

УЧИМСЯ СОЗДАВАТЬ ИГРЫ НА SCRATCH

ЮЛИЯ ТОРГАШЕВА



Программирование для детей





Для кого эта книга?

Эта книга – для всех, кто хочет создать свою собственную компьютерную игру. Она рассказывает о том, что такое среда программирования, как собрать игру, научить героев двигаться. При помощи книги и сайта с ресурсами для сборки игр ребенок сможет самостоятельно сделать четыре игры в среде программирования Scratch. В каждой игре есть герои, которые преодолевают препятствия, ищут сокровища или по-другому взаимодействуют между собой или с предметами. От начала к концу книги сложность игр возрастает, но собирать их можно в любой последовательности.

Что такое Скретч?

Scratch (Скретч) – среда программирования для детей, разработанная в Массачусетском технологическом институте (MIT). В Скретче можно создавать игры, мультфильмы, интерактивные истории. Для сборки программ используются разноцветные блоки-команды с подсказками.

Сайт Скретч – <https://scratch.mit.edu>

ПЕРВЫЕ ШАГИ В СКРЕТЧЕ

Добро пожаловать в Скретч!
Это среда программирования,
в которой можно легко собирать игры
и мультфильмы.

ПОПРОБУЙ – ЭТО ПРОСТО!



Скретч – это специальная программа для детей, в которой можно легко собирать компьютерные игры и мультфильмы.

Герои игр могут примерять разные наряды.



Их можно кормить и занимать играми.



Или научить летать.

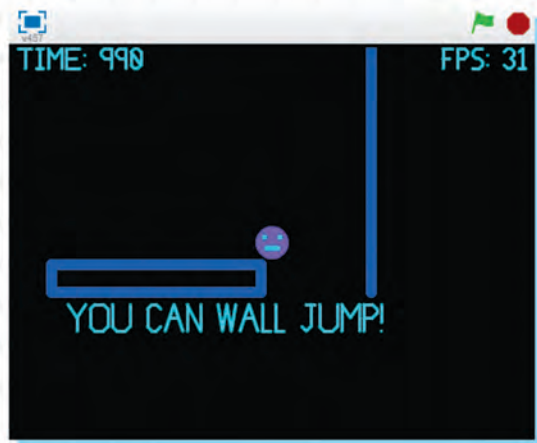


Заставьте его летать

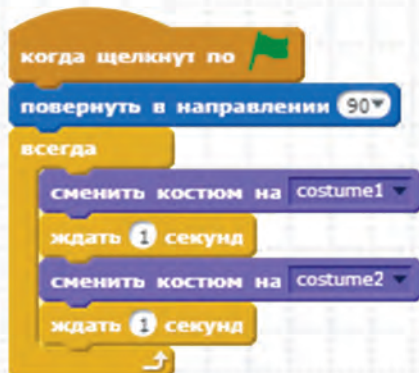
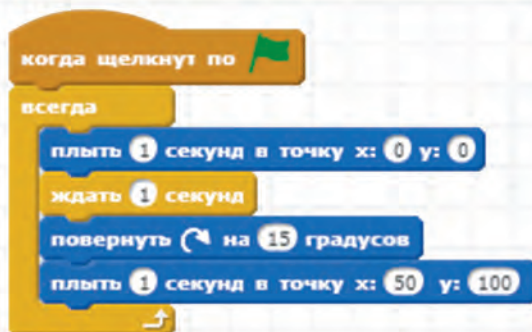
Анимируйте кота Scratch,
Суперкрошек или даже тако!



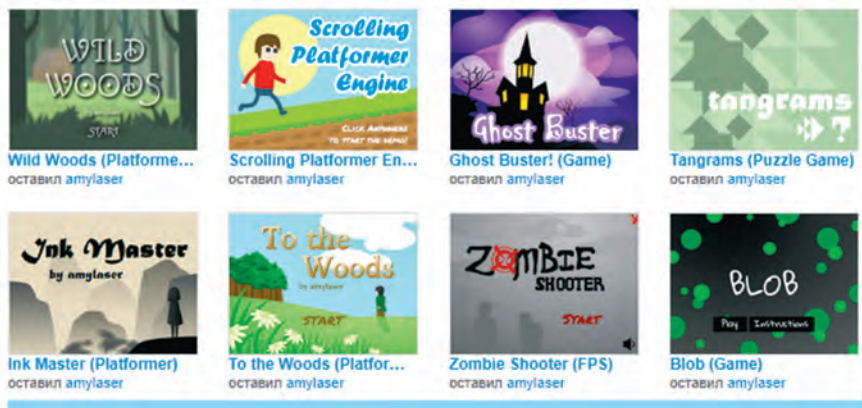
Или создать собственный платформер.



Анимация — это смена и перемещение изображений, при которых герои как будто оживают. В Скретче анимацию может создать каждый при помощи команд-блоков.



На сайте Скретч можно изучать, копировать и использовать проекты, созданные другими пользователями.



А главное —
создавать
свои.
Готовы?



ЧТО ТАКОЕ ЯЗЫК ПРОГРАММИРОВАНИЯ ЧТО ТАКОЕ ПРОГРАММА КАК ДЕЛАЕТСЯ ПРОГРАММА-ИГРА

ЧТО ТАКОЕ ЯЗЫК ПРОГРАММИРОВАНИЯ

Компьютеры (а также телефоны, смартфоны, телевизоры, умные часы и множество других устройств, даже холодильники) должны получать команды, чтобы работать. Они понимают эти команды в двоичном коде — в нем для передачи информации используются только цифры 0 и 1. Например, русская буква А в двоичном коде записывается как 11000000. Человеку писать программы в таком виде неудобно. Ученые, создававшие первые компьютеры,

придумали писать программы на языках, похожих на обычные «человеческие», только сильно сокращенных, чтобы у всех программистов были одни и те же наборы коротких команд. Так появились языки программирования. Специальные программы-компиляторы переводят тексты программ с языков программирования в машинный код.

ЧТО ТАКОЕ СРЕДА ПРОГРАММИРОВАНИЯ

Среда программирования — это специальная программа, которая помогает программисту работать. Тексты больших программ удобно писать в нескольких файлах. В среде программирования программист собирает программу из нескольких таких файлов, подключает (если нужно) графику и звуковые файлы.

КАК ДЕЛАЮТСЯ ИГРЫ

В компьютерных играх программы управляют графикой — картинками с изображением фонов, героев, предметов. Программа обрабатывает команды игрока, перемещения графики по экрану и соприкосновения картинок между собой. Например, когда герой игры дотрагивается до нужного предмета, программа «ловит» соприкосновение картинки героя с картинкой предмета и засчитывает игроку очки.

Как

составить

программу

Для того чтобы составить программу, надо написать в нужном порядке команды для компьютера. В Скретче это разноцветные блоки-команды, которые нужно соединять между собой в правильной последовательности.

Ты научишься

- * Подключать к игре графические файлы и звук.
- * Расставлять героев и предметы в игре в нужные места и управлять их перемещениями.
- * Управлять взаимодействием героев и предметов.
- * Подсчитывать очки игрока.



Как начать работать со **SCRATCH**

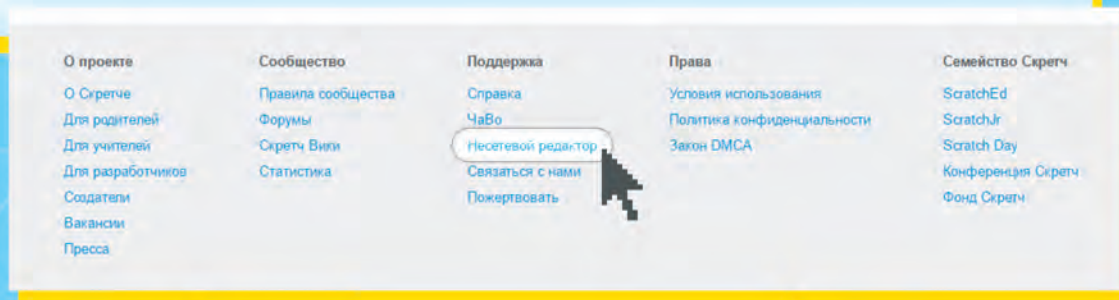
В Скретче можно создавать программы в специальном редакторе на сайте программы – тогда понадобится интернет.

Открыть редактор можно по ссылке:

<https://scratch.mit.edu/projects/editor/>

Загрузка офлайн-редактора и ресурсов для игры

Если хочешь создавать программы без подключения к интернету, то используй несетевой редактор Скретч. Ссылка на страницу для скачивания расположена в нижнем меню («подвале») главной страницы сайта Скретч.



Она ведет сюда:

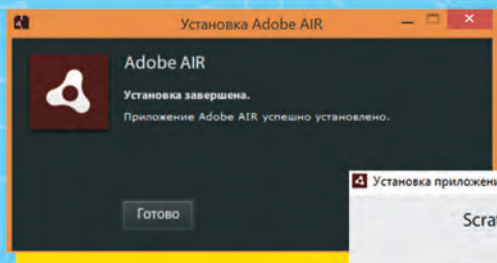
<https://scratch.mit.edu/scratch2download/>



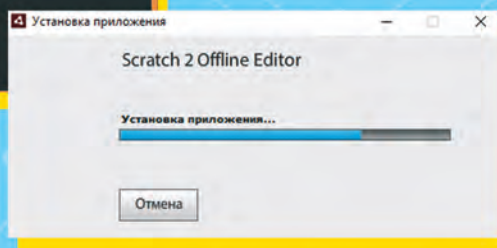
- 1**
Adobe AIR
Если вы еще не сделали этого, скачайте и установите последнюю версию [Adobe AIR](#)
Mac OS X - [Скачать](#)
Mac OS 10.5 и старше - [Скачать](#)
Windows - [Скачать](#)
Linux - [Скачать](#)
- 2**
Офлайн-редактор Scratch
Следующий шаг скачать и установить офлайн-редактор Scratch 2.0
Mac OS X - [Скачать](#)
Mac OS 10.5 и старше - [Скачать](#)
Windows - [Скачать](#)
Linux - [Скачать](#)
- 3**
Вспомогательные материалы
Не уверены с чего начать? Вот несколько полезных ресурсов.
Стартовые проекты - [Скачать](#)
С чего начать - [Скачать](#)
Карточки Scratch - [Скачать](#)

КАК СКАЧАТЬ СКРЕТЧ

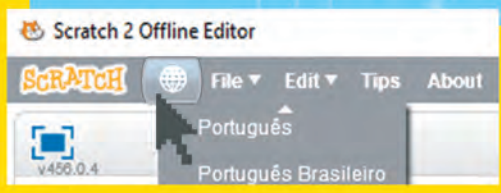
1. Сначала скачиваем программу Adobe Air.



2. Потом сам Скретч.



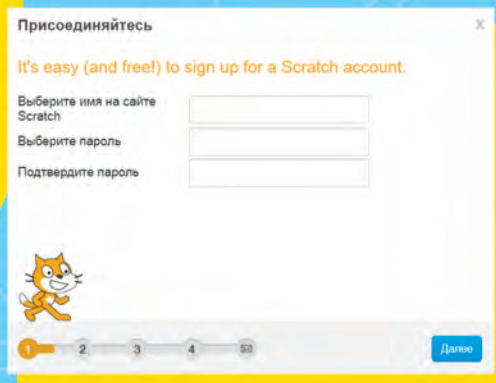
Вот ты и открыл Скретч. Но сначала меню в нем будет на английском языке. Хочешь выбрать русский? Жми на значок земного шара слева вверху и крути колесо мышки вниз. Русский ты найдешь ближе к концу длинного списка языков.



Ресурсы для сборки игры

По этому адресу ты найдешь графику и звуки для игр из этой книги. Зайди на сайт.

scratchbook.piter.com



Ну что, ты готов создать свою первую игру? Начинаем!





ФУТБОЛ

Ура, футбол!
Стадион ревет и ждет твоих голов!
Но вратарь команды соперника
не дремлет! Сможешь его обыграть?

Игра «Футбол»

Сложность: *



Время сборки: 1,5–2 часа

Видеоролик игры можно найти по адресу: scratchbook.piter.com

ОПИСАНИЕ

Футбольный матч на исходе — до конца осталась всего одна минута. Забивай как можно больше голов, и твоя команда победит!

.....

Игрок забивает мяч в ворота, используя мишень для прицела. Ему мешает вратарь — он движется в воротах. самого игрока не видно. Удар по мячу он делает при помощи клавиши ПРОБЕЛ. Время игры ограничено.

ПРАВИЛА ИГРЫ

- * Цель игрока — забить как можно больше голов противнику за ограниченное время.
- * Играет один игрок.
- * Игрок прицеливается при помощи мишени.
- * Игрок двигает мишень клавишами со стрелками ВЛЕВО и ВПРАВО, ВВЕРХ и ВНИЗ.
- * Когда мишень находится в нужном месте, игрок нажимает ПРОБЕЛ — это удар по мячу.
- * При попадании мяча в сетку ворот засчитывается гол — игроку добавляются очки.
- * Если мяч касается вратаря или штанги ворот, гол не засчитывается.

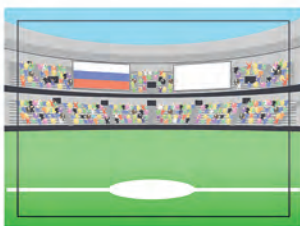


Все необходимые для игры ресурсы можно скачать по адресу:
scratchbook.piter.com. Хотя программу ты будешь делать сам, тебе понадобятся **СПРАЙТЫ** — так в Скретче называются отдельные картинки для каждого героя или предмета в игре.

ВОТ ВСЕ СПРАЙТЫ, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ В ИГРЕ «ФУТБОЛ»



Футболист
в двух вариантах



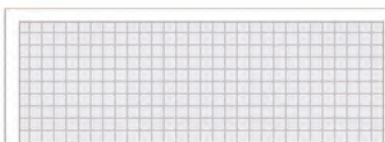
Фон



Мяч



Мишень
для прице-
ливания
в ворота



Ворота



Птичка
в двух вариантах



Рамка
для монитора
голов



Сообщения
о голах
или промахах

ГОТОВ К ИГРЕ? ВЫХОДИМ НА ПОЛЕ!



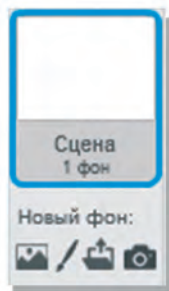


ЗАГРУЖАЕМ РЕСУРСЫ



ВНИМАНИЕ

В Скретче пространство, в котором происходят действия игры, называется сцена. У сцены есть фон – картинка на заднем плане. Фоны могут меняться, но в игре «Футбол» мы ограничимся одним.



Для того чтобы загрузить в игру фон – картинку стадиона, нажми на значок папки под сценой слева.

Теперь игра будет идти, как ей и положено, на стадионе, а не на фоне белого экрана.



Загрузи файл Фон.jpg из папки ресурсов с сайта scratchbook.piter.com. Теперь в поле справа на вкладке «Фоны» появился новый фон. Старый нам больше не нужен – его можно удалить, нажав на крестик.

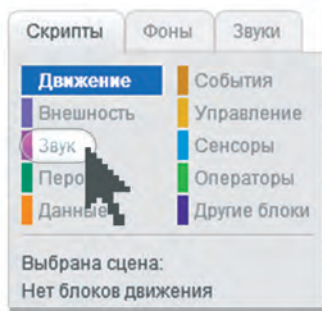


ЗАГРУЖАЕМ ФОНОВЫЙ ЗВУК



Конечно, можно играть и без звука, но со звуком гораздо интереснее – так игра оживает. Да и какой футбол в тишине! Добавим в игру шум стадиона, чтобы игрок чувствовал поддержку болельщиков.

Перейдем на вкладку «Скрипты». Именно здесь мы будем собирать программу. Для сцены и для каждого из героев в игре собирается отдельная программа-скрипт.

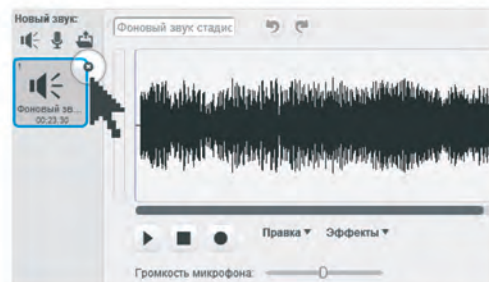


Перейдем в меню «Звук». Здесь собраны все команды, которые управляют звуками в игре. В этом меню можно подключить звуковые файлы, настроить их продолжительность, громкость. Можно даже собрать мелодию из отдельных нот или записать свои собственные звуки для игр!

Мы подключим к игре фоновый звук с шумом стадиона.



На вкладке «Звуки» нажми на значок папки и открой файл «Фоновый звук стадиона» из папки с ресурсами с сайта scratchbook.piter.com. На вкладке «Звуки» появится новый звук. Ненужные звуки можно удалить, нажав на крестик.



На вкладке «Скрипты» перейди в меню «События» и перетащи в окно справа блок.



Потом перетащи под него блок со вкладки «Звук».



Блоки соединятся друг с другом.

ВНИМАНИЕ



Звуки, которые мы загружаем для сцены или спрайта, становятся доступны в окошках внутри блоков команд. Пока звук не загружен, его там не видно.

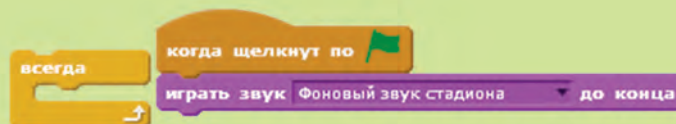
НАСТРАИВАЕМ ФОНОВЫЙ ЗВУК. ПЕРВЫЙ ЗАПУСК ИГРЫ

Когда попадаешь на стадион в первый раз, кажется, что можно оглохнуть от шума! Но футболист сосредоточен на игре. Окружающие звуки для него становятся все тише.

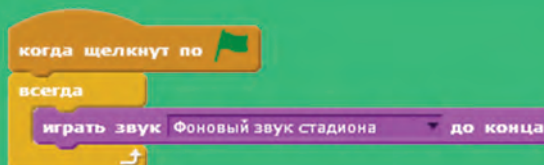
Звук в Скретче можно сделать тише или громче. Можно запустить его один раз, можно несколько раз. Сделаем так, чтобы фоновый звук, был постоянно до конца игры.



Нажми на вкладку «Управление». Перетащи блок «Всегда» на правое поле поближе к программе. Он как бы наденется на блок «Играть звук».



Получится так:

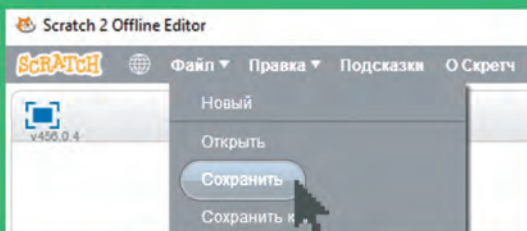


Пора запустить игру в первый раз! Для этого нажми на зеленый флажок вверху экрана игры или на блоке.



Для того чтобы оставить работу программы, нажми на красный кружок.

Мы уже проделали большую работу. Самое время сохранить проект. Для этого нажми команду «Файл» в верхнем меню Скретча и сохрани программу с любым именем. Например, «Футбол». Сохранить ее можно в любое место на компьютере.



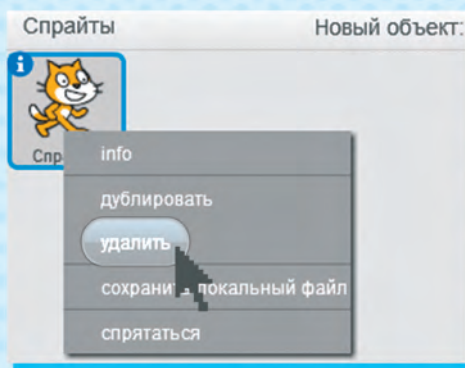
СТАВИМ ВОРОТА

Для того чтобы попасть в ворота, одной меткости мало. Для начала ворота должны быть на поле!

Как мы уже говорили, герои и предметы в Скретче называются спрайты. Это графические файлы, команды для которых мы собираем в поле справа. Сейчас мы будем загружать нужные нам спрайты в игру, и они начнут появляться на игровом поле.

Пока же вместо них на экране ты видишь кота. Это символ Скретча, но сейчас он нам не нужен, так что мы его уберем.

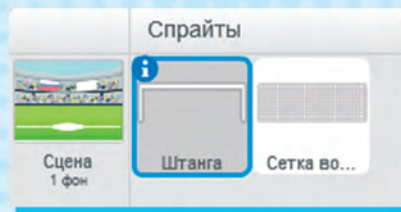
Для удаления спрайта нажми на его изображение внизу игрового поля правой кнопкой мыши. Откроется меню. Выбери команду «Удалить». Все, спрайт исчез!



Загрузим ворота. Ты уже легко с этим справишься: все делается так же, как при загрузке фона. Нажми на значок папки и выбери из папки с ресурсами файл под названием «Сетка ворот.svg».



Еще нам понадобится файл под названием «Штанга.svg».



ВНИМАНИЕ

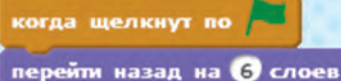


Ворота состоят из двух частей: самих ворот и сетки ворот. Это нужно для того, чтобы отдельно программировать действия при попадании в штангу и в створ ворот.



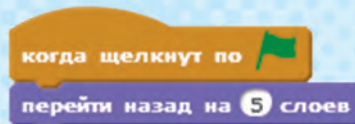
Создадим скрипты для сетки и штанги. Они простые — указывают, в каком порядке спрайты должны показываться на экране. Нам понадобятся блоки с вкладок «События» и «Внешность».

Для сетки ворот соберем такой скрипт.



И похожий на него — для штанги. Обрати внимание: различаются в них только номера слайдов. Фон слайда не считается. Сетка ворот должна находиться на самом нижнем слайде.

Поверх нее будет лежать штанга. А перед ней мы оставили несколько слайдов для мишени, вратаря, мяча и сообщений о голах.



ВНИМАНИЕ



Перетаскивать спрайты на игровом поле можно просто мышкой.

Совмести штангу и сетку. Поставь готовые ворота в центре поля.



ВНИМАНИЕ



Спрайт, который ты берешь мышкой, оказывается сверху. Но при запуске игры все спрайты разместятся в тех слоях, которые мы указали.

СТАВИМ МИШЕНЬ

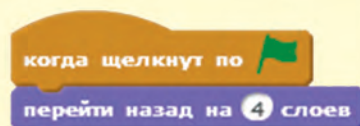
Добавим еще один спрайт — мишень. Принцип тот же, что при загрузке фона и ворот. Нажми на значок папки и выбери из папки с ресурсами файл под названием «Мишень».

СОВЕТ

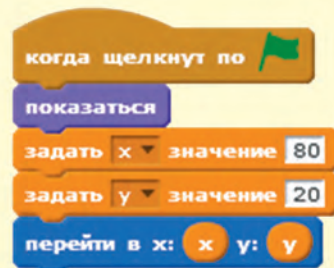
После сборки каждого скрипта сохраняй программу в меню «Файл» — «Сохранить».



Мишень должна появляться перед воротами, чтобы игрок мог прицелиться. Соберем скрипт указания слоя:



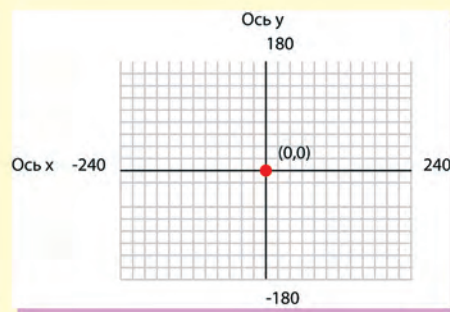
Во время игры мишень будет передвигаться по командам игрока. Но при запуске игры она должна всегда оказываться в определенном месте. Для этого нужно указать ее стартовые координаты. Соберем такой скрипт:



Перейдем в поле программы, во вкладку «Скрипты». Нам нужны блоки из групп «События», «Внешность», «Данные» и «Движение».

В Скретче центр игрового поля обозначается координатами (0,0). По горизонтали идет ось икс («x»), по вертикали — ось игрек («y»). Направления вверх и вправо обозначаются положительными числами, а вниз и влево — отрицательными, то есть числами со знаком «-» (минус).

Ширина сцены в Скретче составляет 480 точек, то есть вправо от нуля 240 точек и влево тоже 240. Общая высота поля — 360 точек, то есть по 180 точек вверх и вниз от центра экрана.



СОЗДАЕМ ПЕРЕМЕННЫЕ ДЛЯ КООРДИНАТ МИШЕНИ

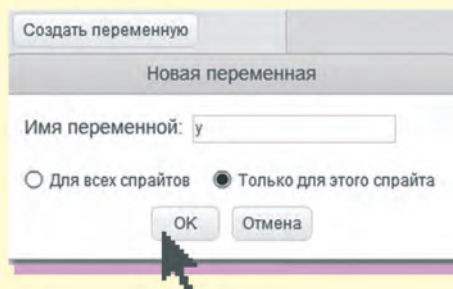
Переменные в игре нужны, чтобы хранить разные числовые значения: координаты героев и предметов, скорость их перемещения и многое другое. Значение переменной может меняться, потому-то она так и называется.

Создадим переменные для изменения положения мишени. Во вкладке «Данные» нажми на кнопку «Создать переменную».

ВНИМАНИЕ



Выбери пункт «Только для этого спрайта». У разных спрайтов могут быть разные координаты.



ВНИМАНИЕ



Значения в команде, которые задаются командами и изменяются во время исполнения программы, — это параметры команды. Параметрами могут быть переменные, числа, математические выражения, текстовые строки, цвет.

ВНИМАНИЕ



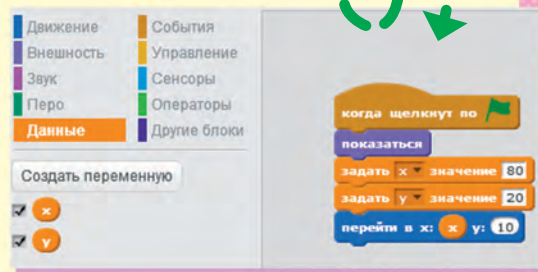
Проверим работу программы. Щелкни по зеленому флажку в скрипте. Что происходит?

Мишень переместилась? Да.

Включился ли фоновый звук? Нет.

Дело в том, что нажатие на флажок в скрипте запускает только сам скрипт!

Для того чтобы запустить всю программу, надо щелкнуть флажок над игровым полем.



Футбол

v456.0.4



< ДВИГАЕМ МИШЕНЬ >

Игрок передвигает мишень клавишами со стрелками ВВЕРХ, ВНИЗ, ВПРАВО, ВЛЕВО. Для сборки скриптов, которые обрабатывают эти нажатия, нам потребуются блоки с вкладок «События», «Данные», «Операторы» и «Движение».

ВНИМАНИЕ



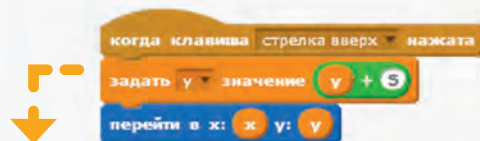
Проверим работу программы!
Щелкни по зеленому флажку сверху и попробуй подвигать мишень.
Работает?



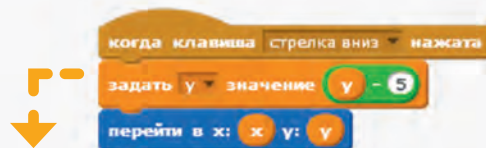
ВНИМАНИЕ



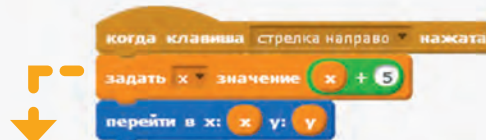
Счет координат идет от центра игрового поля!
Координаты спрайта обозначают центр картинки со спрайтом!



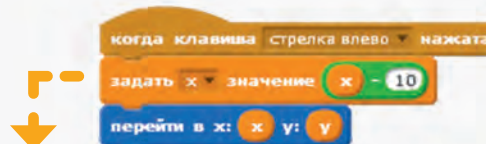
При каждом нажатии клавиши Вверх координата у мишени увеличивается на 5. Мишень перемещается в новую точку.



При каждом нажатии клавиши Вниз координата у мишени уменьшается на 5. Мишень перемещается в новую точку.



При каждом нажатии клавиши Вправо координата х мишени увеличивается на 5. Мишень перемещается в новую точку.



При каждом нажатии клавиши Влево координата х мишени уменьшается на 5. Мишень перемещается в новую точку.

ЗАГРУЖАЕМ ВРАТАРЯ И НАСТРАИВАЕМ ЕГО ДВИЖЕНИЯ

Как и в настоящем футболе, вратарь в нашей игре защищает ворота. Он движется в разные стороны, чтобы успеть поймать мяч.

Для начала выведем вратаря на поле. Ты уже помнишь, как загрузить новый спрайт, – нажми на значок папки под игровым полем.

