3 уровень (часть 1)

Платформер – это жанр компьютерной игры, в которой одной из главных черт игрового процесса является прыганье по платформам, преодоление различных препятствий, собирание предметов и др. Одно из обязательных условий завершения игры-платформера – прохождение уровня от начала и до конца. В большинстве случаев, платформер представлен в виде обычного линейного уровня, в котором главный персонаж начинает двигаться слева направо.

Третий уровень нашей игры будет представлен в виде двухмерного платформера. Марио необходимо перепрыгнуть ямы с лавой, обойти врагов и спасти принцессу. После этого игра заканчивается. Главное отличие 3 уровня от первых двух в том, что компоненты уровня будут двигаться вокруг Марио.

Существует множество различных вариантов игровой физики движений. Рассмотрим основные.

1. Герой двигается свободно и от его движений (влево, направо, вверх, вниз) не зависят остальные компоненты уровня, а также сам уровень.

2. Уровень полностью зависит от движений героя. Компоненты уровня и сам уровень двигаются относительно героя, т.е. герой стоит на месте, а уровень двигается вокруг него так, что создается иллюзия движения героя, а не уровня с компонентами.

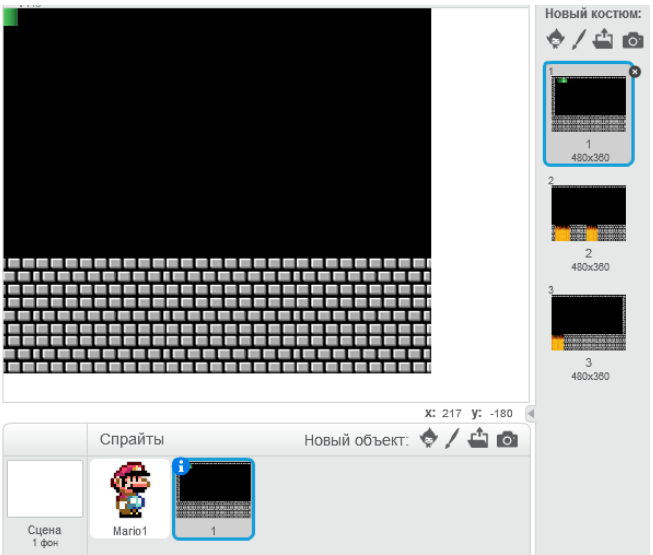
3. Смешанный вариант. Уровень двигается вокруг героя и сам герой может двигаться в определенных координатах.

Наш 3 уровень будет смешанного типа.

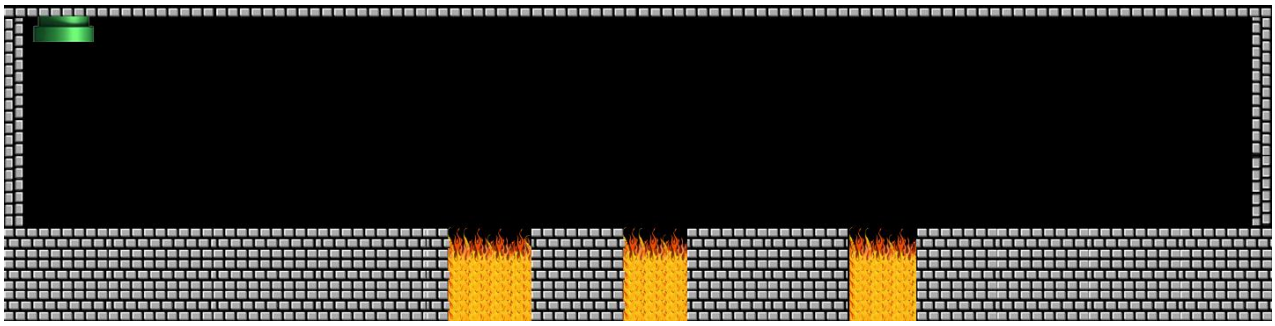
Внимание! 3 уровень игры будет создаваться отдельно от первых двух уровней. Создайте новый проект и назовите его ***Уровень 3***.

