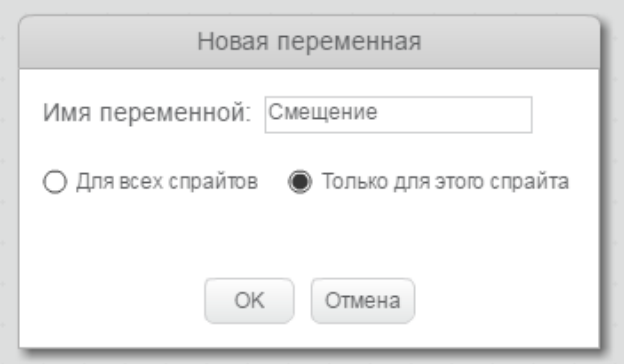
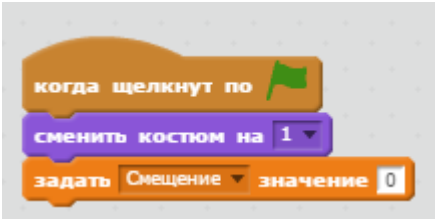
3 уровень (часть 2)

Данная часть инструкции по разработке 3 уровня посвящена динамически сменяемому фону. В зависимости от смещения спрайта ***Марио1*** по оси ***X*** направо или влево, фон смещается на такое же пройденное расстояние, в противоположную, движению Марио, сторону. Сам фон построен по принципу клонов, т.е. спрайт фона ***1*** создает клон себя самого, перемещает его за видимую область экрана, а в нужный момент клон появляется на сцене, но только с последующим костюмом спрайта фона ***1***.

Для начала необходима новая переменная, которая будет задавать значение смещения клонам спрайта фона ***1***. Выделите спрайт фона ***1*** и создайте новую переменную. Новая переменная должна быть только для этого спрайта, т.к. она будет работать только в этом спрайте и нигде более. Назовите переменную ***Смещение***.



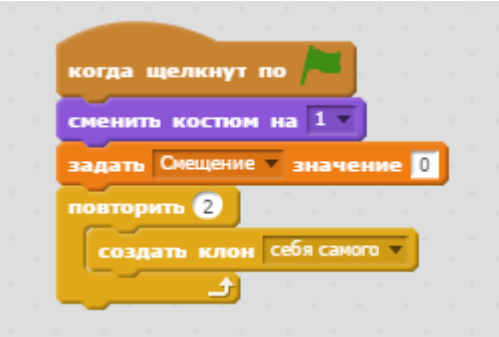
Фон ***1*** будет смещаться. Марио начнет «движение» с фона ***1***. Переменная должна быть задана в начале уровня. В скрипте спрайта фон ***1*** задайте переменной значение ***0***. Блок ***Показаться*** уберите.



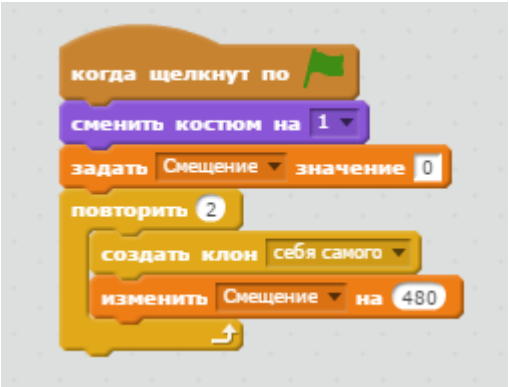
Мы выбрали такое значение переменной, т.к. после запуска уровня отсчет смещения фона легче вести с ***0***.

Спрайт фона ***1*** уже присутствует на сцене, два фоновых рисунка должны идти следом. Спрайт фона ***1*** должен создать два своих клона, чтобы происходила смена фона на уровне. Будут созданы только два клона спрайта, поскольку уровень состоит из трех фоновых рисунков.

Чтобы создать два клона спрайта фон ***1***, задайте в цикле ***Повторить*** значение ***2***. В цикл вставьте команду ***Создать клон себя самого***.