Новый объект:



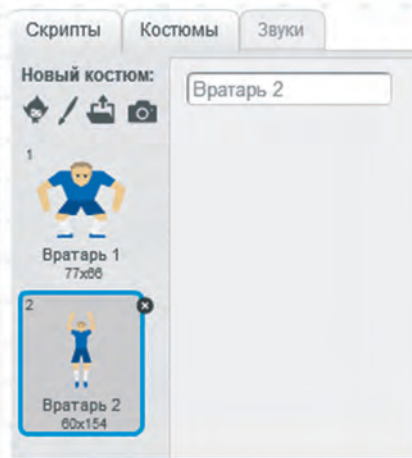
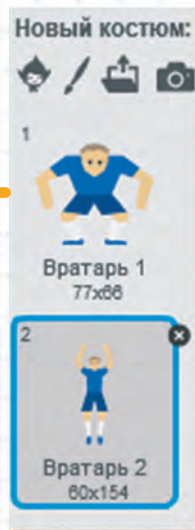
Теперь перейдем на вкладку «Костюмы» в поле программы. Добавим второй костюм для спрайта. Нажми на значок папки и загрузи файл «Вратарь 2.svg» из папки с ресурсами.

ВНИМАНИЕ

В списке спрайтов под игровым полем видно только один костюм спрайта – тот, который последним был выделен в списке костюмов. Для того чтобы узнать, сколько костюмов у спрайта, надо открыть вкладку «Костюмы».

ВНИМАНИЕ

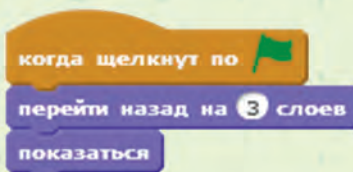
Думаешь, мы заставим вратаря переодеваться во время игры!? Нет! Костюм – это картинка для спрайта. Поведение спрайта задается скриптами, а его внешний вид может меняться. Смена костюма – один из способов это сделать.





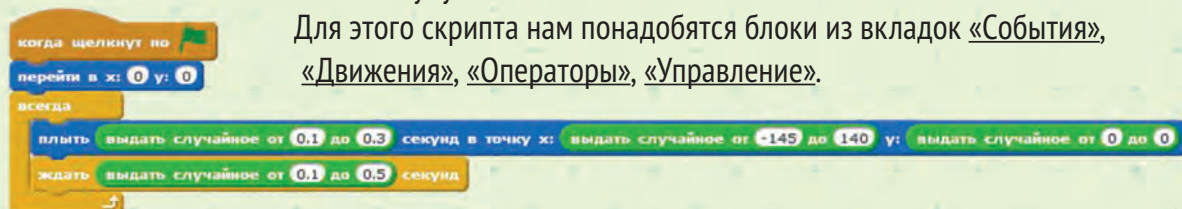
УПРАВЛЯЕМ ДВИЖЕНИЯМИ ВРАТАРЯ

Чтобы вратарь мог исполнять свои обязанности, соберем для него скрипты.

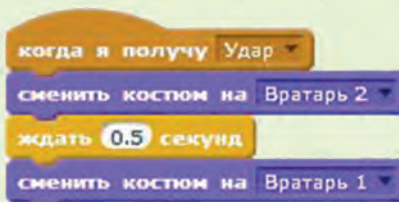


При старте игры вратарь должен оказаться перед воротами и перед мишенью. Затем всю игру он движется вправо-влево перед воротами, чтобы не упустить мяч.

Для этого скрипта нам понадобятся блоки из вкладок «События», «Движения», «Операторы», «Управление».



Что происходит с вратарем? Он через случайные промежутки времени перемещается в случайное место экрана, но движется только вправо и влево. Удар! Что делает вратарь? Он прыгает за мячом. Тут нужен отдельный скрипт, для которого нам понадобятся блоки из вкладок «Внешность» и «События».



ВНИМАНИЕ



Что такое сообщение?

Сообщение передается всем спрайтам, когда в программе происходят какие-либо действия или события. Для каждого спрайта можно указать реакцию на эти сообщения. Если в скрипте у спрайта такая реакция не задана, он пропустит это сообщение «мимо ушей» и не будет ничего делать. Так можно управлять поведением спрайтов.

Сейчас мы создадим сообщение «Удар».

Получив его, спрайт вратаря «узнает», что к нему летит мяч.

Перетащи блок «Когда я получу сообщение» в поле скриптов.

когда я получу

сообщение1 ▾

сообщение1

новое сообщение...

Новое сообщение

Имя сообщения:

Если нажать на стрелку в окошке параметра, появится список сообщений и пункт «Новое сообщение».

Введи любое имя сообщения и нажми ОК. Удобно называть сообщения так, чтобы было понятно, о чем они сообщают спрайтам. Каждый раз, получая сообщение об ударе, вратарь будет прыгать — ему надо поймать мяч! На полсекунды он будет зависать в воздухе, а потом приземляться и принимать обычную позу — и так после каждого удара игрока. Поскольку вратарь обязательно возвращается в исходное положение, то при каждом запуске игры он будет в полуприседе.

ВНИМАНИЕ



Сохраняй игру после создания новых скриптов!

СТАВИМ МЯЧ И НАСТРАИВАЕМ УДАР



Но чем же мы будем бить по воротам? На поле не хватает самого главного — мяча.

Загрузи спрайт мяча из папки с ресурсами — файл называется «Мяч.svg».

Это действительно главный герой игры — от событий, которые с ним происходят, зависит подсчет очков и других спрайтов. Поэтому для него понадобится больше всего скриптов. Заодно мы положим сюда и скрипты, которые действуют для всей игры в целом, а не только для мяча. Их можно положить на поле скриптов любого спрайта, можно на поле скриптов фона, но удобнее держать там, где описывается основное поведение программы — просто чтобы все было под рукой.

Для начала создадим несколько переменных — для подсчета времени матча, ударов и голов. Они будут действовать для всех спрайтов.

Создать переменную

☐ x

☐ y

☒ Голы

☒ До конца матча

☐ Удары

когда щелкнут по

перейти назад на 1 слоев

показаться

перейти в x: 0 y: -139

установить размер 20 %

Галочки слева от переменных означают, что программе нужно показывать переменную на игровом экране. Голы — да, нужно. Время до конца матча — тоже нужно. Удары — можно показывать, а можно нет. Координаты мишени показывать игроку не надо. Торжественный момент: ставим мяч на поле.

Мяч мы расположим на втором сверху слое. Где именно он будет находиться? Решать тебе. Перетащи мяч на нужное место, наведи на него курсор и посмотри координаты в правом нижнем углу. Введи их в окошки параметров блока «Перейти». Теперь при запуске игры мяч всегда будет оказываться в этом месте.





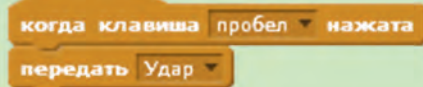
Для сборки этого скрипта нам понадобятся вкладки «Движение», «События», «Управление», «Операторы», «Внешность» и «Сенсоры».

А теперь посмотрим, что происходит с мячом, когда игрок ударяет по нему, то есть нажимает клавишу ПРОБЕЛ.

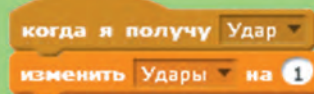
СОБИРАЕМ СКРИПТ УДАРА ПО МЯЧУ



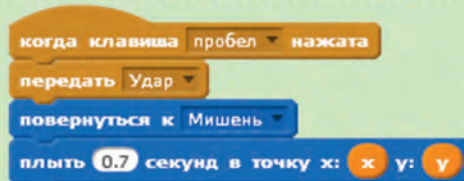
При нажатии клавиши ПРОБЕЛ все спрайты получают сообщение «Удар».



На это сообщение реагирует переменная «Удар» – она увеличивается на 1.
Собери для этого отдельный скрипт в этом же окошке.



Теперь нужно сделать так, чтобы мяч летел к мишени.
Для этого мы добавляем блок «Повернуться».



Настоящий мяч не перемещается мгновенно, и наш тоже не должен. Чтобы он летел в нужную точку с определенной скоростью, нам пригодится блок «Плыть» из вкладки «Движение». Время в нем считается в секундах.

ВНИМАНИЕ

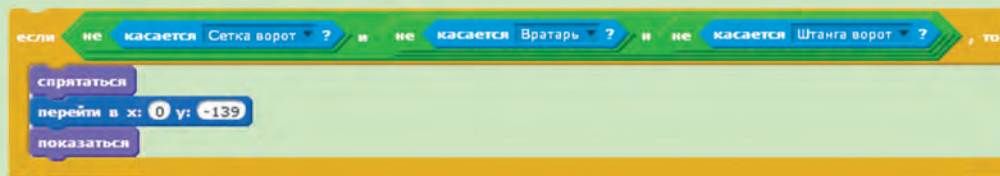


Со скоростью мяча можно экспериментировать.

Параметры команды «Плыть» – переменные x и y , то есть координаты мишени. Их надо перетащить сюда из вкладки «Данные». Эти переменные меняются только при движении мишени, но использовать их значения могут скрипты любых спрайтов.

После удара нужно проверить, не коснулся ли мяч вратаря или штанги? Тут возможны четыре варианта развития событий.

- 1** Мяч пролетел мимо ворот и не коснулся ни сетки, ни штанги, ни вратаря. Это возможно, если мишень находится за пределами ворот.



Мяч как бы улетает за пределы поля, поэтому мы даем ему команду спрятаться. Затем мы возвращаем его в исходную точку с координатами (0, -139) и снова показываем. Больше ничего не происходит – игрок промахнулся.

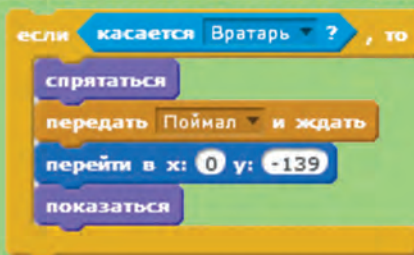


ВНИМАНИЕ

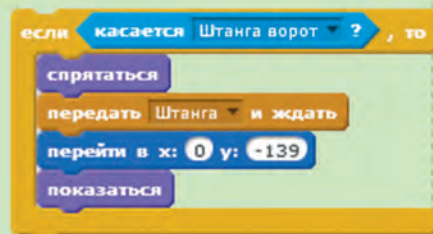
Сохраняй игру!

- 2** Мяч попал во вратаря, который по случайным координатам передвигается вдоль ворот.

Мяч пропадает, затем возвращается в исходную точку и снова появляется. Скрипт передает сообщение «Поймал».

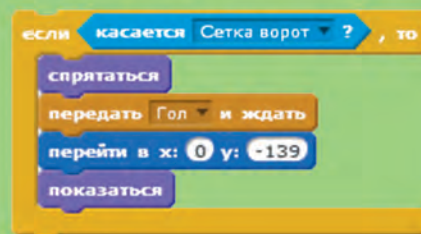


- 3** Мяч коснулся штанги.



Происходит то же самое, что при попадании во вратаря. Скрипт передает сообщение «Штанга», мяч возвращается в исходное положение.

- 4** Мяч попал в сетку ворот! Гол!



Скрипт передает сообщение «Гол», мяч возвращается в исходное положение.

ЗВУКИ УДАРА

Футбольный матч – это бушующий океан звука. Болельщики скандируют, хлопают, гудят, судья свистит. Добавим в игру дополнительные звуки, чтобы игрок мог как следует погрузиться в атмосферу настоящего матча.

На вкладке «Звуки» в поле скриптов мяча загрузи из папки с ресурсами с сайта scratchbook.piter.com еще несколько звуков.

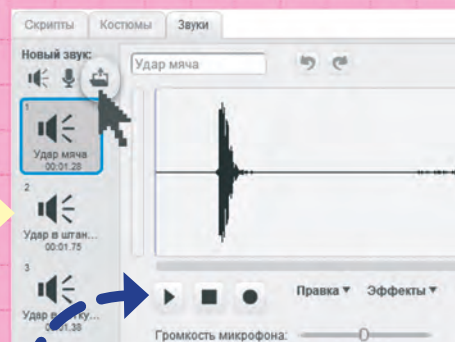
Они появятся в окошке параметров блока «Играть звук».

Звуки нужно добавить в скрипт обработки ударов мяча, чтобы они проигрывались в зависимости от действий игрока.

Звук при ударе.

Звуки при попадании в штангу.

Звуки при попадании в ворота.
Стадион ревет от восторга!



- «Удар мяча»
- «Удар в штангу»
- «Удар в сетку»
- «Недовольный стадион»
- «Довольный стадион»

когда клавиша пробел нажата

играть звук Удар мяча

передать Удар

вернуться к Мишень

плыть 0.7 секунд в точку x: x y: y

если касается Штанга ворот ?, то

играть звук Удар в штангу

играть звук Недовольный стадион

спрятаться

передать Штанга и ждать

перейти в x: 0 y: -139

показаться

если касается Сетка ворот ?, то

играть звук Удар в сетку

играть звук Довольный стадион

спрятаться

передать Гол и ждать

перейти в x: 0 y: -139

показаться

СООБЩЕНИЯ ОБ УДАРЕ

Чтобы игрок тоже видел результат своей игры, добавим дополнительные сообщения о том, чем закончился удар.
Из папки с ресурсами с сайта scratchbook.piter.com загрузи спрайты:



Сообщение
Гол.svg



Сообщение
Штанга.svg

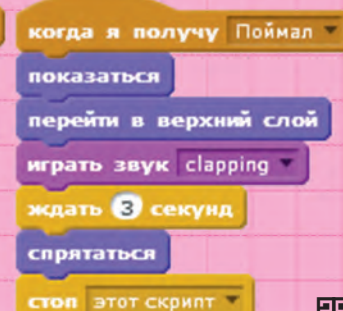
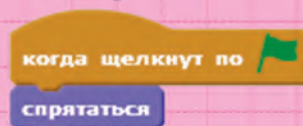
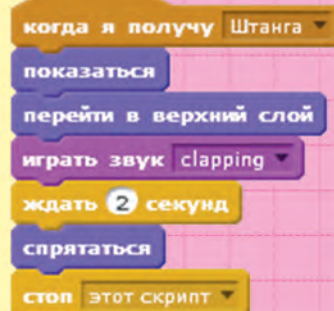
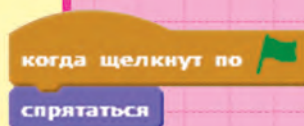
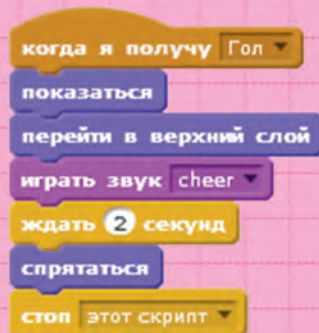
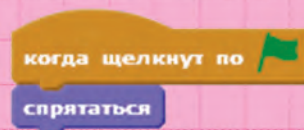


Сообщение
Поймал.svg

Добавим к спрайту «Сообщение Гол» скрипты:

В начале игры сообщения не видно. Когда мяч попадает в сетку и передается сообщение «Гол», спрайт сообщения получает его и показывается поверх всех остальных спрайтов. Пусть игрок не только увидит сообщение, но и услышит восторженные крики болельщиков. Ведь он заслужил! Для этого загрузи звук **cheer** из библиотеки звуков Скретча.

Собери такие же скрипты для спрайтов «Сообщение Штанга» и «Сообщение Поймал», только вместо звука **cheer** загрузи из библиотеки звук **clapping** — слабые хлопки.

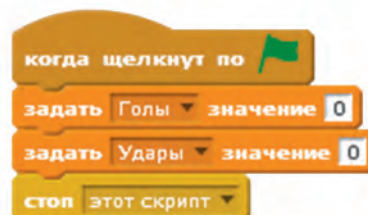


СЧИТАЕМ ГОЛЫ

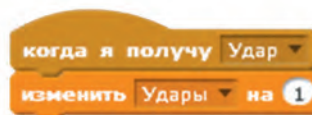
ВНИМАНИЕ

Сохраняй игру!

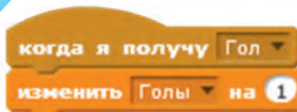
Как ты помнишь, на игровом поле у нас будут счетчики голов и ударов, чтобы они работали, зададим значения переменных «Голы» и «Удары». Собери скрипт:



Блоки «Задать значение» находятся на вкладке «Данные». При нажатии на стрелку в окошке параметров становятся доступны переменные, которые уже созданы. При каждом ударе значение переменной «Удары» будет увеличиваться на единицу. Собери скрипт:



При каждом попадании мяча в ворота значение переменной «Голы» будет увеличиваться на единицу. Собери скрипт:



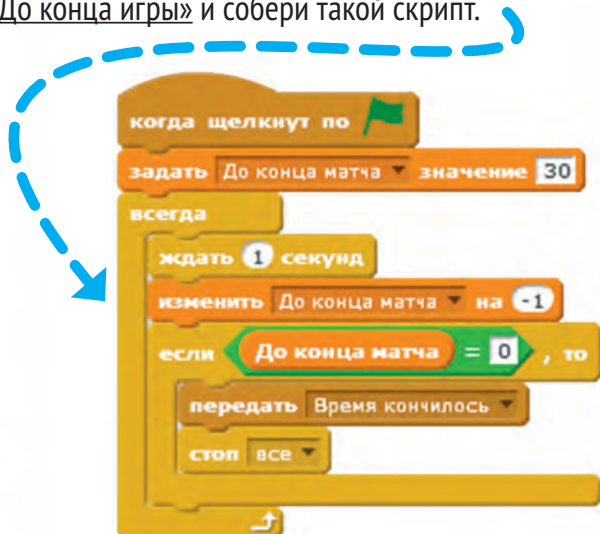
КОНЕЦ ИГРЫ ПРИ ОКОНЧАНИИ ВРЕМЕНИ

Футбольный матч не может быть бесконечным. Добавим таймер, который будет вести обратный отсчет времени и показывать, сколько секунд осталось до завершения игры.

ВНИМАНИЕ

Этот скрипт находится на поле скриптов мяча, но его можно собрать и в другом спрайте, например в скриптах фона.

Во вкладке «Данные» создай переменную «До конца игры» и собери такой скрипт.



При старте игры мы задаем этой переменной значение 30 – время в секундах до конца игры.

Из папки с ресурсами с сайта scratchbook.piter.com загрузи спрайт Сообщение Игра окончена.svg



ВНИМАНИЕ

Сохраняй игру!

ВНИМАНИЕ

Ты можешь попробовать разные значения времени. В скретче есть внутренний счетчик времени – блок «Ждать» со вкладки «Управление». Используем его для обратного отсчета времени. Через каждую секунду значение переменной «До конца матча» будет уменьшаться на один. Когда значение переменной уменьшится до нуля, игра остановится и скрипт мяча передаст всем спрайтам сообщение об этом. Время вышло!

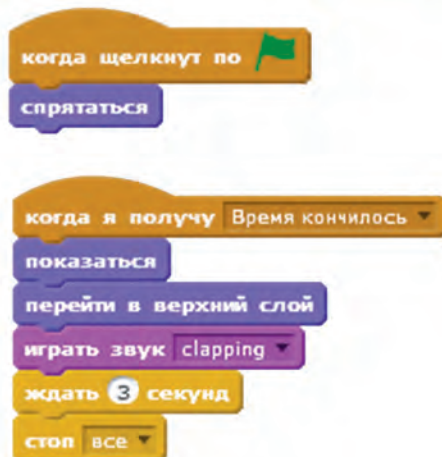
ВНИМАНИЕ

Создай новое сообщение «Время кончилось».

- * Работа всех скриптов всех спрайтов прекращается.
- * Кто «ловит» это сообщение?

Собери для него скрипт.

Вернись в спрайты сообщений и добавь к каждому из них по маленькому скрипту с командой спрятаться при получении сообщения «Время кончилось». Он нужен для того, чтобы скрывать эти сообщения, если они появились меньше чем за 2 секунды до окончания игры. Иначе уведомление об окончании игры будет на них наслаиваться и выглядеть некрасиво.



ВНИМАНИЕ

- * Сохраняй игру!
- * Запусти и проверь игру.



ВСЕ РАБОТАЕТ? ОТЛИЧНО!
ПОЗДРАВЛЯЕМ
С ПЕРВОЙ ИГРОЙ, СОБРАННОЙ
НА СКРЕТЧЕ!

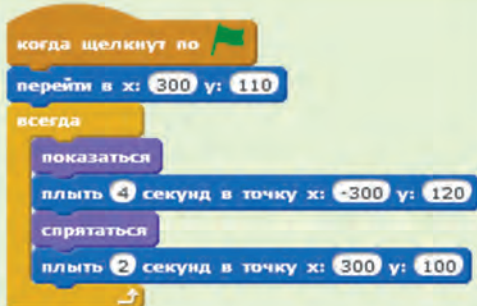
БОНУС. ПТИЧКА

Ой! Нападающий забивает один гол за другим, но вдруг возникает неожиданная помеха. Смелая птичка залетела прямо на поле и мешает ему прицеливаться. Сможет ли он сосредоточиться на игре?

ВНИМАНИЕ

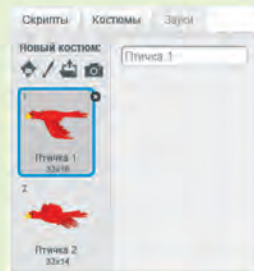
Сохраняй игру!

Добавь из папки с ресурсами с сайта scratchbook.piter.com спрайт птички «Птичка 1.svg».

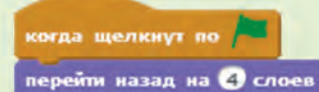


Птичка появляется с правого края экрана и летит влево. Потом она пропадает из поля зрения и снова появляется справа. В программе она на самом деле возвращается слева направо, но ее не видно.

На вкладке «Костюмы» добавь к нему второй костюм – файл «Птичка 2.svg» из той же папки с ресурсами.



Собери для птички скрипты.

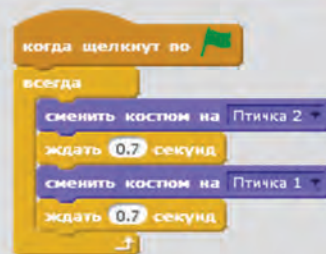


Она должна пролетать на заднем плане по отношению к воротам.

ВНИМАНИЕ



Ты можешь поэкспериментировать со временем пролета птички через экран и координатами точки, в которой она появляется и пропадает.



**ВСЕ РАБОТАЕТ?
ОТЛИЧНО!
ТЕПЕРЬ ТОЧНО
ВСЕ!
ПОЗДРАВЛЯЕМ
С ПЕРВОЙ ИГРОЙ,
СОБРАННОЙ
НА СКРЕТЧЕ!**



Футбол

v450.0.4





ПИНГВИН

• • • • •

НАКОРМИ ПИНГВИНА

Ам, ам, ам... Маленький пингвин потерял родителей и грустно раскрывает клюв — ждет, что его кто-нибудь накормит. Ты можешь это сделать. Только будь внимательнее: пингвину нужна рыба, а от сладкого у него заболит живот!

Игра «Накорми ПИНГВИНА»

Сложность: *

 Время сборки: 1,5–2 часа

Видеоролик игры можно найти по адресу: scratchbook.piter.com

Описание

Голодный пингвин открывает рот, чтобы поймать вкусную еду. Тебе надо накормить его и уложиться в отведенное время. Но будь внимателен! Пингвин очень любит вредные сладости, которые ему так и норовят подкинуть посетители зоопарка. Конфеты и торты падают с неба с разной скоростью — тяжелый торт быстрее, а легкие конфеты медленнее. Управляй пингвином так, чтобы он ел только рыбу, иначе у него заболит живот!

Правила игры

- * Цель игрока — кормить пингвина только рыбками.
- * Играет один игрок.
- * Игрок управляет пингвином при помощи клавиш со стрелками ВЛЕВО и ВПРАВО.
- * Еда появляется наверху экрана в случайных местах и падает вниз с ускорением. Когда пингвин ловит еду, игроку засчитываются очки.
- * За 30 секунд пингвину нужно скормить 10 рыбок. Если он кроме рыбы ел конфеты и съел их слишком много, у него может заболеть живот! Тогда игра заканчивается.
А уж если в клюв пингвину попал торт — все, конец игры наступает сразу. Торты пингвинам есть нельзя.



Все необходимые для игры ресурсы можно скачать по адресу:
scratchbook.piter.com.

ВОТ КАКИЕ СПРАЙТЫ ТЕБЕ ПОНАДОБЯТСЯ

Пингвин в двух видах:



С открытым
клювом



и с закрытым
клювом



Фон



Торт

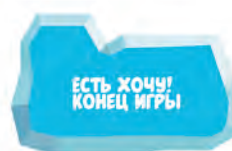
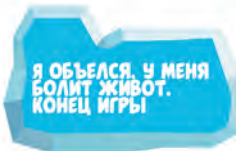


Конфета



Рыбка

Сообщения о событиях:



Рамка-фон для подсчета очков:





ЗАГРУЖАЕМ РЕСУРСЫ



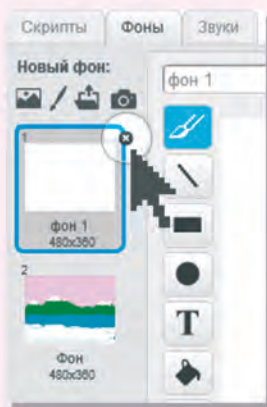
Для того чтобы загрузить в игру фон – картинку со льдинами и горами, нажми на значок папки под сценой слева.



Загрузи файл Фон.jpg из папки ресурсов с сайта scratchbook.piter.com. Новая картинка появится в поле слева внизу и в поле справа на вкладке «Фоны».

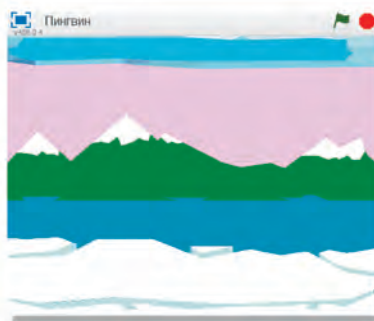
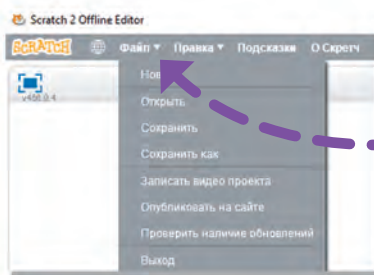


Ненужный старый фон можно удалить.



Теперь игра будет проходить не на скучном белом экране, а на фоне льдин и гор. Они куда больше нравятся пингвину!

Сохраним программу, чтобы не потерять сделанную работу. Нажми на пункт верхнего меню «Файл».



Выбери пункт «Сохранить как» и введи имя проекта – например «Пингвин». Оно появится на плашке над игровым полем.

ВНИМАНИЕ



Сохраняй проект после всех важных изменений! Это можно сделать также в меню «Файл» – командой «Сохранить».

СОЗДАЕМ ФОНОВЫЙ СКРИПТ

Фон — это тоже спрайт. Хотя он и стоит на месте, к нему можно привязать выполнение действий, которые происходят не с конкретными героями, а со всей игрой.

Перейдем на вкладку «Скрипты». Здесь мы будем собирать программу. Для сцены и для каждого из героев в игре собирается отдельная программа-скрипт.

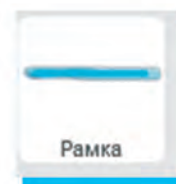
Для начала создадим скрипт, в котором запускается фоновый звук и задаются значения переменных для подсчета очков и отсчета обратного времени.



Сначала загрузим спрайт с рамкой для счетчиков и таймера. Нажми на значок папки под игровым полем.

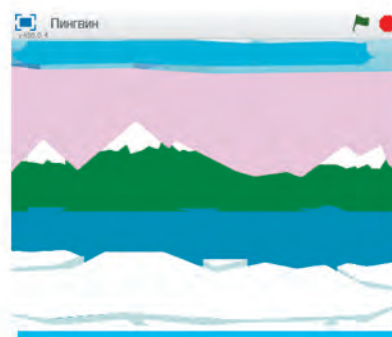


Загрузи файл «Рамка.svg». Изображение рамки появится на игровом поле и в белом прямоугольнике под ним — это поле списка спрайтов.

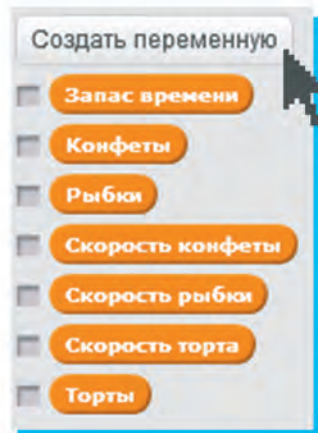


Перетащи его мышкой наверх.
Теперь игровое поле выглядит так:

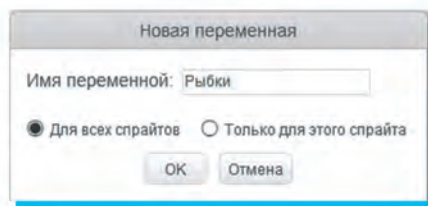
А теперь выбери вкладку «Данные» — в ней мы будем создавать переменные. В этой игре нам понадобятся переменные для подсчета съеденных пингвином рыбок, конфет и тортов. Еще одна будет подсчитывать время, которое осталось до окончания кормления.



