



Програмуємо зі Скретчем



Укладач: Буланова І.М., вчитель інформатики, спеціаліст вищої категорії зі званням «Старший вчитель»

Буланова І.М.

Програмуємо зі Скретчем. Методична розробка для вчителів початкової школи та інформатики, що викладають «Сходинки до інформатики» 2 клас у початковій школі за підручником Г.В.Ломаковська, Г.О. Проценко, Й.Я.Ривкінд, Ф.М.Рівкінд. Скадовськ: Скадовська загальноосвітня школа І-ІІІ ступенів №2 Скадовської міської ради Херсонської області, 2014. – 20с.

.



Як встановити Скретч?

Викачати Scratch на свій комп'ютер і встановити встановимо його в системі.

Декілька підказок:

Сайт Scratch - <http://scratch.mit.edu>

Посилання на скачування Scratch знаходиться в самому низу головної сторінки сайту.

Відеоінструкція по встановленню за посиланням:

<http://www.nachalka.com/book/export/html/1398>

Скретч (Scratch) це нове середовище програмування, яке дозволяє дітям створювати власні анімовані інтерактивні історії, ігри і моделі.

У Скретчі можна грати з різними об'єктами, видозмінювати їх вигляд, переміщати їх по екрану, встановлювати форми взаємодії між об'єктами. Це об'єктно-орієнтоване середовище, в якому блоки програм збираються з різнокольорових цеглинок команд.



Що може Скретч?

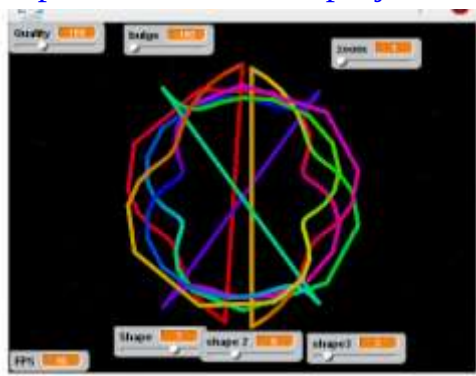
Придумувати, робити і запозичати

- 1.1 [Игры](#) и [мультфильмы](#)
- 1.2 [Музыкальные клипы](#)
- 1.3 [Занимательные \(учебные\) задачи](#)

З чого розпочати?

Подивитися приклади проектів на сайті <http://scratch.mit.edu>

<http://scratch.mit.edu/projects/10121154/>



<http://scratch.mit.edu/projects/13047114/>



Пограти з командами

Подивитися Перелік команд мови Скретч

Придумати сюжет для свого першого фільму і запропонувати його до обговорення - Скретч-історії

Зробити свій перший Скретч-фільм

У Скретч використовується метафора цеглинок Лего, з якої навіть найменші діти можуть зібрати прості конструкції. Діти можуть збирати свої програми-процедури з блоків так само, як вони збирали конструкції з різноколірної цегли.

З конструкцій і структур, що управляють, можуть бути зібрані різні агенти виконуючі прості інструкції.

Ми можемо узяти будь-який об'єкт і здійснити над ним дії:

- видозмінити його зовнішній вигляд за допомогою **фіолетової** команд-цегли зовнішнього вигляду;
- перемістити його за допомогою **синіх** команд цегли переміщення;
- помістити команди переміщення і видозміни всередину **помаранчевих** блоків управління;
- додати до блоків, що управляють, фіолетову цеглу звуків;
- і так далі

В результаті виконання простих команд може складатися складна модель, в якій взаємодіятимуть безліч об'єктів, наділених різними властивостями.

Перелік команд мови Скретч

- [1 Команды движения \(синий ящик\)](#)
- [2 Звуки \(лиловый ящик\)](#)
- [3 Графика \(фиолетовый ящик\)](#)
- [4 Управление \(желтый ящик\)](#)
- [5 Сенсоры \(голубой ящик\)](#)
- [6 Вычисления \(зеленый ящик\)](#)
- [7 Рисование \(темно-зеленый ящик\)](#)
- [8 Переменные \(оранжевый ящик\)](#)

Команди руху (синій ящик)