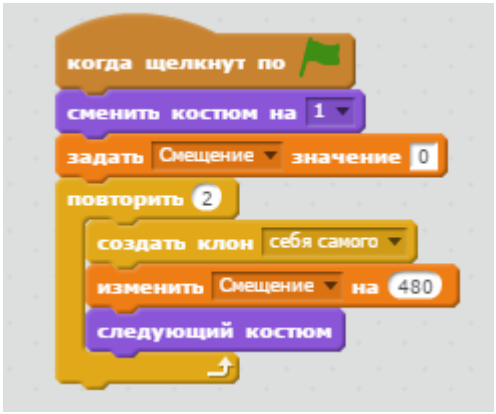
Необходимо, чтобы созданные клоны переместились за пределы сцены. Для этого сместите их, т.е. измените переменную ***Смещение*** на ***480***.



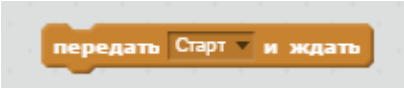
480 – это количество пикселей по горизонтали, т.е. видимая область сцены. Все, что находится за пределами сцены, не видимо для пользователя.

Если запустить проект, то фоны никуда не сместятся. Дело в том, что переменная ***Смещение*** пока не привязана напрямую к координатной оси ***X***. Смещение фонов будет происходить по оси ***X***, в дальнейшем переменная ***Смещение*** будет отвечать за данную ось.

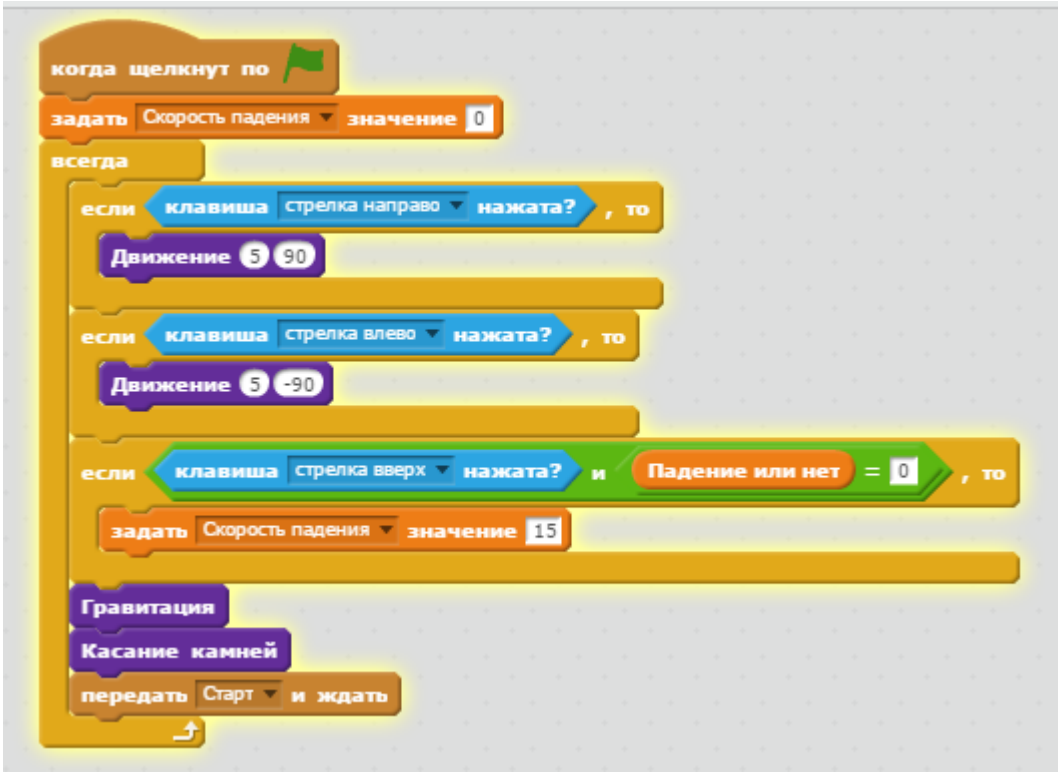
Вставьте команду ***Следующий костюм*** в блок повтора. Спрайт фона один и он не меняется, а меняются последовательно его костюмы, в зависимости от движения Марио и смещения первого фона.