Во вкладке «Данные» нажми на кнопку «Создать переменную».



В открывшемся окошке введи имя переменной:

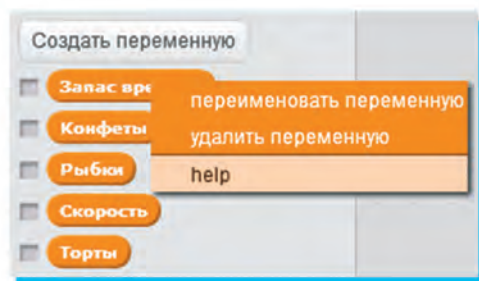


Повтори все эти действия со всеми переменными.

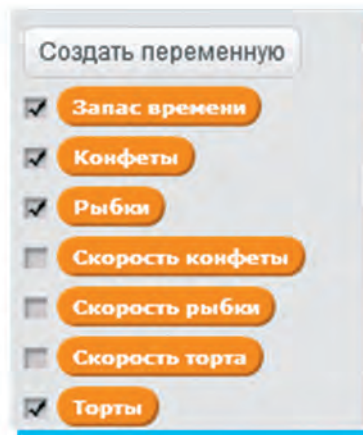
ВНИМАНИЕ

Переменные скорости понадобятся нам для других спрайтов. Удобно создать их сразу, чтобы в нужный момент они были под рукой, но можно сделать это и позже.

Если нужно исправить название переменной, нажми на нее правой кнопкой мыши. Откроется контекстное меню.



Поставь галочки рядом с переменными «Запас времени», «Конфеты», «Рыбки» и «Торты».

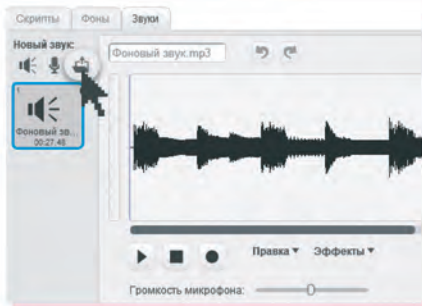
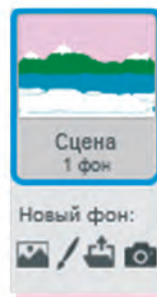


Они появятся на экране. Перетащи их мышкой наверх на рамку, чтобы игровое поле выглядело аккуратно.



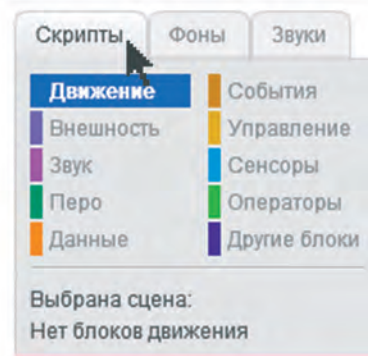
ЗАГРУЗИМ ФОНОВЫЙ ЗВУК ДЛЯ ИГРЫ

Играть в тишине можно, но это скучно. Звуки очень важны для создания атмосферы, поэтому мы добавим их в нашу игру. Начнем с фонового, который будет проигрываться постоянно. Щелкни на значке сцены в правом нижнем углу.



На вкладке «Звуки» загрузи файл «Фоновый звук.wav» из папки ресурсов с сайта scratchbook.piter.com.

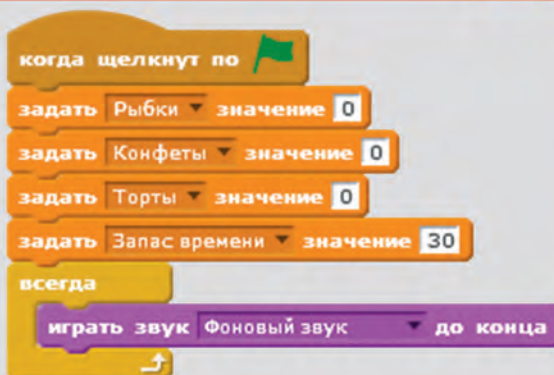
Теперь все готово, чтобы собрать скрипт для фона. Это делается на вкладке «Скрипты» в поле справа. Нам понадобятся вкладки «События», «Данные», «Управление» и «Звук». Перетащи с них нужные блоки, поставь их по порядку и выбери в окошках параметров названия переменных. Укажи во втором окошке параметра значение переменной. Для «Скорости» это 3, для остальных — 0.



ВНИМАНИЕ



Значение переменной при запуске игры – это стартовое значение переменной. Потом оно может меняться.



ВНИМАНИЕ

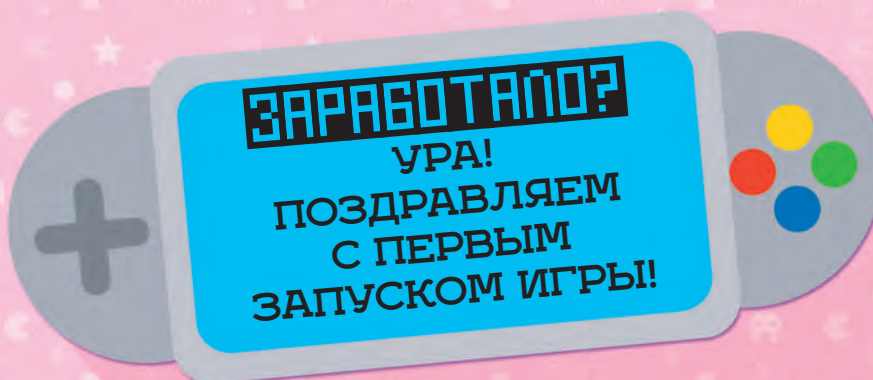


- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Пингвин

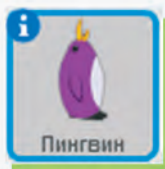
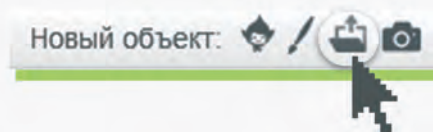
v458 0.4



< ДОБАВЛЯЕМ ПИНГВИНА >

Теперь пора добавить в игру ее главного героя – пингвина. И, конечно же, еду, которую он будет ловить.

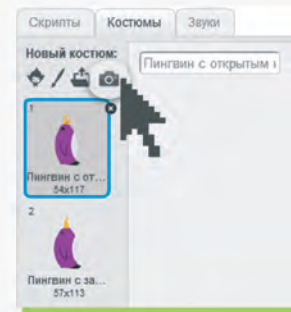
Создай новый спрайт. Нажми на значок папки под игровым полем и загрузи файл «Пингвин с раскрытым клювом.svg» из папки с ресурсами с сайта scratchbook.piter.com.



В списке спрайтов появится пингвин.

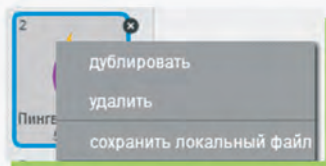
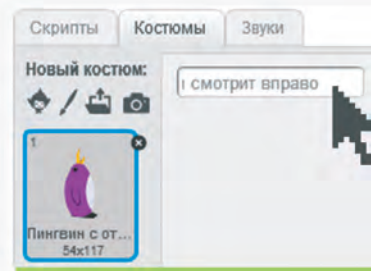
На вкладке «Костюмы» добавим пингвину второй костюм, то есть вариант внешнего вида.

Нажми на значок папки и загрузи файл «Пингвин с закрытым клювом.svg».



Пока наш пингвин – всего лишь неподвижная картинка, но сейчас мы соберем скрипт, который оживит его. Пингвин должен появляться на экране при запуске игры и перемещаться по экрану по команде игрока.

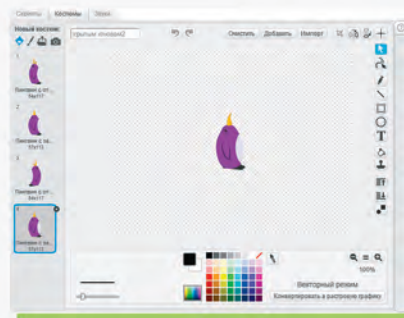
Чтобы не путаться в костюмах, переименуем их. Справа от значка костюма в окошке показывается его название. Щелкни по окошку и допиши: «Пингвин с раскрытым клювом смотрит вправо» и «Пингвин с закрытым клювом смотрит вправо».



Если щелкнуть по костюму правой кнопкой мыши, появится контекстное меню.

Выбери в меню пункт «Дублировать». Появилась копия пингвина! Но нам не нужен еще один точно такой же, поэтому мы его изменим.

С правой стороны экрана находится область графического редактора. Здесь можно самому нарисовать что угодно или изменить загруженный рисунок.





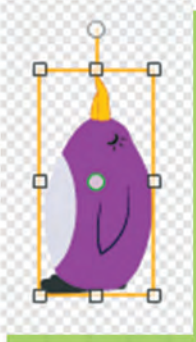
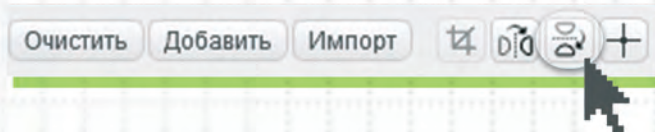
Выдели изображение пингвина полностью. Для этого щелкни по нему – должна появиться желтая рамка вокруг рисунка.

ВНИМАНИЕ



Другой способ выделить весь рисунок – щелкнуть мышкой в любом месте за пределами рисунка и провести по нему.

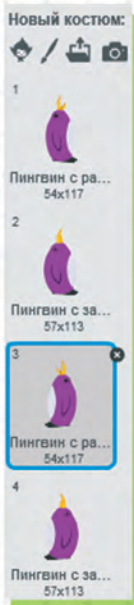
Теперь нажми на значок зеркального отображения картинок.



Пингвин развернулся влево!

Измени название костюма на «Пингвин с закрытым клювом смотрит влево». А теперь сделаем все то же самое, чтобы научить смотреть влево и пингвина с раскрытым клювом:

- * дублируй костюм;
- * в копии разверни пингвина влево;
- * назови его «Пингвин с раскрытым клювом смотрит влево».



У тебя должно получиться четыре костюма пингвина: два с закрытым клювом и два с раскрытым.

ВНИМАНИЕ



Сохрани программу через меню «Файл» – «Сохранить».

СКРИПТ ДВИЖЕНИЯ ПИНГВИНА

Соберем скрипты для управления пингвином.

При старте игры пингвин просто появляется на поле в определенном месте и ждет, когда же его начнут кормить.

когда щелкнут по 

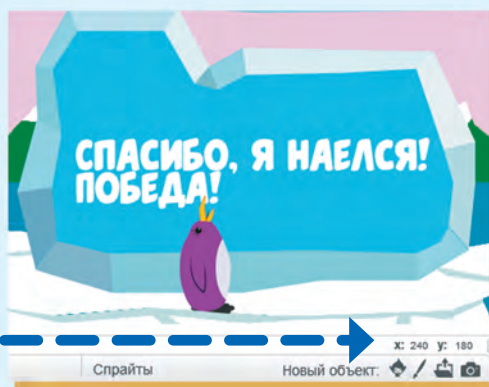
перейти в верхний слой

сменить костюм на Пингвин с раскрытым клювом смотрит вправо

перейти в x: 40 y: -100

Где пингвин будет стоять при старте игры? Решать тебе. Выбери любую точку поля и введи ее координаты в окошки параметров синего блока «Перейти».

Координаты точки, на которую показывает курсор мыши, видно в правом нижнем углу игрового поля.



Пингвин может двигаться вправо и влево по командам — когда игрок нажимает клавиши со стрелками «вправо» и «влево».

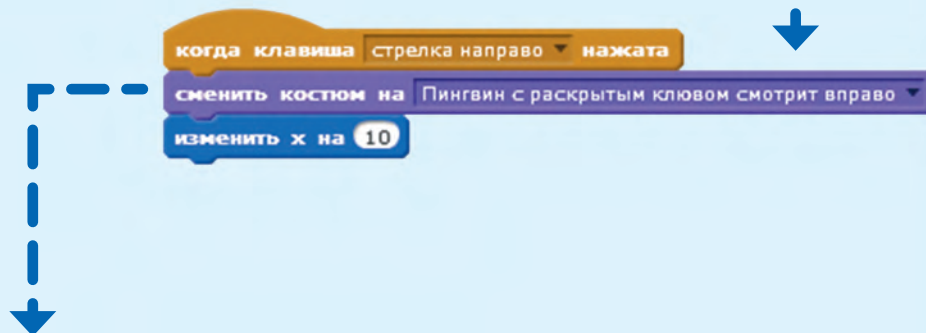
При нажатии клавиши со стрелкой «влево» пингвин поворачивается влево — его костюм меняется на тот, где он смотрит влево. Координата пингвина по оси x уменьшается на 10, а это значит, что он перемещается в левую сторону.

когда клавиша стрелка влево нажата

сменить костюм на Пингвин с раскрытым клювом смотрит влево

изменить x на -10

При нажатии клавиши со стрелкой «вправо» пингвин поворачивается вправо – его костюм меняется на тот, где он смотрит вправо. Координата пингвина по оси x увеличивается на 10 – то есть он движется вправо.



ВНИМАНИЕ



- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



ПИНГВИН ХОДИТ ВПРАВО И ВЛЕВО ПО КОМАНДАМ КЛАВИШ!

ПОЗДРАВЛЯЕМ, ГЛАВНЫЙ ГЕРОЙ ТВОЕЙ ИГРЫ НАУЧИЛСЯ ДВИГАТЬСЯ!

ЗАГРУЖАЕМ ЕДУ

Теперь загрузим в игру еду для нашего пингвина, а то он уже заждался. Для этого нужно создать новые спрайты.

Начнем со здоровой пищи. Нажми на значок папки под игровым полем и загрузи файл «Рыбка.svg» из папки с ресурсами с сайта scratchbook.piter.com.

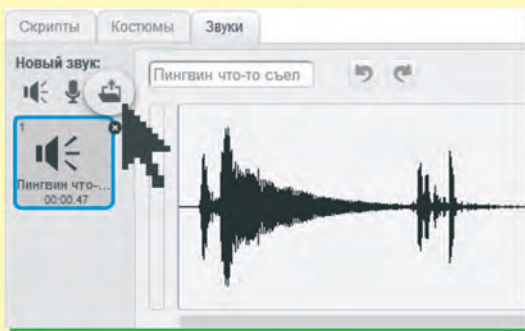
Новый объект:



Спрайт рыбы появился в списке спрайтов под игровым полем.



Загрузим новый звук. На вкладке «Звуки» загрузи файл «Пингвин что-то съел.wav» из папки ресурсов с сайта scratchbook.piter.com.



Соберем скрипты для рыбки. Рыбка все время падает сверху, и пингвин должен ее ловить. Когда игра заканчивается из-за того, что истекло время или пингвин объелся сладостями, рыба исчезает с игрового поля.



Тебе понадобятся блоки со вкладок: «События», «Управление», «Внешность», «Данные», «Операторы», «Сенсоры».



Рыбка должна падать вертикально вниз, поэтому при старте игры она один раз выполняет команду «Повернуться».

Затем рыбка раз за разом, пока не закончится игра, появляется наверху игрового поля и движется вниз. За повторяющиеся команды отвечает блок «Всегда».

Координаты точки, в которой появляется рыбка, определяются двумя числами: по оси x и y .

По оси x координата выбирается случайно. Рыбка может появиться в любой точке игрового поля от левого до правого края. Для этого перетаски с зеленой вкладки «Операторы» блок «Выдать случайное» и добавь его к блоку «Перейти». Введи нужные числа.

перейти в x : выдать случайное от -220 до 220 y : 150

По оси y координата всегда постоянная – рыбка должна появляться наверху экрана, а не возникнуть из воздуха посреди него.

ВНИМАНИЕ

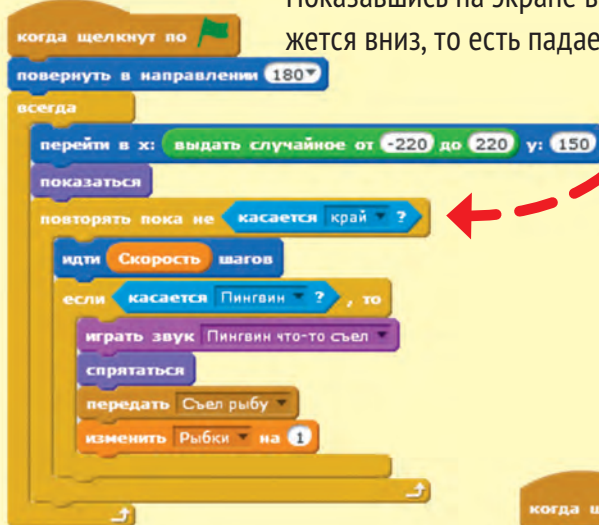


Поворачивается не костюм (картинка) спрайта, а сам спрайт. Это нужно для того, чтобы спрайт правильно выполнял команду «Идти... шагов». В какую сторону при этом будет смотреть голова рыбы, зависит от костюма.

Для этого щелкни по рыбке или обведи ее мышкой, а затем наведи курсор на кружок на линии. Щелкни по нему мышкой и, не отпуская, тяни в любую сторону. Рыбка будет поворачиваться на месте.



Показавшись на экране в случайной точке, рыба движется вниз, то есть падает.



Рыбка движется по направлению «(180) вниз», указанному в блоке «Повернуть». Когда рыбку только бросают сверху, в первое мгновение у нее нулевая скорость. Поэтому при первом исполнении цикла «Повторять пока не» спрайт рыбы делает 0 шагов. Дальше рыбка начинает падать с ускорением. После первого исполнения цикла скорость спрайта увеличивается на 0,2 шага.

Если, сделав несколько шагов, рыба коснулась пингвина, то он ее съедает. При этом в программе происходит сразу несколько действий.

- * Проигрывается звук «Пингвин что-то съел».
- * Рыбка исчезает — пингвин съел ее.
- * Скрипт передает всем сообщение «Пингвин что-то съел».
- * Число съеденных рыбок, которое запоминает переменная «Рыбки», увеличивается на 1.

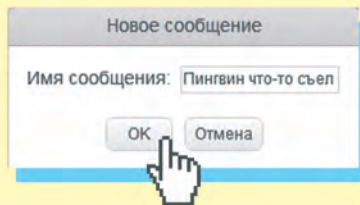
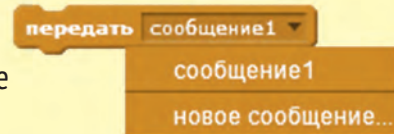


Если после выполнения команды «Иди «скорость» шагов» рыбка не коснется пингвина, блок «Если касается» исполняться не будет.

Тогда спрайт рыбы снова должен сделать несколько шагов, как командует ему блок «Повторять пока не «Касается край»».

Когда рыбка касается нижнего края экрана, программа возвращается к первой команде в блоке «Всегда», и рыбка снова появляется в случайной точке наверху экрана.

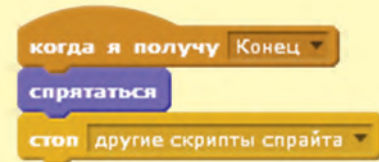
Возможно, ты обратил внимание, что скрипт должен передать сообщение «Пингвин что-то съел», а мы его еще не создали. Нужно это сделать. Чтобы создать сообщение «Пингвин что-то съел», перетащи блок «Передать» со вкладки «События» в поле скриптов и щелкни по стрелке в окошке параметров.



Откроется контекстное меню. Выбери пункт меню «Новое сообщение». Введи любое имя и нажми ОК. Удобно называть сообщения так, чтобы было понятно, о чем они сообщают спрайтам.

Добавим рыбе еще один скрипт, чтобы в конце игры падающая сверху еда не зависала в воздухе.

Создай сообщение «Конец» тем же способом, что и предыдущее – через контекстное меню в окошке параметра блока.



ВНИМАНИЕ



- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



ДОБАВЛЯЕМ ВРЕДНУЮ ЕДУ

Посетители зоопарка любят сладкое. Они так и норовят подкинуть пингвину конфеты и торты. Но ему нельзя их есть ни за что на свете!

ВНИМАНИЕ



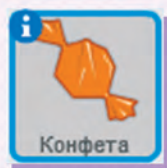
Зачем в игре нужны сложности? Для того чтобы она получилась интересной. Если все слишком просто, если игроку не встречаются опасности, играть попросту скучно.

Добавим спрайты конфет и торта и посмотрим, что будет происходить с пингвином. Нажми на значок папки под игровым полем и загрузи файл «Конфета.svg» из папки с ресурсами с сайта scratchbook.piter.com.

Новый объект:



Спрайт конфеты появился в списке спрайтов под игровым полем.



Соберем скрипт для конфеты. Он похож на скрипт рыбы — конфета тоже появляется в случайном месте экрана и падает с ускорением. Только у нее своя скорость, за которую отвечает отдельная переменная.



Для того чтобы рыба и конфета не появлялись одновременно, добавим задержку времени перед первым появлением конфеты.

Пришло время для торта! Нажми на значок папки под игровым полем и загрузи файл «Торт.svg» из папки с ресурсами с сайта scratchbook.piter.com.



Спрайт торта появился в списке спрайтов под игровым полем.



Соберем скрипт для торта. Он похож на скрипт рыбы и конфеты – торт тоже появляется в случайном месте экрана и падает с ускорением. У него тоже своя отдельная переменная для значения скорости и задержка времени.



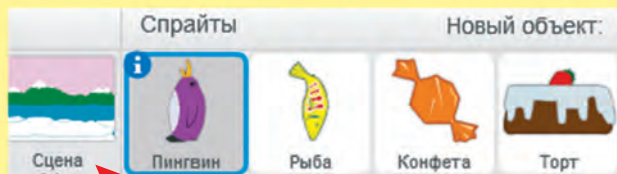
ВНИМАНИЕ



Обрати внимание на разные значения координаты y для конфет, рыбки и торта. У первых двух спрайтов $y = 150$, у торта $y = 140$. Дело в разных размерах спрайтов. Торт больше, и, если его поставить при первом появлении выше, он будет задевать верхний край экрана. Сработает условие «касается край?», и торт будет раз за разом появляться наверху экрана, вместо того чтобы падать вниз.

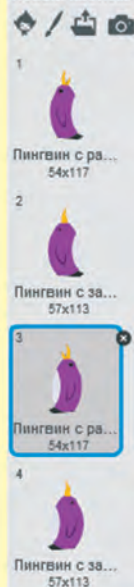
ЧТО ПРОИСХОДИТ, КОГДА ПИНГВИН ЧТО-ТО СЪЕЛ?

Вернемся в скрипты пингвина. Для этого надо просто щелкнуть по пингвину в списке спрайтов под игровым полем.



Когда пингвин съедает угощение, оно не должно просто исчезать. Научим нашего героя есть почти по-настоящему и соберем для этого еще один скрипт.

Новый костюм:

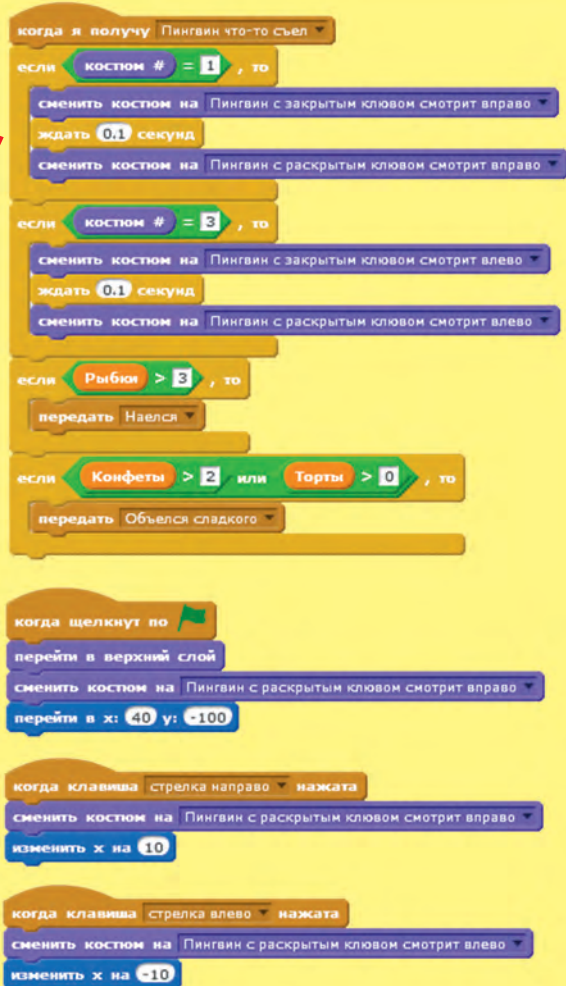


Во-первых, поймав еду, пингвин должен закрыть клюв. Для этого мы будем менять костюм нашего героя – вместо пингвина с раскрытым клювом появится пингвин с закрытым клювом. Если пингвин с открытым клювом смотрел вправо, то должен загрузиться соответствующий костюм с закрытым клювом, и так же для левого варианта.

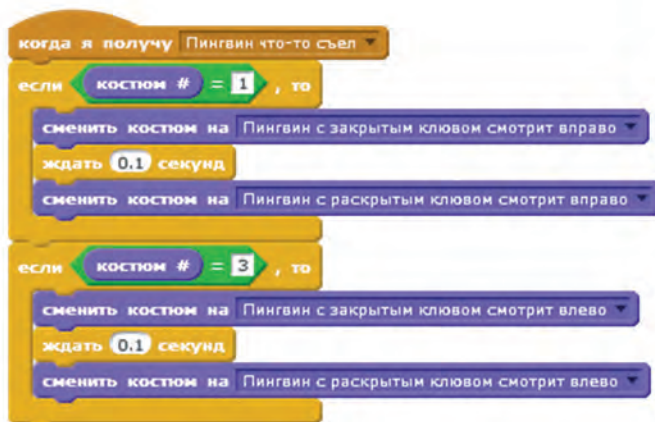
ВНИМАНИЕ



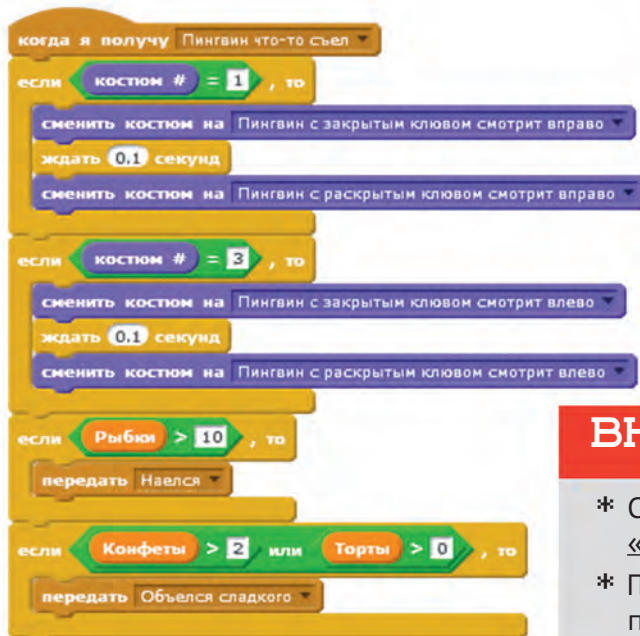
Проверь, под какими номерами стоят костюмы в списке на вкладке «Костюмы». Если порядок костюмов надо поменять, перетаскивай их мышкой.



Соберем скрипт для смены костюма. Когда скрипт получает сообщение, что пингвина коснулись рыба, конфета или торт, наш подопечный будет закрывать рот на долю секунды и потом снова открывать.



Добавим к скрипту команды для проверки количества съеденных рыбок, конфет и тортов.



Получив сообщение о касании рыбы, конфеты или торта, скрипт будет проверять, сколько каждого из угощений пингвин уже съел. Если значение переменной «Рыбки» 5 или меньше, ничего не происходит. Как только рыбок становится 6, скрипт передает сообщение «Наелся» – пингвин сыт, игрок достиг цели и одержал победу.

То же самое происходит с подсчетом конфет и тортов. Пока конфет меньше 2 или 2, а тортов 0, ничего не происходит. Как только пингвин съедает третью конфету или один торт, скрипт отправляет сообщение «Обьелся сладкого» – игра завершается поражением игрока, у пингвина болит живот.

ВНИМАНИЕ



- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.

СООБЩЕНИЯ О КОНЦЕ ИГРЫ ПО ОЧКАМ

Когда скрипт отправляет сообщения «Наелся» или «Объелся сладкого», игра завершается. Но ее конец нужно обозначить не только для программы, но и для игрока. Для этого создадим новые спрайты с уведомлениями о результате игры. Загрузи файлы «Сообщение Наелся.svg» и «Сообщение Объелся.svg» из папки с ресурсами с сайта scratchbook.piter.com.