Команда	Призначення
move	пройти вперед указанное число шагов. Например: move(-10) – сделать десять шагов назад
turn()	повернуться - стрелочка указывает по часовой или против часов стрелки выполняется поворот. Например: turn(45) повернуться на 45 градусов (стрелочка указывает по или против часовой стрелки)
point in direction()	повернуться в указанном направлении. Например: point in direction(180)
point towards ()	повернуться в направлении другого существа или координат мышки. После команды всегда существует перечень объектов, которые в данный момент присутствуют в системе и на которые можно реагировать. В самом простом случае, когда других объектов нет, предлагается повернуться в сторону, где находится указатель мышки.
change x by ()	изменить положение по оси X на указанное число шагов. Например: change x by (-10) – объект смещается влево на 10 шагов
set x to ()	установить объект положение по оси x. Например: set x to (50)
go to x() y()	переместиться в точку с указанными координатами Например: go to x(70) y(-50)
glide x() y() in () secs	плавное перемещение в точку с указанными координатами за указанное время. На перемещение затратится время указанное в секундах. Например: glide x(90) y(90) in (20) secs – объект медленно (в течение 20 секунд) будет переползает в указанную точку.

go to ()-	перейти в точку, где расположен указатель мыши или другая фигура. Например: go to (mousepointer)
if on edge, bounce	- если попадаешь в край экрана, то отразись от него. Это очень полезно, если Вы не хотите потерять своего героя
x position	возвращает значение по оси X Например: set x (xposition)
y position	возвращает значение по оси y Например: set y (yposition)
direction -	возвращает направление Например: point in direction (-direction)

Звуки (лиловый ящик)

Команда	Назначение
play sound()	воспроизвести звук (можно выбрать звук). При этом звук можно выбрать в библиотеке – там этих звуков достаточно много. Так же как и библиотека картинок, библиотека звуков расширяется и к ней можно добавлять свои местные звуки в формате wav
set instrument()	выбрать инструмент, который будет играть. Инструментов в Скретче множество в несколько прокруток экрана
play drum () for () sec	
play note () for () sec	играть определенную ноту указанное количество времени (в секундах). Ноты записаны в цифрах, но против каждой цифры стоит ее звучание.
stop all sounds	Убрать все звуки

Графика (фиолетовый ящик)

Команда	Назначение
switch to costume ()	перейти к костюму (можно выбрать костюм героя)

Управление (желтый ящик)

Команда	Назначение
when clicked ()	когда нажмут на (зеленый флажок, на форму героя)
when () key pressed	запускает выполнение блока команд в ответ на нажатие выбранной клавиши. Позволяет передать управление на клавиатуру. Например: when (g) key pressed move(10)
when I received ()	запускает выполнение блока команд в ответ на полученное сообщение. Работает в сочетании с broadcast
wait()	команда ожидания. Параметр указывает сколько секунд следует ждать.
Forever	– блок команд, заключенных внутри конструкции будет выполняться постоянно. Например: forever (move(20) ; rt (45)) объект будет постоянно двигаться и поворачиваться, пока мы не нажмем кнопку Stop
If ()	– условие, при выполнении которого должны выполняться команды, заключенные внутри конструкции if Если не выполняется, то никаких действий

if () [] else []	– условие, при выполнении которого должны выполняться команды, заключенные внутри конструкции if . Если не выполняется, то нужно перейти к действиям внутри else
repeat()	повторение. Параметр указывает, сколько раз нужно повторить блоки команд, заключенные внутри блока repeat()
broadcast()	Передать сообщение. Переданное сообщение может запускать активность другого исполнителя. Работает в сочетании с when I received ()
forever if ()	Выполняется пока условие в if верно
wait until ()	Ждать пока не выполнится условие
stop script	Остановить выполнение программы для данного исполнителя
stop all	Остановить выполнение всех программ

Сенсоры (голубой ящик)

Команда	Назначение
mouse x	возвращает значение указателя мыши по оси x
mouse y	возвращает значение указателя мыши по оси y
touching color () ?	касается ли наш объект указанного цвета Например: touching color () ? –
Color (1) is over (2) ?	цвет 1 накладывается на цвет 2
Mouse down?	нажата ли управляющая клавиша мышки?
Touching (mouspointer) ?	касается ли наш объект мышки или другого существа касается ли наш объект цвета() соприкасается ли цвет() с цветом()

Вычисления (зеленый ящик)