Удалите ***Рыжего Кота*** и загрузите в проект спрайт ***Марио1***. Для спрайта импортируйте костюм ***Марио2***. В свойствах спрайта ***Марио1*** выберите стиль вращения ↔.

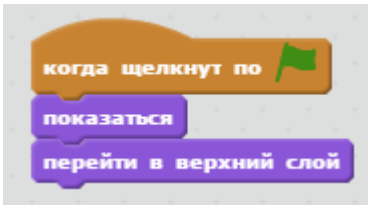
Фон уровня будет динамически сменяемым, т.е. являться спрайтом и двигаться относительно ***Марио1***. Загрузите в директорию ***Спрайты*** фон ***1*** (разрешение 480×360). Фоны ***2*** и ***3*** загрузите как костюмы фона 1.



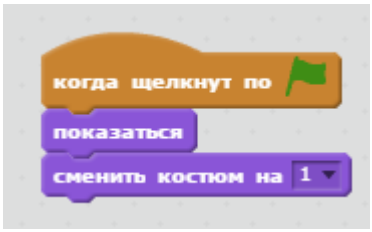
Все три фона имеют одинаковое разрешение (480×360), но изначально фон создавался в графическом редакторе и представлял собой цельное изображение (размеры 1440×360).



Чтобы во время программирования не создавалась путаница между спрайтом фона ***1*** и ***Марио1***, вставьте в спрайт Марио следующие команды:



В спрайт фона ***1*** добавьте команды:

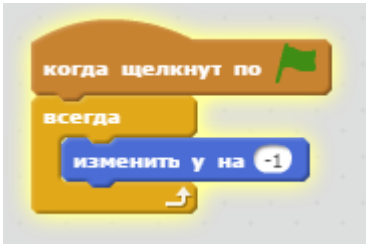


Теперь при запуске проекта, всегда будет появляться Марио и нужный фон (первый сегмент фона для начала уровня).

Предлагаем начать разработку уровня с программирования гравитации, которая понадобится для последующего создания управления ***Марио1***.

Гравитация в Scratch – это изменение Y-координаты, при котором спрайт движется сверху вниз, т.е. значение Y-координаты спрайта постоянно уменьшается.

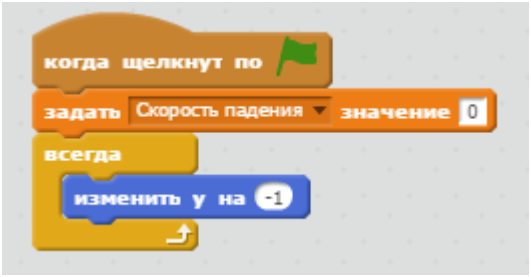
Т.к. спрайт ***Марио1*** будет подчиняться гравитации, следовательно, его Y-координата тоже должна уменьшаться. Для этого необходимо постоянно изменять Y-координату на какое-то отрицательное значение (или отнимать от координаты какое-то значение). Для этого в спрайте ***Марио1*** составьте следующие команды:



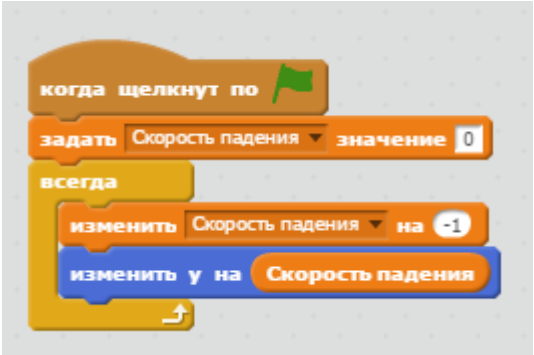
Значение изменения Y-координаты равно -1. При запуске 3 уровня спрайт Марио равномерно «падает» вниз сцены.