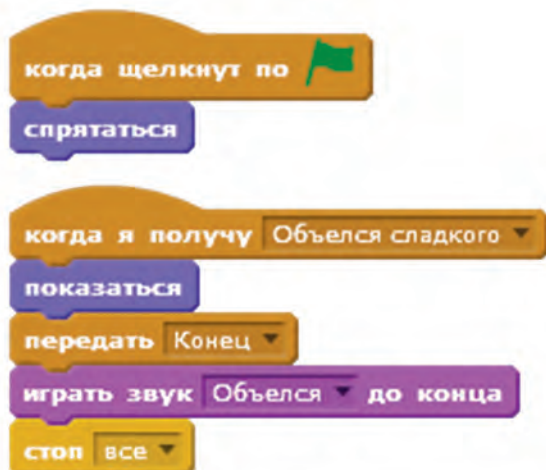
Новый объект:



Спрайты сообщений появятся в списке спрайтов под игровым полем.

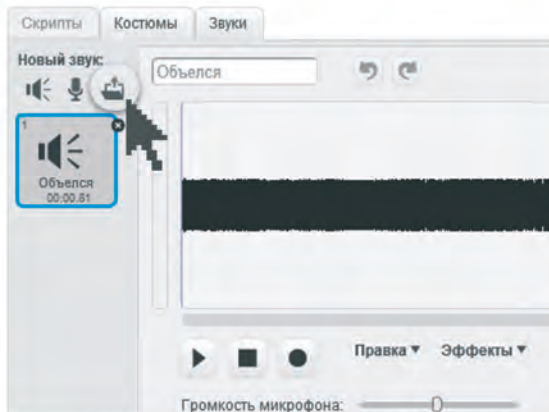


Соберем скрипт для окончания игры в случае, если пингвин съел слишком много сладкого.



При старте игры спрайт «Объелся» скрывается. Но если спрайт получает сообщение «Объелся сладкого», то он показывается на экране и передает всем спрайтам сообщение «Конец», чтобы рыбки, конфеты и торты перестали падать, а фоновая музыка затихла. Сам же спрайт проигрывает звук «Объелся».

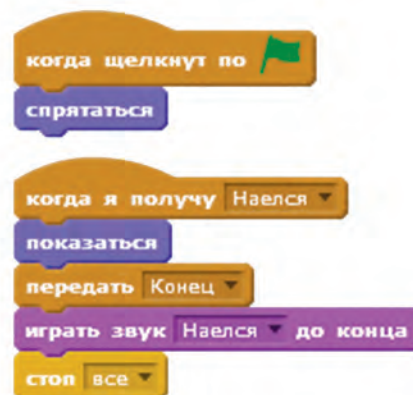
Для того чтобы добавить звук «Объелся», загрузи файл «Объелся.wav» на вкладке «Звуки» из папки ресурсов с сайта scratchbook.piter.com.



Соберем похожий скрипт для случая, когда игрок справился с задачей и пингвин сыт. Щелкни по изображению спрайта «Наелся» в списке спрайта под игровым полем.



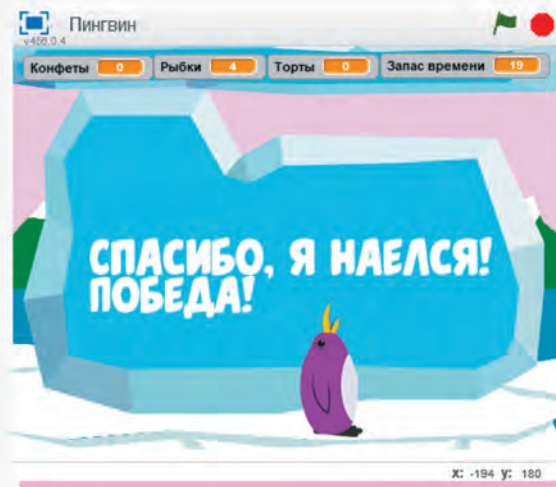
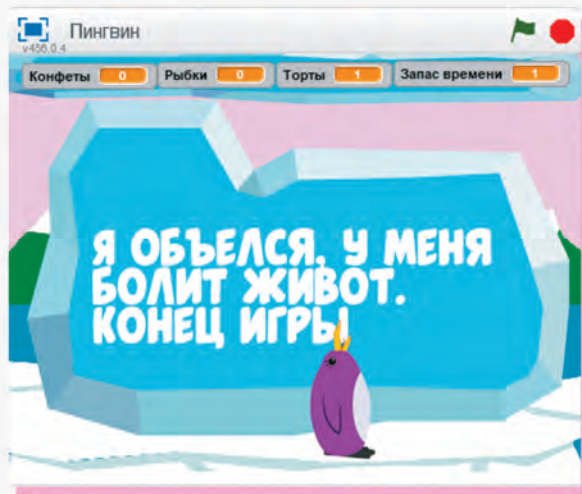
Его скрипт выглядит так:



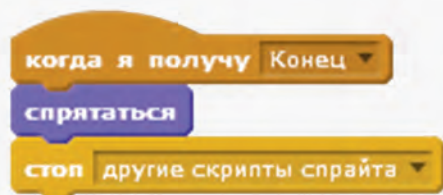
Для того чтобы добавить звук «Наелся», загрузи файл «Наелся.wav» на вкладке «Звуки» из папки ресурсов с сайта scratchbook.piter.com.

◀ ПИНГВИН ПОЯВЛЯЕТСЯ ▶ НА ЭКРАНЕ В КОНЦЕ ИГРЫ

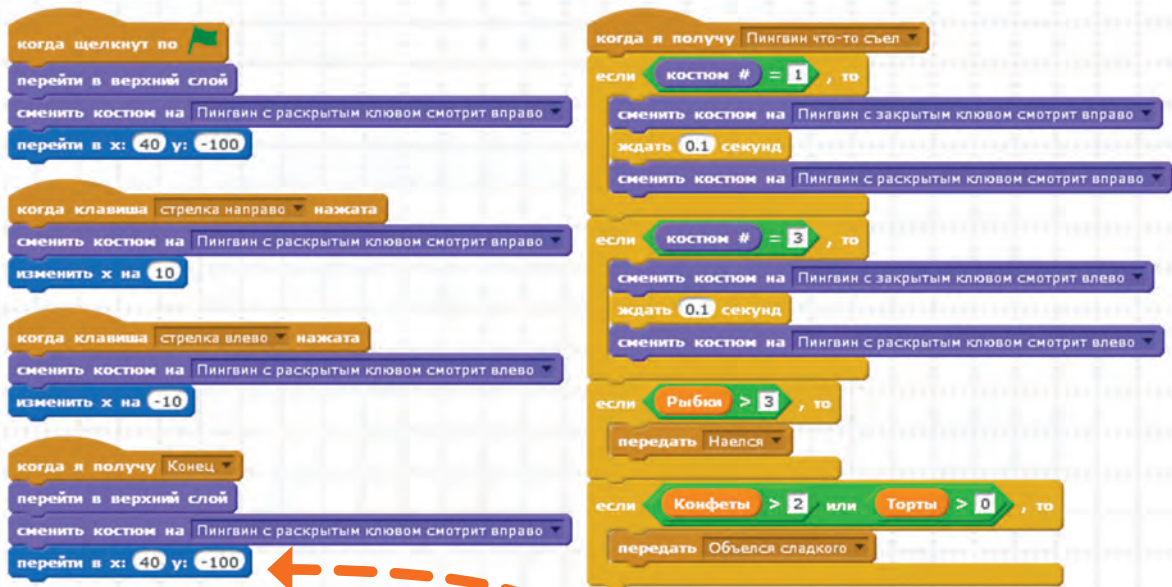
В конце игры появляются такие картинки:



Спрайты, которые получили сообщение «Конец», при этом скрываются. Вот зачем мы создавали для рыбки, конфет и торта эти скрипты:

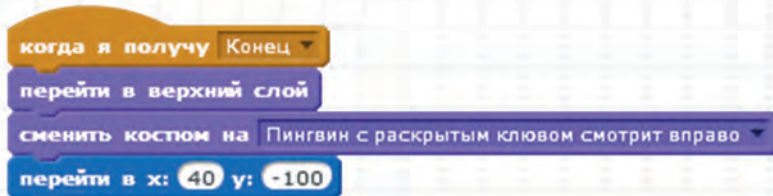


Что происходит в это время с пингвином? Он тоже получает сообщение «Конец», но, наоборот, появляется в центре игрового поля, чтобы попрощаться с игроком.



Вернемся в скрипты пингвина и добавим ему еще один короткий скрипт.

Теперь, даже если финал игры застал пингвина на краю экрана развернутым вбок, он вернется на передний план, как только получит сообщение «Конец».



ВНИМАНИЕ



- * Сохрани программу через меню «Файл» — «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Пингвин

v455 0 4





СЧЕТЧИК ВРЕМЕНИ И КОНЕЦ ИГРЫ ПО ВРЕМЕНИ

Играть можно было бы бесконечно, но... голодный пингвин слишком долго ждать не станет. У нашей игры есть конечное время, за которое его нужно накормить. Создадим новый спрайт. Загрузи файл «Сообщение Есть хочу.svg» из папки с ресурсами с сайта scratchbook.piter.com.

Новый объект:



Спрайт появится в списке спрайтов под игровым полем. Соберем для него скрипт.



При старте игры картинка с сообщением «Есть хочу! Конец игры» будет скрываться. Затем начнется выполнение цикла. Подождав секунду, программа уменьшит значение переменной на 1.



Вспомним, где мы задали значение этой переменной при старте игры — в скрипте фона:



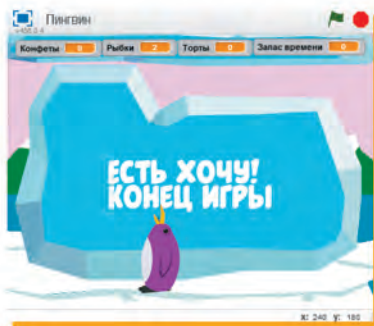
Здесь его можно изменять при разработке игры.

ВНИМАНИЕ

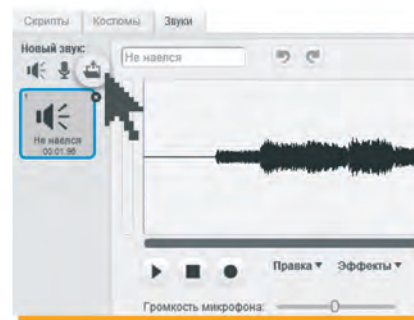


Экспериментируй с разными значениями времени игры.

Когда значение переменной «Запас времени» меняется, скрипт проверяет, не равно ли это значение нулю. Если нет, то он ждет еще секунду, снова изменяет значение переменной и проверяет его. Если оно равно нулю, то скрипт показывает картинку спрайта, передает всем сообщение «Конец», чтобы движение других спрайтов остановилось, и проигрывает звук «Не наелся». Игра заканчивается. Получится так:



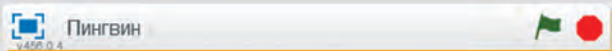
Для того чтобы добавить звук «Объелся», загрузи файл «Объелся.wav» на вкладке «Звуки» из папки ресурсов с сайта scratchbook.piter.com.



ВНИМАНИЕ



- * Сохрани программу через меню «Файл» — «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.

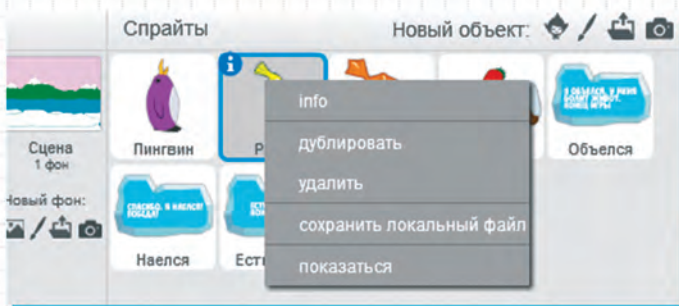


БОНУС – ВТОРАЯ РЫБКА

Что, если в зоопарк придет больше посетителей и пингвину будут бросать больше рыбы?

Тогда игроку удастся накормить его быстрее!

Добавим второй спрайт рыбы. Для этого нажми правой кнопкой мыши на спрайт рыбы и выбери пункт меню «Дублировать».

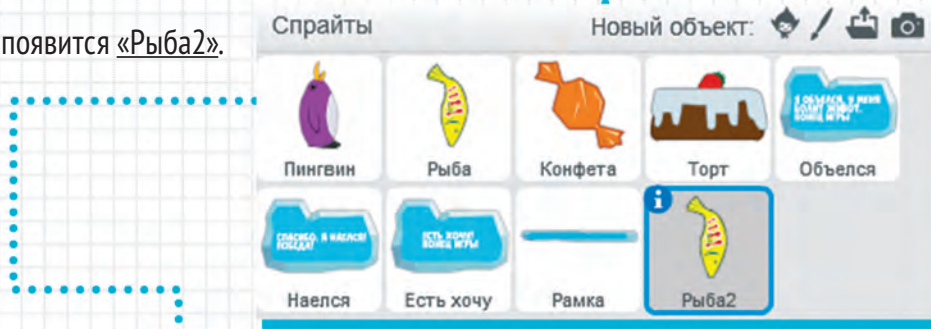


ВНИМАНИЕ



Сохрани программу через меню «Файл» – «Сохранить».

В списке спрайтов появится «Рыба2».



Когда спрайт дублируется, создается полная его копия с теми же самыми костюмами и, главное, скриптами. Это очень удобно – не надо заново собирать повторяющиеся скрипты.

Запусти игру. Что происходит?



Две рыбы появляются наверху экрана одновременно и на одной линии по горизонтали, только в разных местах. Нам нужно, чтобы они падали по очереди.

Для этого перейдем в скрипты спрайта «Рыба2» и добавим задержку по времени, чтобы рыбы появлялись на экране не одновременно.

ВНИМАНИЕ



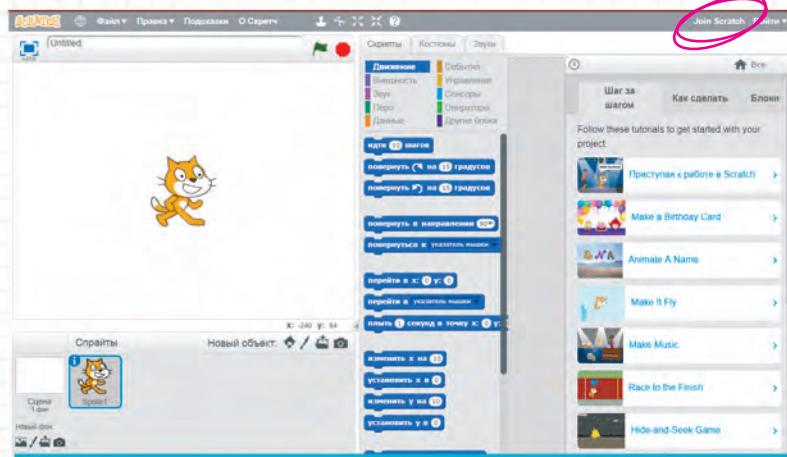
Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.

Пингвин

ВНИМАНИЕ



Экспериментируй с разными значениями скорости предметов, времени игры и времени задержки, чтобы найти самое лучшее сочетание. Игра должна быть азартной, но проходимой.

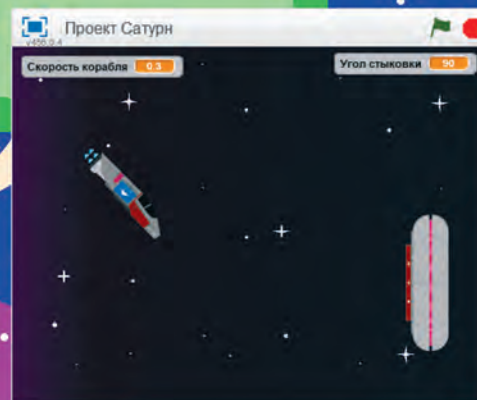


УРА! ИГРА ГОТОВА!

Если у тебя получилась хорошая игра, ты наверняка захочешь поделиться ею с другими. И тебе не обязательно ограничиваться при этом своими друзьями! Если ты зарегистрируешься на сайте Скретч, то сможешь сохранять свои проекты и показывать их другим пользователям.

ПРОЕКТ «САТУРН»

Космический челнок проекта «Сатурн» возвращается на станцию после трудной экспедиции, в которой мужественные астронавты исследовали опасную планету. Сможешь ли ты провести точную стыковку челнока со станцией, которая медленно движется в космосе?



Игра «Проект „Сатурн“»

Сложность: * *



Время сборки: 1,5–2 часа

Видеоролик игры можно найти по адресу: scratchbook.piter.com

Описание

Космический корабль приближается к станции для стыковки. Его надо аккуратно подвести к нужному месту: так, чтобы люк красного стыковочного отсека точно совпал с красным стыковочным шлюзом станции. Если корабль будет двигаться слишком быстро, если он коснется корпуса станции или угол стыковки окажется слишком большим, произойдет катастрофа. Игроку понадобятся терпение и внимание, чтобы не допустить гибели челнока!

Правила игры

- * Цель игрока — успешно провести стыковку челнока и станции.
- * Играет один игрок.
- * Игрок управляет челноком при помощи клавиш со стрелками. Клавиши ВПРАВО и ВЛЕВО — поворот челнока по часовой и против часовой стрелки, ВВЕРХ и ВНИЗ — ускорение и замедление.
- * Челнок не может двигаться назад.
- * Станция медленно движется по экрану и при этом разворачивается.
- * Для того чтобы стыковка была успешной, нужно посадить корабль на станцию под углом меньше 5 и со скоростью меньше 3.

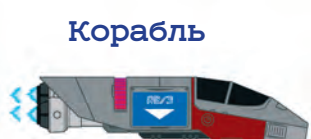


Все необходимые для игры ресурсы можно скачать по адресу:
scratchbook.piter.com.

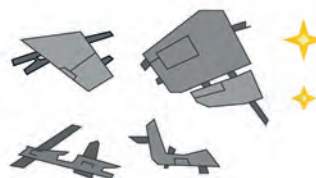
ВОТ КАКИЕ СПРАЙТЫ ПОНАДОБЯТСЯ ТЕБЕ В ЭТОЙ ИГРЕ



Фон

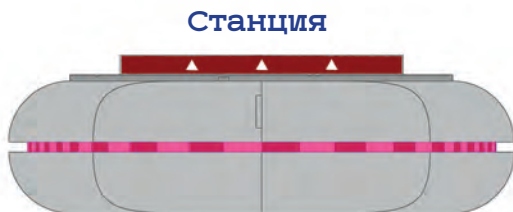
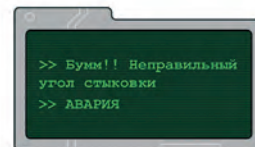
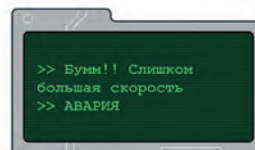
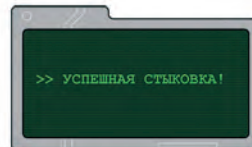
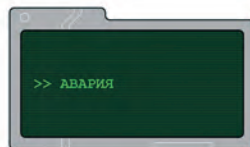
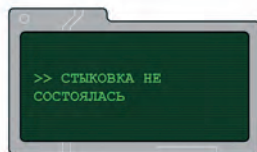


Корабль



и его обломки

Сообщения о событиях



Станция



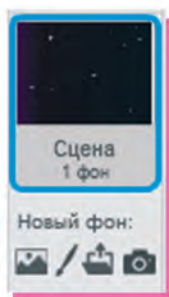
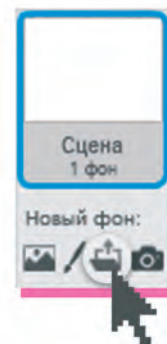


ЗАГРУЖАЕМ РЕСУРСЫ

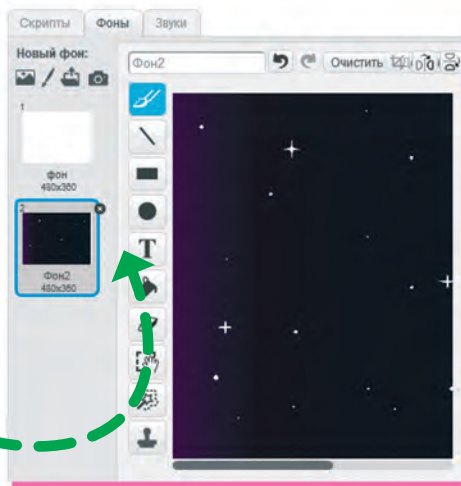


Для того чтобы загрузить в игру фоновую картинку с видом на просторы космоса, нажми на значок папки под сценой слева.

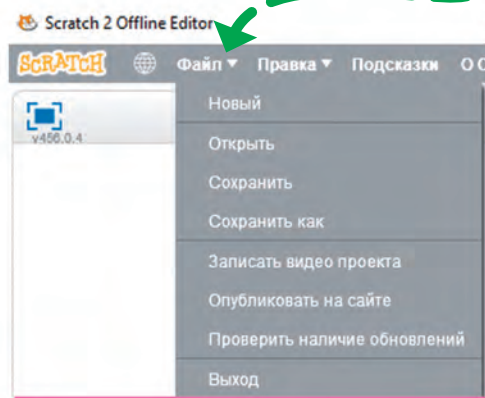
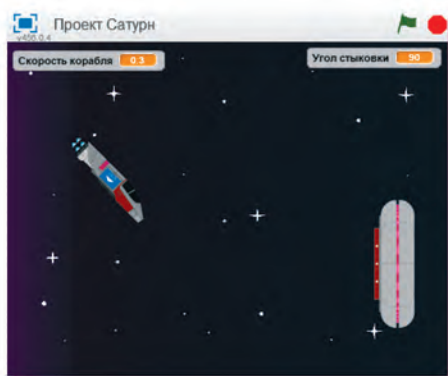
Загрузи файл «Фон.jpg» из папки ресурсов с сайта scratchbook.piter.com. В поле слева внизу и в поле справа на вкладке «Фоны» появится новая картинка.



Ненужный фон
можно удалить.



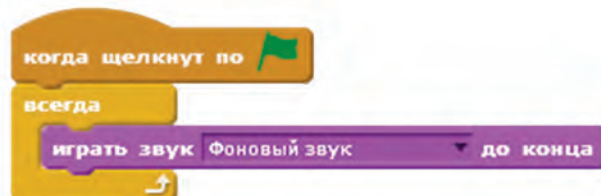
Теперь наша опасная миссия будет выполняться на фоне темных просторов космоса. Сохраним программу, чтобы не потерять сделанную работу. Нажми на пункт верхнего меню «Файл».



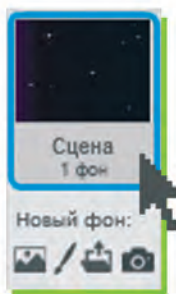
СОЗДАЕМ ФОНОВЫЙ СКРИПТ

Как ты помнишь, фон – это тоже спрайт. К нему можно привязать выполнение действий, которые происходят не с конкретными героями или объектами, а со всей игрой. Перейдем на вкладку «Скрипты». Здесь мы будем собирать программу. Для сцены и для каждого из героев собирается отдельная программа-скрипт.

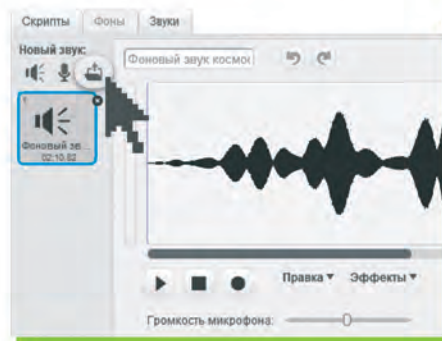
Для начала создадим скрипт, в котором запускается фоновый звук для игры «Проект „Сатурн“».



Щелкни на значке сцены в правом нижнем углу.



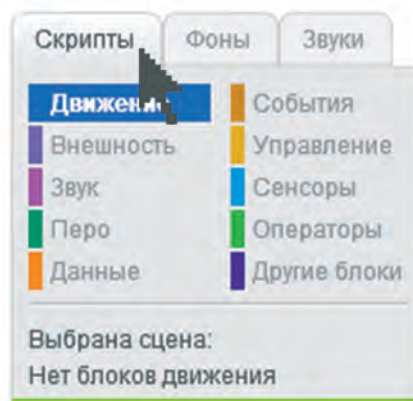
На вкладке «Звуки» загрузи файл «Фоновый звук» из папки ресурсов с сайта scratchbook.piter.com.



Ненужный звук можно удалить, нажав на крестик.



На вкладке «Скрипты» в поле справа соберем скрипт для фона.



Нам понадобятся вкладки «События», «Управление» и «Звуки». Перетащи с них нужные блоки, поставь их по порядку и выбери в окошке параметров команды «Играть» название файла «Фоновый звук.wav».

ВНИМАНИЕ



- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Проект Сатурн

v456 0.4



★ ЗАРАБОТАЛО? УРА!
ПОЗДРАВЛЯЕМ
С ПЕРВЫМ ЗАПУСКОМ ИГРЫ! •

ДОБАВЛЯЕМ КОСМИЧЕСКУЮ СТАНЦИЮ

Основными «героями» в нашей игре будут космическая станция и челнок. Сейчас мы добавим их в программу, и они появятся на экране. Создай новый спрайт. Нажми на значок папки под игровым полем и загрузи файл «Станция.svg» из папки с ресурсами с сайта scratchbook.piter.com.

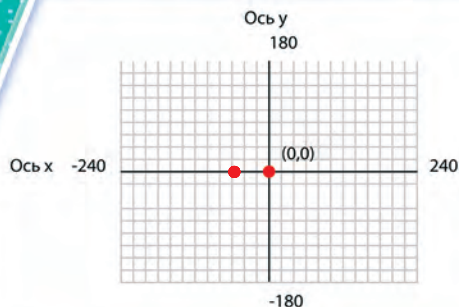


В списке спрайтов появится станция.



Соберем скрипт для станции, чтобы запустить ее в космос. Она должна появляться на экране при запуске игры и затем двигаться по нему, одновременно поворачиваясь.



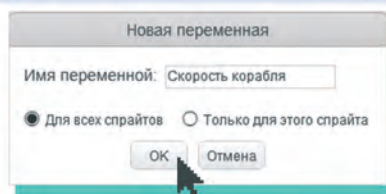
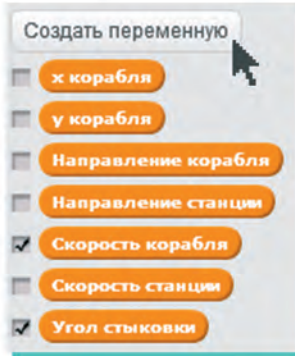


При старте игры станция будет появляться в нижнем правом углу экрана.

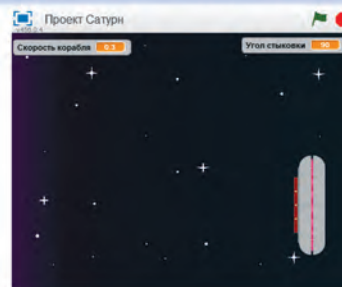
СОБИРАЕМ СКРИПТ СТАНЦИИ

Для сборки скрипта нам понадобятся вкладки «События», «Управление», «Данные», «Внешность», «Движение», «Операторы».

Сначала создадим переменные для скрипта станции и для всей игры. Для этого надо открыть вкладку «Данные» и нажать кнопку «Создать переменную».