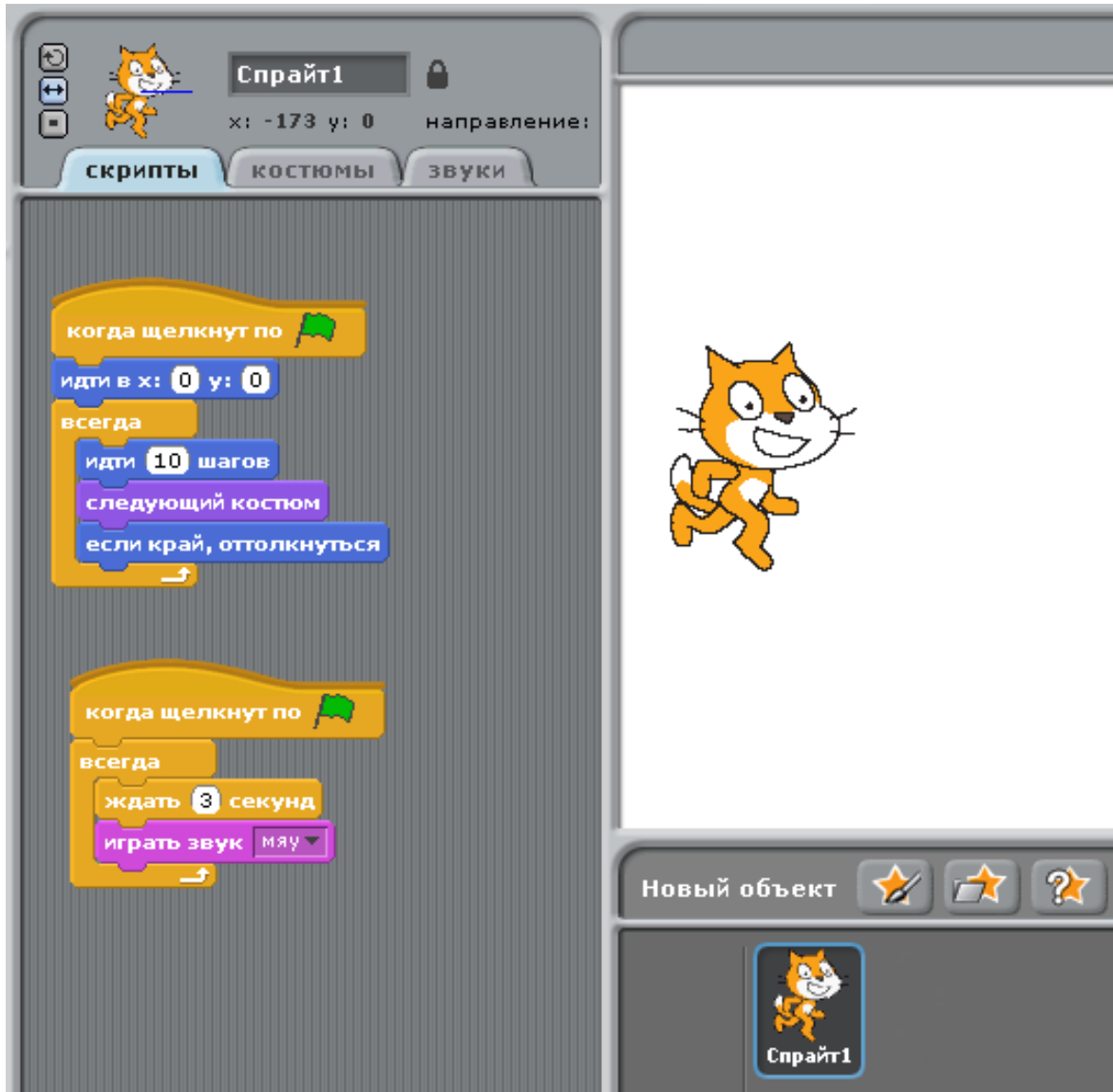
Команда	Назначение
() + ()	операция сложения
() – ()	вычитание
() * ()	умножение
() / ()	деление
() < ()	Сравнение больше, равно, меньше. Например: ((5) * (5)) < ((4) * (10)) – вполне разумное выражение, результат которого должен выдавать отрицание (false)
pick random () to ()	pick random () to () случайное число в интервале от и до. Например: pick random (0) to (10) возвратит 1 или 3, или 9 -
логические операторы And, Or, Not	Эти блоки содержат входные окошки, куда вставляются кирпичики сравнений. Например: AND () ()
abs()	
round()	Округленное значение

Рисование (темно-зеленый ящик)