Во многих играх при падении герой начинает ускоряться, т.е. скорость падения увеличивается в зависимости от высоты падения. Чтобы спрайт ***Марио1*** начал ускоряться, необходимо ввести переменную, которая будет каждый раз изменяться на отрицательное значение при падении. Значение изменения переменной, в свою очередь, должно складываться с предыдущим значением и передаваться в команду изменения Y-координаты пока Марио падает.

Создайте переменную и назовите её ***Скорость падения***. В начале скрипта задайте переменной значение ***0***.



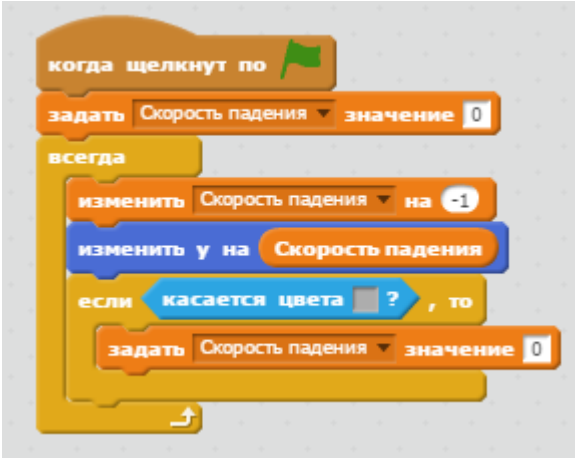
Далее переменная должна изменяться, например, на ***-1*** пока Марио падает. Измененное значение переменной должно передаваться в команду ***Изменить y на...***



При запуске проекта спрайт Марио1 падает вниз с ускорением.

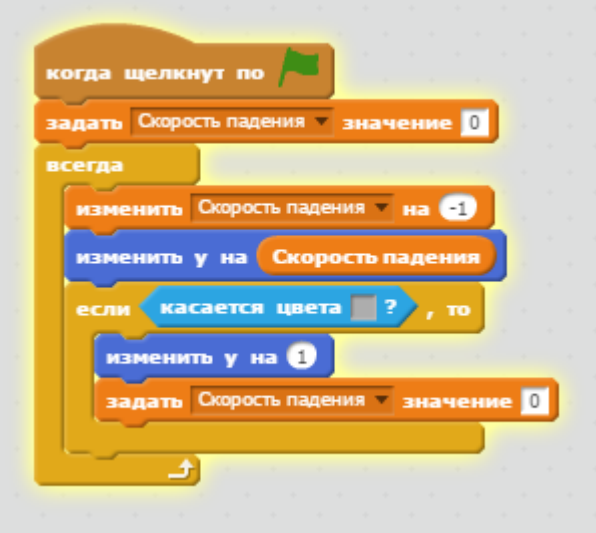
При падении Марио «проваливается» вниз сцены, но он должен «приземляться» на камни. Необходимо, чтобы при касании камней Y-координата Марио (переменная ***Скорость падения***) перестала изменяться, т.е. нужно задать переменной новое значение в скрипте (***0***).

Создайте условие, по которому при касании Марио серого цвета значение переменной ***Скорость падения*** станет равно ***0***.



После проверки условия видно, что Марио, падая с высоты, ускоряется, но касании камней всё равно падает, но уже с равномерной скоростью. Когда спрайт Марио касается цвета камней и скорость падения становится равным ***0***, программа не завершает свое действие и не останавливает Марио, а снова выполняет цикл, вначале которого указано, что необходимо изменить Y-координату на ***-1***, и потом переходит к условию.

Вставьте в начало условия команду ***Изменить у на…***, которая будет «приподнимать» спрайт Марио пока он касается камней, т.е. постоянно увеличивать его Y-координату на 1 при касании. Команда ***Изменить у на…***должна быть равна ***1***, так как Y-координата постоянно изменяется на ***-1***.