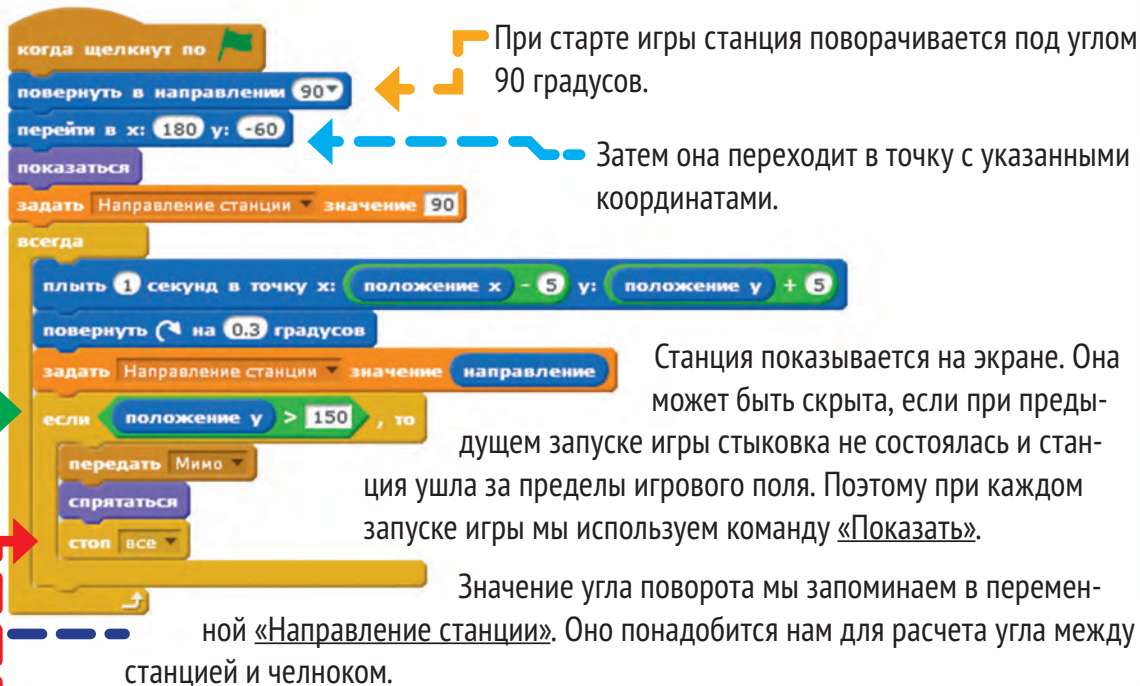


Откроется новое окошко. Введи нужное имя переменной и нажми ОК.



Создай семь переменных, как на рисунке сверху, и отметь галочками две из них, которые должен видеть игрок: «Скорость корабля» и «Угол стыковки». Они появятся на игровом поле. Перетащи их наверх экрана.

Давай разберемся: что происходит в скрипте станции?



После этого начинается выполнение цикла. Во время каждого повтора цикла станция совершает движение по экрану. Команда «Плыть» отличается от команды «Перейти» тем, что спрайт будет перемещаться по экрану постепенно и плавно, а не мгновенно оказываться в другой точке. Чем больше значение времени в окошке «<> секунд», тем медленнее будет «Плыть» спрайт.

При каждом повторе координаты станции меняются: она смещается на 5 точек влево (координата x уменьшается на 5) и на 5 точек вверх (координата y увеличивается на 5).

Затем станция совершает поворот по часовой стрелке. Значение угла ее поворота меняется. Мы запоминаем его в переменной «Направление станции».

После этого проверяем, не дошла ли станция до верхнего края игрового экрана.

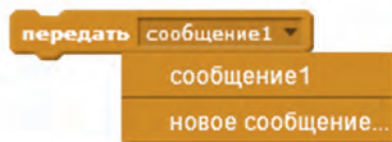
Если координата y станции не больше 150, то ничего не происходит и цикл повторяется еще раз. Если станция вышла за верхний край, то скрипт станции отправляет сообщение «Мимо», а станция пропадает с экрана. Выполнение скриптов спрайта останавливается.

ВНИМАНИЕ

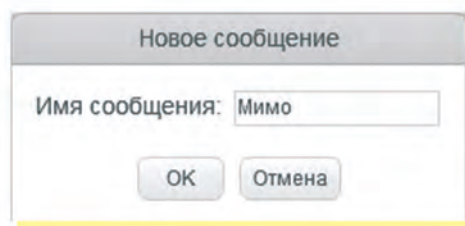


Внутри Скретча значение угла поворота для спрайта всегда хранится в синей переменной «Направление». Но она действует только для своего спрайта. Если в программе нужно будет использовать углы поворота для нескольких спрайтов, то сделать это при помощи синих переменных «Направление» не получится – мы не сможем собрать их в одном скрипте. Поэтому мы создали отдельные переменные для углов поворота станции и корабля и будем присваивать им значение синих переменных «Направление».

Но чтобы передать сообщение «Мимо», его надо сначала создать. Перетащи блок «Передать» со вкладки «События» в поле скриптов и щелкни по стрелке в окошке параметров.



Откроется контекстное меню. Выбери пункт меню «Новое сообщение». Введи любое имя сообщения и нажми ОК. Удобно называть сообщения так, чтобы было понятно, о чем они сообщают спрайтам.



ВНИМАНИЕ



- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.

Проект Сатурн



**СТАНЦИЯ
ДВИГАЕТСЯ ПРИ
ЗАПУСКЕ ИГРЫ!
ПОЗДРАВЛЯЕМ!**

ЗАГРУЖАЕМ КОСМИЧЕСКИЙ КОРАБЛЬ

Чтобы в игре появился космический корабль, нам понадобятся новые спрайты. Нажми на значок папки под игровым полем и загрузи файл «Корабль.svg» из папки с ресурсами с сайта scratchbook.piter.com.

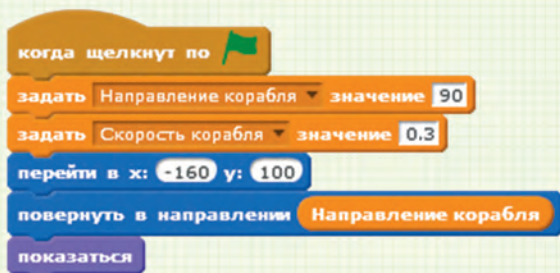
Новый объект:



Спрайт корабля появился в списке спрайтов под игровым полем.

Теперь соберем скрипты, которые будут управлять нашим кораблем. Нам понадобятся блоки со вкладок: «События», «Управление», «Внешность», «Движение», «Операторы».

При старте игры корабль появляется на верхнем правом углу экрана и получает определенные значения скорости и направления.

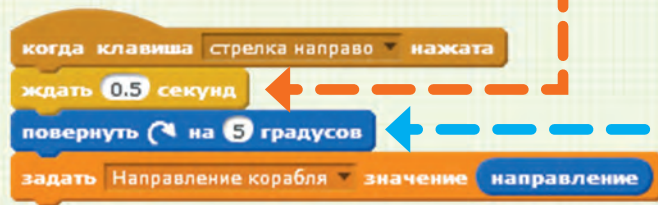


Корабль неуклонно движется к станции, а игрок должен управлять им: поворачивать, ускорять и замедлять. За исполнение этих команд отвечают четыре скрипта.

ВНИМАНИЕ

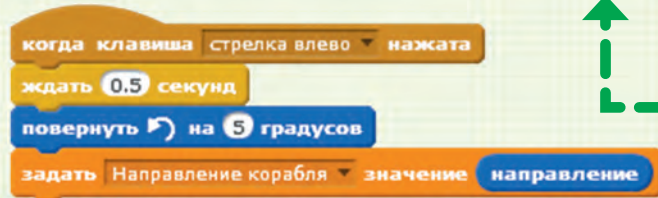


Измени стартовые координаты корабля, если хочешь, чтобы он появлялся в другом месте.

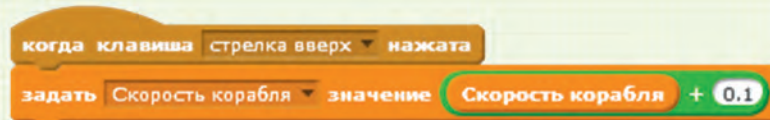


Когда игрок нажимает клавишу «вправо», корабль поворачивается по часовой стрелке на 5 градусов.

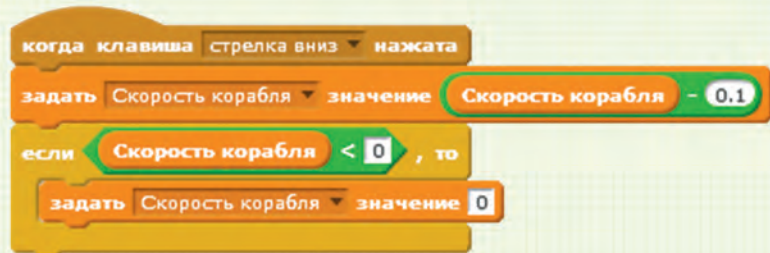
Но перед этим проходит полсекунды, ведь огромный космический корабль не может повернуться мгновенно.



Так же, как в случае с космической станцией, мы запоминаем значение направления, в котором движется корабль, в специальной переменной «Направление корабля».



При нажатии стрелок «вверх» и «вниз» скорость корабля меняется.



Корабль не может двигаться назад, поэтому при нажатии стрелки «вниз» скрипт проверяет значение скорости. Если она стала меньше нуля, скрипт назначает ей значение 0.

СОВЕТ

Поэкспериментируй с углом поворота, временем задержки поворота, скоростью корабля. От всего этого зависит, насколько сложной получится игра.

ВНИМАНИЕ

- * Сохрани программу через меню «Файл» — «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Проект Сатурн

v455.0.4

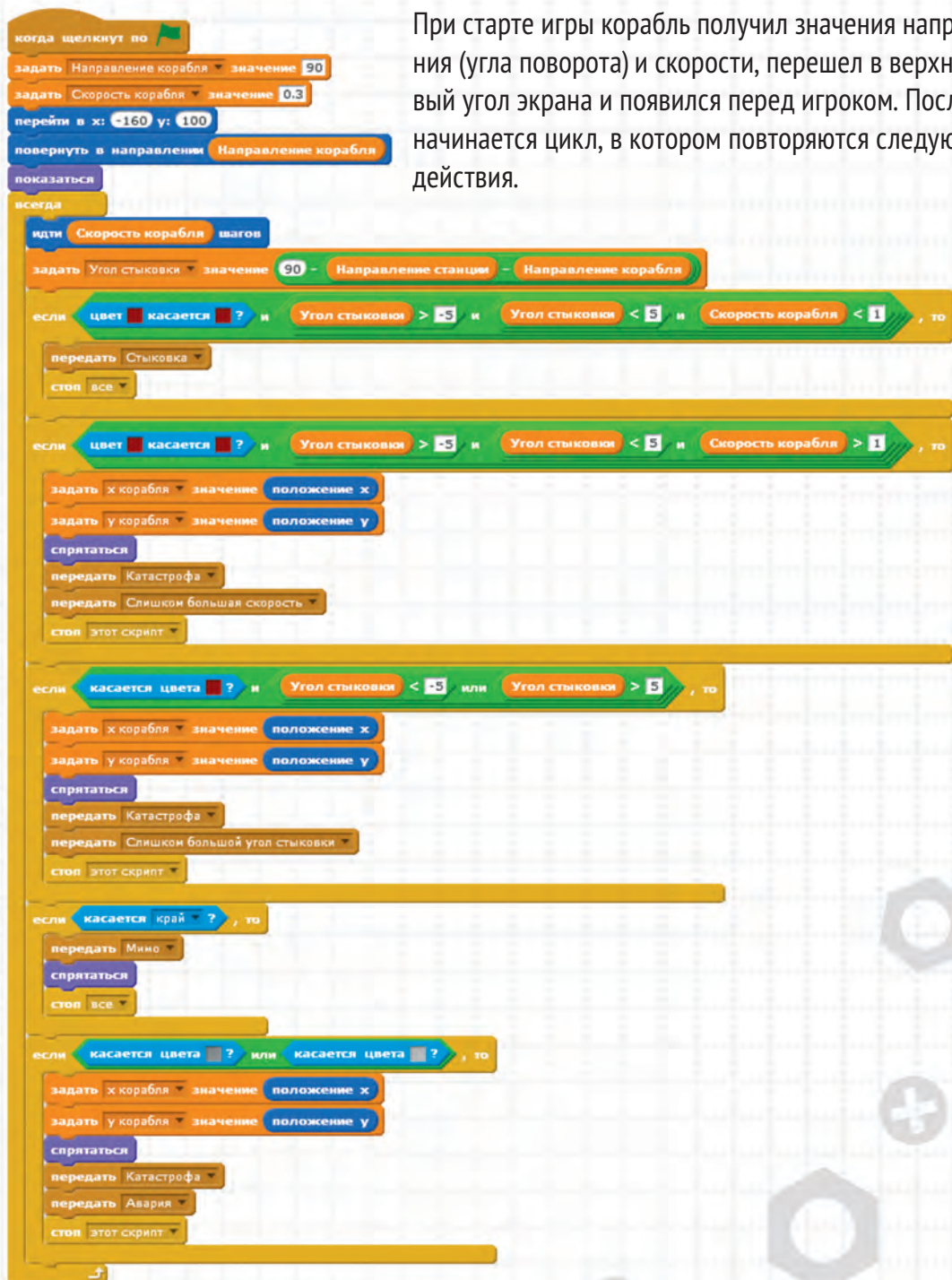


КОРАБЛЬ ИДЕТ НА СТЫКОВКУ

✕ Для того чтобы корабль не просто поворачивался на месте, но и двигался по экрану, добавим к первому скрипту корабля несколько команд. Они выглядят сложными, но на самом деле все просто. Каждый блок «Если...» отвечает за проверку, произошла ли стыковка или челнок врезался в станцию. Поехали?

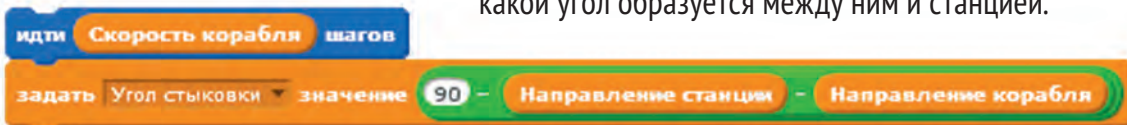
В ИГРЕ МОЖЕТ БЫТЬ НЕСКОЛЬКО ИСХОДОВ

- * Корабль правильно стыкуется со станцией. Это произойдет, если корабль и станция соприкоснутся друг с другом красными участками — «стыковочными шлюзами». При этом угол между кораблем и станцией не должен быть слишком большим, и корабль должен подойти к станции на достаточно малой скорости. Стыковка прошла успешно, победа!
- * Корабль подходит к станции под слишком большим углом. Они сталкиваются, корабль разлетается на куски. Конец игры.
- * Корабль подходит к станции слишком быстро. Они сталкиваются, и корабль разлетается на куски. Конец игры.
- * Корабль подходит к станции не «шлюз к шлюзу», а другими частями корпуса. Они сталкиваются, и корабль разлетается на куски. Конец игры.
- * Корабль проходит мимо станции и вылетает за пределы экрана. Стыковки не произошло. Конец игры.
- * Станция выходит за пределы экрана. Стыковки не произошло. Конец игры.
- * Последний из этих исходов — с выходом станции за экран — мы обработали в скрипте станции (см. стр. 69). Теперь займемся кораблем.



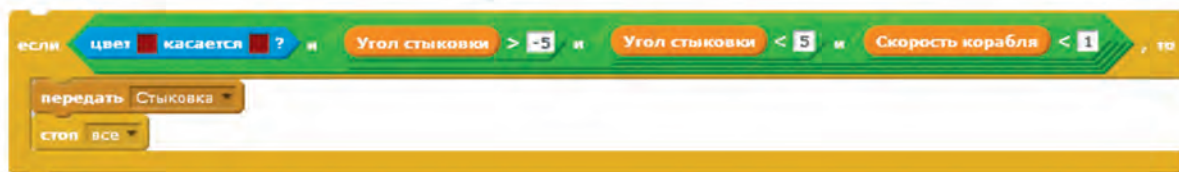
При старте игры корабль получил значения направления (угла поворота) и скорости, перешел в верхний левый угол экрана и появился перед игроком. После этого начинается цикл, в котором повторяются следующие действия.

Корабль движется — спрайт совершает несколько шагов в соответствии с заданной скоростью. Скрипт считает, какой угол образуется между ним и станцией.

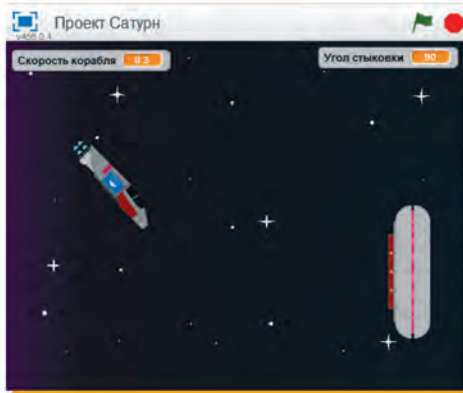


Теперь скрипт проверяет, не касается ли красный шлюз корабля красного шлюза станции. При этом значение угла между кораблем и станцией должно быть больше -5 градусов и меньше 5 градусов. Скорость корабля должна быть меньше единицы, иначе катастрофы не избежать.

ПРОВЕРКА № 1



Для того чтобы указать цвет в голубом блоке сенсора, щелкни курсором по первому цвету. Стрелка курсора превратится в изображение руки с указательным пальцем. Щелкни им по нужному фрагменту на корпусе корабля.



Проделай то же самое со станцией. Щелкни курсором по второму цвету, потом по красному цвету на шлюзе станции.

Собрать сложное условие «и...и...и» можно из нескольких блоков. Их надо поместить друг в друга.

Если корабль касается станции и выполняются все условия, то скрипт передает сообщение «Стыковка» и останавливает выполнение всех скриптов.

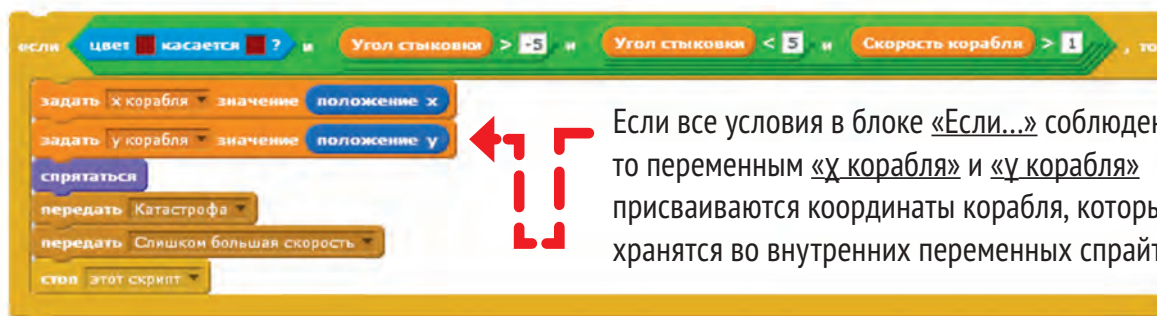
Если нет, то скрипт переходит к следующему блоку «Если...».



АВАРИЯ, ВЗРЫВ, КАТАСТРОФА

ПРОВЕРКА № 2

Следующий блок срабатывает, если корабль соприкоснулся со станцией «красным к красному» — шлюзу, и это произошло под нужным углом, но со слишком большой скоростью. Другими словами, игрок не справился с задачей, и случилась авария.



Если все условия в блоке «Если...» соблюдены, то переменным «х корабля» и «у корабля» присваиваются координаты корабля, которые хранятся во внутренних переменных спрайта.

Корабль исчезает — «прячется». Скрипт передает сообщения «Слишком большая скорость» и «Катастрофа», а затем останавливает свою работу.

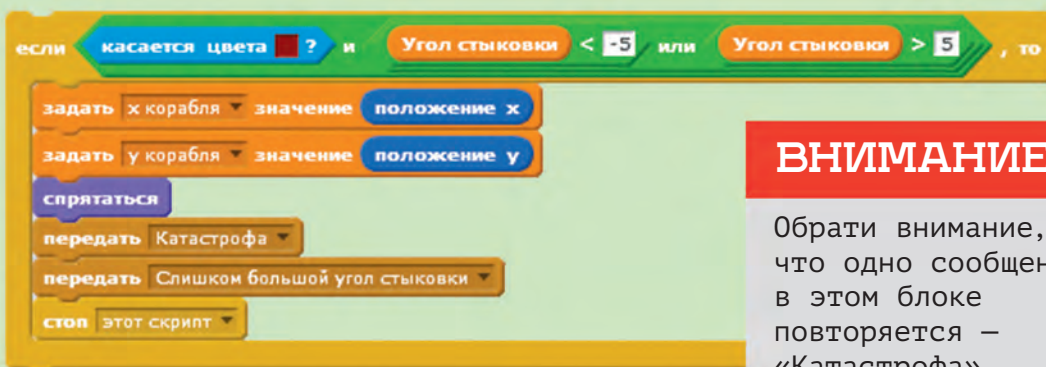
ВНИМАНИЕ



В этом блоке команда «Стоп» действует только для этого скрипта. Если после касания корабля и станции происходят еще какие-то события — например, станция продолжает двигаться или разлетаются осколки корабля, то останавливать другие спрайты не нужно. В этом случае в блоке «стоп» мы используем параметр «этот скрипт». Если другие спрайты должны остановиться, то используем параметр «все».

ПРОВЕРКА № 3

Если корабль и станция соприкоснутся под слишком большим углом, катастрофа тоже неизбежна, поэтому программа должна проверить угол их касания. Эта проверка аналогична предыдущей, только условий здесь меньше, и блок «касается...» выглядит по-другому.



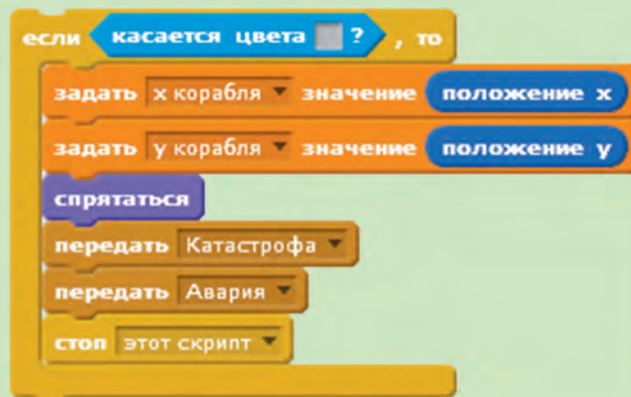
ВНИМАНИЕ

Обрати внимание, что одно сообщение в этом блоке повторяется — «Катастрофа», а другое новое — «Слишком большой угол стыковки».

Мы используем другой сенсор касания. Он проверяет не соприкосновение двух цветов друг с другом, а соприкосновение всего спрайта корабля с красным цветом, то есть со шлюзом станции. Почему так? Потому что если корабль слишком сильно наклонен по отношению к станции, то он сначала коснется ее носом. Если оставить в скрипте сенсор «цвет к цвету», то он работает, только когда корабль уже войдет носом в станцию.

ПРОВЕРКА № 4

Теперь соберем блок, который обрабатывает еще одну аварийную ситуацию: когда корабль касается корпуса станции за пределами красного шлюза.

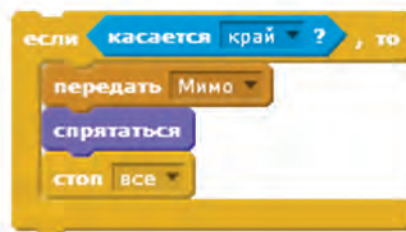


Здесь скрипт корабля проверяет, не коснулся ли корабль серого цвета, то есть основного корпуса станции, за пределами шлюза. При соприкосновении срабатывает этот блок. В нем такие же команды, как и в других блоках проверки касания, только появляется еще одно новое сообщение: «Авария».

ПРОВЕРКА № 5

Остался еще один неудачный исход игры – корабль может промахнуться мимо станции. Эту ситуацию тоже обрабатывает отдельный скрипт.

Как только корабль касается края с любой стороны экрана, скрипт передает сообщение «Мимо», а корабль пропадает. Все скрипты останавливаются. Миссия провалена, игра окончена!



ВНИМАНИЕ



- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Проект Сатурн

v458.0.4



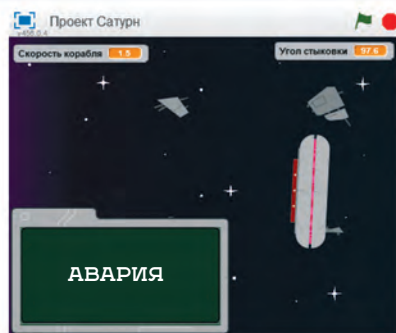
ОБЛОМКИ КОРАБЛЯ РАЗЛЕТАЮТСЯ ПО ВСЕЛЕННОЙ

Столкновение челнока и станции выглядит драматично: корабль разрушается, а его обломки разлетаются в разные стороны.

Корабль при удачной и неудачной стыковке и при выходе за пределы игрового поля отправляет несколько сообщений другим спрайтам.

Их получают одновременно обломки корабля и спрайты сообщений об исходе игры. Все обломки появляются на экране разом, если получено сообщение «Катастрофа». Панели с разными уведомлениями об исходе игры появляются в зависимости от того, какое сообщение получено.

Вот как выглядит обмен сообщениями в игре «Проект „Сатурн“».



Для того чтобы такой сценарий заработал, понадобится создать отдельные спрайты для обломков — да-да, для каждого свой! Загрузим новые спрайты и соберем для них скрипты.



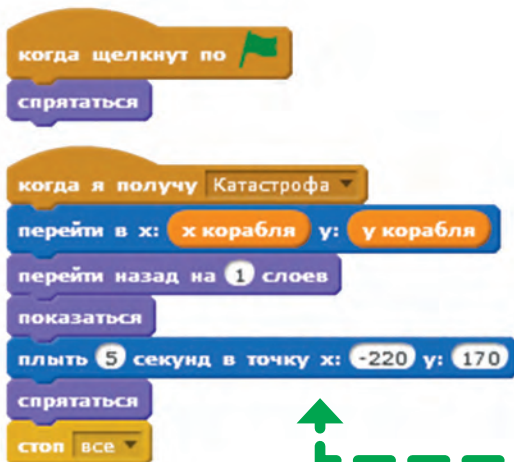
В списке спрайтов под игровым полем появится новый спрайт.

ОБЛОМОК № 1

Нажми на значок папки под игровым полем и загрузи файл «Обломок 1.svg» из папки с ресурсами с сайта scratchbook.piter.com.



Соберем для него скрипт.

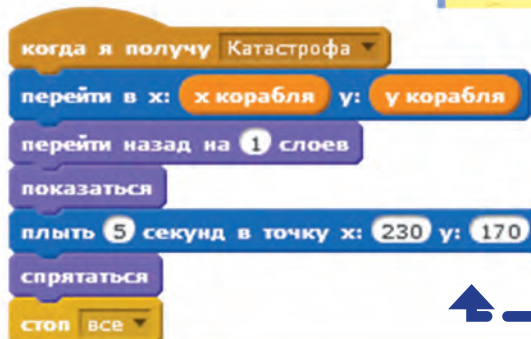
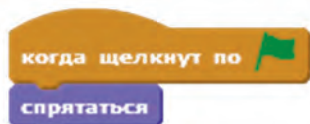


При старте игры обломок скрыт от игрока. Когда корабль, врезавшийся в станцию, рассылает сообщение «Катастрофа», обломок переходит в точку с координатами корабля и показывается на игровом поле.

Затем блок перемещается с видимой глазу скоростью — «плывет» — в точку в левом верхнем углу игрового экрана. Чем больше время перемещения в этом блоке, тем медленнее будет плыть обломок.

Дойдя до угла экрана, обломок пропадает.

ОБЛОМОК № 2



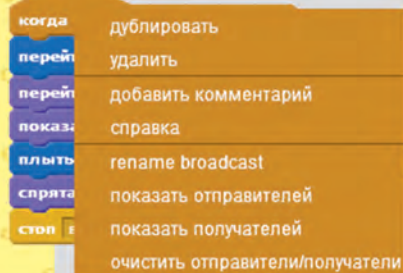
Нажми на значок папки под игровым полем и загрузи файл «Обломок 2.svg» из папки с ресурсами с сайта scratchbook.piter.com. Он появится в списке спрайтов под игровым полем.



Скрипт для него похож на скрипт первого обломка. Но этот будет плыть уже в другом направлении, поэтому нужно задать другие координаты конечной точки. Она находится в правом верхнем углу.



ВНИМАНИЕ



Воспользуйся функцией «Дублировать скрипт». Для этого нажми правой кнопкой мыши на верхний блок скрипта и выбери пункт меню «Дублировать». После этого в списке спрайтов под игровым полем щелкни по спрайту, в скрипты которого надо перенести копию скрипта. Дубликат появится в поле скриптов.

ОБЛОМОК № 3

И снова нажми на значок папки под игровым полем и загрузи файл «Обломок 3.svg» из папки с ресурсами с сайта scratchbook.piter.com. Новый спрайт появится в списке под игровым полем.



```

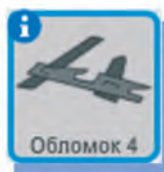
когда щелкнут по [флаг]
  спрятаться

когда я получу Катастрофа
  перейти в x: x корабля y: y корабля
  перейти назад на 1 слоев
  показаться
  плыть 5 секунд в точку x: 230 y: -170
  спрятаться
  стоп все

```

Скрипт этого обломка также ничем не отличается от первых двух, кроме координат конечной точки. Здесь это нижний правый угол игрового поля.

ОБЛОМОК № 4



Это последний! Нажми на значок папки под игровым полем и загрузи файл «Обломок 4.svg» из папки с ресурсами с сайта scratchbook.piter.com. Новый спрайт появится в списке под игровым полем.

Последний обломок будет двигаться в левый нижний угол.

```

когда щелкнут по [флаг]
  спрятаться

когда я получу Катастрофа
  перейти в x: x корабля y: y корабля
  перейти назад на 1 слоев
  показаться
  плыть 5 секунд в точку x: -230 y: -170
  спрятаться
  стоп все

```

ВНИМАНИЕ



Если корабль при столкновении со станцией будет находиться не в центре игрового поля, то при одинаковом времени разлета обломки будут плыть к своей конечной точке с разной скоростью. Поэкспериментируй со временем разлета, например попробуй, сделать его разным для разных обломков.

ВНИМАНИЕ



В скриптах всех обломков последним стоит блок «Стоп все». Именно эти блоки будут завершать игру. Первым команду «Стоп все» даст блок, который первым придет в свою конечную точку. Если для обломков настроено разное время разлета, то первым команду даст обломок, у которого это время меньше всего.

ВНИМАНИЕ



- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



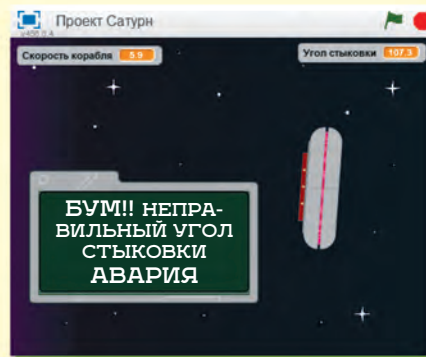
v456.0.4

Проект Сатурн



СООБЩЕНИЯ О РЕЗУЛЬТАТЕ СТЫКОВКИ

После того как корабль успешно пристыковался к станции, прошел мимо или врезался в нее, игрок должен получить сообщение об этом.



```
когда щелкнут по [флаг]
  спрятаться

когда я получу [Стыковка]
  сменить костюм на Сообщение Успешная стыковка
  перейти в верхний слой
  показаться
  играть звук computer beeps1 до конца
  стоп все
```

```
когда я получу [Слишком большая скорость]
  сменить костюм на Сообщение Слишком большая скорость
  перейти в верхний слой
  показаться
  играть звук buzz whirl
```

```
когда я получу [Авария]
  сменить костюм на Сообщение Авария
  перейти в верхний слой
  показаться
  играть звук buzz whirl до конца
```

```
когда я получу [Мимо]
  сменить костюм на Сообщение Стыковка не состоялась
  перейти в верхний слой
  показаться
  играть звук buzz whirl
  ждать 1 секунд
  стоп все
```

```
когда я получу [Слишком большой угол стыковки]
  сменить костюм на Сообщение Неправильный угол стыковки
  перейти в верхний слой
  показаться
  играть звук buzz whirl
```

Для этого мы создадим один спрайт с несколькими костюмами и скриптами. Создадим новый спрайт для сообщений и соберем для них скрипты.



Нажми на значок папки под игровым полем и загрузи файлы картинок с сообщениями о стыковке из папки с ресурсами с сайта scratchbook.piter.com.

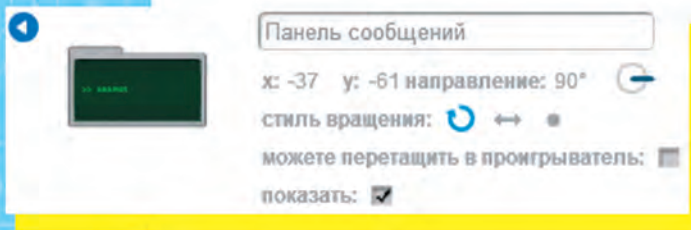
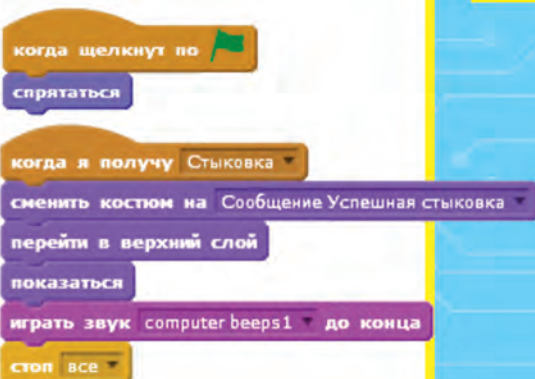
Новый объект:



СООБЩЕНИЕ № 1 УСПЕШНАЯ СТЫКОВКА

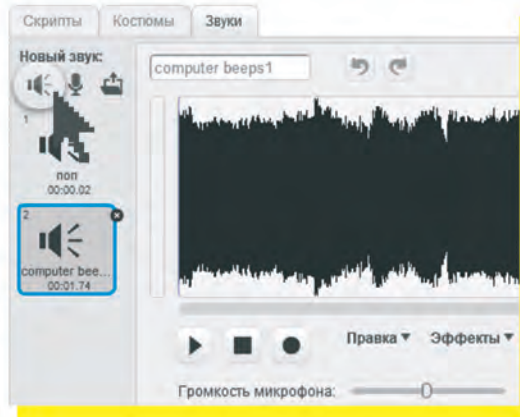
Начнем с самого радостного финала. Для начала загрузи файл «Сообщение Успешная стыковка.svg». В списке под игровым полем появится новый спрайт. Назовем его «Панель сообщений».

Скрипт сообщения о стыковке состоит из блоков со вкладок «События», «Управление», «Внешность» и «Звук».



При старте игры сообщение будет скрыто. Но если игрок справится с задачей и скрипт получит сообщение о стыковке, уведомление появится на переднем плане и раздастся звук победы. Действие всех скриптов будет остановлено.

Хотя на самом деле в космосе звуки не распространяются, без них играть было бы немного скучно. Поэтому добавим в игру спецэффектов и начнем со звука стыковки. На вкладке «Звук» нажми значок мегафона.



Загрузи файл
«Computer beeps1»
из библиотеки
звуков Скретч.

Библиотека звуков

Категория

Все

Животное

Эффекты

Электроника

Человек

Инструменты

Музыкальная петля

Примечания к музыке

Перкуссия

Звуки



bell toll



bird



birthday bells



birthday



boing



chord



clap beatbox



clapping



computer beeps1



computer beeps2



cymbal echo



cymbal



dance around

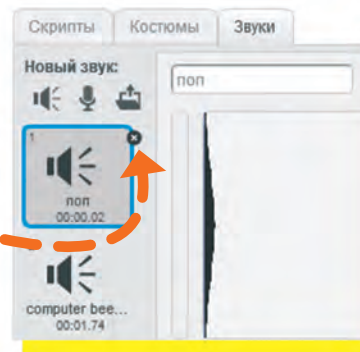


dance celebrate



dance chill out

Ненужный звук можно удалить.
Щелкни по нему и нажми
на крестик.



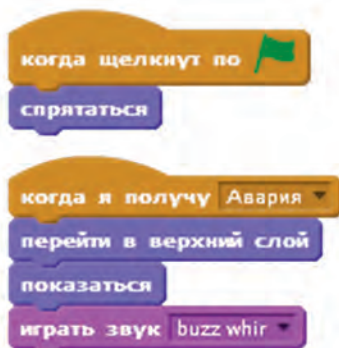
СООБЩЕНИЕ № 2

КОРАБЛЬ ВРЕЗАЛСЯ В КОРПУС СТАНЦИИ. АВАРИЯ



Перейди на вкладку «Костюмы» и загрузи с сайта файл «Сообщение Авария.svg». В списке костюмов появится новый костюм.

Скрипт у этого костюма еще короче, чем у сообщения о стыковке.



ВНИМАНИЕ



Команду «Стоп» всем спрайтам в случае аварии будут рассылать обломки.

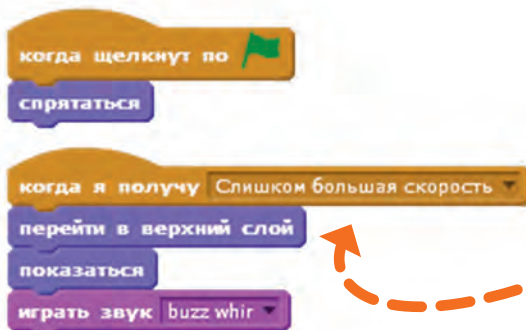
Столкновение корабля со станцией тоже будет сопровождаться специальным звуком. Загрузи файл «Buzz whirl» из библиотеки звуков Скретч.

СООБЩЕНИЕ № 3

СТЫКОВКА СО СЛИШКОМ

БОЛЬШОЙ СКОРОСТЬЮ. АВАРИЯ

На вкладке «Костюмы» загрузи файл «Сообщение Слишком большая скорость.svg».



Библиотека звуков

Категория

Все
Животное
Эффекты
Электроника
Человек
Инструменты
Музыкальная петля
Примечания к музыке
Перкуссия
Звуки

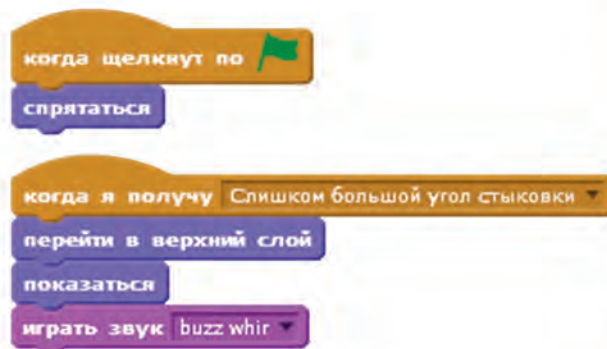


Скрипт для него похож на предыдущий, только он получает другое сообщение от корабля.

СООБЩЕНИЕ № 4

СТЫКОВКА ПОД СЛИШКОМ БОЛЬШИМ УГЛОМ. АВАРИЯ

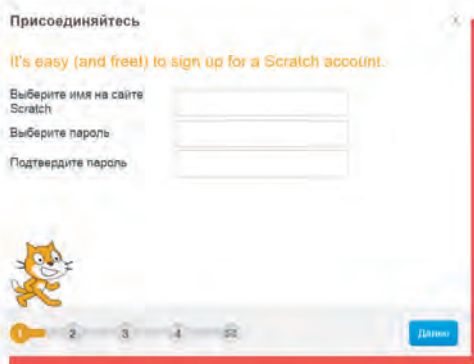
На вкладке «Костюмы» загрузи файл «Сообщение Неправильный угол стыковки.svg». Скрипт этого костюма тоже должен получить свое собственное сообщение. —————>



СООБЩЕНИЕ № 5

СТЫКОВКА НЕ СОСТОЯЛАСЬ

На вкладке «Костюмы» загрузи файл «Сообщение Стыковка не состоялась.svg». Скрипт этого костюма получает сообщение, когда корабль касается края экрана и когда станция уходит за верхний край экрана. В отличие от скриптов столкновения со станцией, он прекращает действие остальных спрайтов.



ВНИМАНИЕ

- * Сохрани программу через меню «Файл» — «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.

Проект Сатурн

**УРА! ИГРА ГОТОВА! МОЖЕШЬ СМЕЛО ОТПРАВЛЯТЬСЯ
В КОСМИЧЕСКОЕ ПУТЕШЕСТВИЕ!**



ЗАТЕРЯННЫЙ ГОРОД

ПОИСК СОКРОВИЩ В ЗАГАДОЧНОМ ГОРОДЕ В ДЖУНГЛЯХ

Знаменитый ученый-археолог обнаруживает в джунглях затерянный город. Изучая развалины, он попадает в запутанный лабиринт с ядовитыми змеями и пауками, ловушками и препятствиями. Сможешь ли ты вывести героя из лабиринта?

Игра «Затерянный город»

Сложность: * * *



Время сборки: 1,5–2 часа

Видеоролик игры можно найти по адресу: scratchbook.piter.com

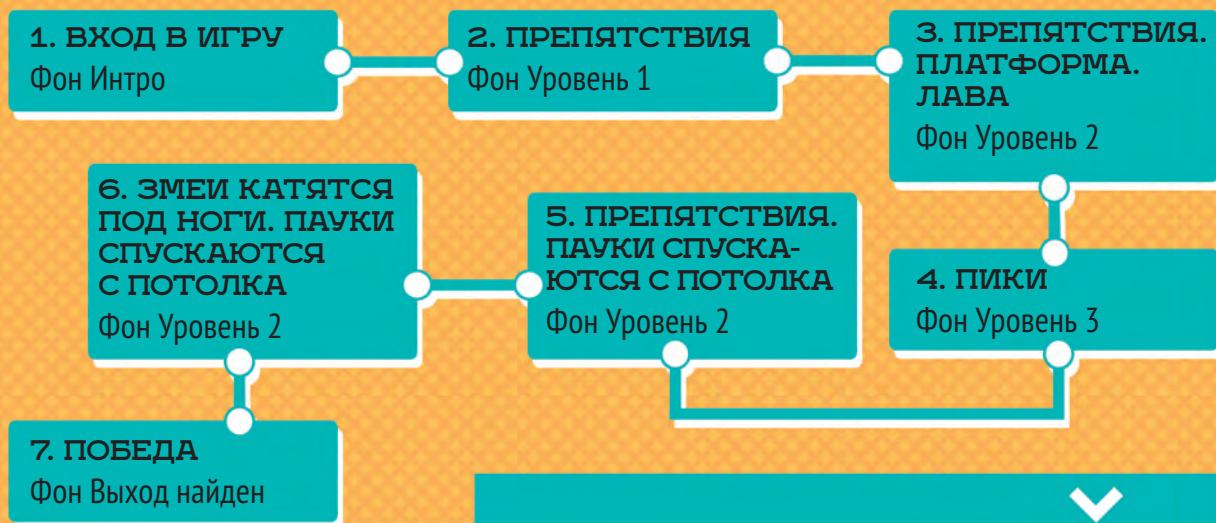
Описание

Игра-платформер с несколькими уровнями с повышающейся сложностью. Игрок может прыгать, взбираться на небольшую высоту по отвесным стенам. Для перехода на следующий уровень он должен коснуться серого камня. У героя пять жизней. Жизнь теряется, когда персонаж касается лавы, пик, ядовитых змей или пауков. При потере жизни игрок возвращается в начало уровня, при потере всех пяти жизней игра заканчивается. Игроку понадобятся внимание и быстрая реакция, чтобы пройти все уровни!

Правила игры

- * Цель игрока — успешно пройти по лабиринтам.
- * Играет один игрок.
- * Игрок управляет героем при помощи клавиш со стрелками. Клавиши ВПРАВО и ВЛЕВО — поворот и движение героя, ВВЕРХ — прыжок на 8 пунктов вверх и замедление.
- * Перемещение на следующий уровень происходит, если коснуться серого камня на правом краю экрана.
- * Для того чтобы выйти из лабиринта, надо пройти пять уровней. У героя есть три жизни.
- * «Смерть» героя происходит, когда он касается зеленой лавы, пауков, пик или змей. Каждый раз после этого герой возвращается к началу уровня.
- * У игры два исхода: **КОНЕЦ ИГРЫ**, если персонаж потратил три жизни; **ПОБЕДА** — выход из лабиринта при прохождении последнего уровня.

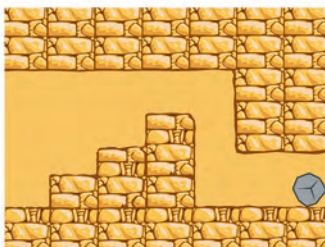
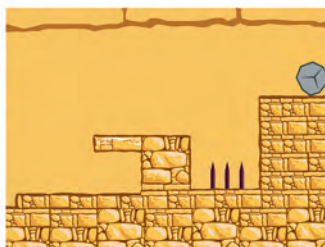
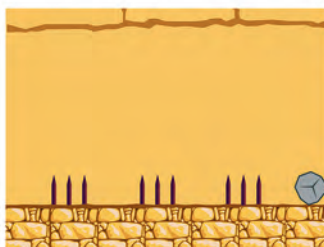
СПИСОК УРОВНЕЙ



Скачай нужные для игры ресурсы по адресу:
scratchbook.piter.com

ВОТ КАКИЕ СПРАЙТЫ ТЕБЕ ПОНАДОБЯТСЯ ДЛЯ ЭТОЙ ИГРЫ

ФОНЫ





Ученый



Кнопка
входа в игру



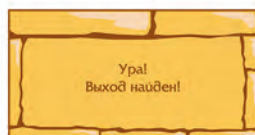
Надгробный
камень



Клубок
змей

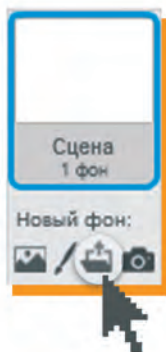


Паук



Сообщения о конце игры

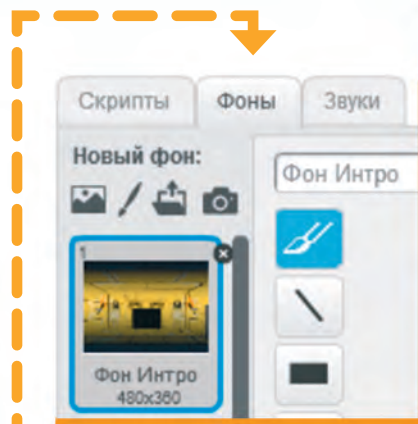
ЗАГРУЖАЕМ РЕСУРСЫ



Для того чтобы загрузить в игру фоны, нажми на значок папки под сценой слева.

Загрузи файлы из папки ресурсов с сайта scratchbook.piter.com, семь фонов:

- * Фон Интро
- * Фон Уровень 1
- * Фон Уровень 2
- * Фон Уровень 3
- * Фон Уровень 4
- * Фон Уровень 5
- * Фон Выход найден



В поле справа на вкладке «Фоны» появятся новые картинки.



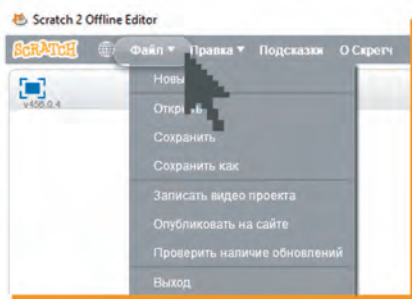
ВНИМАНИЕ



Важно! Фоны должны идти сверху вниз именно в таком порядке – от «Интро» до «Выход найден».

Сохраним программу, чтобы не потерять сделанную работу.

Нажми на пункт верхнего меню «Файл».



ВНИМАНИЕ



Фоны появляются в списке в том порядке, в каком их загружали. Если нужно поменять порядок, перетаскивай их мышкой в нужное место.

Выбери пункт «Сохранить как» и введи имя проекта – например «Затерянный город».

Оно появится на плашке над игровым полем.



ПОДКЛЮЧАЕМ ФОНОВЫЙ ЗВУК

В первую очередь соберем фоновые скрипты, которые будут работать на протяжении всей игры.

когда щелкнут по [флаг]
 сменить фон на Фон Интро
 играть звук Интро до конца

когда я получу Следующий уровень
 остановить все звуки
 всегда
 играть звук Фоновый звук уровня до конца

когда я получу Победу
 сменить фон на Фон Выход найден
 стоп другие скрипты сцены

когда я получу Поражение
 стоп другие скрипты сцены

когда фон меняется на Фон Уровень 1
 задать Уровень значение 1

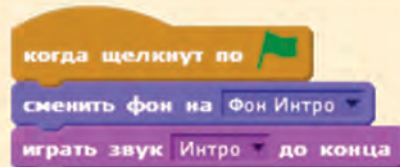
когда фон меняется на Фон Уровень 2
 задать Уровень значение 2

когда фон меняется на Фон Уровень 3
 задать Уровень значение 3

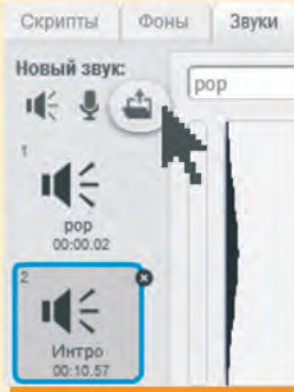
когда фон меняется на Фон Уровень 4
 задать Уровень значение 4

когда фон меняется на Фон Уровень 5
 задать Уровень значение 5

Сначала создадим скрипты, в которых запускаются два фоновых звука для игры «Затерянный город». Один будет звучать, только когда показывается «Фон Интро» – на этом фоне игрок входит в игру. Это «мистический», тревожный звук – ведь наш герой попадает в подземные лабиринты затерянного города и еще не знает, с чем он столкнется.



На вкладке «Звуки» загрузи файл «Интро.wav» из папки ресурсов с сайта scratchbook.piter.com.

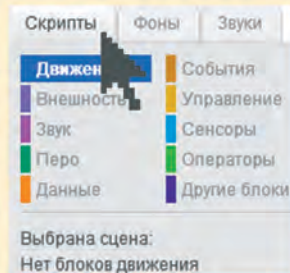


Новый звук:

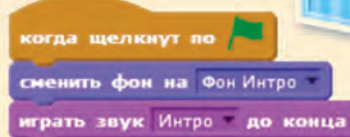


Ненужный звук можно удалить, нажав на крестик.

Для сборки первых скриптов нам понадобятся блоки «События», «Управление», «Внешность» и «Звук» с вкладки «Скрипты».

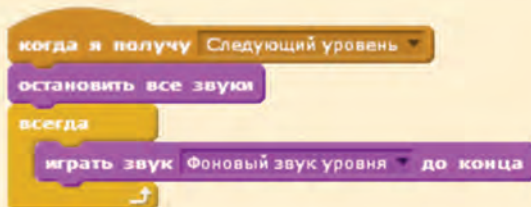


Перетащи с них нужные блоки, поставь их по порядку и выбери в окошке параметров команды «Играть» название звукового файла «Интро.wav».



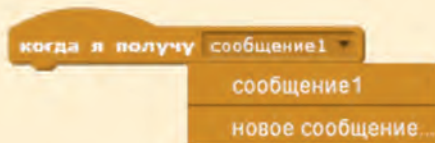
Список фонов появится в окошке параметров команды «Сменить фон». Он открывается при нажатии на черную стрелку.

Второй скрипт будет включать фоновый звук при переходе на каждый из игровых уровней. Это традиционная для аркад и платформеров энергичная мелодия.

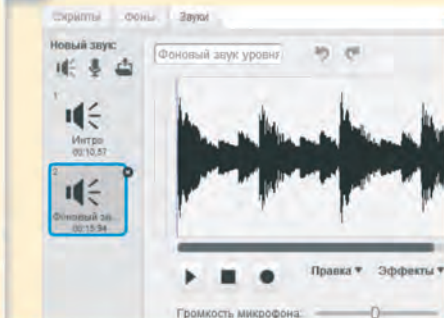
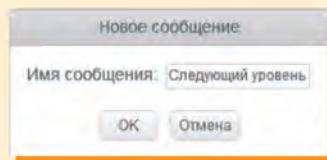


Загрузи файл «Фоновый звук уровня.wav» из папки ресурсов с сайта scratchbook.piter.com.

Для того чтобы передать сообщение «Следующий уровень», его надо создать. Перетащи блок «Когда я получу сообщения» со вкладки «События» в поле скриптов и щелкни по стрелке в окошке параметров.



Откроется контекстное меню. Выбери в нем пункт «Новое сообщение». Введи любое имя сообщения и нажми ОК. Удобно называть сообщения так, чтобы было понятно, о чем они сообщают спрайтам.



ВНИМАНИЕ



- * Сохрани программу через меню «Файл» — «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



**ЗАРАБОТАЛО? УРА!
ПОЗДРАВЛЯЕМ С ПЕРВЫМ
ЗАПУСКОМ ИГРЫ!**

ДОБАВЛЯЕМ КНОПКУ СТАРТА ИГРЫ

В эту игру мы добавим кое-что, чего не делали раньше, — экран заставки. Запустив программу, игрок услышит мистическую музыку, которая поможет ему проникнуться атмосферой приключения, и увидит специальный фон,

кнопку старта Play и правила игры. Для того чтобы добавить кнопку старта, создадим новый спрайт. Из папки ресурсов с сайта scratchbook.piter.com загрузи файл «Кнопка старта.svg».

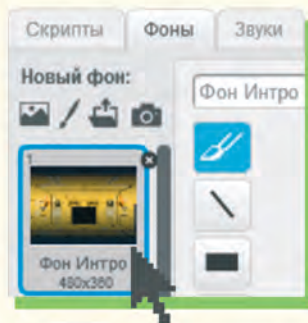
Он появится в окне спрайтов под игровым экраном:



А сама кнопка появится на игровом экране. Перейди на «Фон Интро» в списке фонов.

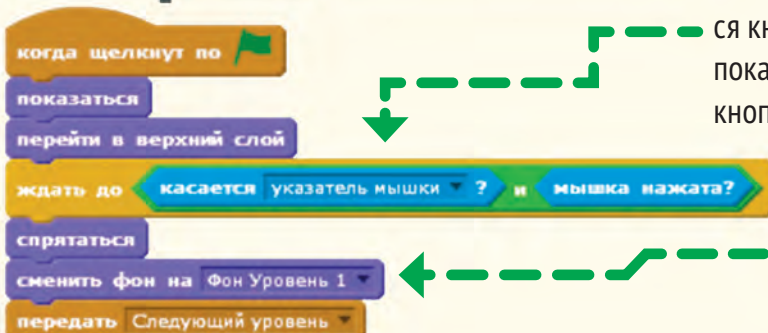


Мышкой перетащи кнопку старта на черный прямоугольник посреди «Фона Интро».



Соберем скрипт для спрайта кнопки старта. Перейди на вкладку «Скрипты». Нам понадобятся блоки из меню «События», «Внешность», «Управление», «Сенсоры» и «Операторы».

Если нажать на зеленый флажок и запустить игру, появляется первый фон — «Фон Интро» (об этом заботится скрипт фона). На нем показывается кнопка старта Play. Программа ждет, пока пользователь не коснется спрайта кнопки курсором и не нажмет на него.



После этого кнопка старта пропадает, а фон меняется на картинку первого игрового уровня.

Затем скрипт фона получает сообщение «Следующий уровень» и включается звук игрового уровня.



ВНИМАНИЕ



- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Затерянный город

v456.0.4



ДОБАВЛЯЕМ УЧЕНОГО И НАСТРАИВАЕМ ЕГО ДВИЖЕНИЕ



Теперь пора добавить нашего главного героя – ученого. А затем мы расставим на его пути те опасные препятствия, с которыми будет сталкиваться игрок.

Создай новый спрайт. Нажми на значок папки под игровым полем и загрузи файл «Ученый.svg» из папки с ресурсами с сайта scratchbook.piter.com.

Новый объект:



В списке спрайтов появится фигурка ученого.



Соберем скрипты для нашего героя. Он должен ходить по игровому полю и прыгать, чтобы преодолевать препятствия и уворачиваться от опасностей. Чтобы он освоил все эти нехитрые навыки, нам понадобятся четыре скрипта и четыре пользовательских блока.



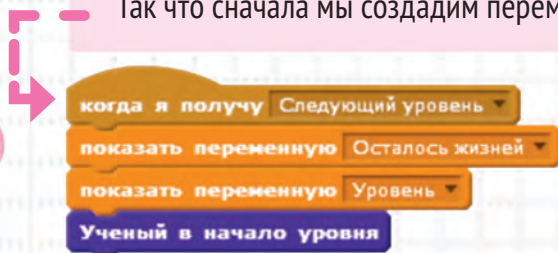
ВНИМАНИЕ



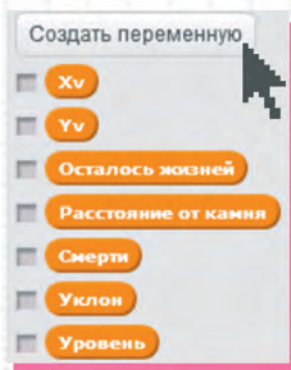
Пользовательский блок – это команда, которую пользователь Скретча может собрать из других команд с собственными параметрами. В пользовательский блок удобно собирать цепочки команд, которые используются в программе несколько раз. Так скрипты получаются короче, а если цепочку нужно изменить, то это придется сделать только один раз.

СКРИПТ № 1. ЗАДАЕМ ЗНАЧЕНИЯ ПЕРЕМЕННЫХ

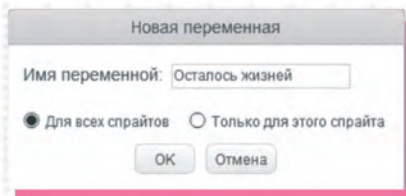
Первый скрипт ученого задает значения переменных, которые используются в игре. Так что сначала мы создадим переменные для этого скрипта и для всей игры.



Для этого надо открыть меню «Данные» и нажать кнопку «Создать переменную».



Появится окно создания новой переменной.



При запуске игры будут заданы значения переменных «Уровень», «Смерти» и «Осталось жизней». Показывать игроку мы будем только переменные «Осталось жизней»

и «Уровень». Но их надо скрыть во время демонстрации заставки, то есть до входа в игру.

«Xv» — скорость изменения положения игрока по оси X.

«Yv» — скорость изменения положения игрока по оси Y.

«Уровень» — номер уровня, на котором находится игрок.

