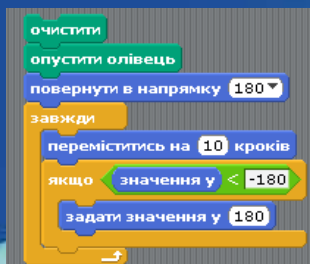


Алгоритми з розгалуженням.

Базова алгоритмічна структура розгалуження.

Складання та виконання алгоритмів з розгалуженням у середовищі Scratch



Розділ 3. Алгоритми з повторенням і розгалуженням

У цьому розділі ви дізнаєтеся про:

цикли (повторення) в природі, у повсякденному житті, під час вивчення навчальних предметів

«цикл з лічильником»

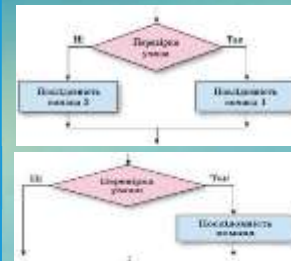
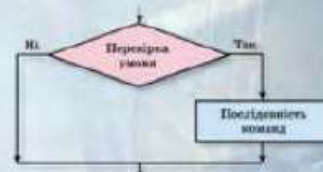
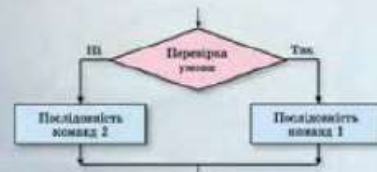
алгоритми з повторенням

висловлювання, істинні та хибні висловлювання

розгалуження

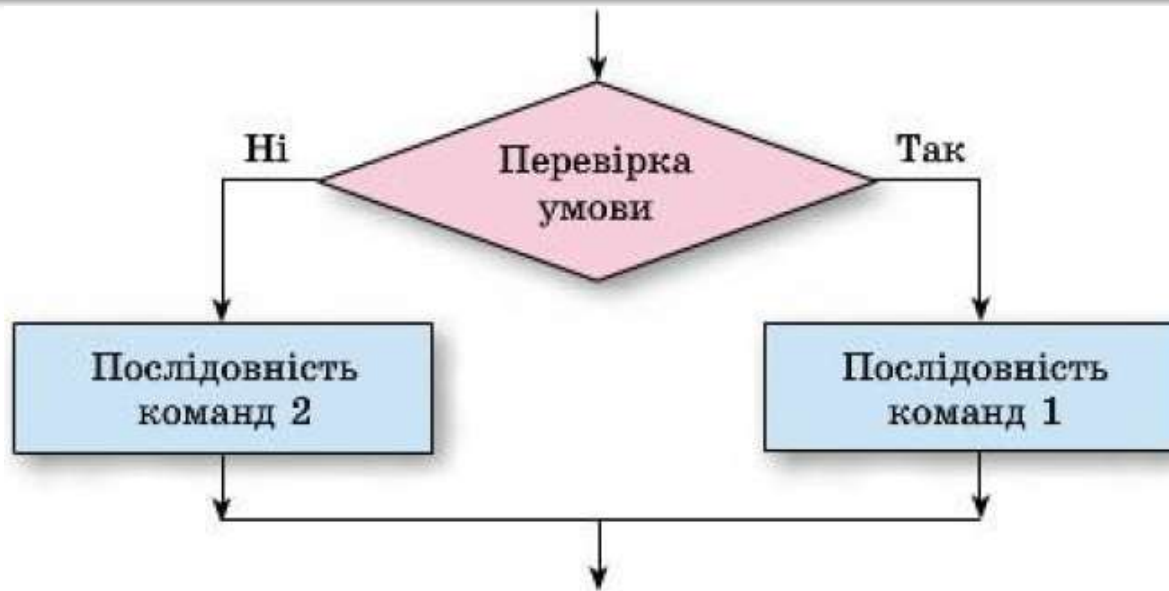
алгоритми з розгалуженням

«цикл з передумовою»



Повне розгалуження

В алгоритмах використовують розгалуження **двох видів**: **повне розгалуження** і неповне розгалуження.