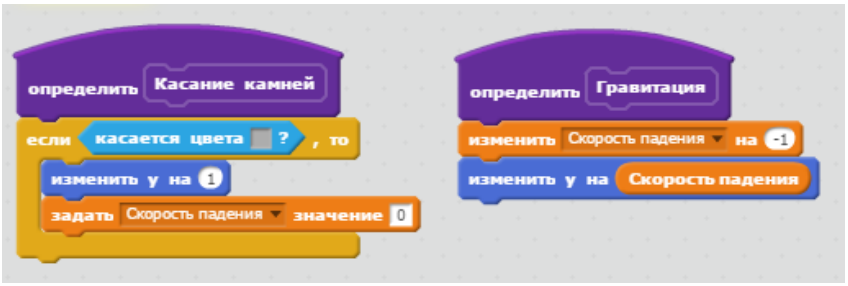


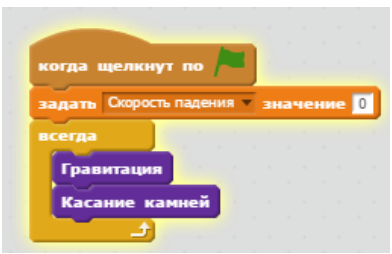
При падении спрайт Марио1 «приземляется» на камни.

Скрипт гравитации понадобится при создании 3 уровня. Чтобы каждый раз не повторять команды гравитации в других скриптах, создайте два ***Других блока*** и назовите их ***Гравитация*** и ***Касание камней***. Параметры блоков задавать не нужно.

Процесс создания и принцип работы ***Других блоков*** подробно рассматриваются в PDF-документе ***Оптимизация скрипта при помощи Других блоков***.

В спрайте ***Марио1*** разберите скрипт ***Когда щелкнут по***  и составьте новые:

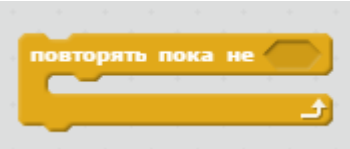




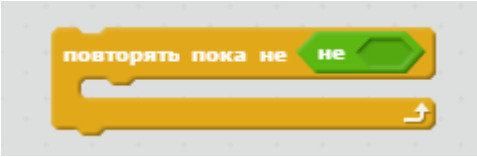
При запуске уровня Марио «падает вниз и немного «вклинивается» в текстуры камня. В зависимости от высоты и ускорения, он то больше, то меньше «вклинивается» в камни (срабатывает эффект торможения). Условие ***Касание камней*** создано верно, но ему не хватает команды, которая бы проверяла, постоянно ли стоит Марио на камнях или «парит в воздухе».

Созданные скрипты создают гравитацию и условие, когда гравитация перестает работать (падение Марио и условие, при котором он больше не падает). При касании камней перестает работать гравитация, т.е. останавливается падение, но не достигается задача, чтобы Марио ходил или стоял ровно на камнях.

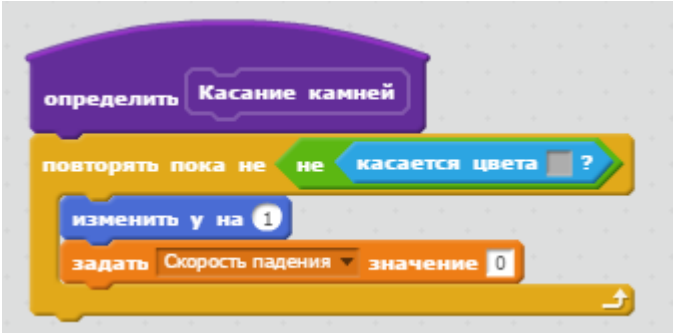
Команда условия ***Если…, то*** лишь единожды срабатывает в цикле и не действует в нем постоянно. Новая команда должна постоянно проверять, касается ли Марио камней. Если Марио касается поверхности, то команда приподнимет его на ***1***. В этот же момент включится гравитация. Действия будут компенсировать друг друга, таким образом, создастся видимость, что Марио стоит ровно на поверхности (на самом деле каждую секунду будут выполняться десятки операций по поднятию и опусканию спрайта Марио, фактически не заметные для человеческого глаза). Новая команда должна постоянно проверять условие и постоянно повторять действия условия (повторять действия пока соблюдается условие). Воспользуйтесь блоком ***Повторять пока не...***