«Смерти» — количество прикосновений к источнику опасности. После каждого касания игрок будет возвращаться в начало уровня, после пяти касаний игра закончится.

Значение переменной «Осталось жизней» всегда зависит от значения переменной «Смерти».

«Расстояние от камня» — расстояние по Y от последней точки, где герой коснулся камня. Эта переменная понадобится, чтобы ограничивать прыжок героя при нескольких нажатиях стрелки «вверх».

«Уклон» — расстояние по Y в ситуации, когда герой уперся в стену и пытается взобраться на нее.

На стену можно поползти, если она высотой не больше 8 пикселей.

ВНИМАНИЕ

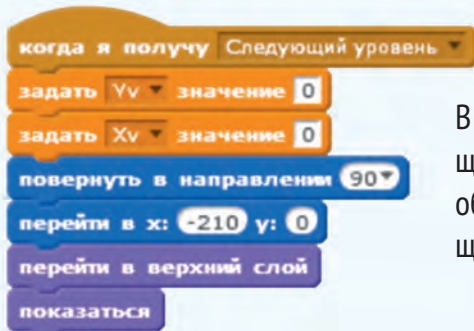


Переменные «Yv», «Xv», «Расстояние от камня» и «Уклон» будут заданы в других скриптах спрайта.

СКРИПТ № 2. ЗАДАЕМ ПРАВИЛА ДВИЖЕНИЯ

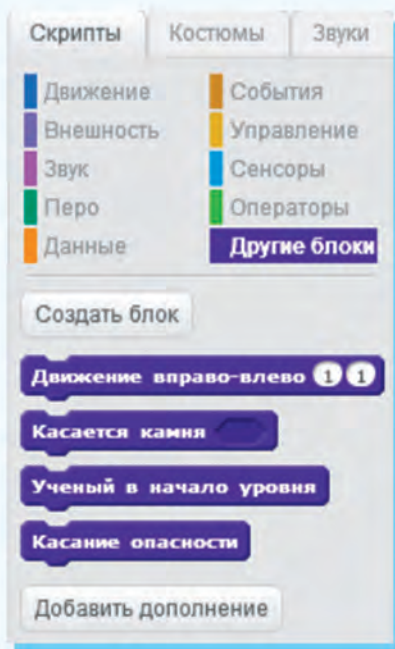
Основа нашей игры, как и любого платформера, – взаимодействие героя с препятствиями. От них он отталкивается при ходьбе и прыжке. Скрипт, обрабатывающий движение героя, будет запускаться при переходе на каждый игровой уровень.

При запуске игрового уровня ученый должен переместиться в его начало. Скрипт для этого выглядит так:



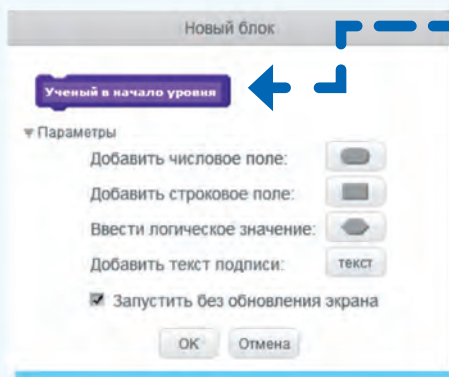
В начало уровня герой игры будет перемещаться каждый раз, когда коснется опасного объекта, поэтому выведем команды перемещения в пользовательский блок.

БЛОК № 1. УЧЕНЫЙ В НАЧАЛО УРОВНЯ

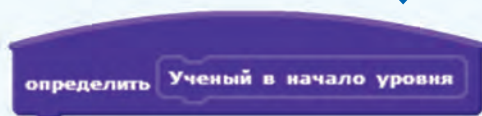


Чтобы создать пользовательский блок, нужно перейти в меню «Другие блоки» и нажать на кнопку «Создать блок».

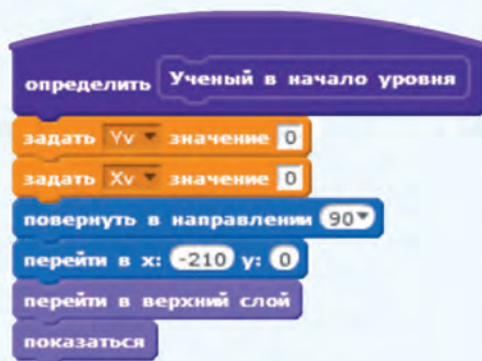
Откроется окно создания нового блока. Введи его имя в сиреновом поле, затем нажми на стрелку рядом с полем «Параметры» и поставь галочку в поле «Запустить без обновления экрана». Нажми ОК.



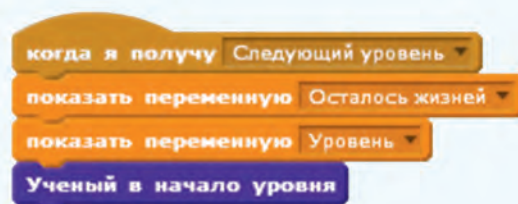
На поле скриптов появится шапка нового блока, а в списке пользовательских блоков – одноименная команда.



Перетащи к шапке блока команды перемещения героя в начало уровня.



Перетащи команду «Ученый в начало уровня» из меню «Другие блоки» в скрипт «Когда я получу „Следующий уровень“».



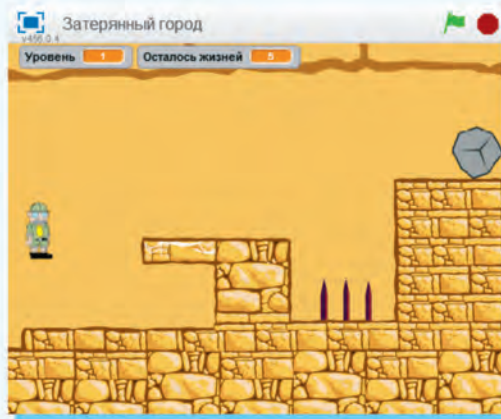
ВНИМАНИЕ



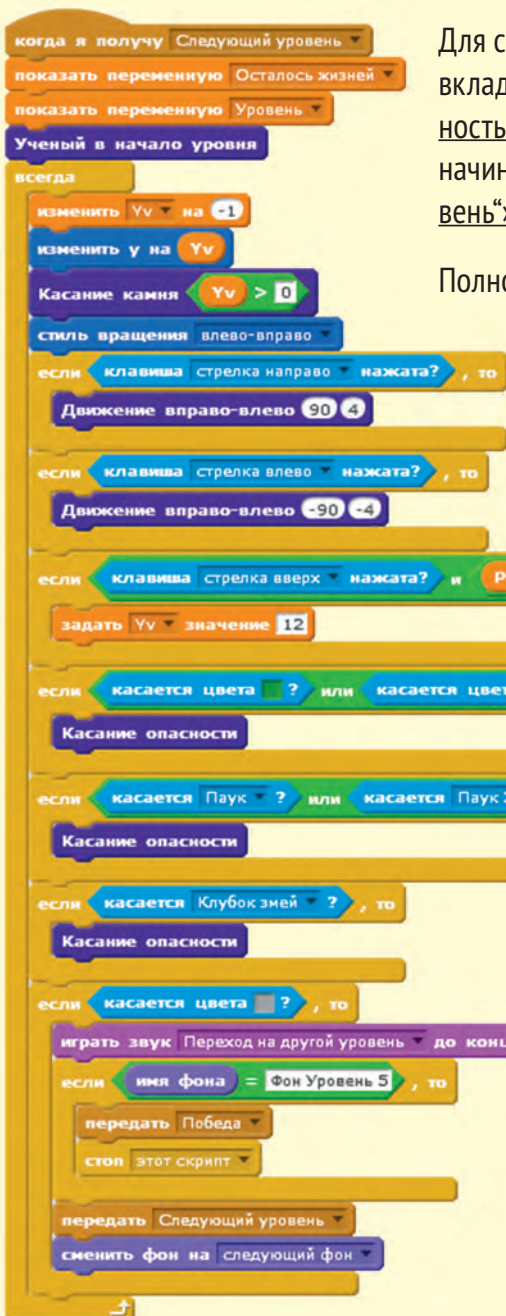
- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Что получилось? При запуске игрового уровня герой появляется на экране и зависает на одном месте. Больше пока ничего не происходит.

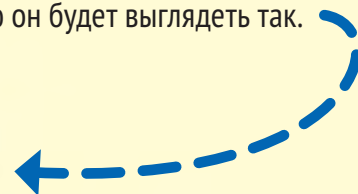


УЧИМ ГЕРОЯ ХОДИТЬ И ПРЫГАТЬ



Для сборки нового скрипта ученого нам понадобятся вкладки «События», «Управление», «Данные», «Внешность», «Движение», «Операторы». К скрипту, который начинается с блока «Когда я получу «Следующий уровень»», мы будем добавлять другие команды.

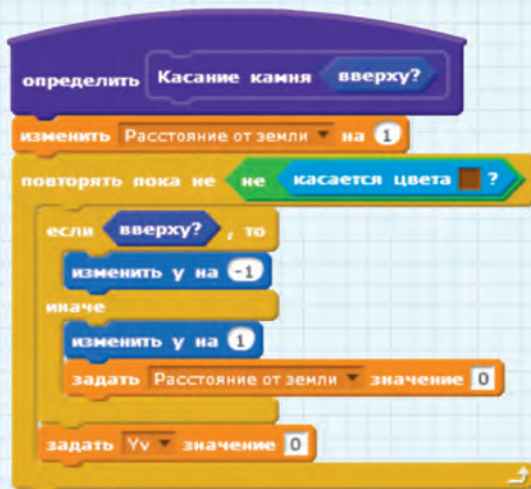
Полностью он будет выглядеть так.



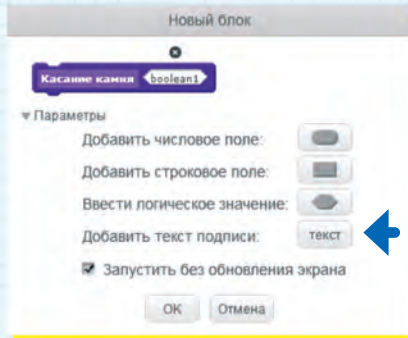
Герой должен уметь перемещаться вправо и влево, касаться каменного пола или потолка и понимать, что перед ним стена.

БЛОК № 2. КАСАНИЕ КАМНЯ

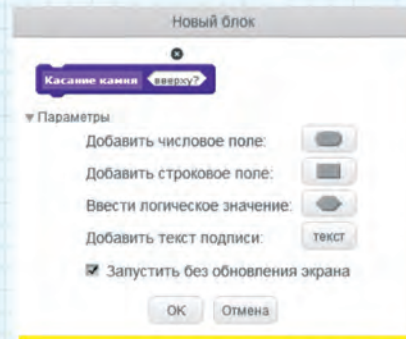
Сначала соберем блок «Касание камня».



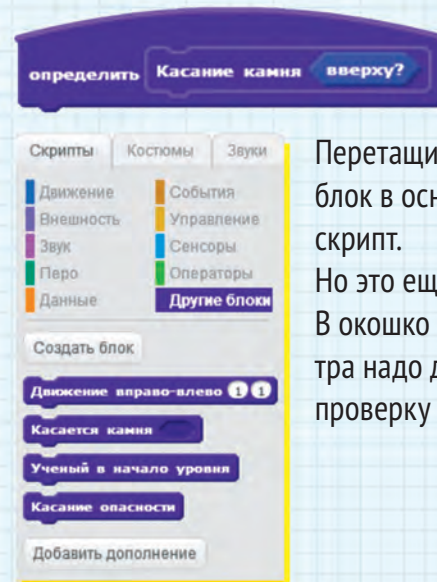
В меню «Другие блоки» перейди на вкладку «Скрипты» и нажми кнопку «Создать блок». Введи название «Касание камня» и разверни строку параметров.



Щелкни один раз по окошку поля «Ввести логическое значение» — при наведении курсора оно будет подсвечиваться синим цветом. Поле переместится в блок наверху. Дважды щелкни в поле «Boolean1» и введи в нем «вверх?». Нажми ОК.



В поле скриптов появится шапка блока «Касание камня», а в списке блоков в меню «Другие блоки» — одноименный блок.



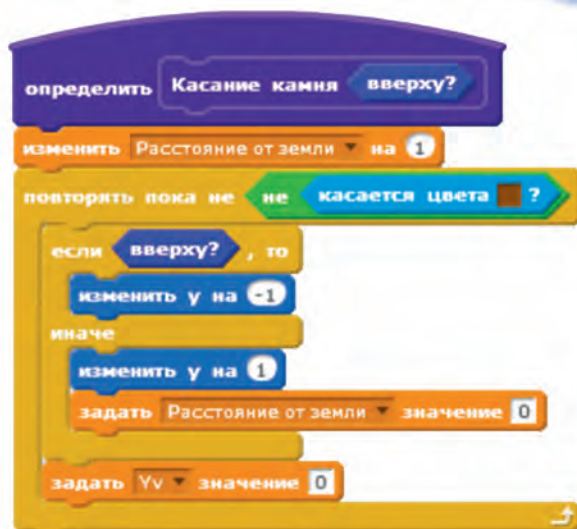
Перетащи этот блок в основной скрипт. Но это еще не все. В окошко параметра надо добавить проверку условия.

При исполнении этой команды основной скрипт передает в блок «Касание камня» параметр — условие «Переменная Yv больше 0».



ЧТО ПРОИСХОДИТ В БЛОКЕ «КАСАНИЕ КАМНЯ»?

При каждом исполнении команд в блоке «Всегда» спрайт ученого перемещается вниз со скоростью 1 пиксель за шаг, пока не коснется камня.



Затем блок «Касание камня» проверяет, коснулся ли спрайт камня. Если спрайт коснулся камня (коричневого цвета), то значение y изменяется на 1.

Получается, что спрайт оттолкнулся от камня и завис на высоте 1 пиксель выше пикселей коричневого цвета. Так происходит бесчисленное количество раз — при каждом прохождении цикла «Всегда». Но глазу это движение не видно.

Если спрайт касается коричневого цвета, когда движется вверх (переменная $Yv > 0$), то он отталкивается от потолка и движется вниз.

Каждый раз, когда положение спрайта изменяется на Yv , переменная «Расстояние от земли» изменяется на 1. Как только спрайт касается земли, она обнуляется. Эта переменная понадобится нам для того, чтобы ограничить высоту прыжков.

Для того чтобы ввести параметр «вверху?» в блок «Если, то», перетащи его мышкой из шапки блока «Касание земли».

Для того чтобы указать цвет, на касание которого должен реагировать сенсор, щелкни мышкой по квадрату в блоке «Касается цвета». Курсор поменяется на изображение ладони. Теперь один раз щелкни им по нужному оттенку на игровом экране – это коричневый цвет на границе камня и фона.



БЛОК № 3. ДВИЖЕНИЕ ВПРАВО-ВЛЕВО

Теперь ученый умеет твердо стоять на камне – неплохо для начала, но явно недостаточно для хорошего приключения. Сейчас мы научим нашего героя двигаться. Нам понадобится создать пользовательский блок для движения вправо и влево.

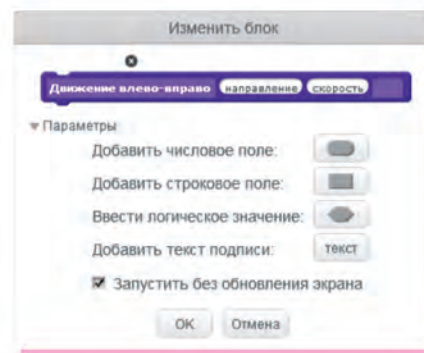
В меню «Другие блоки» на вкладке «Скрипты» нажми кнопку «Создать блок». Введи название «Движение вправо-влево» и разверни строку параметров. Щелкни по числовому полю, и оно появится в сиреновом поле. Потом еще раз – нам их нужно два.

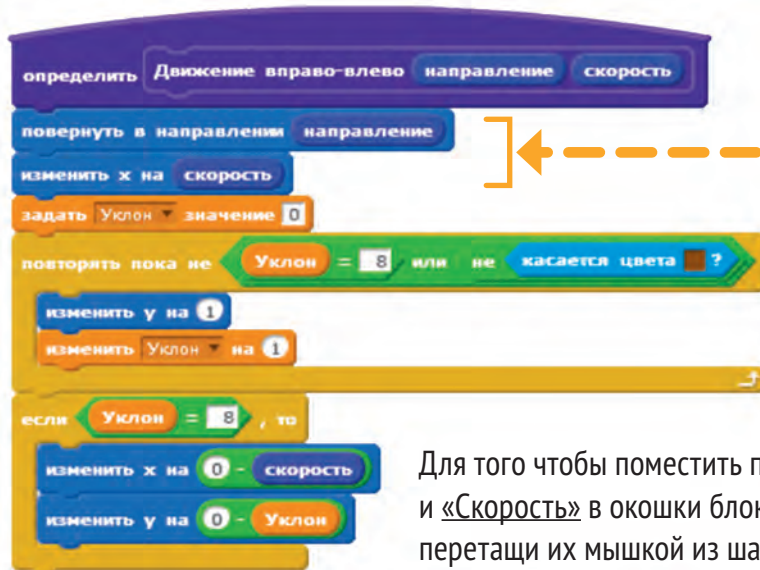
Движение вправо-влево number1 number2

На поле скриптов появится шапка блока.

определить Движение вправо-влево направление скорость

Дважды щелкни по каждому числовому полю. В первом окошке введи «Направление», во втором «Скорость». Поставь галочку в поле «Запустить без обновления экрана» и нажми кнопку ОК.





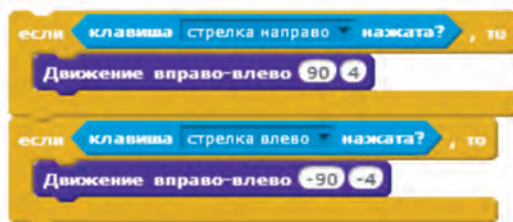
Соберем блок полностью.

Для того чтобы поместить параметры «Направление» и «Скорость» в окошки блоков «Повернуть» и «Изменить», перетащи их мышкой из шапки блока.

При нажатии стрелки ВЛЕВО или ВПРАВО этот блок будет работать с разными параметрами.

1. При нажатии клавиши со стрелкой вправо герой поворачивается в правую сторону (в положение 90 градусов) и перемещается на 4 пикселя вправо.
2. При нажатии клавиши со стрелкой влево герой поворачивается в левую сторону (в положение -90 градусов) и перемещается на 4 пикселя влево.

Вот так эти блоки должны выглядеть в основном скрипте. Перетащи команду «Движение вправо-влево» из меню «Другие блоки» в основной скрипт и введи в окошки параметров нужные значения.



При нажатии клавиш со стрелками блок «Движение вправо-влево» получает от основного скрипта параметры «Направление» -90 или 90, и «Скорость» – изменение x . Герой поворачивается в заданном направлении и с заданной скоростью.

Переменная «Уклон» нужна для того, чтобы обработать ситуацию, когда спрайт касается наклонной или вертикальной стены. Представим, что вертикальная стена — это горка, и спрайт отталкивается от нее так же, как от камня под ногами.



При каждом шаге спрайт поднимается на 1 пиксель. При этом он уже не касается камня внизу, и скрипт проверяет, есть ли другие касания. Если да, то спрайт поднимается еще на пиксель. Если после этого он уже не касается камня, то может переместиться на шаг вправо или влево. Если касается, то поднимается еще на пиксель — и так до 8 раз. Если после подъема на 8 пикселей он все еще касается коричневого цвета, то возвращается к исходному положению — взобраться на стену не удалось.

ВНИМАНИЕ



- * Сохрани программу через меню «Файл» — «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.

ПРЫЖОК, ЕЩЕ ПРЫЖОК



Затерянный город



Чтобы герой мог преодолевать препятствия, его надо научить прыгать. За это отвечает короткий блок в основном спрайте.



Прыжок будет получаться только при условии, что герой находится на высоте не более 4 пикселей от земли — это помешает игроку «летать» в воздухе и сделает игру более сложной. Высота прыжка не будет превышать 12 пикселей.

ВНИМАНИЕ



Поэкспериментируй с разными сочетаниями высоты прыжка и предельного расстояния от земли. Что меняется?

Попробуй поменять также высоту стены, на которую может вскарабкаться игрок – максимальную величину переменной «Уклон». Какие изменения делают игру проще, а какие сложнее?

С какими вариантами значений играть интереснее всего?

ВНИМАНИЕ



* Сохрани программу через меню «Файл» – «Сохранить».

* Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Затерянный город

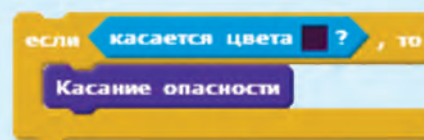
v455.0.4



ПЕРВЫЕ ОПАСНОСТИ

Даже очень умелому игроку может не хватить ловкости, чтобы обойти опасности, которыми наполнен затерянный город. Что будет происходить, когда герой касается препятствия?

На первом уровне задача игрока – не упасть на пики, торчащие в провале.



Для того чтобы выбрать цвет, при касании которого будет исполняться блок «Касание опасности», щелкни один раз по окошку параметров в блоке «Касается цвета?», а потом один раз по пику на игровом фоне.



В окошке появится нужный цвет.

БЛОК № 4. КАСАНИЕ ОПАСНОСТИ

Соберем пользовательский блок

«Касание опасности».

В меню «Другие блоки» нажми «Создать блок» и назови его «Касание опасности».

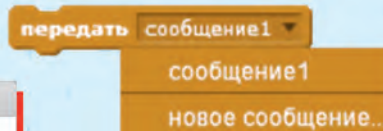
У этого блока не будет собственных параметров – он просто собирает цепочку команд, чтобы было удобнее обрабатывать столкновения игрока с источниками опасности.



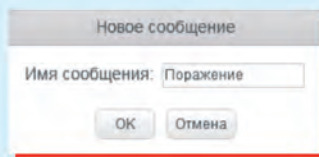
Если герой коснется пикселей цвета пики, будет исполнен блок «Касание опасности».

Ученый переместится в начало уровня, прозвучит резкий звук «screech». Переменная «Смерти» увеличится на 1, а переменная «Осталось жизней», наоборот, уменьшится. Если она станет равна 0, то блок передаст сообщение «Поражение».

Чтобы создать новое сообщение, перетащи из меню «События» блок «Передать сообщение», нажми на стрелку и выбери пункт меню «Новое сообщение».

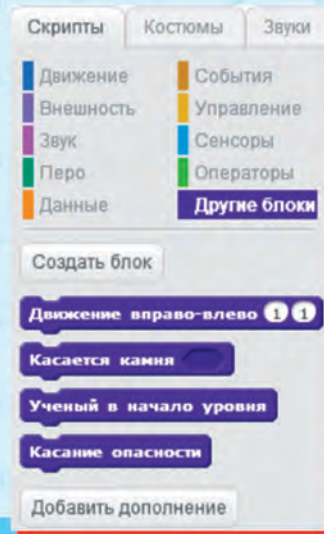
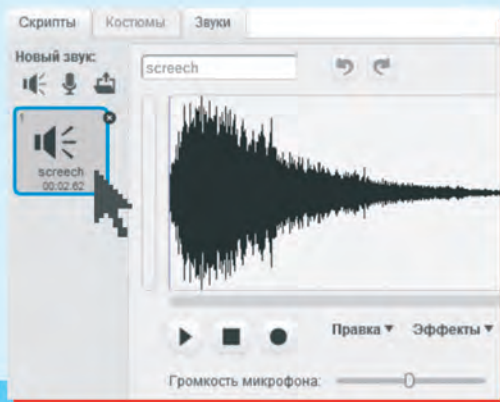


Назови сообщение нужным именем.

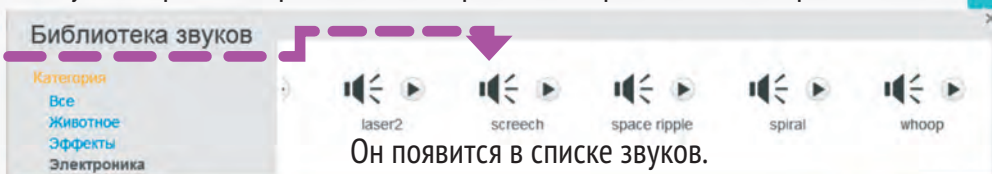


Перетащи команду «Ученый в начало уровня» из меню «Другие блоки».

Чтобы подключить звук «screech», перейди на вкладку «Звуки» и щелкни по значку мегафона.



Откроется библиотека звуков Скретча. Перейди в категорию «Электроника» и выбери из нее нужный звук.



Он появится в списке звуков.

ВНИМАНИЕ



- * Сохрани программу через меню «Файл» — «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.

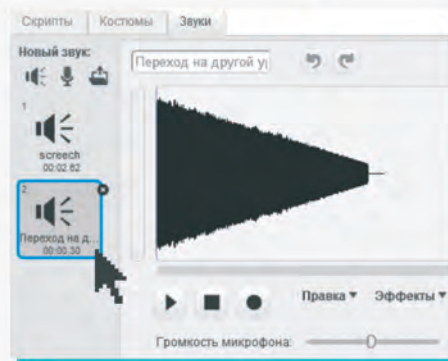
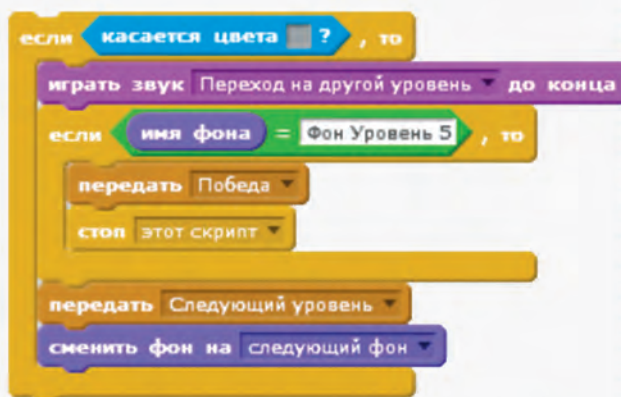
ПЕРЕХОД НА СЛЕДУЮЩИЙ УРОВЕНЬ



Игрок переходит на следующий уровень, как только добирается до серого камня у правого края экрана. За это в основном скрипте отвечает последний блок «Если».

Этот блок будет обрабатывать ситуацию, когда спрайт ученого касается серых пикселей — цвета камня. Если это событие произойдет, то программа проиграет короткий звук перехода на новый уровень. Скрипт передаст сообщение «Следующий уровень», его получит скрипт фона, и заново раздастся фоновый звук уровня. Фоновая картинка уровня при этом поменяется на следующую в списке. Если игрок коснулся серого камня на последнем игровом уровне, это означает, что он дошел до конца игры! Победа!

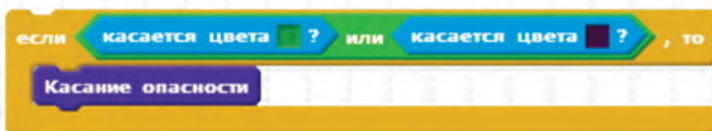
На вкладке «Звуки» добавь звук «Переход на другой уровень» из папки ресурсов с сайта scratchbook.piter.com.



НОВЫЕ ОПАСНОСТИ

На втором игровом фоне героя ждут новые опасности — ямы с кипящей лавой.

Добавим в блок обработки касания опасности новое условие — зеленый цвет лавы. Теперь блок выглядит так:



Каждое условие можно вынести в отдельный блок «Если», чтобы было удобнее.

А можно собирать их все в один блок при помощи условного оператора «или». При переходе на третий уровень героя ждут еще более серьезные испытания — ему придется перепрыгивать пики.

Здесь будет срабатывать тот же условный блок «Касается цвета?», что и в первом уровне.



ВНИМАНИЕ



- * Сохрани программу через меню «Файл» — «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Затерянный город

v456 0.4



НЕЖДАННЫЕ ГОСТИ

На четвертом уровне герою предстоит столкнуться не с опасными препятствиями, а с падающими на голову ядовитыми пауками. Сможет ли он от них увернуться?

Добавим новые спрайты. Загрузи из папки ресурсов с сайта scratchbook.piter.com файл «Паук.svg». Спрайт появится в списке спрайтов под игровым полем.

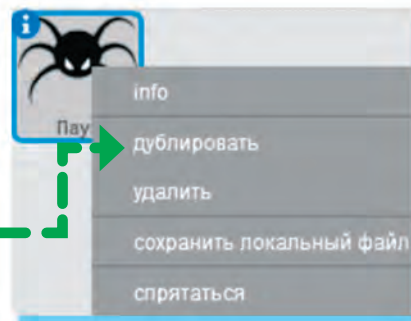


Соберем для него скрипты.



При запуске игры паук прячется. При переходе на четвертый уровень он появляется в случайном месте наверху экрана и движется вниз, а потом снова появляется наверху. Получив сообщение об окончании игры, он опять исчезает с экрана.

С одним пауком опытный игрок, скорее всего, легко справится. Чтобы азарта было больше, запустим на игровое поле второго. Для этого щелкни по изображению спрайта паука под игровым полем правой кнопкой мыши и выбери пункт меню «Дублировать».





В списке спрайтов появится «Паук 2».

Чтобы пауки не появлялись одновременно, добавим в основной скрипт второго паука задержку времени 2 секунды. Все остальное остается так же, как у первого.



Оба паука появляются с предсказуемой задержкой времени, но в непредсказуемых местах экрана. Так что подготовиться к встрече с ними нелегко!

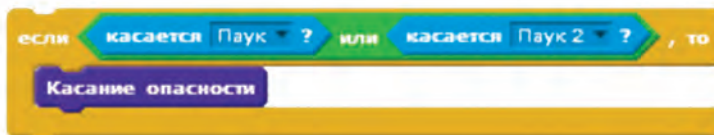
Добавим в основной скрипт ученого проверку касания пауков. Ее можно добавить к блоку проверки касания цвета, воспользовавшись несколькими вложенными блоками «или».

А можно просто добавить отдельный блок.

ВНИМАНИЕ



- * Сохрани программу через меню «Файл» — «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Затерянный город



ПОСЛЕДНИЙ БОЙ

Успешно избежав укусов ядовитых пауков, герой переходит на пятый уровень игры. Угроза миновала? Не тут-то было – к паукам добавляются клубки змей, которые выкатываются прямо под ноги. Они тоже смертельно опасны.

Добавим спрайт со змеями. Загрузи его из папки с ресурсами с сайта scratchbook.piter.com, файл «Клубок змей.svg». Спрайт появится в списке под игровым полем.



Змеи будут появляться при переходе на пятый уровень. Соберем для них скрипт.

когда щелкнут по

спрятаться

когда фон изменится на Фон Уровень 5

показаться

всегда

ждать 0.1 секунд

перейти в x: 213 y: -85

повторять пока не $\text{положение } x < -200$

изменить x на -5

повернуть на 15 градусов

перейти в x: 213 y: -85

когда я получу Поражение

спрятаться

когда я получу Победу

спрятаться

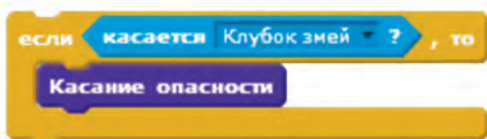
Как и пауки, змеи скрываются при запуске игры и появляются только на пятом уровне. Они в правом нижнем углу экрана и катятся влево. Чтобы они перекатывались, мы заставим спрайт поворачивать на 15 градусов против часовой стрелки после каждого перемещения на 5 пикселей влево. Когда змеи докатятся до левого края экрана, они исчезнут и с задержкой 0,1 секунды снова появятся справа. Получив сообщение об окончании игры, змеи исчезают.

ВНИМАНИЕ

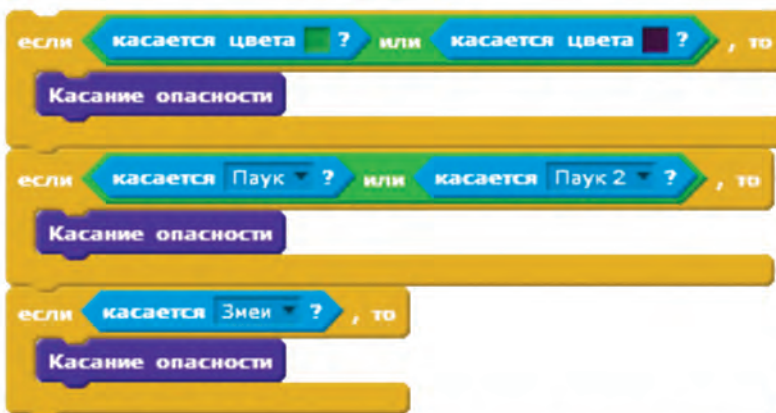


Поэкспериментируй с шагом перемещения и временной задержкой, чтобы изменить скорость появления змей. Чем чаще они появляются справа и быстрее катятся, тем труднее играть.

Добавим проверку касания змеи в основной скрипт ученого.



Таким образом, в основном скрипте блок проверки опасностей повторяется три раза.



На этом основной скрипт ученого завершен!

СКРИПТЫ № 3 И № 4

ВНИМАНИЕ



- * Сохрани программу через меню «Файл» — «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Затерянный город



Для того чтобы игра завершалась специальным сообщением, а не просто останавливалась, создадим новый спрайт.

Загрузи из папки с ресурсами с сайта

scratchbook.piter.com файл «Сообщение Победа.svg».

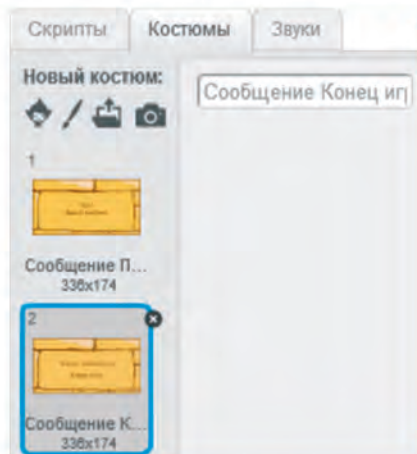
Спрайт появится в списке спрайтов под игровым полем.



Добавим для него еще два коротких скрипта. Они такие же, как у пауков и змей: заставляют спрайт ученого пропадать с экрана, когда получено сообщение об окончании игры.



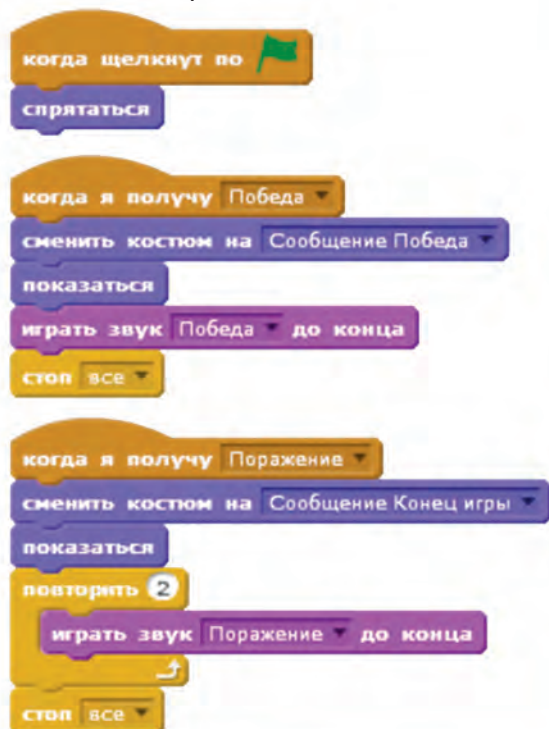
ДОБАВЛЕНИЕ СООБЩЕНИЙ ОБ ОКОНЧАНИИ ИГРЫ



На вкладке «Костюмы» добавим второй костюм для этого же спрайта.

Нажми на значок папки и загрузи из папки с ресурсами с сайта scratchbook.piter.com файл «Сообщение Конец игры.svg».

Соберем скрипты для спрайта сообщений, чтобы программа показывала нужное в зависимости от исхода игры.

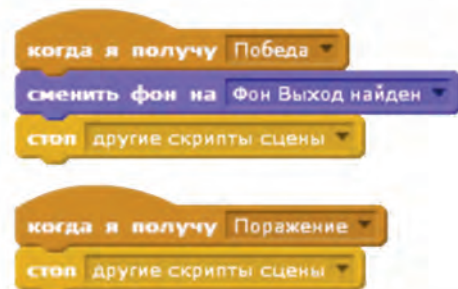


При старте игры панели с сообщениями не видно. Если получено сообщение «Победа», появляется панель с надписью «Ура! Выход найден!» и играет короткая радостная мелодия. Действие всех скриптов в игре останавливается. Если получено сообщение «Поражение», появляется панель с надписью «Жизни закончились. Конец игры» и дважды проигрывается печальный затухающий звук. Действие всех скриптов в игре останавливается.

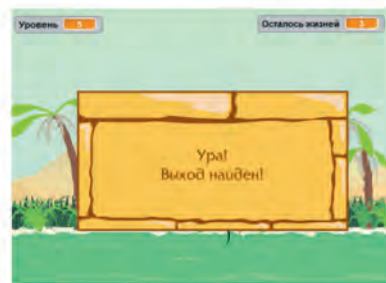
А теперь вернемся в скрипты фона. Для этого щелкни по изображению сцены слева под игровым экраном и перейди на вкладку «Скрипты».



Добавим два коротких скрипта, обрабатывающих сообщения о победе и поражении.



Теперь при получении сообщения «Победа» фон в игре поменяется на пейзаж с изображением реки и пирамид — мы нашли выход из лабиринта!

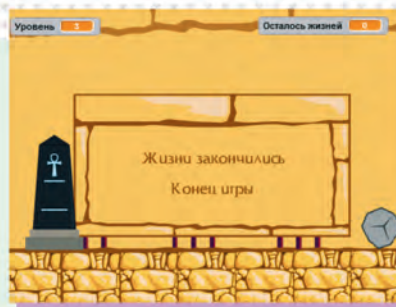


При получении сообщения «Поражение» фон не меняется. Но на нем кроме панели с сообщением появится изображение надгробного камня. Увы, ученый не смог выбраться из опасного лабиринта...

ВНИМАНИЕ



- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



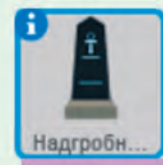
Затерянный город



ПОСЛЕДНИЕ ШТРИХИ

Добавим спрайт надгробного камня (файл «Нагробный камень.svg»).

Спрайт появляется только при получении сообщения «Поражение». Осталось добавить скрипты для изменения переменной «Уровень». Для игры она не нужна, но хочется знать, на каком уровне ты находишься. Перейди в скрипты фона и собери там пять однотипных скриптов.

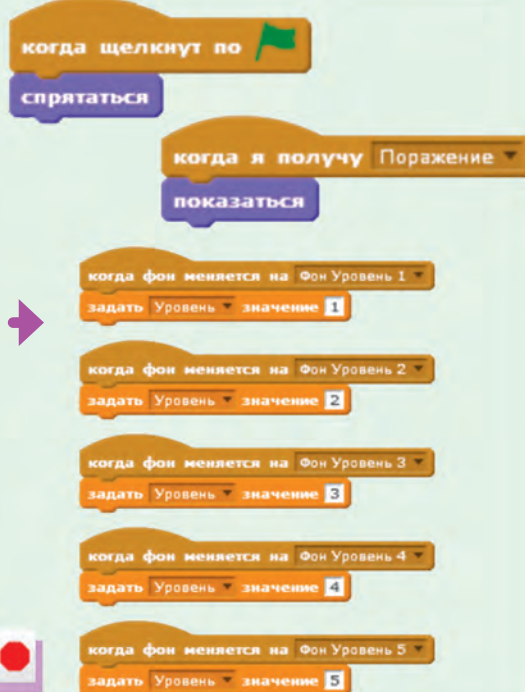


Теперь переменная «Уровень» обновляется при каждой смене фона.

ВНИМАНИЕ

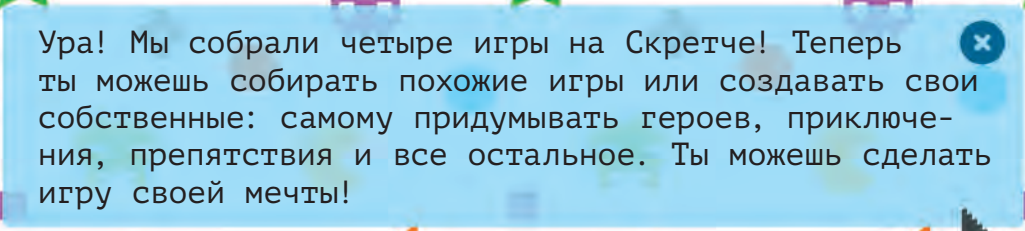


- * Сохрани программу через меню «Файл» – «Сохранить».
- * Проверим работу программы. Щелкни по зеленому флажку в скрипте или на плашке над игровым полем.



Затерянный город





Ура! Мы собрали четыре игры на Скретче! Теперь ты можешь собирать похожие игры или создавать свои собственные: самому придумывать героев, приключения, препятствия и все остальное. Ты можешь сделать игру своей мечты!

СОЗДАЙ СВОЮ ИГРУ

Хочешь поиграть во что-то другое?

Ты сможешь собрать любую игру сам. Используй ресурсы Скретч или нарисуй собственную графику и запиши звуки — главное, чтобы игра была интересной.

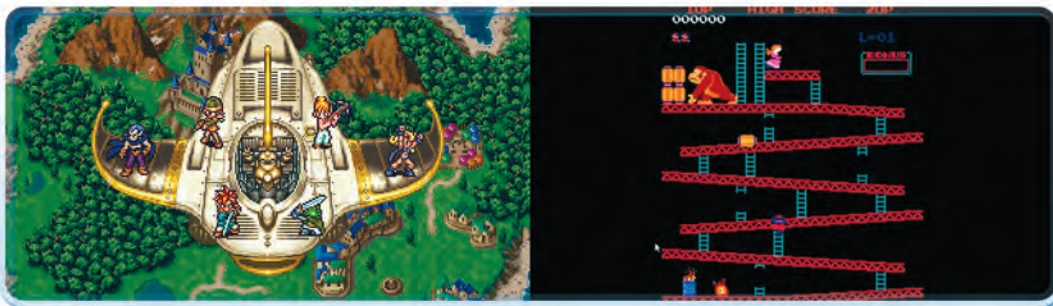
ИСТОРИЯ КОМПЬЮТЕРНЫХ ИГР



Компьютерные игры появились на свет вместе с компьютерами. Одной из первых видеоигр считается ОХО — крестики-нолики, в которые в 1952 году с человеком играла огромная вычислительная машина EDSAC. Технологии быстро развивались, и уже через 15–20 лет видеоигры, сначала на телевизионных приставках и аркадных автоматах, а чуть позже и на компьютерах, стали доступны всем желающим. Космические сражения, посадка на Луну, шахматы, баскетбол, теннис, битвы с пришельцами — вот на чем основывались сюжеты первых видеоигр.



Возможность соревноваться, сражаться, искать клад и разгадывать тайны, изучать неизвестные планеты — все, о чем мечтают люди в обычной жизни, появляется в сюжетах компьютерных игр.



В Скретче ты можешь собрать собственную игру – с любым сюжетом, музыкой и графикой. Платформер, квест или просто мультфильм с любимыми героями – все это совсем нетрудно сделать в Скретче.

ВНИМАНИЕ

Используй для разработки прототипов готовые спрайты и фоны из библиотеки Скретч.

ВНИМАНИЕ

Сохрани программу через меню «Файл» – «Сохранить».

Для вдохновения изучай проекты других пользователей на <https://scratch.mit.edu>. На главной странице сайта показаны игры, разбитые по категориям: самые популярные, недавно опубликованные и так далее. Любой проект можно скопировать вместе с кодом, чтобы использовать в собственных разработках или разобраться, как он устроен.

А при помощи поиска по проектам можно искать игры по жанрам или темам.





СОЗДАЕМ СТОРИБОРД



Игра начинается с идеи у тебя в голове. Но прежде чем открывать программу, нужно все хорошо продумать и распланировать — именно так делают настоящие разработчики. При создании игры удобно использовать сториборд — план разработки.

ОПИСАНИЕ

☐ Экшн
или приключение

☐ Платформер

☐ Драка
или сражение

☐ Аркада

ИГРОКИ

☐ Одиночный игрок

☐ Двое игроков

☐ Несколько
игроков

ЦЕЛИ

Каковы цели игры? _ _ _ _ _

Где происходит игра? _ _ _ _ _

Кто ее герои? _ _ _ _ _

Какова их миссия или цель? _ _ _ _ _

Есть ли у героя друг или враг? _ _ _ _ _

Как он прогрессирует в игре? _ _ _ _ _

Когда он выигрывает и проигрывает? _ _ _ _ _

ПРАВИЛА

Как проходит игра? _ _ _ _ _

Как двигается игрок? _ _ _ _ _

Как игрок будет побеждать врагов или собирать
предметы? _ _ _ _ _

Сколько очков выигрывает игрок? _ _ _ _ _

Есть ли ограниченное время? _ _ _ _ _

Нарисуй, как примерно может выглядеть экран по время игры.

ЭСКИЗ ЭКРАНА	КРАТКОЕ ОПИСАНИЕ
ЭСКИЗ ЭКРАНА	КРАТКОЕ ОПИСАНИЕ
ЭСКИЗ ЭКРАНА	КРАТКОЕ ОПИСАНИЕ

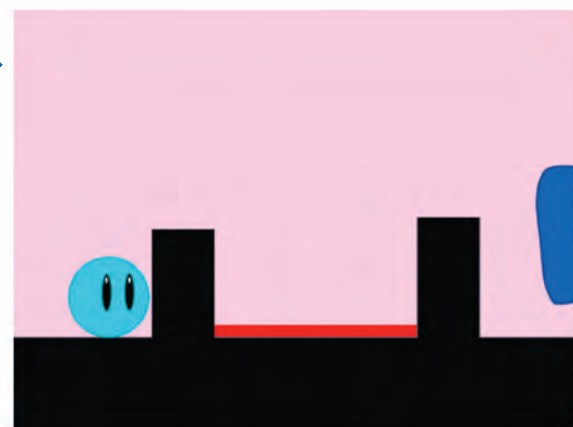
СОЗДАЕМ ПРОТОТИП. ЗАГРУЖАЕМ ГРАФИКУ ИЗ БИБЛИОТЕКИ СКРЕТЧ

После того как сюжет и правила игры в общих чертах готовы, пора собирать прототип игры. Здесь героев обычно изображают простые геометрические фигуры, можно обойтись без специальных фонов и музыки. Задача автора игры на этапе прототипа – понять, получается ли воплотить в игре придуманный сюжет и хорошо ли работают правила. Прототип помогает настроить движения и взаимодействие персонажей.

Например, так может выглядеть прототип игры «Проект “Сатурн”».



А так – прототип игры «Затерянный город».



А так выглядит прототип игры «Футбол».

Прототип помогает определить:

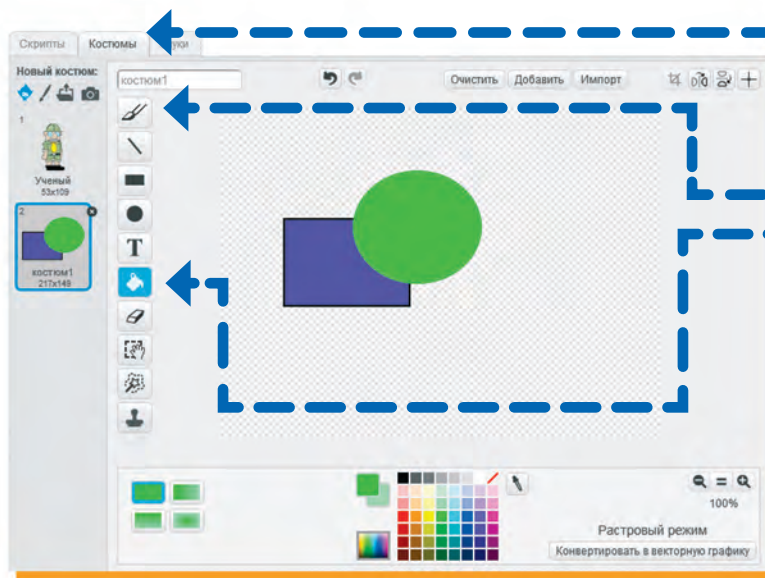
- * примерные размеры спрайтов;
- * их положение при запуске игры;
- * количество фонов, препятствий;
- * принципы взаимодействия спрайтов, то есть что происходит, когда они касаются друг друга;
- * путь и скорость перемещения героя;
- * принцип подсчета очков.

СОВЕТ

- * Начинай с простого сюжета.
- * Сначала создай спрайты, которые управляют основными движениями и взаимодействием персонажей, потом добавляй более сложные движения и дополнительные детали.
- * Сохраняй проект после всех изменений.
- * Проверяй, как работает проект, добавляя и удаляя по одному-два блока – так ты сможешь отыскать ошибку, если что-то пойдет не так..

СОЗДАЕМ СОБСТВЕННЫЕ ГРАФИКУ И ЗВУК

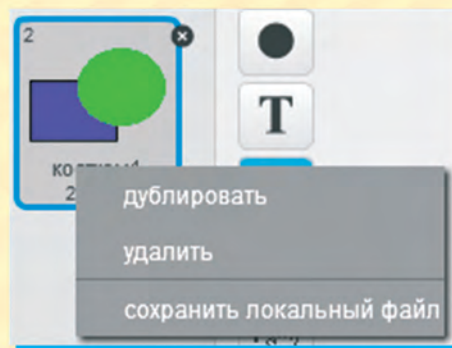
До этого мы пользовались готовыми картинками и звуками, но ты можешь создать собственные, каких не будет больше ни у кого! Графический редактор Скретча позволяет создавать фоны и костюмы спрайтов, а музыкальный редактор – записывать и редактировать музыку. Для того чтобы открыть графический редактор, надо просто перейти на вкладку «Костюмы».



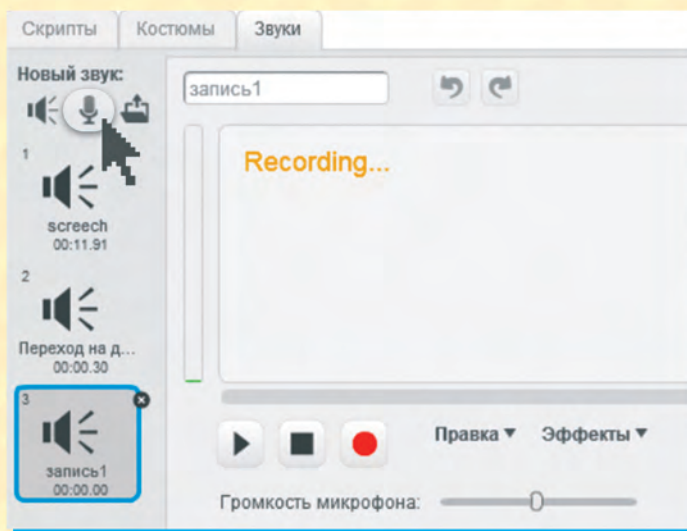
Чтобы создать новый костюм или фон, нажми на значок кисти под шапкой вкладки «Костюмы».

Здесь можно увеличить спрайт, чтобы изменить его детали. Можно нарисовать собственные фигуры при помощи инструментов: карандаша, заливки, прямых и кривых линий, многоугольников. Можно писать текст (правда, только английскими буквами).

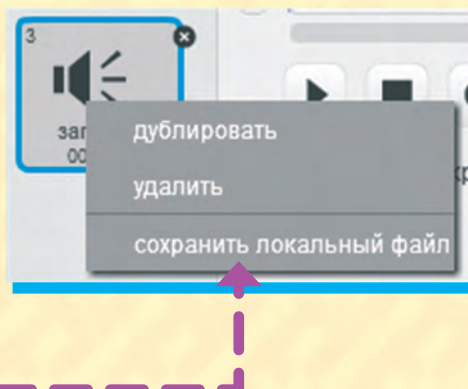
После создания спрайта или фона его можно сохранить в файл, который будет храниться на твоём компьютере. Нажми правой кнопкой мыши на значок спрайта и выбери пункт меню «Сохранить локальный файл». Готово!



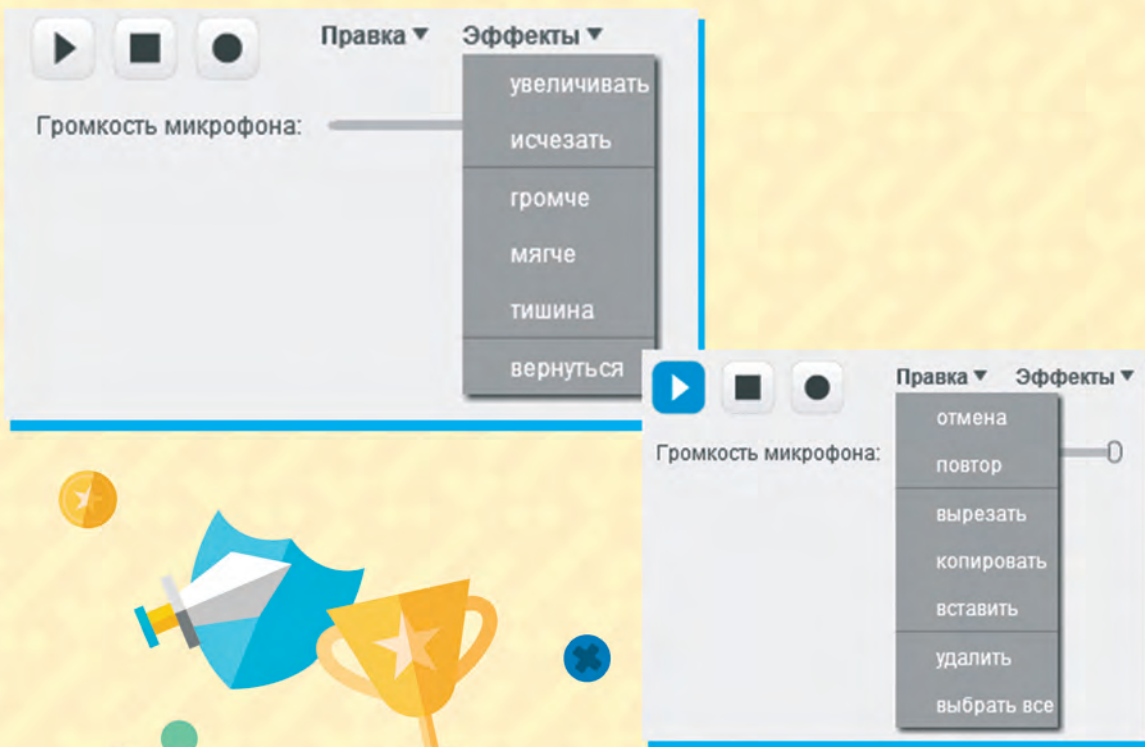
На вкладке «Звуки» можно записать собственный звук. Для этого понадобится микрофон. В ноутбуках он обычно встроенный, в настольных компьютерах его надо специально подключить. Для записи звука нажми на значок микрофона под шапкой вкладки «Звуки».



Так же, как и картинку, созданный в редакторе звуковой файл можно сохранить на компьютер.



Звуковые файлы можно редактировать.



**УДАЧИ
В СОЗДАНИИ
ИГР!**



Юлия Торгашева

Программирование для детей. Учимся создавать игры на Scratch

Серия «Вы и ваш ребенок»

Заведующая редакцией
Ведущий редактор
Литературный редактор
Художники
Корректоры
Верстка

Ю. Сергиенко
Н. Римицан
О. Нестерова
Д. Семенова, А. Сергиенко
Н. Сидорова, Г. Шкатова
Д. Семенова

ББК 32.973.2-018

УДК 004.3

Торгашева Ю.

Т60 Программирование для детей. Учимся создавать игры на Scratch. — СПб.: Питер, 2018. — 128 с.: ил. — (Серия «Вы и ваш ребенок»).

ISBN 978-5-4461-0619-6

Мы уверены, что ты любишь смотреть мультфильмы, играть на компьютере, планшете и в телефоне. А знаешь, почему герои игр двигаются? Ими управляют программы — наборы команд, которые говорят персонажам, когда им убежать, хватать, прятаться, ловить и многое другое.

А теперь — самое главное. Ты сам можешь создавать такие программы! Приготовься, ведь тебе придется постараться, чтобы запустить первый спутник, найти сокровища, спрятанные в египетских пирамидах, пристыковать космический корабль к станции, накормить пингвина вкусным обедом и, наконец, помочь сборной России по футболу одержать победу.

6+ (Для детей старше 6 лет. В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ISBN 978-5-4461-0619-6

© ООО Издательство «Питер», 2018

© Серия «Вы и ваш ребенок», 2018

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

Изготовлено в России. Изготовитель: ООО «Прогресс книга». Место нахождения и фактический адрес:

191123, Россия, г. Санкт-Петербург, ул. Радищева, д. 39, к. Д, офис 415. Тел.: +78127037373.

Импортер в Беларуси: ООО «ПИТЕР М», РБ, 220020, г. Минск, ул. Тимирязева, д. 121/3, к. 214, тел./факс 208 80 01

Дата изготовления: 03.2018. Наименование: детская литература. Срок годности: не ограничен.

Налоговая льгота — общероссийский классификатор продукции ОК 034-2014, 58.11.13.000 — Книги печатные для детей.

Подписано в печать 14.03.18. Формат 84×108/16. Бумага офсетная. Усл. п. л. 13,440. Тираж 3000. Заказ 0000.

Отпечатано в полном соответствии с качеством предоставленных издательством материалов в Типографии «Вятка».

610033, Киров, ул. Московская, 122.