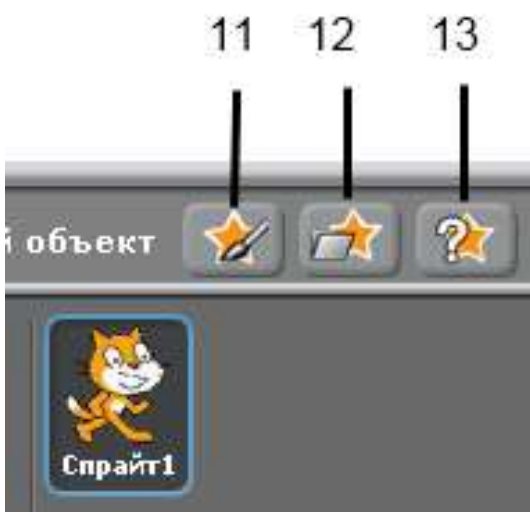
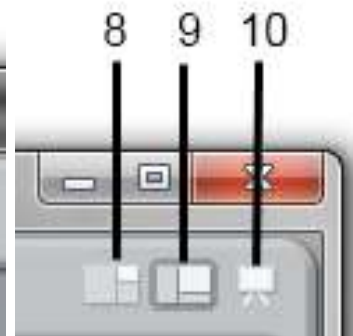
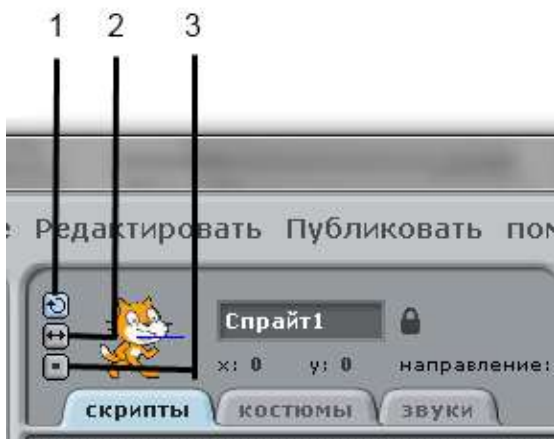
Команда	Назначение
clear	очистить экран от всех следов, которые на нем оставили объекты
pen down	опустить карандаш. После этой команды за движущимся объектом будет

Створюємо перший скрипт і досліджуємо середовище Скретч

Створимо свій скрипт, який керує кошеням.



Коли створите скрипт, проведіть дослідження:

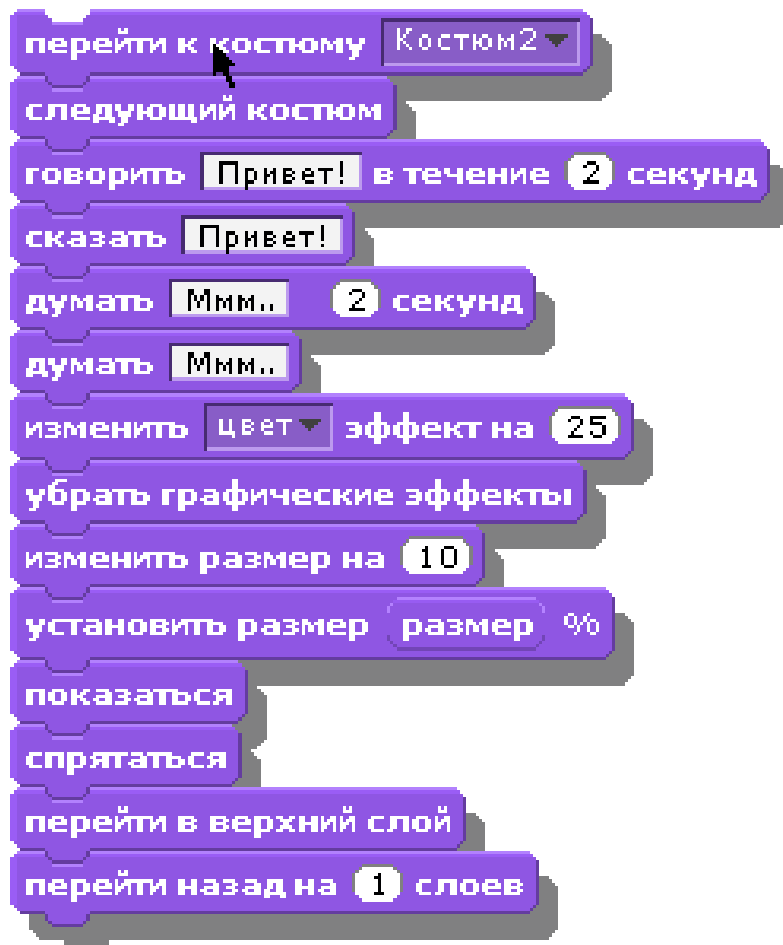


На малюнках вказані і пронумеровані кнопки.

Ваше завдання: З'ясувати, для чого потрібна кожна з них?

Цеглинки контролю, руху і зовнішнього вигляду

Зовнішній вигляд об'єкту(фіолетовий ящик)



Усі команди однозначні. Найчастіше вживаються команди "Змінити()" і "Прибрати графічні ефекти". Переходи між шарами найлегше пояснити на прикладі 2-го однакового спрайту кошеня. У одного з них пофарбуємо очі у блакитний колір. Якщо ми тепер попросимо одного з котенят міняти шар, то вийде проста мультиплікація - котеня мінятиме колір очей.



Можемо збільшити число об'єктів.

Синій ящик - Рух



В синьому ящику складені будівельні блоки, що управляють переміщенням об'єктів по екрану.

Центральна точка екрану відповідає координатам $X=0$, $Y=0$. Вправо від нульової точки X зростає і це область позитивних значень. Вліво - область негативних значень.

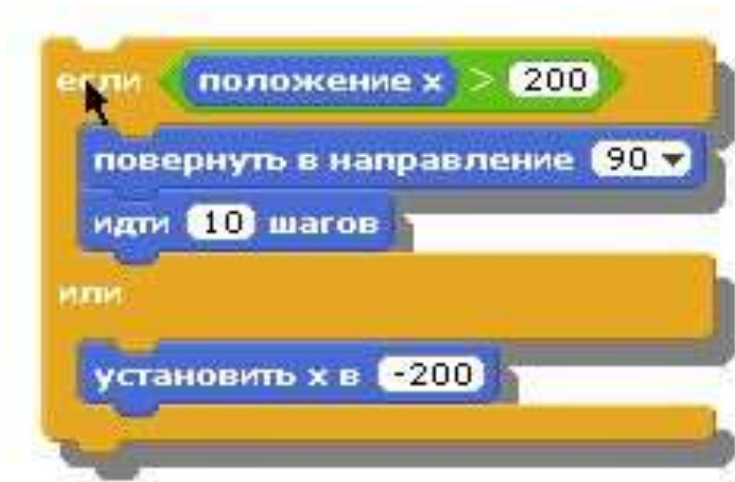
Навчимося використати блоки, які не є командами, а повертають значення - положення x , y і напрям. Простий приклад:

Ми хочемо, щоб кошеня послідовно переходило з нижнього правого кута в лівий верхній кут.