Преобразуем команду ***Повторять пока не***... в «повторять пока…». К сожалению, такой блок в Scratch отсутствует, поэтому в условие придётся добавить второе отрицание. Переместите оператор ***Не…*** в блок ***Повторять пока не...***



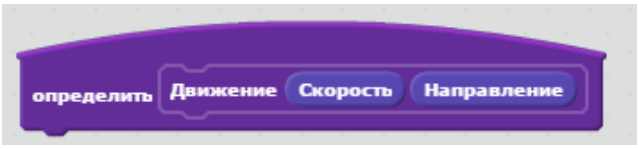
В скрипте ***Касание камней*** замените условие ***Если…, то*** на блок ***Повторять пока не...*** Сенсор касания цвета и команды ***Изменить у на 1*** и ***Задать скорости падения значение 0*** переместите в новое условие.



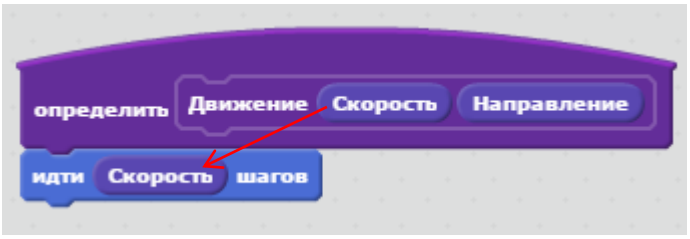
Марио ровно стоит на поверхности, но при падении на мгновение погружается в неё, а затем «выталкивается» обратно. В некоторых случаях возможно дребезжание спрайта на сцене.

Чтобы устранить проблему, в контекстном меню блока ***Определить Касание камней*** выберите ***Редактирование – Параметры***. Поставьте метку в чек-боксе ***Запустить без обновления экрана*** и нажмите ***ОК***.

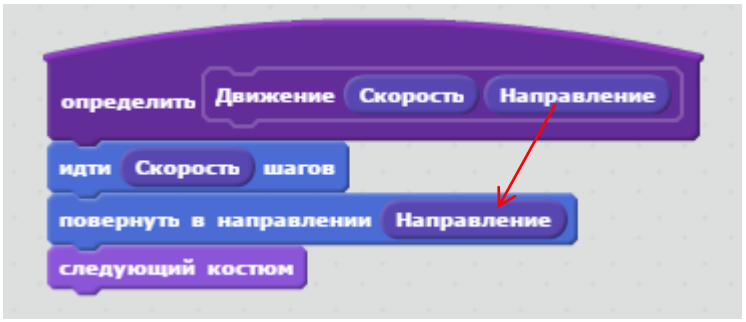
Задайте для спрайта ***Марио1*** управление влево и вправо. В группе ***Другие блоки*** создайте новый блок и назовите его ***Движение***. При создании блока добавьте два параметра (см. рисунок). Первую форму назовите ***Скорость***, вторую – ***Направление***.



Параметр ***Скорость*** отвечает за движение спрайта. К блоку ***Движение*** прикрепите команду ***Идти… шагов*** и интегрируйте в него ***Скорость***.