В зависимости от движения спрайта Марио, фоны должны смещаться, а затем появляться. Перейдите в спрайт ***Марио1*** и создайте новое сообщение ***Старт***. Блок будет передавать сообщение о том, что Марио сместился и теперь программе необходимо рассчитать значение смещения фона, и, как следствие, сместить фон.



Команду-сообщение ***Передать…и ждать*** вставьте в конец скрипта движения Марио. Блок должен передать все команды движения в скрипте, а не часть их.



Можно обойтись и без данного сообщения и просто передать значения движения Марио какому-то скрипту для расчета смещения фона, но могут возникнуть определенные трудности с визуализацией данного процесса.

Команда ***…и ждать*** в блоке передачи сообщения предназначена для того, чтобы фон плавно смещался, без прерываний, которые могут возникнуть, если смещение фона не успевает за расчетами значений движений Марио. Как следствие движения Марио могут выглядеть прерывисто.

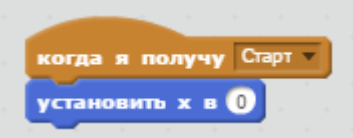
Перейдите в спрайт фон ***1***.

Сообщение о том, что спрайт Марио двигается существует, но его необходимо передать. Следующий шаг – это создание скрипта, в котором будет рассчитываться, сколько на экране должно появиться того или иного фона при движении Марио, в какой координате должен появиться фон, а также условие появления фона (движения Марио не являются условием появление фона, т.к. они заставляют срабатывать это условие).

Старт передает сигнал о том, что спрайт Марио движется, следовательно, скрипт должен начинаться с неё.

Костюмы фона ***1*** должны «понимать», в какой координатной точке им появляться (показываться) при смещении предыдущего фона в очереди. Для этого им нужно

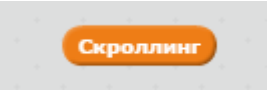
установить координаты той точки, в которой они должны появиться. Т.к. фоны не будут двигаться по вертикали, следовательно, нужно установить только координаты по оси ***X***. Задайте команду ***Установить x в…***



Нельзя задать фиксированные координаты по оси ***X***, так как фоны постоянно смещаются. Поэтому необходимо найти значение, которое будет меняться в зависимости от движения спрайта ***Марио1***.

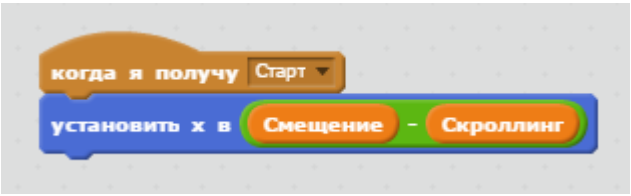
Марио проходит *n*-ое расстояние в любую сторону, следовательно, фон в этот же момент должен сместиться ровно на столько, на сколько сместился Марио, но в обратном направлении. Значение смещения фона должно быть равно пути, пройденному Марио.

Из всего этого следует, что надо подсчитать, сколько проходит спрайт Марио, чтобы сместить фоны. Марио каждый раз может пройти разный путь, следовательно, и значение будет разным. Поэтому необходимо ввести новую переменную, которой будет передаваться информация, сколько проходит Марио. Создайте новую переменную и назовите её ***Скроллинг***.

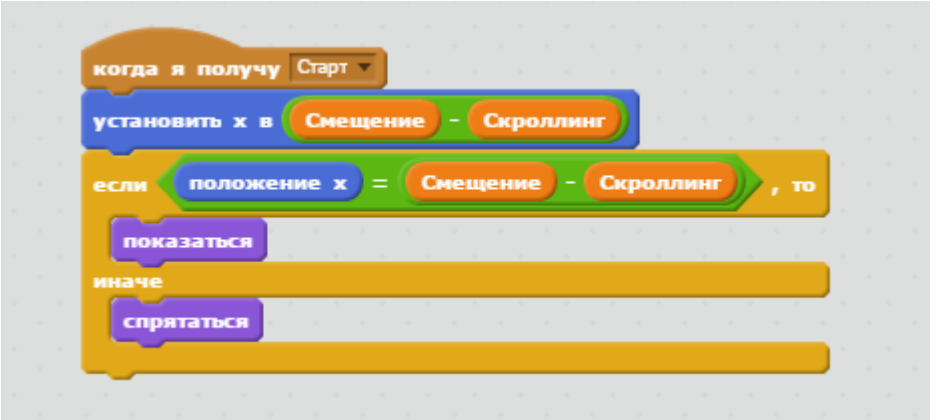


Однако одной переменной, подсчитывающей путь, пройденный Марио, мало. Дело в том, что пройденный спрайтом Марио путь не есть значение смещения фона по оси ***X***. Мы только подсчитываем, сколько проходит Марио и передаем в переменную ***Скроллинг***, которая также не является смещением фона. За значение смещение фона по оси *X* отвечает другая переменная ***Смещение***. Спрайт-фон ***1*** начинает со значением смещения ***0***, следующий костюм – со значением ***480***, а третий костюм – со значением ***960***, следовательно, все фоны нужно смещать по оси ***X*** на какое-то значение ***Смещения*** минус путь, пройденный Марио:

Этот скрипт (формулу) вставьте в команду ***Установить x в...***



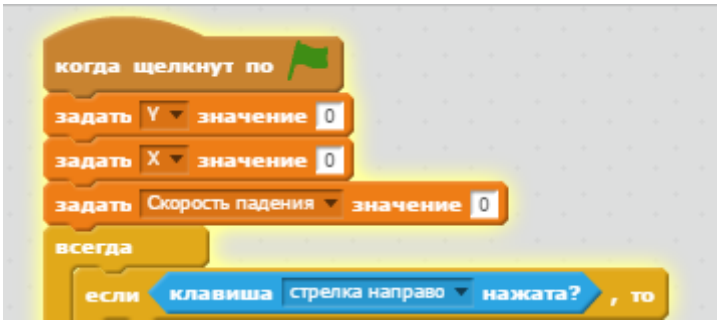
Фоны «умеют» смещаться, но «не знают», в каких координатах по оси ***X*** они должны появляться и исчезать на экране, когда Марио их достигает или проходит. Для этого введите условие в скрипте.



Если положение фона по оси X равно формуле «Смещение» минус «Скроллинг», значит фон должен показаться, а иначе спрятаться.

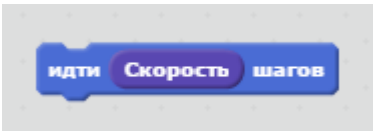
Остается главный вопрос: «Как привязать ***Скроллинг*** к движению спрайта ***Марио1***?» У Марио, так же, как и у фона, должно быть своё какое-то определенное положение. Спрайт Марио умеет перемещаться в двух плоскостях ***X*** и ***Y***, следовательно, чтобы узнать его положение на сцене, нужно знать эти координаты. Но пока мы их не знаем. Чтобы их узнать, перейдите в спрайт ***Марио1*** и создайте соответствующие переменные ***X*** и ***Y***.

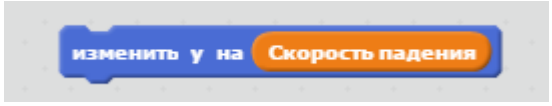
Задайте переменным значения по 0 в начале уровня (в скрипте движения ***Марио1***).



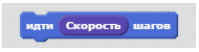
Созданные переменные необходимо поставить туда, где раньше использовались «синие» команды движения ***X*** и ***Y***.

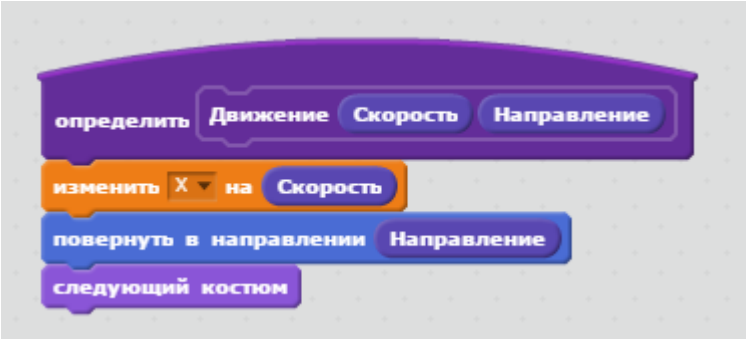
Спрайт Марио двигается по определенным координатам. За установление этих координат в скриптах ***Движение*** и ***Гравитация*** отвечают команды ***Идти Скорость шагов*** и ***Изменить Y на Скорость падения***.



