використовуйте два спрайти.

ЗВУК: ВСТАНОВИТИ ГУЧНІСТЬ

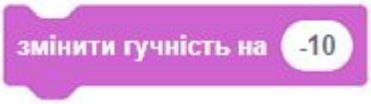


Встановлює задане значення гучності звука спрайта. Гучність змінюється від 0 до 100, за замовчуванням гучність 100. Ви можете налаштувати рівень гучності звука окремо для кожного спрайта.

Для одночасного відтворення двох звуків з різною гучністю скористайтеся двома спрайтами.



ЗВУК: ЗМІНИТИ ГУЧНІСТЬ

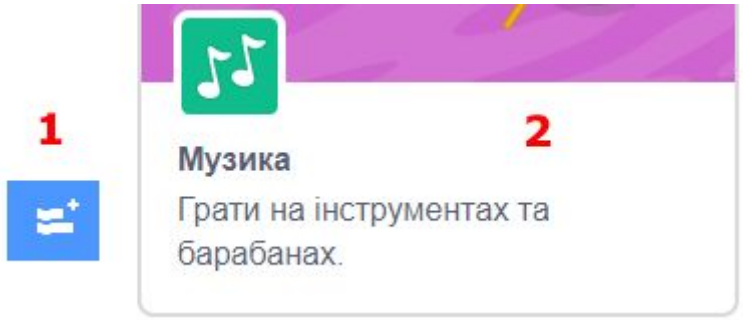


Змінює гучність звука для спрайта на зазначену величину. Гучність змінюється від 0 до 100, за замовчуванням гучність 100. Ви можете встановити рівень гучності окремо для кожного спрайта.

Для відтворення двох звуків одночасно з різною гучністю використовуйте два спрайти. Приклад:

ДОВІДНИК: МУЗИКА

У новому Scratch категорія блоків МУЗИКА захована в розширеннях. Для того щоб її додати, необхідно натиснути кнопку додавання розширення та вибрати розширення МУЗИКА.



МУЗИКА: ПРОГРАТИ НА БАРАБАНІ

Грає обраний музичний інструмент певну кількість часу. Слово «drum» у назві блока помилково переклали з англійської як «барабан», насправді ж йдеться про «інструмент».

Ви можете вибрати музичний інструмент зі списку, натиснувши на трикутник. Також можна ввести число в межах від 1 до 18. Якщо ввести негативне число, 0 або число більше 18, то вибереться інструмент №3.

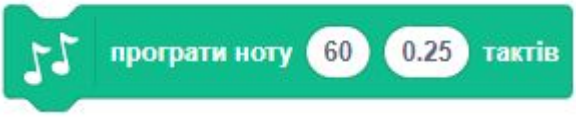
Тривалість удару встановлюється за допомогою ВСТАНОВИТИ ТЕМП.

Темпом є швидкість (кількість ударів за хвилину) програвання нот і барабана.

МУЗИКА: ПАУЗА..ТАКТІВ

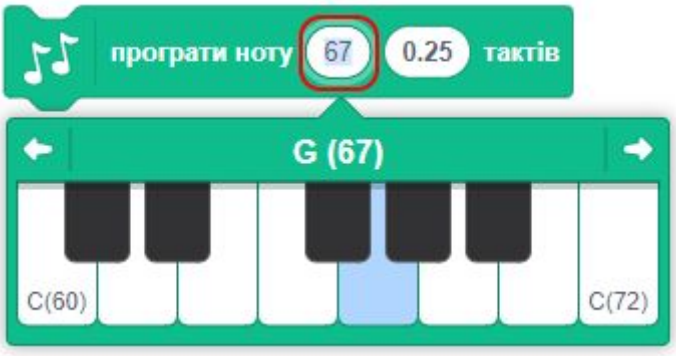
Пауза - музика припиняється на вказану кількість тактів. Приклад:

МУЗИКА: ПРОГРАТИ НОТУ



Грає музичну ноту вказане число тактів.

Ви можете вибрати, яку ноту грати, у вікні, що випадає:



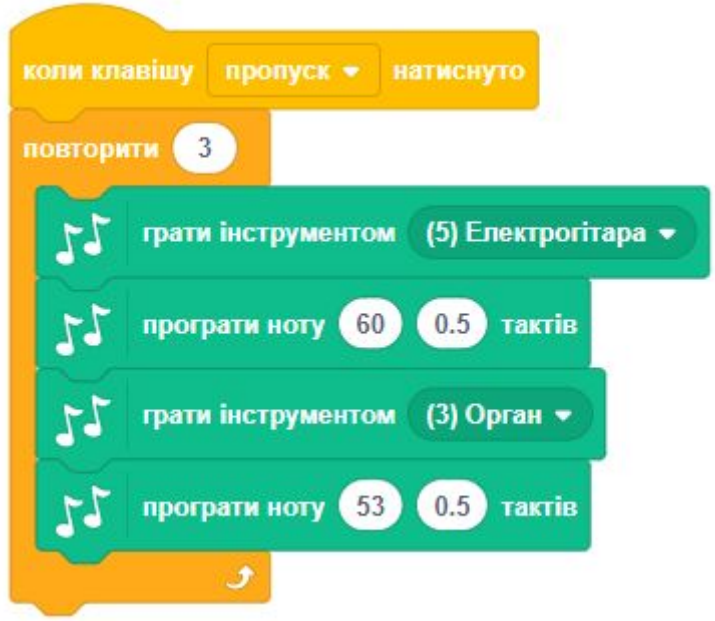
Також ви можете ввести ноту за допомогою клавіатури (числа від 0 до 130). Використовуйте великі числа для більш високих нот.

Довжину такту можна встановити за допомогою ВСТАНОВИТИ ТЕМП.

МУЗИКА: ГРАТИ ІНСТРУМЕНТОМ

Встановлює тип інструменту, який спрайт використовує для відтворення нот. Кожен спрайт має свій власний інструмент.

Клацніть на параметрі, щоб вибрати його з меню. До бібліотеки додано 20 різних інструментів.



МУЗИКА: ТЕМП, ВСТАНОВИТИ/ЗМІНИТИ ТЕМП

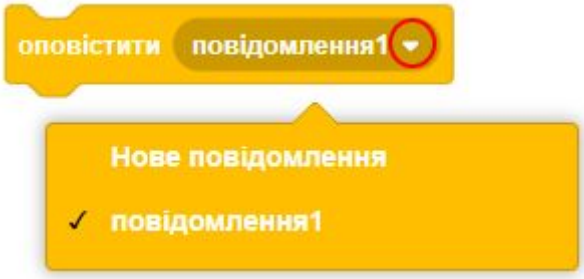
Темп - це швидкість (bpm = тактів за хвилину) програвання нот і барабана в скретч. Чим більше значення темпу, тим швидше будуть грати ноти і барабан.

Другий блок встановлює темп програвання музики для спрайта. Третій блок змінює темп на задану величину (додає її до поточного темпу).

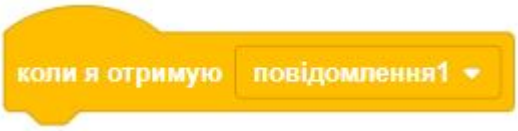
ДОВІДНИК: ПОВІДОМЛЕННЯ

- Спрайти та сцена можуть спілкуватись між собою, надсилаючи один одному повідомлення. Для оповіщення та прийому повідомлень використовуйте ці блоки з категорії ПОДІЇ:

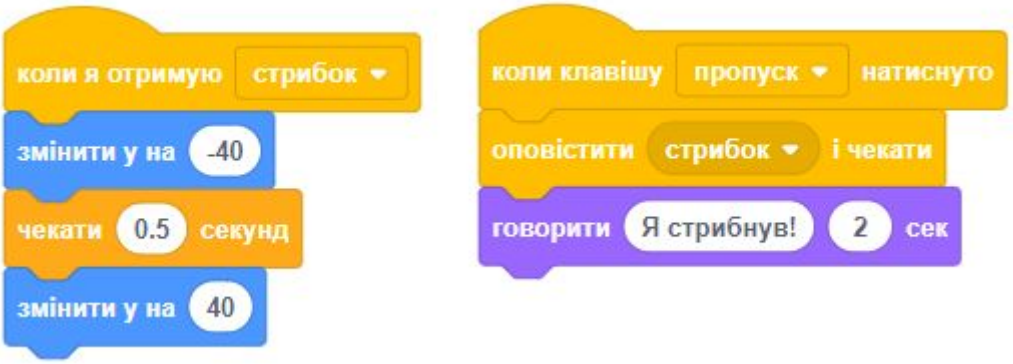
- Кожен з цих блоків має параметр, в якому вказується назва повідомлення.
- Щоб ваш скрипт був зрозумілий, назва повідомлення має відповідати його змісту.
- Назви повідомлень не показуються користувачам, ними користуються тільки ви та ваша програма.
- Клацніть на трикутник, щоб вибрати повідомлення, яке буде відправлено або отримання якого буде очікуватися. Або виберіть НОВЕ ПОВІДОМЛЕННЯ і введіть його назву у вікні створення повідомлення.



ПОДІЇ: КОЛИ Я ОТРИМАЮ



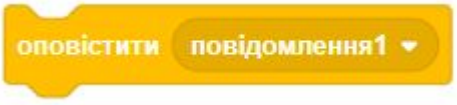
- Виконує під'єднаний знизу скрипт, якщо отримано певне повідомлення. В цьому прикладі спрайт стрибає, коли натиснута клавіша ПРОПУСК, а потім каже, що він стрибнув.



ПОДІЇ: ОПОВІСТИТИ І ЧЕКАТИ

- Надіслати певне повідомлення всім спрайтам і сцені, а потім почекати, поки вони не закінчать виконання скриптів, які обробляють надіслане повідомлення.
- Якщо ви не хочете чекати, поки всі спрайти оброблять повідомлення, використовуйте блок ОПОВІСТИТИ.

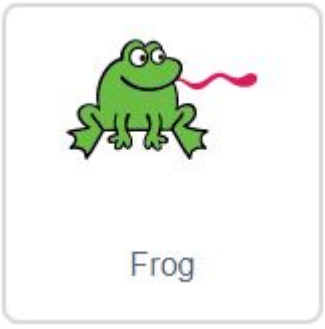
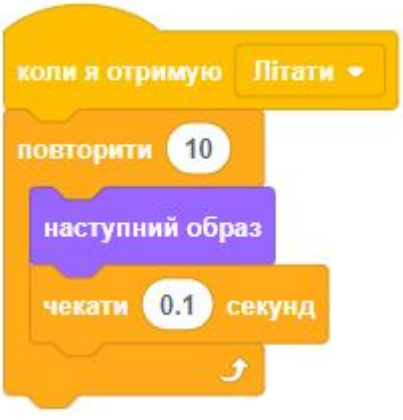
ПОДІЇ: ОПОВІСТИТИ



- Надіслати певне повідомлення всім спрайтам та сцені.
- Блок НЕ ЧЕКАЄ, поки повідомлення буде надіслано і оброблено.
- Якщо ви хочете почекати, поки всі спрайти оброблять повідомлення, використовуйте блок ОПОВІСТИТИ І ЧЕКАТИ.

СИСТЕМА ПОВІДОМЛЕНЬ

- Як працює система повідомлень? Будь-який спрайт або сцена можуть надіслати повідомлення за допомогою одного з блоків: ОПОВІСТИТИ або ОПОВІСТИТИ І ЧЕКАТИ.
- Після виклику блока оповіщення, запускаються всі скрипти в усіх спрайтах, які починаються з блока КОЛИ Я ОТРИМУЮ.
- Оповіщення отримують усі спрайти та сцена, але реагуватимуть на нього лише ті, які мають блок КОЛИ Я ОТРИМУЮ.
- Ви можете використати систему повідомлень, щоб скоординувати роботу багатьох спрайтів.
- Давайте наведемо приклад. Припустимо, що у вашій грі є такі спрайти: динозавр, сова, жаба, кажан.
- Сова передає повідомлення «стрибати», і воно надходить до всіх спрайтів. І сова не є винятком.
- Кожний спрайт стрибає, як вміє, і виконує свій власний скрипт.
- Кажан отримує повідомлення про стрибок, але не може відгукнутися на нього, бо не знає, якими мають бути його дії.

Жаба вміє тільки стрибати. 	
Кажан вміє тільки літати. 	

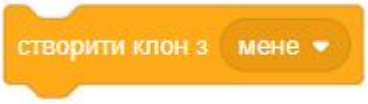
Динозавр знає, як йому ходити та стрибати.

ПЕРЕМОГА ТА ПОРАЗКА

- Перемога та поразка зазвичай супроводжуються різними подіями. Кожен зі спрайтів знає, що йому робити в разі перемоги, а що – при поразці. Наприклад, ваші спрайти можуть зникнути зі сцени, а тло сцени може змінитися на якесь інше.
- Зробити це дуже просто, вам потрібно надіслати повідомлення про перемогу або про поразку. Таким чином усі ваші спрайти та сцена дізнаються, що відбувається.
- Коли спрайти отримують повідомлення, то можуть ігнорувати його або почнуть виконувати потрібні дії.

ДОВІДНИК: КЛОНИ

КЕРУВАННЯ: СТВОРИТИ КЛОН



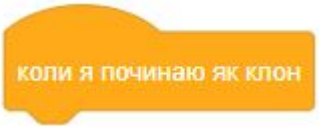
Створення клону (тимчасового дубліката) певного спрайту.

Клон існує тільки під час виконання проекту. Виберіть спрайт для клонування в меню, що випадає, натиснувши трикутник.

Якщо ви не бачите створений клон, покажіть його та перемістіть так, щоб основний спрайт не перекривав його. Якщо цього не зробити, клон з'являється в тому ж місці, що й основний спрайт.

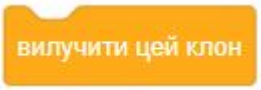
Ви можете створити приблизно 300 клонів або менше, якщо у вас старий комп'ютер. Якщо кількість клонів стає більшою за це число, нові клони перестають створюватись. Чим більше клонів створено, тим повільніше працює ваша гра.

КЕРУВАННЯ: КОЛИ Я ПОЧИНАЮ ЯК КЛОН



Це стартова подія. Під цим блоком ви можете задати скрипт, який буде виконуватися, коли клон спрайту буде створений.
Цей блок дуже подібний до блока КОЛИ НАТИСНУТО ЗЕЛЕНИЙ ПРАПОР, але працює для клонів.
Під цим блоком ви можете показати клон або перемістити в інше місце, змінити його образ або зробити щось інше.

КЕРУВАННЯ: ВИЛУЧИТИ ЦЕЙ КЛОН



Видаляє поточний клон спрайту.

Клони треба видаляти, коли вони більше не потрібні. Якщо клони не видаляти, а тільки ховати, ваша гра може перестати працювати правильно.

Всі клони автоматично знищуються при зупинці програми.

ЗАВДАННЯ

ПОЧАТКОВІ

1. Введіть значення двох змінних і визначте їх суму, різницю між першим і другим, добуток першого та другого і частку від ділення першого на друге.
2. У початковій школі ми виконували дію ділення із залишком. Наприклад: 7:4 = 1 (3). Напишіть програму, що ділить перше введене з клавіатури число на друге.
3. Ввести з клавіатури число, порівняти його з числом 10. Результат порівняння вивести на екран.
4. Ввести з клавіатури 2 числа, порівняти їх. Результат порівняння показати на екрані.
5. Сума довжин будь-яких двох сторін трикутника більше довжини третьої його сторони. Програма попросить ввести три числа і визначить, чи можуть вони бути довжинами сторін трикутника.
6. Попросіть ввести з клавіатури число і визначте, чи ділиться воно на 3. Скористайтеся тим, що якщо перше число ділиться на друге, то залишок від ділення дорівнює 0.
7. Ввести з клавіатури два числа. Визначити, чи ділиться одне з них на інше, якщо ділиться, то виконати це ділення.
8. Ввести з клавіатури три числа і знайти найменше.
9. Напишіть програму, яка запитує число і проводить лінію зазначеної довжини.
10. Напишіть програму, яка запитує число, проводить лінію зазначеної довжини і градуює її, позначаючи кожен 10 блок на цій лінії. А потім стирає цю лінію з кінця по одному блоку (скористайтеся затримкою).
11. Заповнення списку і показування значень його елементів.
12. Поміняйте місцями значення двох змінних.

13. Напишіть програму, яка запитує зріст Матроскіна і Шаріка, конструює "блоками" горизонтальні лінії так, що зросту найвищого відповідає лінія в 20 блоків. Зросту другого персонажа відповідає пропорційно менша лінія.
14. Програма генерує два випадкових числа. Потім запитує третє число. Обчислює, до якого з випадкових чисел ближче третє число. І визначає, чи розташоване третє число на відрізку від найменшого до найбільшого з випадкових чисел. Наприклад: випадкові числа 5 і 9, людина назвала 11. Програма повідомляє, що найближче число 9, а 11 не лежить на відрізку від 5 до 9.
15. Дано тризначне число. Знайдіть добуток його цифр.
16. Дано п'ятизначне число. Знайдіть різницю двох чисел. Перше число дорівнює сумі цифр вихідного числа, що стоять на парних місцях. Друге число - сума цифр, що стоять на непарних місцях.
17. Дано тризначне число. Виведіть нове число, отримане з вихідного шляхом перестановки цифр у зворотному порядку.
18. Дано тризначне число. Замініть середню цифру на нуль.
19. Дано шестизначне число. Поміняйте першу та останню цифри.
20. Дано п'ятизначне число. Цифри на парних позиціях обнулити. Наприклад, з 12345 виходить число 10305.
21. Дано два тризначних числа. Знайдіть шестизначне число, утворене з двох даних чисел шляхом дописування другого числа до першого справа.
22. Дано два числа. Виведіть на екран більше з них.
23. Дано два числа. Якщо вони не рівні, то знайти їх суму, інакше знайти їх добуток.
24. Дано три числа. Якщо перше число більше двох інших, то знайти їх суму, інакше знайти різницю між 1 і 3 числами.
25. Дано три числа. Якщо сума перших двох чисел дорівнює третьому числу, то знайти суму квадратів трьох чисел, інакше знайти різницю між першим числом і третім.
26. Дано номер місяця першого півріччя. Вивести назву місяця.
27. Дано число. Якщо воно більше 10, то вивести його, якщо ж воно менше 3, то збільшити його на 11 і вивести, в інших випадках зменшити число на 100.
28. Дано три числа. Якщо сума двох з них дорівнює третьому, то виведіть цю рівність, якщо ні, то виведіть добуток цих чисел.

29. Дано номер дня тижня. Виведіть назву дня тижня.
30. Дано номер місяця. Виведіть на екран пору року (осінь, зима, літо, весна).
31. Дано два числа. Якщо обидва більше 8, то вивести їх добуток, якщо хоча б одне менше 5, то вивести їх суму.
32. Дано три числа. Якщо всі числа більші нуля, виведіть їх суму. Якщо два з них менші нуля, то виведіть їх добуток. Якщо два з них дорівнюють нулю, то виведіть нуль.
33. Дано число. Якщо воно в діапазоні від 4 до 10 включно, то збільшити його в 2 рази. Якщо воно менше 0 або в діапазоні від 1 до 3 включно, то зменшити на 55.
34. Дано три числа. Знайти найбільшу суму двох із них.
35. Дано чотиризначне число. Переставити цифри так, щоб вийшло найбільше з можливих число.
36. Вивести на екран 8 разів фразу "Loading ...".
37. Вивести на екран числа 10, 20, 30, ...150.
38. Вивести N рядків, кожен з яких складається з M нулів.
39. Знайдіть суму 1 + 3 + 5 + 7 + ... + 37.
40. Знайдіть добуток $1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$.
41. Виведіть на екран n рядків, кожен з яких складається з такої кількості нулів, як номер рядка.
42. Користувач вводить число. Виведіть на екран квадрат цього числа, куб цього числа.
43. Користувач вводить три числа. Збільшіть перше число в два рази, друге число зменшіть на 3, третє число зведіть у квадрат і потім знайдіть суму нових трьох чисел.
44. Користувач вводить три числа. Знайдіть середнє арифметичне цих чисел, а також різницю між подвоєною сумою першого і третього чисел і потроєним другим числом.
45. Користувач вводить кількість тижнів, місяців, років і отримує кількість днів за цей час. Рахувати, що в місяці 30 днів.
46. Користувач вводить ціни 1 кг цукерок і 1 кг печива. Знайдіть вартість: а) однієї покупки 300 грам цукерок і 400 грам печива; б) трьох покупок, кожна покупка складається з 2 кг печива і 1 кг 800 грам цукерок.

47. Відомо, що X кг цукерок коштують A доларів. Визначте, скільки коштує У кг цих цукерок, а також скільки кілограмів цукерок можна купити на К доларів. Числа вводить користувач.
48. Дано три змінні A, B і C. Змінити значення цих змінних так, щоб в A зберігалося значення A + B, у B зберігалася різниця старих значень C - A, а в C сума старих значень A + B + C. Наприклад, A = 0, B = 2, C = 5, тоді нові A = 2, B = 3 і C = 7.

РОЗГАЛУЖЕННЯ

49. Дано число. Якщо воно більше 3, то збільшити число на 10, інакше зменшити на 10.
50. Дано число. Якщо воно менше 7, то вивести Yes. Якщо воно більше 10, то вивести No, а якщо дорівнює 9, то вивести Error.
51. Користувач вводить номер місяця, вивести назву місяця.
52. Дано два числа. Вивести найбільше з них.
53. Дано два числа. Вивести Yes, якщо вони відрізняються на 100, інакше вивести No.
54. Дано два числа. Якщо перше число більше другого, то вивести Yes, інакше поміняти значення у цих змінних і вивести їх.
55. Дано число. Якщо воно від -10 до 10 включно, то збільшити його на 5, інакше зменшити на 10.
56. Дано число. Якщо воно більше 100 або менше -100, то обнулити його, інакше збільшити на 1.
57. Дано число. Якщо воно від 2 до 5 включно, то збільшити його на 10. Якщо воно від 7 до 40, то зменшити на 100. Якщо воно не більше 0 або більше 3000, то збільшити в 3 рази (тобто помножити на 3). Інакше обнулити це число.
58. Користувач вводить два числа. Якщо вони не рівні 10 і перше число парне, то вивести їх суму, інакше вивести їх добуток.
59. Користувач вводить три числа. Якщо всі числа більше 10 і перші два числа діляться на 3, то вивести Yes, інакше No.
60. Користувач вводить три числа. Знайти суму тих чисел, які діляться на 5. Якщо таких чисел немає, то вивести Error.
61. Дано три числа. Знайдіть найбільше з них.
62. Дано три числа. Знайдіть ті два з них, сума яких найбільша.
63. Користувач вводить чотири числа. Знайдіть найбільше парне число серед них. Якщо воно не існує, виведіть фразу "not found"

64. Дано три числа. Написати "Yes", якщо серед них є однакові.
65. Дано три числа. Написати "Yes", якщо можна взяти якісь два з них і в сумі отримати третє.
66. Дано чотири числа, якщо перші два числа більше 5, третє число ділиться на 6, четверте число не ділиться на 3, то вивести Yes, інакше No.
67. Дано два числа. Якщо хоча б одне з них більше 30, то вивести Yes, інакше No.
68. Дано три числа. Якщо рівно два з них менше 5, то вивести Yes, інакше вивести No.
69. Дано три числа. Знайти кількість додатних чисел (більших за нуль) серед них.
70. Робот може переміщатися в чотирьох напрямках («11» - північ, «12» - захід, «13» - південь, «14» - схід) і приймати три цифрові команди: «0» - продовжувати рух, «1» - поворот ліворуч, «-1» - поворот праворуч. Дано число (11, 12, 13 або 14) - вихідний напрямок руху робота і число N (0, 1 або -1) - послана йому команда. Вивести напрямок руху робота після виконання отриманої команди (тобто північ, захід, південь чи схід).
71. Дана дата із трьох чисел (день, місяць і рік). Вивести Yes, якщо така дата існує (наприклад, 12 02 1999 - Yes, 22 13 2001 - No). Вважати, що в лютому завжди 28 днів.
72. Дано дві дати, кожна з яких складається з трьох чисел (день, місяць і рік). Вивести Yes, якщо перша дата менша за другу, інакше вивести No.
73. Дано чотиризначне число. Чи вірно, що цифри в ньому розташовані по спадаючій? Наприклад, 4311 - ні, 4321 - так, 5542 - ні, 5631 - ні, 9871 - так.
74. Дано тризначне число. Переставте першу і останню цифри.
75. Дано чотиризначне число. Треба визначити, чи має воно однакові цифри, та вивести на екран результат перевірки.
76. Дано п'ятизначне число. Цифри на парних позиціях обнулити. Наприклад, з 12345 виходить число 10305.
77. Дано два тризначні числа. Знайдіть шестизначне число, утворене з двох даних чисел шляхом дописування другого числа до першого справа.

78. Дано чотиризначне число. Якщо воно читається зліва направо і справа наліво однаково, то вивести Yes, інакше No.
79. Дано чотиризначне число. Переставте місцями цифри так, щоб спочатку опинилися цифри, менші за п'ять.
80. Дано два тризначні числа. Отримайте нове число приєднанням другого числа праворуч до першого без останніх цифр у кожного. Наприклад, 123 і 456. Відповідь: 1245.
81. Дано чотиризначне число. Поміняйте місцями найменшу і найбільшу цифри.

ЦИКЛИ

82. Виведіть на екран 10 разів фразу "Ласкаво просимо!"
83. Виведіть на екран n разів фразу "Мовчання - золото". Число n вводить користувач.
84. Виведіть на екран прямокутник з нулів. Кількість рядків вводить користувач, кількість стовпців дорівнює 5.
85. Вивести на екран фігуру із зірочок:

    ****
    ****
    ****
    ****

    (квадрат з n рядків , в кожному рядку n зірочок).
86. Виведіть на екран числа 1, 2, 3, 4, ...20.
87. Вивести на екран ряд чисел 1001, 1004, 1007, ...1025.
88. Вивести на екран числа 100, 96, 92... до останнього позитивного включно.
89. Виведіть на екран числа 1.2, 1.4, 1.6, ...2.8.
90. Виведіть такі рядки. Перший: 25 25.5 24.8.
    Другий: 26 26.5 25.8. Останній рядок: 35 35.5 34.8.
91. Користувач вводить курс долара в національній валюті. Показати таблицю цін 1 $, 2 $, ..., 100 $ у національній валюті.
92. Для даного n знайти суму 1 + 2 + 3 + ... + n.
    Наприклад, для n = 10 відповідь дорівнює 55.
93. Знайти суму 10 + 11 + 12 + 13 + ... + 88. (Відповідь: 3871)
94. Знайти добуток $5 \cdot 6 \cdot 7 \cdot \ldots \cdot 13$.
95. Знайти суму 1 + 4 + 7 + 11 + ... + 112.
96. Знайти суму 23 + 34 + 45 + ... + 910.

97. Вивести на екран сто перших сум виду 1 + 2 + 3 + ... + n.
98. Знайдіть суму квадратів перших n чисел.
99. Знайдіть суму 1 + 12 + 13 + ... + 1n.
100. Почавши тренування, лижник в перший день пробіг 10 км. Кожен наступний день він збільшував пробіг на 10% від пробігу попереднього дня. Визначте: а) пробіг лижника за другий, третій, ...десятий день тренувань; б) який сумарний шлях він пробіг за перші 7 днів тренувань; в) сумарний шлях за n днів тренувань; г) в який день йому слід припинити збільшувати пробіг, якщо він не повинен перевищувати 80 км?
101. Вивести числа від 1000 до 9999 такі, в яких всі цифри різні.
102. Вивести на екран числа від 1000 до 9999 такі, в яких серед цифр немає 5 і 6.
103. Вивести всі п'ятизначні числа, які діляться на 2, в яких середня цифра непарна, і сума всіх цифр ділиться на 4.
104. Вивести на екран числа від 1000 до 9999 такі, в яких є трійка.
105. Знайдіть 3-значні числа, що дорівнюють сумі кубів своїх цифр.
106. Скільки існує чотиризначних чисел, які в 600 разів більше суми своїх цифр?
107. Виведіть на екран рядки (в останньому рядку n зірочок):

```
*
**
***
****
*****
```

108. Виведіть на екран такі рядки:

```
*******
****
*******
```

(всього n рядків, зірочок або 7, або 4 по черзі)

109. Вивести на екран:

```
AAABBBAAABBBAAABBB
BBBAAABBBAAABBBAAA
AAABBBAAABBBAAABBB
```

(таких рядків n, у кожному рядку m трійок AAA).

110. Виведіть на екран квадрат з нулів і одиниць, причому нулі знаходяться тільки на діагоналі квадрата. В квадраті сто цифр.

111. Вивести на екран:
AAAAAAAAAAAAAAA
ABBBBBBBBBBBBBA
ABBBBBBBBBBBBBA
ABBBBBBBBBBBBBA
AAAAAAAAAAAAAAA
(кількість рядків вводить користувач, ширина прямокутника в два рази більше висоти).
112. Вивести на екран 20 рядків. У рядках з парними номерами вивести по 10 чисел, рівних номеру рядка. У рядках з непарними номерами вивести десять одиниць.
113. Виведіть на екран таблицю множення для чисел від 1 до 10.
114. Скільки цілих чисел від a до b включно, які діляться на 12?
115. Користувач вводить ненульові числа до тих пір, поки не введе нуль. Знайдіть суму цих чисел.
116. Користувач вводить ненульові цілі числа до тих пір, поки не введе нуль. Знайдіть кількість парних чисел, які він ввів.
117. Знайдіть чотиризначні числа, сума цифр яких дорівнює 15.
118. Дано число. Знайдіть кількість парних цифр.
119. У числі знайдіть кількість цифр, які більше 3, але менше 8.
120. Для даного числа знайдіть число, цифри якого записані в зворотному порядку.
121. Дано число k. Визначте, чи існує таке число n, що 1 + 2 + 3 + ... + n = k.
122. Дано n цегли. Ви і комп'ютер ходите по черзі. За один хід можна взяти 1, 2 або 3 цегли. Програв той, кому немає чого брати. Реалізуйте гру з комп'ютером. Комп'ютер ходить випадково (без аналізу виграшної стратегії), та якщо у нього є хід, який є останнім для його виграшу, то він його здійснює.
123. Комп'ютер загадує число від 1 до 100. Користувач пробує відгадати. Після кожної невдалої спроби комп'ютер повідомляє, менше або більше задумане число.

СПИСКИ

124. Додавання елемента в початок списку.
125. Додавання елемента в кінець списку.
126. Показ всіх елементів списку.

127. Видалення всіх елементів списку.
128. Визначення кількості елементів списку.
129. Перевірка списку на порожнечу.
130. Видалення першого елемента.
131. Видалення останнього елемента.
132. Пошук даного значення в списку.
133. Пошук найбільшого і найменшого значень у списку.
134. Видалення елемента списку з даними значенням.
135. Видалення всіх елементів списку з даними значенням.
136. Зміна всіх елементів списку з даним значенням на нове.
137. Визначення, чи є список симетричним.
138. Визначення, скільки різних значень міститься в списку.
139. Видалення зі списку елементів, значення яких вже зустрічалися в попередніх елементах.
140. Зміна порядку елементів на зворотний.
141. Заповнити масив нулями, крім першого і останнього елементів, які повинні дорівнювати одиниці.
142. Заповнити масив нулями та одиницями, при цьому дані значення чергуються, починаючи з нуля.
143. Заповнити масив послідовними непарними числами, починаючи з одиниці.
144. Сформувати зростаючий масив із парних чисел.
145. Сформувати регресний (значення елементів зменшуються) масив із чисел, які діляться на 3.
146. Заповнити масив заданої довжини різними простими числами. Натуральне число, яке більше одиниці, називається простим, якщо воно ділиться тільки на себе і на одиницю.
147. Створити масив, кожен елемент якого дорівнює квадрату свого номера.
148. Створити масив, на парних місцях в якому стоять одиниці, а на непарних місцях - числа, що дорівнюють залишку від ділення їхнього номера на 5.
149. Заповніть масив випадковим чином нулями та одиницями так, щоб кількість одиниць було більше кількості нулів.
150. Створіть масив, в якому кількість від'ємних чисел дорівнює кількості додатних, і додатні числа розташовані на випадкових місцях у масиві.

АНАЛІЗ ЕЛЕМЕНТІВ МАСИВА

151. Визначити, чи містить масив дане число x.
152. Знайти кількість парних чисел у масиві.
153. Знайти кількість чисел у масиві, які діляться на 3, але не діляться на 7.
154. Визначте, яких чисел у масиві більше: які діляться на перший елемент масиву або які діляться на останній елемент масиву.
155. Знайдіть суму і добуток елементів масиву.
156. Знайдіть суму парних чисел масиву.
157. Знайдіть суму непарних чисел масиву, які не перевищують 11.
158. Знайдіть суму чисел масиву, які розташовані до першого парного числа масиву. Якщо парних чисел у масиві немає, то знайти суму всіх чисел за винятком крайніх.
159. Знайдіть суму чисел масиву, які стоять на парних місцях.
160. Знайдіть суму чисел масиву, які стоять на непарних місцях і при цьому перевищують суму крайніх елементів масиву.
161. Знайти найбільший елемент масиву.
162. Знайдіть суму найбільшого і найменшого елементів масиву.
163. Знайдіть кількість елементів масиву, які відрізняються від найбільшого елемента не більше, ніж на 10%.
164. Знайдіть найменший парний елемент масиву.
165. Серед елементів з непарними номерами знайдіть найбільший елемент масиву, який ділиться на 3.
166. Дано масив і число P. Знайдіть два різних числа в масиві, сума яких максимально близька до P.
167. Дано масив. Знайдіть 2 сусідні елементи, сума яких мінімальна.
168. Дано масив. Знайдіть три послідовних елемента в масиві, сума яких максимальна.
169. В даному масиві знайти кількість чисел, сусіди яких відрізняються більш ніж у 2 рази.
170. Знайдіть кількість чисел, кожне з яких дорівнює сумі квадратів своїх сусідів і при цьому не є найбільшим у масиві.
171. Перевірте, чи містить масив з n чисел усі числа від 1 до n.
172. Перевірте, чи є даний масив зростаючим або спадним.
173. Знайдіть кількість різних елементів даного масиву.
174. Визначте кількість змін знаків елементів масиву.
175. У масиві знайти максимальну кількість однакових елементів.

176. Знайдіть в масиві елемент, який найчастіше зустрічається.
177. Напишіть програму, яка вводить з клавіатури непорожній масив цілих чисел і виводить число локальних максимумів (елемент є локальним максимумом, якщо він не має сусідів, більших за себе).
178. У даному масиві знайти два найменших елемента.
179. Визначте, чи є в масиві елементи, що повторюються.
180. У даному масиві знайдіть найбільшу серію елементів, що розміщені поспіль в порядку зростання.

ПЕРЕТВОРЕННЯ МАСИВУ

181. У масиві замінити всі числа, що більші за вказане число, на середнє арифметичне всіх чисел масиву.
182. Дано масив. Замінити всі числа, менші за останній елемент масиву, на перший елемент.
183. Поміняти місцями найбільший та найменший елементи масиву.
184. Знайти найбільший парний елемент масиву і поміняти місцями з найменшим непарним елементом. Якщо одного з цих елементів немає, то всім елементам масиву надати значення - нуль.
185. Замінити кожен елемент масиву з парним номером на наступний елемент.
186. Видалити в масиві перший і останній елементи.
187. Видалити в масиві всі числа, які повторюються більше 2 разів.
188. Знайти в масиві всі серії однакових елементів, що розміщені поспіль, і видалити з них всі елементи крім одного.
189. Видалити в масиві всі найбільші елементи.
190. Переставити елементи масиву в зворотному порядку.
191. Дано масив A. Сформувати новий масив B такого ж розміру, щоб елемент B[K] дорівнював сумі перших елементів масиву A до номера K включно.
192. В даному масиві знайти всі нульові елементи і замінити їх разом з сусідніми елементами на 3.
193. Дано 2 масиви. Сформувати третій масив, що складається з тих елементів, які: а) є в обох масивах; б) є тільки в одному.
194. Дано два масиви. Визначте, чи існують в першому масиві такі два елементи, що їх сума дорівнює сумі будь-яких трьох елементів другого масиву.

195. Дано масив, в якому кількість від'ємних елементів дорівнює кількості додатних. Поміняйте місцями перший від'ємний і перший додатний, другий від'ємний і другий додатний і так далі.
196. Дано масив. Сформувати новий масив, в якому йдуть спочатку від'ємні елементи, потім нулі, потім додатні.
197. Дано масив. Переставте його елементи випадковим чином.
198. У даному масиві кожен елемент дорівнює 0, 1 або 2. Переставити елементи масиву так, щоб спочатку розміщувалися всі нулі, потім всі одиниці і, нарешті, всі двійки. Додатковий масив не використовувати.
199. Визначити кількість повторень кожної з цифр 0, 1, 2, ...9 в числі n, де n - натуральне число, n> 100.
200. Дано масив A розміру N. Сформувати новий масив B того ж розміру за таким правилом: елемент B[K] дорівнює середньому арифметичному елементів масиву A з номерами від K до N.
201. Дано цілочисельний масив. Збільшити всі парні числа, що містяться в масиві, на початкове значення першого парного числа. Якщо парні числа відсутні, то залишити без змін.
202. Дано масив. Замінити кожен елемент масиву на середнє арифметичне цього елемента і його сусідів.
203. З елементів масиву A довжиною 2n, отримати масиви B і C. Вибрати в масиві A два найбільш близьких за значенням елемента, менший з них помістити в масив B, а більший - в C. Виключити з розгляду в масиві A ці елементи і продовжити вибір з решти елементів.
204. Дано масив. Після кожного негативного елемента масиву вставити елемент з нульовим значенням.
205. Дано результати щоденних вимірювань кількості опадів. Протягом якого з тижнів, рахуючи з початку періоду вимірювань, випала найбільша кількість опадів?
206. Заданий масив, що складається з невід'ємних чисел. Знайдіть в ньому індекс елемента, для якого сума елементів, що стоять перед ним, найменше відрізняється від суми елементів, що стоять після нього.
207. У грі такі правила. В один ряд лежать 25 монет. За хід дозволяється брати 1 або 2 монети, що лежать поруч. Програє той, у кого немає, що брати. Реалізувати гру з комп'ютером.

208. Імітувати перетасування колоди карт. Кожне тасування складається з трьох етапів: розбиття колоди на дві підколоди, поділ кожної підколоди на частини, перемішування обраних частин, об'єднання всіх карт в одну колоду.
209. Гра починається з числа 0. За хід можна додавати до наявного числа будь-яке число від 1 до 10. Виграє той, хто першим отримає число 100. Реалізувати можливість гри з комп'ютером.

РЯДКИ

210. Користувач вводить англійську літеру, вивести наступні три за алфавітом. Якщо алфавіт закінчився, то вивести циклічно з початку алфавіту, тобто якщо z, то abc. Виводити тільки маленькі літери. Користувач може ввести велику літеру.
211. Вивести англійський алфавіт по 5 літер у рядку.
212. Вивести квадрат 7 на 7 з випадкових літер.
213. Користувач вводить позитивне ціле число. Зашифрувати кожну цифру серією з літер (конкретний принцип складання серії літер розробити самостійно).
214. Згенеруйте рядок символів довжиною від 3 до 10, в якій рівно два символа "!"
215. Згенерувати пароль. Вимоги: довжина від 6 до 20 символів, має бути рівно 1 символ підкреслення, хоча б 2 великі літери, не більше 5 цифр, будь-які 2 цифри поспіль неприпустимі.
216. Дано рядок. Вивести його три рази через кому і показати кількість символів в ньому.
217. Дано рядок. Вивести перший, останній і середній символ.
218. Дано рядок. Вивести перші три символа і останні три символа, якщо довжина рядка більше 5. Інакше вивести перший символ стільки разів, як довжина рядка.
219. Сформувати рядок з 10 символів. На парних позиціях повинні знаходиться парні цифри, на непарних позиціях - літери.
220. Дано рядок. Показати номери символів, які збігаються з останнім символом рядка.
221. Дано рядок. Показати 3, 6, 9,... (і так далі) символи.
222. Дано рядок. Визначте загальну кількість символів '+' і '-' в ньому. А також скільки таких символів, після яких йде нуль.

223. Дано рядок. Визначте, який символ в ньому зустрічається раніше: 'x' або 'w'. Якщо якогось із символів немає, вивести повідомлення про це.
224. Дано два рядки. Вивести більший за довжиною рядок стільки разів, на скільки символів відрізняються рядки.
225. Дано рядок. Якщо він починається з 'abc', то замінити їх на 'www', інакше додати в кінець рядка 'zzz'.
226. Дано рядок. Якщо його довжина більше 10, то залишити в рядку тільки перші 6 символів, інакше доповнити рядок символами 'o' до довжини 12.
227. Дано рядок. Розділити рядок на фрагменти по три символа поспіль. У кожному фрагменті середній символ замінити на випадковий символ, який не збігається ні з одним з символів цього фрагмента. Показати фрагменти.
228. Дано рядок. Замінити кожен парний символ або на 'a', якщо символ НЕ дорівнює 'a' чи 'b', або на 'c' в іншому випадку.
229. В даному рядку знайти кількість цифр.
230. Визначити, чи містить рядок тільки символи 'a', 'b', 'c', чи ні.
231. Замініть у рядку всі входження 'word' на 'letter'.
232. Видаліть у рядку всі літери 'x', за якими йдуть літери 'abc'.
233. Видаліть у рядку всі 'abc', за якими йде цифра.
234. Знайдіть кількість входжень 'aba' в рядок.
235. Видалити в рядку всі зайві пропуски, тобто серії пропусків, що йдуть поспіль, замінити на одиничні пропуски. Крайні пропуски в рядку видалити.
236. Дано рядок, що складається зі слів, розділених символами, які перераховані в другому рядку. Показати всі слова.
237. Дано текст. Знайдіть в ньому найбільшу кількість цифр, що йдуть поспіль.
238. Дано текст. Знайти суму наявних в ньому цифр.
239. Дано текст. Знайти слова, що складаються з цифр, і суму чисел, які утворюють ці слова.
240. Дано текст. Знайдіть найбільшу кількість пропусків у ньому, що йдуть поспіль.
241. Дано два слова. Знайдіть тільки ті символи, які зустрічаються в обох словах лише один раз.
242. Дано рядок. Замінити всі символи 'a' і 'b' на 'A' і 'B' відповідно.

243. Дано текст. Сформувати рядок із символів, розташованих між першою і другою комами даного тексту.
244. Дано два рядки. Визначте, чи міститься менший за довжиною рядок у більшому.
245. Дано два рядки. Визначте, чи можна з деяких символів першого рядка скласти другий рядок.
246. Дано два рядки. Визначте, чи можна з деяких символів першого рядка і всіх символів другого рядка скласти новий рядок, в якому кожний символ зустрічається рівно два рази.
247. Видалити в рядку всі цифри.
248. Видаліть у рядку всі символи "!"
249. У рядку знайдіть всі серії пропусків, що йдуть поспіль, і замініть кожну серію на один пропуск.
250. Дано рядок, що складається зі слів, розділених пропусками. Визначте кількість слів у рядку.
251. У рядку вставити після кожного символа 'a' символ 'b'.
252. Дано 2 рядки. Видалити в 1 рядку 1 входження 2 рядка.
253. Рядок складається зі слів, розділених одним або декількома пропусками. Знайдіть слово максимальної довжини.
254. Дано email у рядку. Визначити, чи є він коректним (наявність символа @ і точки, наявність не менше двох символів після останньої точки і т. д.).
255. Дано натуральне число. Отримати рядок, в якому трійки цифр цього числа розділені пропуском, починаючи з правого кінця. Наприклад, число 1234567 перетворюється в 1 234 567.
256. Дано рядок. Вставити після кожного символа пропуск.
257. Дано рядок. Вставити після кожного символа два випадкових.
258. Дано рядок. Якщо в рядку більше трьох різних символів, то видалити з рядка три будь-які різні символа, інакше додати в рядок у довільних місцях нові елементи, щоб кількість різних елементів була більше трьох.
259. Написати алгоритм генерації пароля.
260. Написати генерацію рядків довжиною 10, причому перші 4 символа - цифри, наступні 2 символа - різні літери, ще 4 символа - нулі або одиниці, причому одна 1 точно є.
261. Рядок складається зі слів, розділених одним або декількома пропусками. Поміняйте місцями найбільше і найменше слова.

262. Написати генерацію рядків довжиною 12, перші 5 символів якої - парні цифри, наступні 5 символів - букви 'a' - 'z', наступні 2 символа - "AB", якщо серед перших п'яти символів рядка є цифра 8, та "XY" - якщо немає.
263. Дано два рядки. Для кожного слова першого речення визначте кількість його входжень у друге речення.
264. Дано рядок, який містить натуральні числа, знаки чотирьох арифметичних дій (додавання, віднімання, множення, ділення) і дужки. Обчисліть значення виразу.
265. Дано рядок, в якому немає початкових і кінцевих пропусків. Необхідно змінити його так, щоб довжина рядка стала дорівнювати заданій довжині, що більша за поточну довжину рядка. Це слід зробити шляхом вставки між словами додаткових пропусків. Кількість пропусків між окремими словами не повинна відрізнятися більше ніж на один пропуск (тобто пропуски додаються рівномірно).
266. Дано рядок, що містить повне ім'я файлу (наприклад, 'c:\WebServers\home\testsite\www\myfile.txt').
Виділіть з цього рядка ім'я файлу без розширення.
267. Дано рядок, що містить літери і дужки '(', ')', '[', ']', '{', '}'. Якщо дужки розставлені правильно (тобто кожній дужці, що відкриває, відповідає той же вид дужки, що закриває), то вивести 'yes', інакше вивести номер першої помилкової дужки або -1, якщо відсутня дужка, що закриває.
268. Дано три рядки. Замінити всі входження другого рядка в перший на третій рядок.
269. Дано текст. Знайдіть у ньому всі числа, що оточені пропусками, і додайте перед ними '<' і після них '>'.
270. Дано текст. Деякі його фрагменти виділені групами символів ##. Замінити виділення групами символів '<' і '>)'. Приклад: 'Це ##приклад## для завдання ##на## рядки' перетворюється в 'Це <приклад> для завдання <на> рядки'.
271. Дано текст і список слів. Знайти в тексті всі слова, кожне з яких відрізняється від деякого слова зі списку однією літерою, і виправити такі слова на слова зі списку.
272. Дано текст. Замінити всі цифри відповідними словами.

273. Дано текст. Замінити всі входження найбільшої цифри її словесним написанням.
274. Виключити з рядка групи символів, розташовані між символами «/*», «*/» включаючи межі. Передбачається, що немає вкладених дужок.
275. Перевірити, чи витримується в заданому тексті баланс круглих дужок, що відкривають і закривають, тобто, чи можна встановити взаємно однозначну відповідність дужок, що відкривають і закривають, причому дужка, що відкриває, завжди передує відповідній дужці, що закриває.
276. Написати програму, що реалізує процедуру видалення K символів з позиції номер N із рядка S.
277. Визначити найдовше спільне слово у двох заданих реченнях.
278. Дано три речення. Визначити найдовше зі слів першого речення, яке є в другому і третьому реченнях.
279. У заданому списку слів знайдіть найдовше слово, утворене іншими словами, що входять у список. Наприклад, для слів dog, cat, mouse, dogcat, mouseandcat відповіддю є слово dogcat.
280. Дано масив змінних у camelCase. Перетворити їх в snake_case.
281. Реалізуйте метод, який здійснює стиснення рядків на основі лічильника символів, що повторюються.
Наприклад, рядок aaabbcccc повинен перетворитися в a3b2c5.
Якщо стиснутий рядок виявився довшим за початковий, то результатом роботи алгоритму повинен бути початковий рядок.
Вирішити завдання з припущенням, що в заданому рядку немає цифр. Вирішити завдання для довільного рядка.