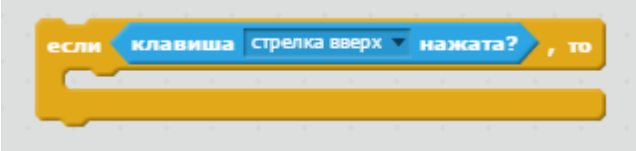


Параметр ***Направление*** работает с командой ***Повернуть в направлении...*** Проделайте те же операции, что и со скоростью, а в конец скрипта вставьте блок ***Следующий костюм***.



В скрипт, где в цикле ***Всегда*** находятся блоки ***Гравитация*** и ***Касание камней*** вставьте два условия ***Если…, то…*** и сенсоры для управления спрайтом ***Марио1*** при помощи клавиш ***Стрелка влево*** и ***Стрелка направо***. Внутрь условий из ***Других блоков*** перетащите команды ***Движение*** со значениями ***5*** и ***90*** для клавиши ***Стрелка направо*** и ***5*** и ***-90*** – для клавиши ***Стрелка влево***. Очередность значений в команде соответствует очередности параметров в блоке ***Движение***. Значение ***5*** – это количество шагов для Марио, ***90*** — поворот в правую сторону, ***-90*** – в левую сторону.

В цикле ***Всегда*** создайте условие для прыжка. В сенсоре ***Клавиша…нажата?*** выберите ***Стрелка вверх***.



«Прыжок» в Scratch – это изменение Y-координаты в большую сторону (со знаком «плюс»). В проекте за изменение Y-координаты отвечает переменная ***Скорость падения***, через блок ***Гравитация*** переменная изменяет значение Y-координаты. В условие для прыжка вставьте команду ***Задать скорость падения…*** Введите значение, например ***15***.

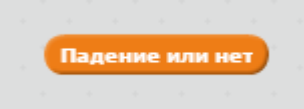
Внимание! Если прыжок не работает, то перезагрузите проект.

Если несколько раз нажать на прыжок, то спрайт Марио будет подниматься всё выше и выше, пока не «застрянет» в верхней части сцены. Это происходит потому, что нажать

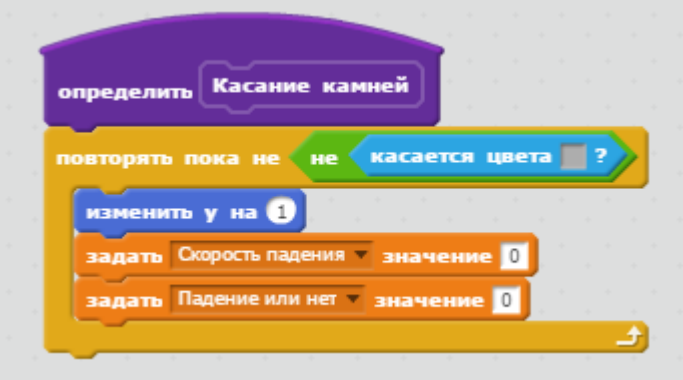
прыжок можно сколько угодно раз, т.е. команда прыжка не ограничена одним нажатием, когда Марио «находится в воздухе». По умолчанию, у большинства платформеров прыжок можно нажать только один раз, когда герой отрывается от поверхности.

Необходимо создать ограничение в скрипте прыжка. Ограничение будет построено по принципу проверки: находится ли спрайт Марио в полёте или нет. В зависимости от положения Марио, скрипт «ограничения» будет задавать и изменять значение. Скрипт прыжка, проанализировав это значение, будет выполняться либо нет (будет срабатывать или не срабатывать прыжок при нажатии клавиши ***Стрелка вверх***).