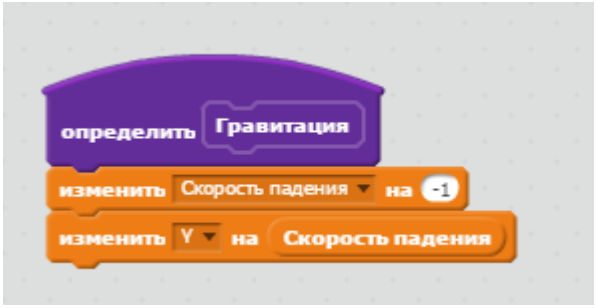


Хоть эти команды и просчитывают движение Марио, они не могут передать значения координат спрайта. Поэтому их необходимо заменить новыми переменными.

В скрипте ***Движение*** вместо команды ***Идти Скорость шагов***  поставьте команду ***Изменить x на Скорость*** шагов:

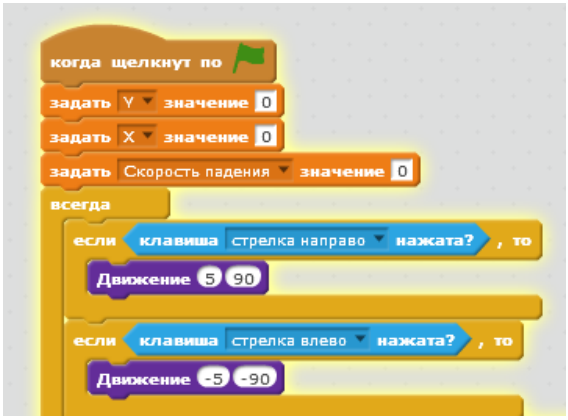


В ***Гравитации*** вместо ***Изменить y на Скорость падения***  поставьте ***Изменить y на Скорость падения***:



В скрипте ***Касание камней*** замените блок  на команду  .

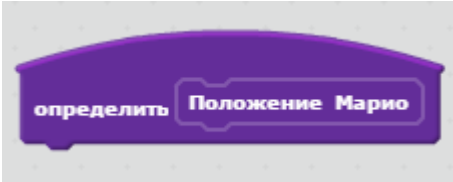
Движение будет считаться не в шагах, а в координатах. В управлении движением спрайта ***Марио1*** в условии ***Если клавиша стрелка влево нажата…*** измените для блока ***Движение*** значение с ***5*** на ***-5***. Движение влево – это отрицательное значение по оси ***X***.



Переменные ***X*** и ***Y*** необходимо преобразовать в настоящие координаты.

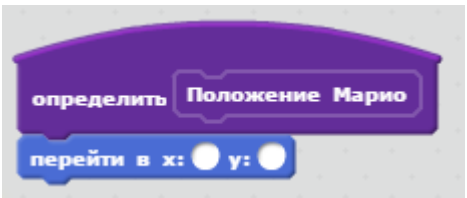
Программа должна считывать положение спрайта Марио по координатам не только когда он двигается, но и когда касается камней или просто находится в покое. Можно отдельно создать скрипт, в котором переменные ***X*** и ***Y*** будут равняться положениям по соответствующим координатам. Однако создание такого скрипта приведет к увеличению числа строк в кодом и возможным ошибкам. Проще создать отдельный блок, который будет отвечать за положение спрайта Марио.

Создайте новый блок и назовите его ***Положение Марио***.



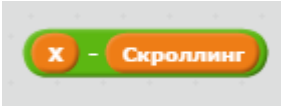
Переменные ***X*** и ***Y*** необходимо приравнять к настоящим координатам. Приравнивание этих переменных к координатам через команды ***Положение X*** и ***Положение Y*** не приведет к перемещению Марио в координаты. Спрайт Марио только получит данные о координатах, но он ещё должен в них перейти. Если спрайту не задать переход в эти координаты, то он будет стоять на месте. Следовательно, в блоке, кроме приравнивания переменных к координатам, нужно еще задать спрайту Марио перемещение в эти координаты.

Данную конструкцию скрипта можно также упростить одной командой. Это блок ***Перейти в x… y…*** Команда позволяет одновременно задавать координаты и перемещать спрайт Марио в эти координаты. Прикрепите команду к блоку ***Положение Марио***.

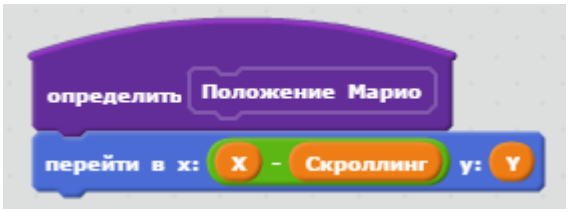


Если с координатой по оси ***Y*** все просто, т.к. кроме самого спрайта Марио больше нечему перемещаться по этой оси, то координату ***X*** необходимо вычислять. Марио двигается по поверхности и поверхность (фон) смещается. Спрайт Марио тоже должен сместиться на столько, на сколько была смещена поверхность. В противном случае, спрайт Марио будет «обгонять» перемещаемый фон и окажется в конце экрана, смещая при этом фон. Если

значение переменной ***X*** больше, чем скорость смещения фона, то нужно отнять от положения спрайта Марио по оси ***X*** то расстоение, которое он проходит.

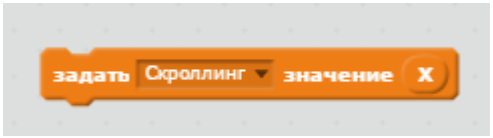


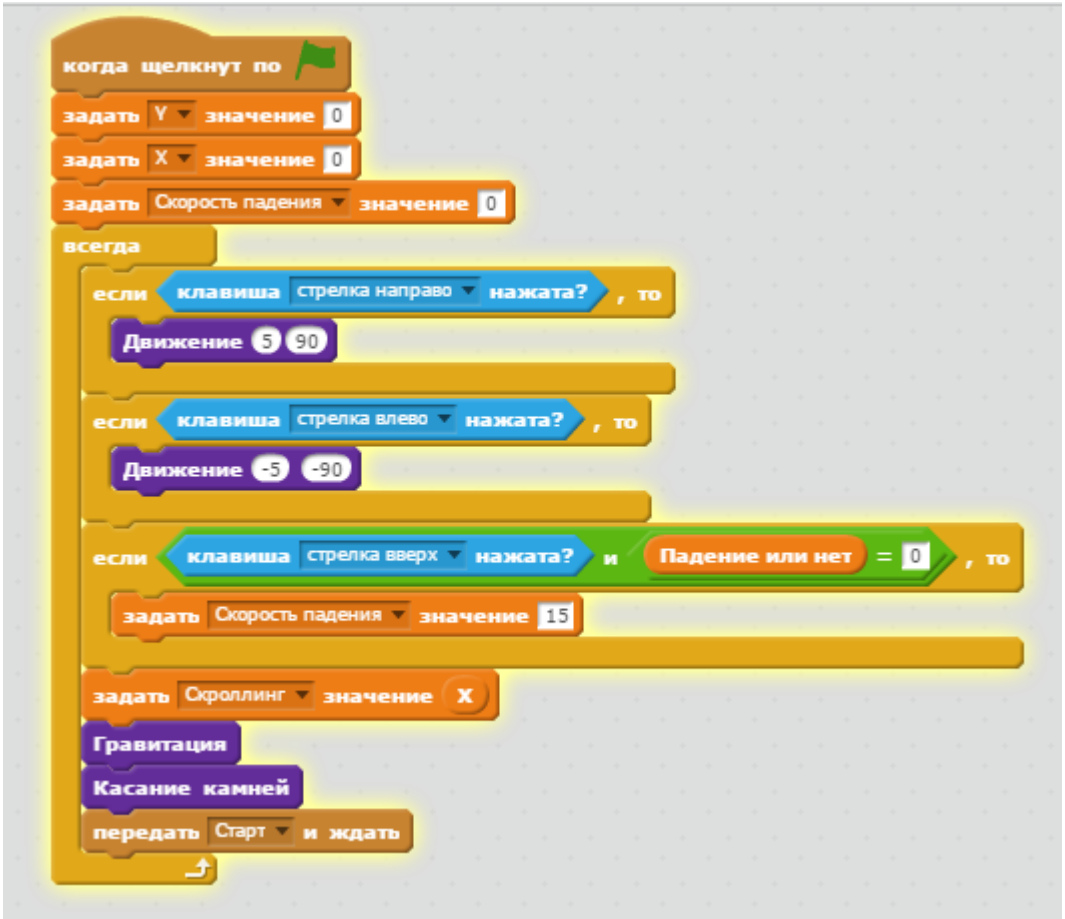
Переменную ***Y*** и формулу вставьте в блок.



Теперь координаты положения спрайта Марио передаются и принимают значения коррдинат.

Далее необходимо связать переменные ***X*** и ***Скроллинг,*** т.к. последняя напрямую связана с перемещением спрайта Марио. Для этого нужно задать значение переменной ***Скроллинг*** значение ***X*** и вставить в скрипт движения, т.к. именно при движении меняется переменная ***X***.



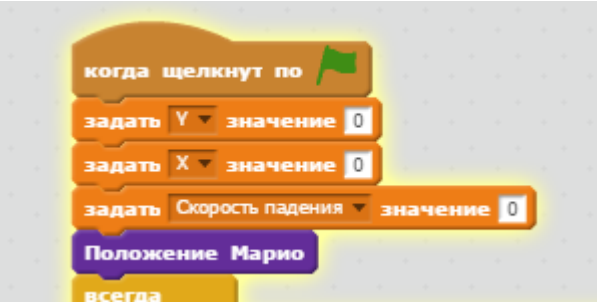


Как будет двигаться Марио, если этим скриптом происходит компенсирование его движений по оси ***X***?

Ранее в инструкции были приведены примеры построения уровней платформера. Один из них, когда все двигается вокруг главного персонажа. В нашем случае, был произведен переход от самого движения персонажа к движению обьктов, в частности фонов, вокруг него. Значение движения по оси ***X*** вычисляется и передется в переменную ***X*** (при этом значение еще не является координатой). Далее происходит компенсирование переменной, чтобы персонаж стоял на месте и не сдвигался, и передача ее в координаты оси ***X***. Но переменная ***X*** далее вычисляется по коду программы, т.к. от нее зависит переменная ***Скроллинг***, которая, в свою очередь, заставляет двигаться фон. При этом создается эффект движения Марио, но по настоящему все двигается вокруг него.

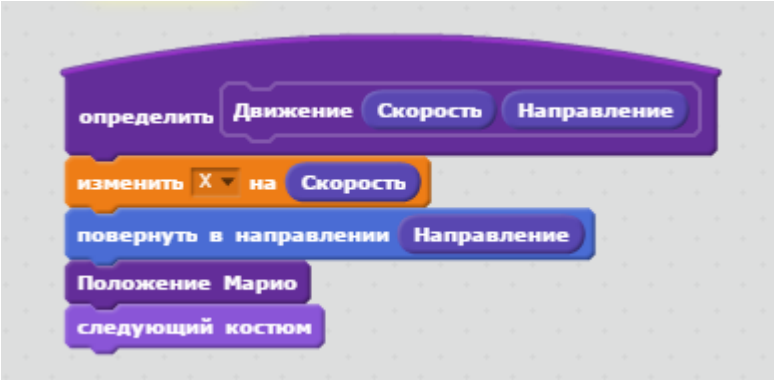
Если все было сделано по инструкции, то при запуске игры, Марио должен примерно находится на середине сцены, а при движении фоны должны сменяться. Однако Марио «пока не знает», в каких координатах он должен находиться, поэтому нужно расставить блоки ***Положение Марио*** в скрипте.

1. Поставьте блок ***Положение Марио*** в начало уровня, до цикла:



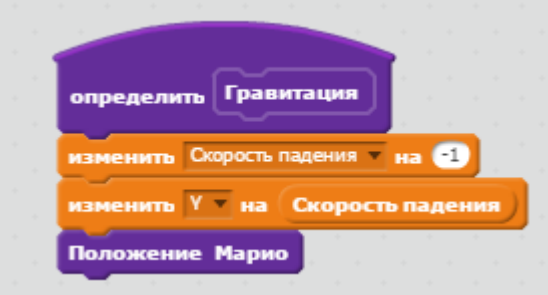
2. Поставьте блок ***Положение Марио*** после того, как будет получено значние ***Скроллинга***:

3. В скрипте определения движения ***Марио1***:



4. В скрипте ***Касание камней***:

5. В скрипт ***Гравитация***:



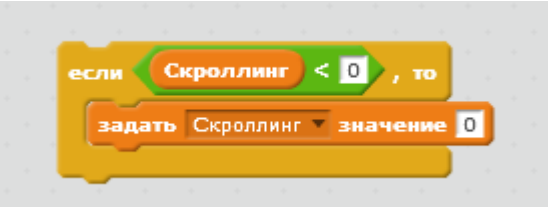
Запустите и проверьте проект. При движении Марио влево, спрайт фона ***1*** смещается и отображается белый фон сцены. Чтобы скрыть белый фон, создайте новый скрипт, который будет останавливать смещение спрайта-фона ***1***.

Ранее было отмечено, что игровая физика уровня построена по принципу «все движется вокруг персонажа». Чтобы фон не смещался, необходимо на определенный момент вернуть предыдущую игровую физику, когда спрайт фона ***1*** статичен, а спрайт ***Марио1*** двигается.

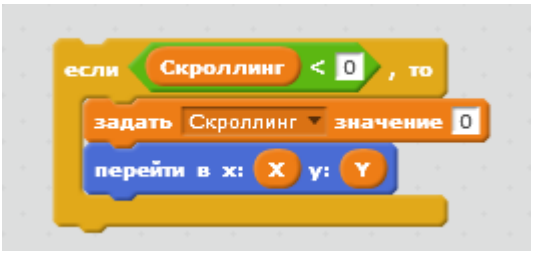
Для этого не нужно переписывать заново скрипт, необходимо лишь задать определенные условия, при которых смещение фона остановится, а Марио сможет смещаться по оси ***X***.

Можно заметить, что когда спрайт Марио смещается влево, то значение ***Скроллинга*** принимает отрицательное значение и спрайт фона ***1*** при этом начинает смещаться. Из этого следует, что ***Скроллинг*** не должен принимать отрицательное значение, а значит

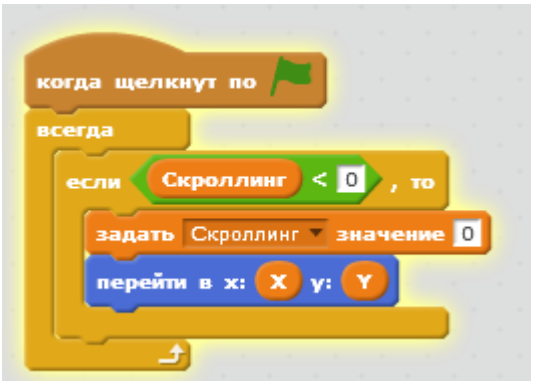
спрайт фона ***1*** не сможет сместиться. Условие будет выглядеть следующим образом: если ***Скроллинг*** меньше нуля, то ему задается значение равное ***0***.



Смещение фона остановилось, однако и Марио в статичном положении. Спрайт Марио должен двигаться по направлению к стене фона. Нужно вставить команду, которая при данном условии разрешит Марио смещаться в координатах по осям. Фон остановлен (не смещается), следовательно, спрайту ***Марио1*** можно задать движения простой командой ***Перейти в х… у…***, в которой находятся переменные координат, и вставить в условие.



В ***Марио1*** поставьте блок, запускающий скрипт, а также цикл, чтобы условие выполнялось непрерывно.



В конце уровня происходит смещение фона (отображается белый фон). Необходимо ввести похожее условие. В данном случае ***Скроллинг*** должен перестать изменяться при значении ***960***. Спрайт Марио будет двигаться вперед к стене, следовательно, необходимо от значения переменной ***X*** отнять ***Скроллинг***, иначе Марио при достижении значения ***Скроллинга*** в ***960***, переместится из середины в конец сцены.

При помощи курсора разместите спрайт фона ***1*** так, чтобы не было видно белого фона сцены. Проверьте проект.

Фон уровня создан, осталось добавить спрайты врагов и условия прохождения 3 уровня.