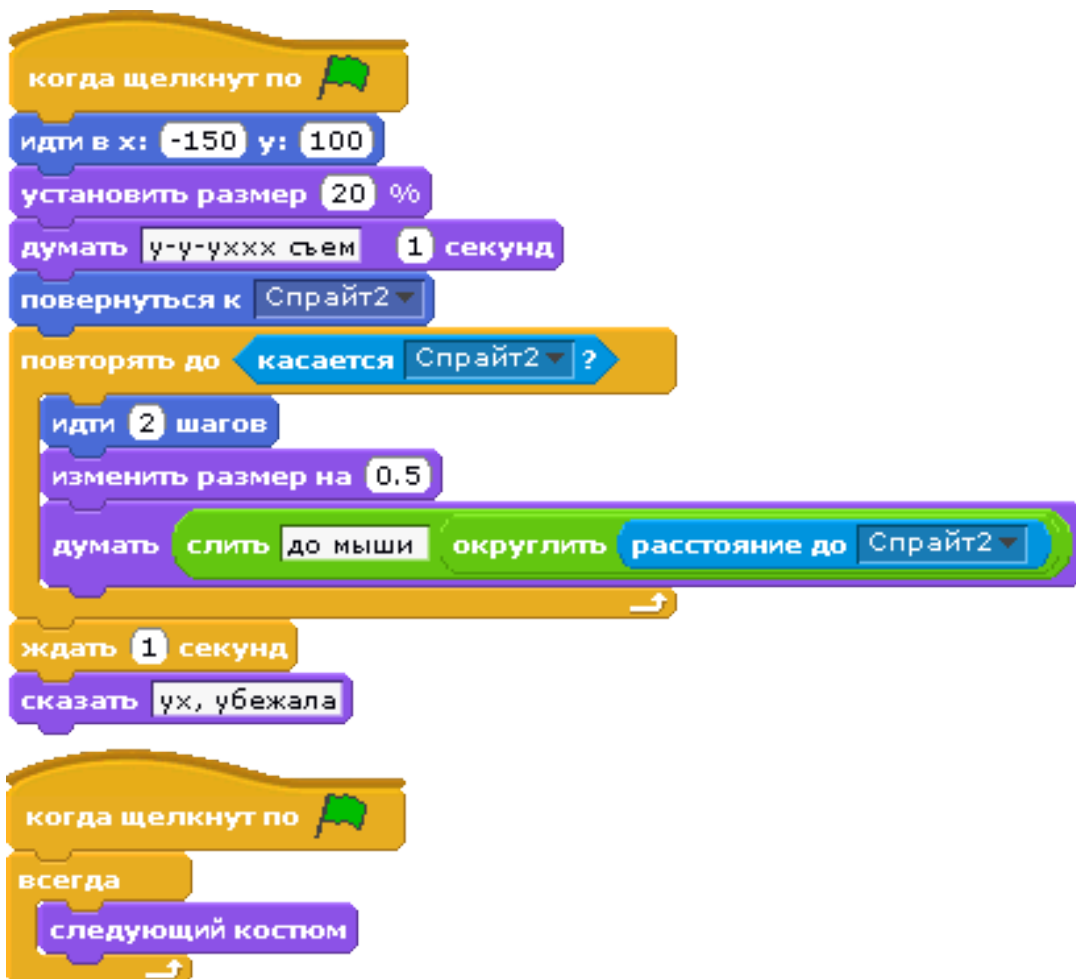


Цегла контролю(жовтий ящик)

Цегла контролю дозволяє збирати блоки команд і управляти послідовністю виконання скриптів. Вони визначають те, за яких умов здійснюється те або інша дія. Вони дозволяють спрайту передавати, приймати і реагувати на повідомлення, що управляють.



Виконайте такий проект:

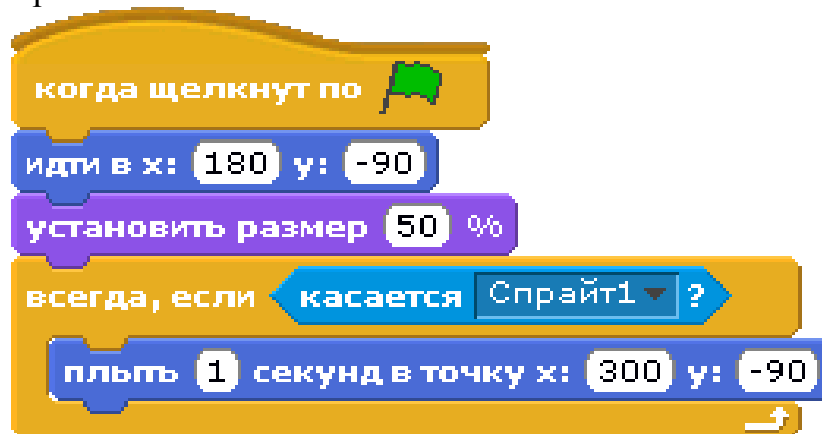


Це скрипт кажана. Особливості:

Перші 4 команди(Йти ... Обернутися) встановлюють початкове положення і стан миші.

Далі, поки кажан не торкнеться звичайної миші, вона рухатиметься до неї, поступово збільшуючись в розмірі. Тим самим створюється ефект наближення.

Другий скрипт - Завжди наступний костюм - служить для того, щоб кажан махав крилами.



Це скрипт звичайної миші. Особливості:

Перші дві команди описують початкове положення і стан об'єкту. Використовується форма "Завжди, якщо." для того, щоб скрипт постійно стежив за тим, чи торкається спрайт. Як тільки вони торкнуться, миша втече.

Завдання:

1) Слоненя здалека підходить до м'ячика. Как-будто ударяє по ньому і м'яч відлітає. Слоненя радісний кричить: "Урааа"! Підрахувати кількість кроків, зроблених слоненям до м'яча. Застосувати ефекти наближення і видалення.

2) Придумайте свою подібну історію.

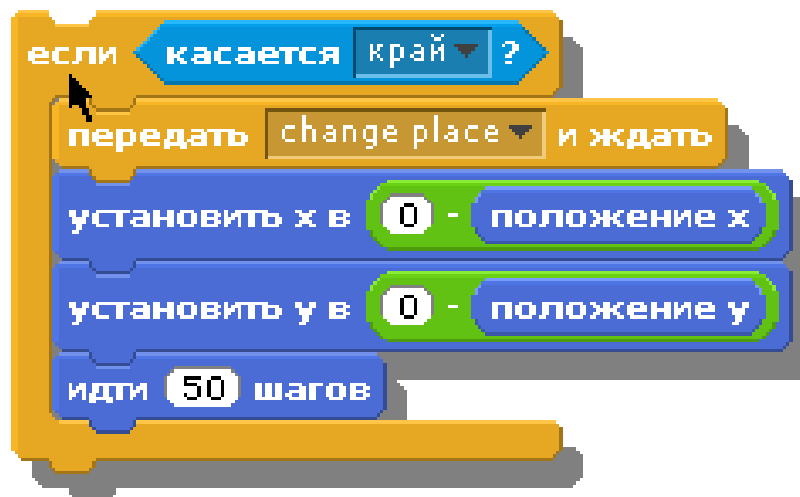
Скретч-квест

Квест припускає, що герой переміщається з одного простір в інше. Число просторів або кімнат може бути дуже великим, але для прикладу ми побудуємо світ з 5 пов'язаних між собою кімнат. Кішка управляється клавішами - вниз, вгору, управо і вліво.

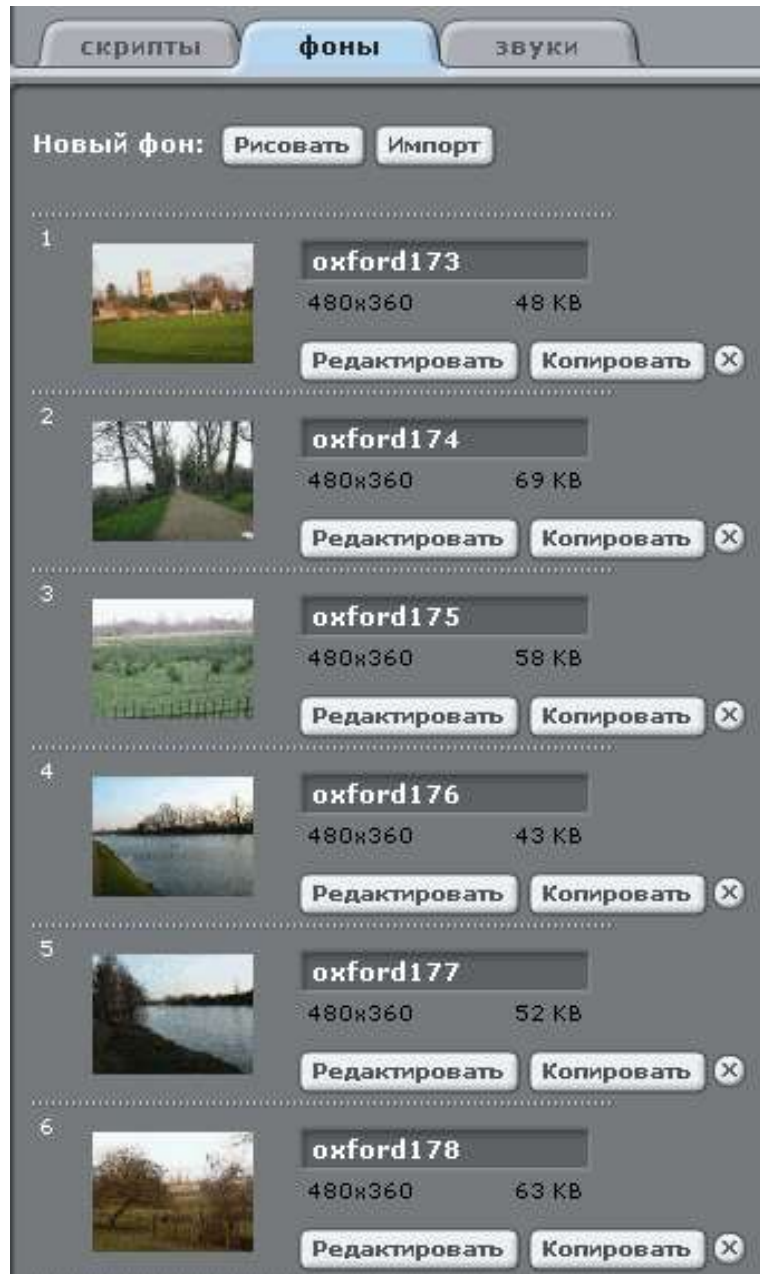


З кішкою все ясно. А ось світ навколо кішки влаштований складніше, він повинен пам'ятати звідки йде кішка і куди вона повинна потрапляти при її зіткненні з екраном.

Як тільки кішка торкається екрану, вона посилає світу сигнал - Broadcast "Change place"



У відповідь на цей сигнал середовище починає аналізувати положення(напрямок) кішки і фон, на якому вона знаходиться.



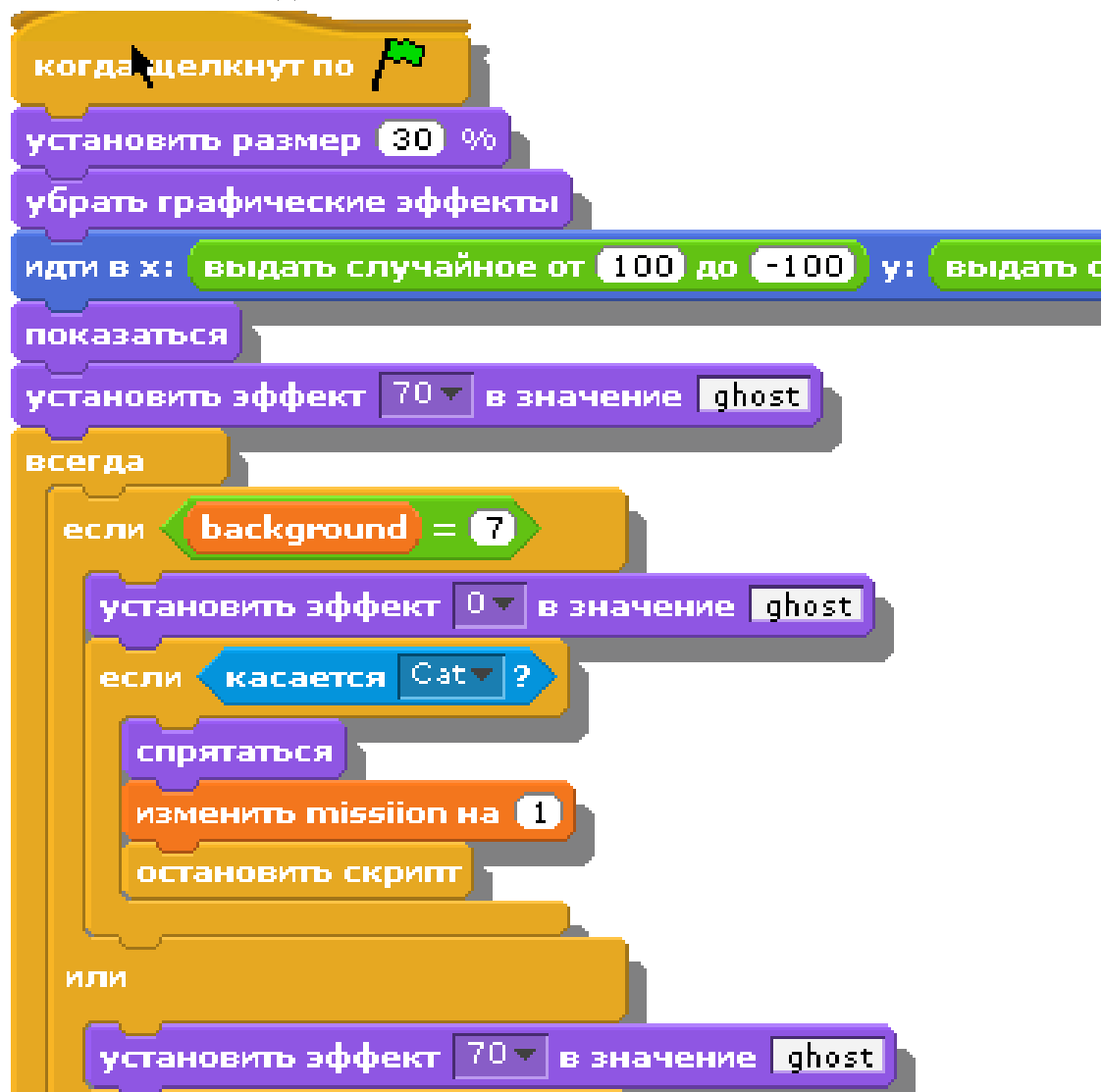
Залежно від того, куди дивиться кішка і з якої кімнати вона прийшла, світ підсовує під кішку нове зображення кімнати.



Тепер ми можемо подорожувати з кімнати до кімнати або з області в область. Ускладнимо гру, додавши до кішки двох злісних демонів. Кожен з них постійно обертається у бік кішки і рухається до неї. Якщо один з демонів торкається іншого, він завмирає. Якщо демон торкнеться кішки, то ми програли.



Додамо кішці місію, яку вона повинна виконати, переходячи з однієї області в іншу: знайти і зібрати сім зелених кульок. Кульки мають бути видимі в деяких областях і невидимі в інших.



Тепер у нас практично повноцінний квест, в якому гравець бродить по полях, збирає зелені кульки і вислизає від супротивників.

Розвиток програми : - ми можемо додавати нові об'єкти, з якими кішка зустрічатиметься в різних кімнатах квеста.