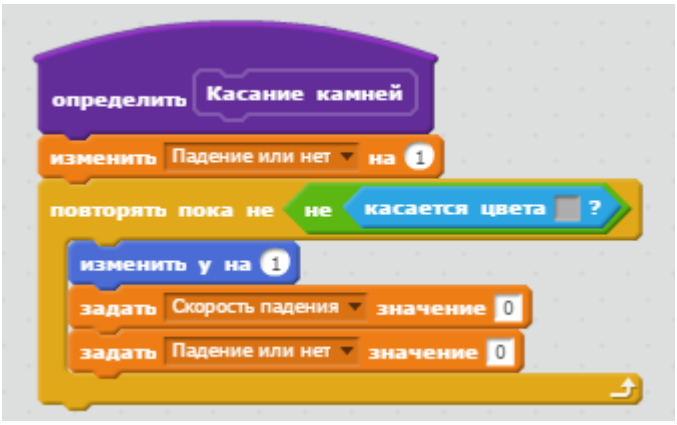
Создайте новую переменную ***Падение или нет***, которая будет сохранять значение.



Задайте значение для этой переменной. Если Марио стоит на поверхности, то он не падает (пусть при этом условии значение переменной равняется ***0***). Скрипт проверки касания поверхности уже есть, это ***Касание камней***. Вставьте в скрипт команду ***Задать… значение…***



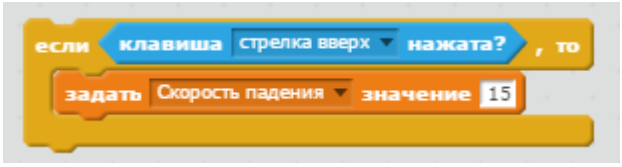
Если Марио не касается поверхности, то значение переменной должно поменяться на ***1***. Команду изменения переменной поставьте в начало скрипта ***Касание камней***. Переменная будет постоянно изменяться, а когда спрайт Марио коснется поверхности, то значение изменится на ***0*** (когда Марио подпрыгнет, условие перестанет работать, т.е. значение не будет равняться 0, а станет снова изменяться).



Это необходимо для того, чтобы старое значение переменной при остановке игры или после окончания игры не сохранялось и не передавалось команде ***Изменить… на 1*** в

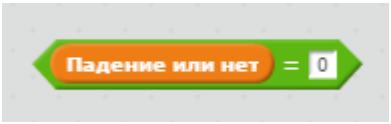
скрипте ***Касание камней***. Изменение переменной всегда должно начинаться от ***0***, иначе будут ошибки, которые повлекут за собой неправильную работу скрипта прыжка.

Теперь нужно задать проверку переменной ***Падение или нет***, в зависимости от её значения будет работать или не работать прыжок. По условию прыжок происходит, когда нажата стрелка вверх.



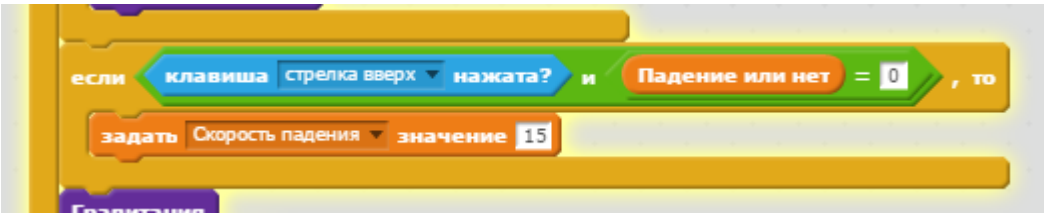
Мы уже знаем, что, когда Марио находится в полёте, переменная ***Падение или нет*** приобретает значение больше ***0***. Следовательно, если переменная будет больше ***0***, то прыжок не должен действовать, а именно, не должно выполняться условие, по которому при нажатии стрелки вверх переменная ***Скорость падения*** примет значение ***15***, и Марио изменит свою Y-координату.

Условие прыжка дополните командой, которая сравнивает переменную и при значении ***0*** разрешит действие прыжка. Вставьте переменную ***Падение или нет*** в оператор ***Равно***, в поле справа введите значение равное ***0***.



Этот небольшой скрипт теперь необходимо вставить в условие прыжка, но нужно, чтобы выполнялись нажатие ***Стрелки вверх*** и значение переменной равное ***0***. Чтобы реализовать данный алгоритм, воспользуйтесь оператором ***И***.