Виконання **повного розгалуження** відбувається так: виконавець виконує команду перевірки умови: **якщо результат** виконання цієї команди **істина**, то виконавець виконує **послідовність команд 1**, після чого переходить до виконання першої команди наступного фрагмента алгоритму; **якщо ж результат** виконання цієї команди **хиба**, то виконавець виконує **послідовність команд 2**, після чого також переходить до виконання першої команди наступного фрагмента алгоритму.

Неповне розгалуження

В алгоритмах використовують розгалуження **двох видів**: повне розгалуження і **неповне розгалуження**.



Виконання **неповного розгалуження** відрізняється від виконання повного розгалуження тим, що **за результату** виконання команди перевірки **умови** **хиба** виконавець одразу переходить до **виконання першої команди наступного фрагмента** алгоритму.

Розгалуження в Scratch



У середовищі **Scratch** можна використати команду **якщо-інакше** для організації **повного розгалуження** і команду **якщо** для організації **неповного розгалуження**.



Складемо алгоритм, за яким виконавець рухатиметься **Сценою** вправо. Після того як виконавець дійде до правої границі **Сцени**, він повинен з'явитися біля лівої границі **Сцени** і продовжити рух вправо.

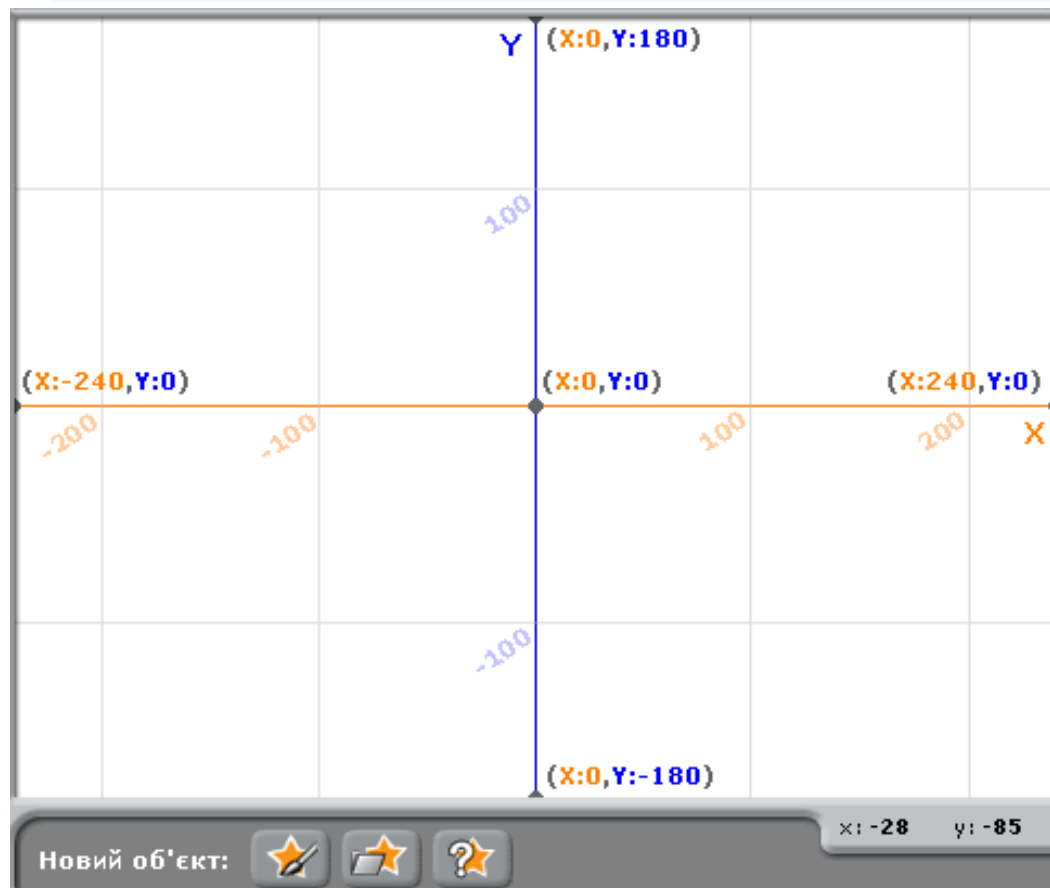


Розгалуження в Scratch



На **Сцені** введено **систему координат** для того, щоб положення виконавця на **Сцені** можна було задавати значеннями двох координат: **X** і **Y**.

(для виклику системи координат вибрати: **сцена** → **фони** → **імпортувати** → **xy-grid** → **Гаразд**)



За замовчуванням **довжина** видимої частини **Сцени 480 кроків** виконавця, а **ширина** — **360 кроків**.

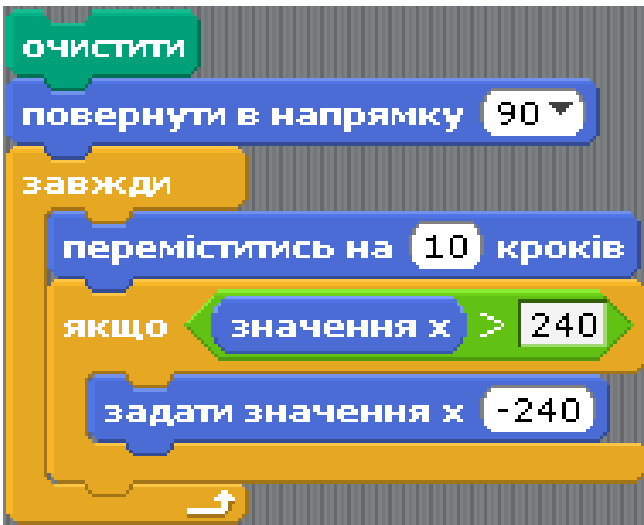
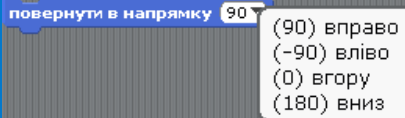
Тому якщо виконавець розміщується у видимої частині **Сцени**, то **абсциса** його положення **може змінюватися від -240 до 240**, а **ордината** - **від -180 до 180**.



Алгоритм для організації руху виконавця вправо

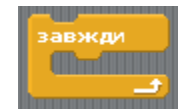
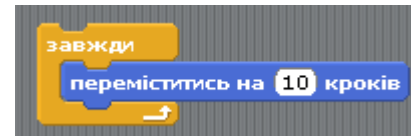
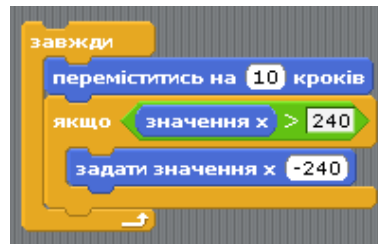


На початку алгоритму, наведеному на рисунку 3.22, використано команду з групи **Рух** для орієнтації виконавця на **Сцені** в потрібному напрямку **(90) вправо**.



Щоб виконавець багаторазово виконував команду **переміститися на 10 кроків**, її включено в тіло циклу **Завжди**. Команди тіла цього циклу виконуються доти, поки не буде натиснута кнопка **Зупинити все** .

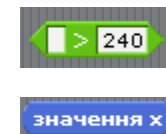
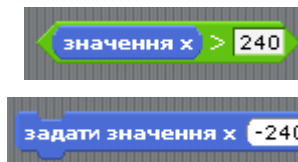
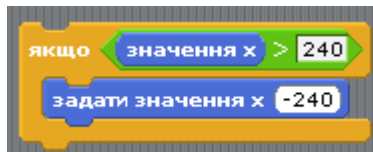
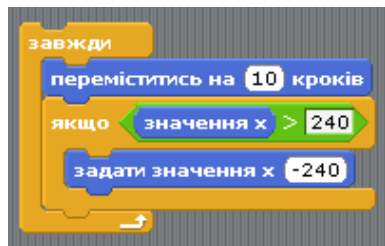
Рис.3.22



Алгоритм для організації руху виконавця вправо



У тілі цього циклу використано команду **розгалуження**



У ході кожного виконання в тілі циклу цієї команди виконується команда перевірки умови значення $x > 240$.

Якщо результат виконання цієї команди **Так** (тобто виконавець вийшов за праву границю **Сцени**), то виконується команда **задати значення $x -240$** і виконавець з'являється біля лівої границі **Сцени**. Після цього команди тіла циклу починають виконуватися в черговий раз і виконавець переміщується вправо.

Якщо результат виконання команди перевірки умови **Ні** (тобто виконавець ще не вийшов за праву границю **Сцени**), то команда **задати значення $x -240$** не виконується. Після цього команди тіла циклу починають виконуватися в черговий раз і виконавець переміщується вправо.





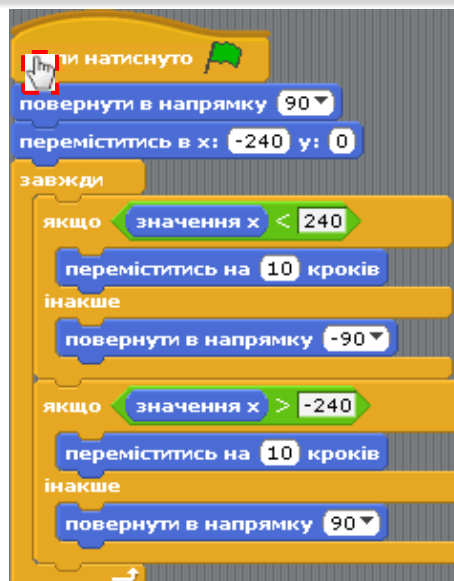
Працюємо з комп'ютером

Практична робота

«Складання та виконання алгоритмів з розгалуженням у середовищі **Scratch**»

1. Запустіть програму **Scratch**.
2. Складіть алгоритм, за яким виконавець рухається вправо, доходить до правої границі **Сцени**, повертається для руху вліво, рухається вліво, доходить до лівої границі **Сцени**, повертається для руху вправо і так далі.
3. Збережіть створений проект у своїй папці у файлі з іменем **практична 5.1**.

Приклад алгоритму



Увага!

Для того, щоб виконавець не перевертався під час руху необхідно натиснути кнопку  «**Приймати тільки з ліва на право**».





Працюємо з комп'ютером

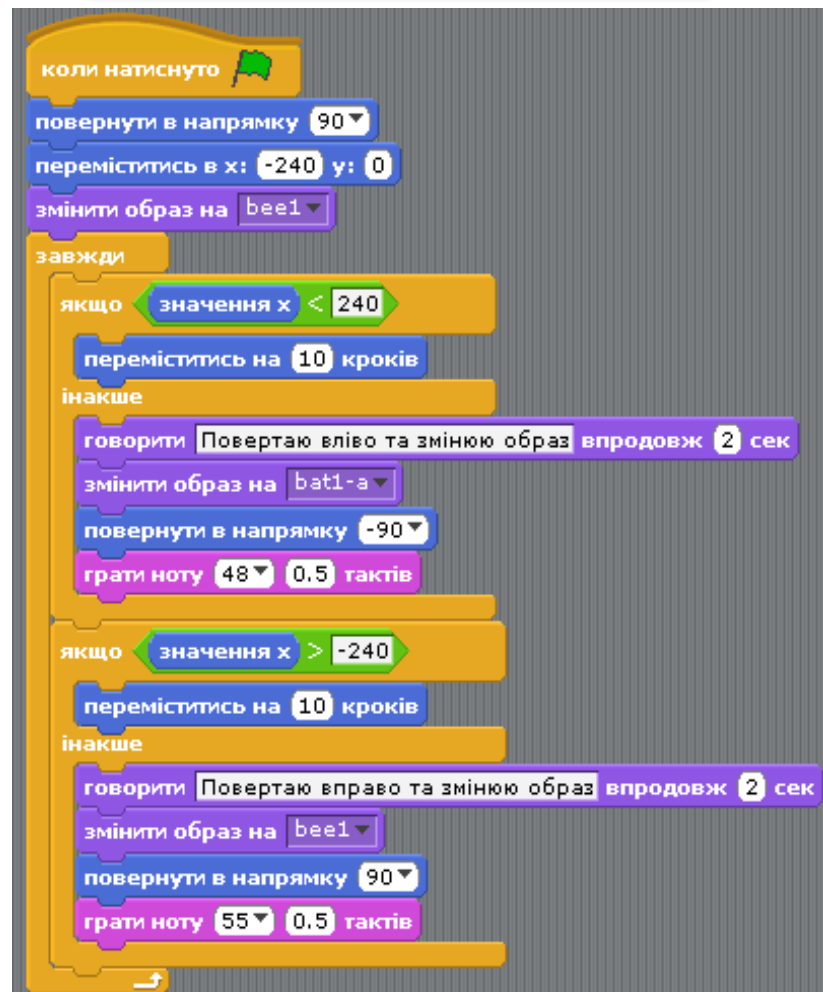
Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм

4. Додайте до складеного алгоритму команди для **змінення образу**, появи відповідного **текстового повідомлення** і **звукового сигналу** під час змінення напрямку переміщення.

5. Збережіть створений проект у своїй папці у файлі з іменем **практична 5.2.**



Приклад алгоритму





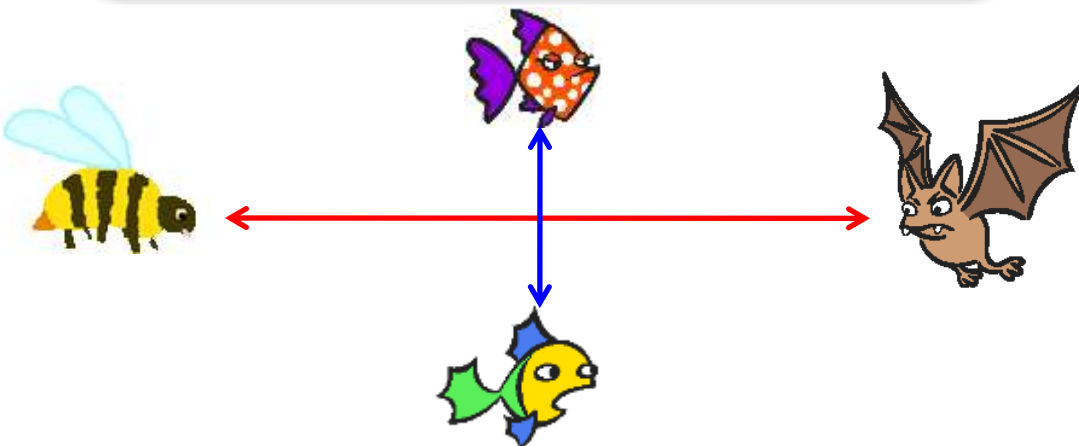
Працюємо з комп'ютером

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм

6. Додайте до складеного алгоритму команди, щоб виконавець рухався між границями **Сцени** в таких напрямках: **зліва направо, справа наліво, зверху вниз, знизу вверху.**

7. Збережіть створений проект у своїй папці у файлі з іменем **практична 5.3.**

8. Закрийте вікно програми **Scratch.**



Приклад алгоритму

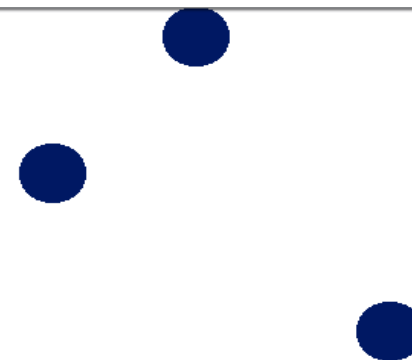
```

коли натисну клавішу space
  повернути в напрямку 90
  переміститись в x: -240 y: 0
  змінити образ на bee1
  завжди
    якщо значення x < 240
      переміститись на 10 кроків
    інакше
      говорити Правий край сцени, зміна образу впродовж 2 сек
      змінити образ на bat1-a
      повернути в напрямку -90
      грати ноту 48 0.5 тактів
    якщо значення x > -240
      переміститись на 10 кроків
    інакше
      говорити Лівий край сцени, зміна образу впродовж 2 сек
      змінити образ на fish1
      грати ноту 55 0.5 тактів
      повернути в напрямку 0
      переміститись в x: 0 y: -180
    якщо значення y < 180
      переміститись на 10 кроків
    інакше
      говорити Верх сцени, зміна образу впродовж 2 сек
      змінити образ на fish2
      повернути в напрямку 180
      грати ноту 65 0.5 тактів
    якщо значення y > -180
      переміститись на 10 кроків
    інакше
      говорити Низ сцени, зміна образу впродовж 2 сек
      змінити образ на bee1
      повернути в напрямку 90
      грати ноту 72 0.5 тактів
      переміститись в x: -240 y: 0
  
```



Задача 1. Виконавець рухається по сцені на якій безперервно падають сторонні об'єкти. Необхідно забезпечити зупинку(паузу) у русі виконавця, якщо він може зіткнутися з об'єктом.

Скрипт для руху об'єктів вверх-вниз



Скрипт виконавця

```

коли натиснуто [ ]
  повернути в напрямку 180
  завжди
    переміститись на 10 кроків
    чекати 0.1 секунд
    якщо доторкається границя ?
      повернути на 180 градусів
  
```

```

коли натиснуто [ ]
  завжди
    переміститись на 10 кроків
    наступний образ
    чекати 0.05 секунд
    якщо < відстань до Об'єкт2 < 100 > або < відстань до Об'єкт3 < 100 > або < відстань до Об'єкт4 < 100 >
      чекати 0.5 секунд
    якщо границя, відбити
  
```

Вправа 1. Пішохід.

Завдання. Склади проект, у якому виконавець *Пішохід* переходить дорогу з трьома смугами руху без пішохідного переходу, якщо відстань до автомобіля, що рухається, більша, ніж 100 кроків.

1. Сплануй створення проекту. Перевір, що в бібліотеці зображень можна дібрати зображення трьох автомобілів та людини. Додай ці об'єкти до проекту. Перейменуй їх на **Пішохід**, **Авто 1**, **Авто 2**, **Авто 3**.
2. Намалюй розмітку дороги на сцені проекту, скориставшись вбудованим графічним редактором Скретч, наприклад, як на малюнку:



Вправа 2. Гра з м'ячем.

Завдання. Склади проект, у якому виконавець переміщується до м'яча, що з'являється на ігровому полі випадково, або залишається на своєму місці, якщо відстань до м'яча більша, ніж 100 кроків.

1. Створи проект, у якому задай об'єкти та фон сцени, як показано на малюнку.



Число - Scratch

SCRATCH

Файл Редагувати Відправити допомога

Рух Керувати
Вигляд Датчики
Звук Оператори
Олівець Змінні

Об'єкт1
x: 50 y: 0 напрямок: 120

Скрипти Образи Звуки

коли натиснуто

запитати Ввести число A та чекати
надати A значення відповідь

запитати Ввести число B та чекати
надати B значення відповідь

якщо $A - B < 0$
говорити з'єднати A з'єднати менше B

якщо $A - B > 0$
говорити з'єднати A з'єднати більше B

якщо $A - B = 0$
говорити з'єднати A з'єднати дорівнює B

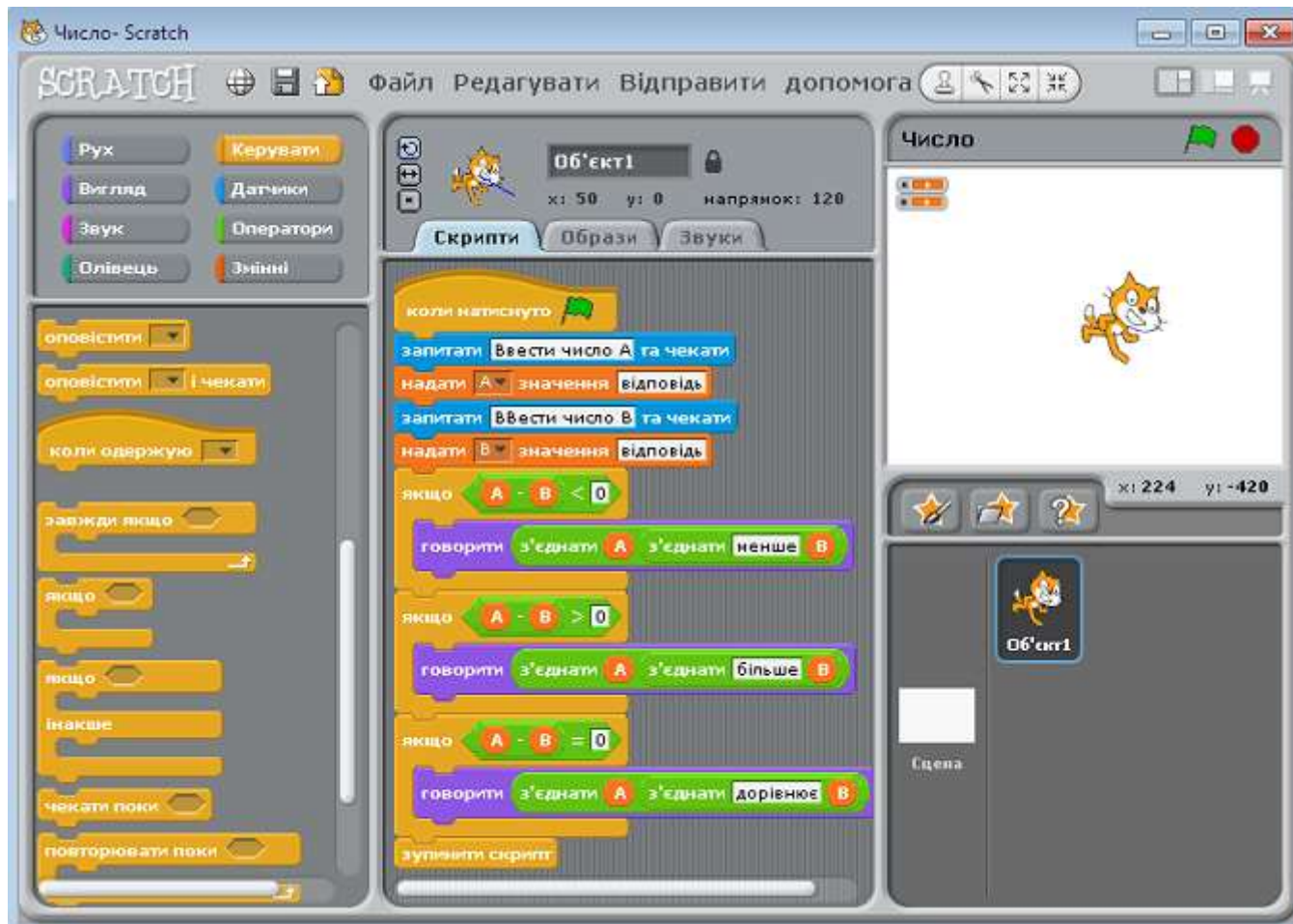
зупинити скрипти

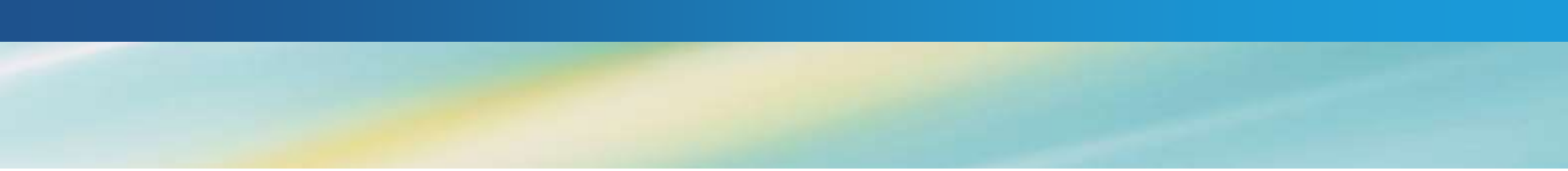
Число

x: 224 y: -420

Об'єкт1

Сцена





This area is intentionally left blank for content.