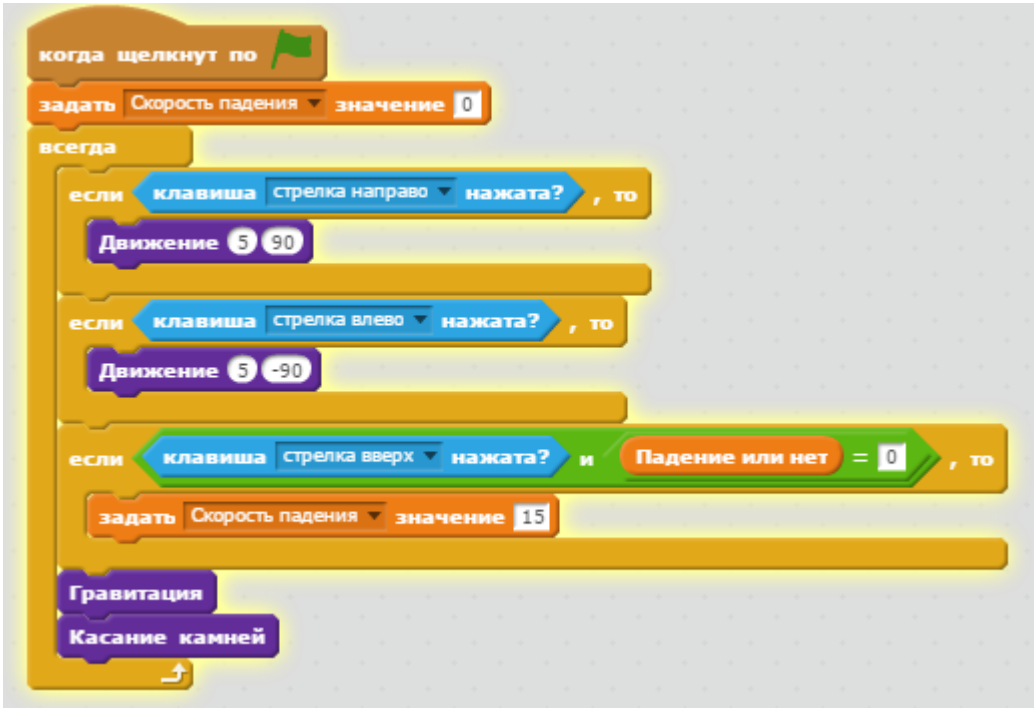
В область слева вставьте сенсор ***Клавиша стрелка вверх нажата?***, а в другую форму – значение переменной ***Падение или нет = 0***.



Игровая физика для спрайта Марио создана.

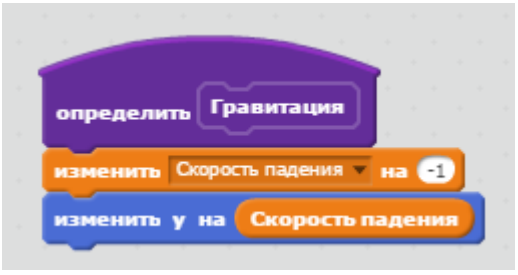
Скрипты, которые должны быть созданы на данном этапе в спрайте Марио.

1. Скрипт движения спрайта ***Марио1***:



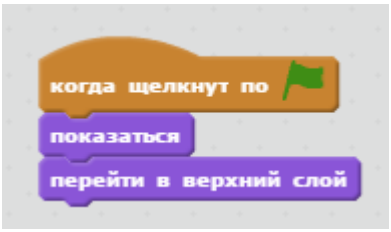
2. Скрипт ***Движение***:

3. Скрипт ***Гравитация***:



4. Скрипт ***Касание камней***:

5. Скрипт перехода ***Марио1*** на верхний слой:



Скрипт, который должен быть создан на данном этапе в спрайте фона ***1***: