



Co-funded by the  
Erasmus+ Programme  
of the European Union



University-Enterprises Cooperation  
in Game Industry in Ukraine

# Тренинг GameHub: Разработка компьютерных игр в Unity 3D.

## Основы визуального программирования в PlayMaker Unity 3D

Юлия Трояновская,  
ведущий инженер лаборатории ИС ОНПУ

*561728-EPP-1-2015-1-ES-EPPKA2-CBHE-JP, Одесса – 28 апреля 2017 года*

# PlayMaker - визуальный скриптовый интерфейс

PlayMaker - визуальный скриптовый интерфейс, разработанный Ntong Games.

PlayMaker позволяет разрабатывать игры со сложным поведением и взаимодействиями без необходимости программирования.

Благодаря использованию PlayMaker мы можем сосредоточиться на логике игры, не увязнув в синтаксисе конкретных языков.

Это захватывающая возможность, создавая игры, получать навыки моделирования логики, которые могут нам понадобиться для овладения языком программирования.



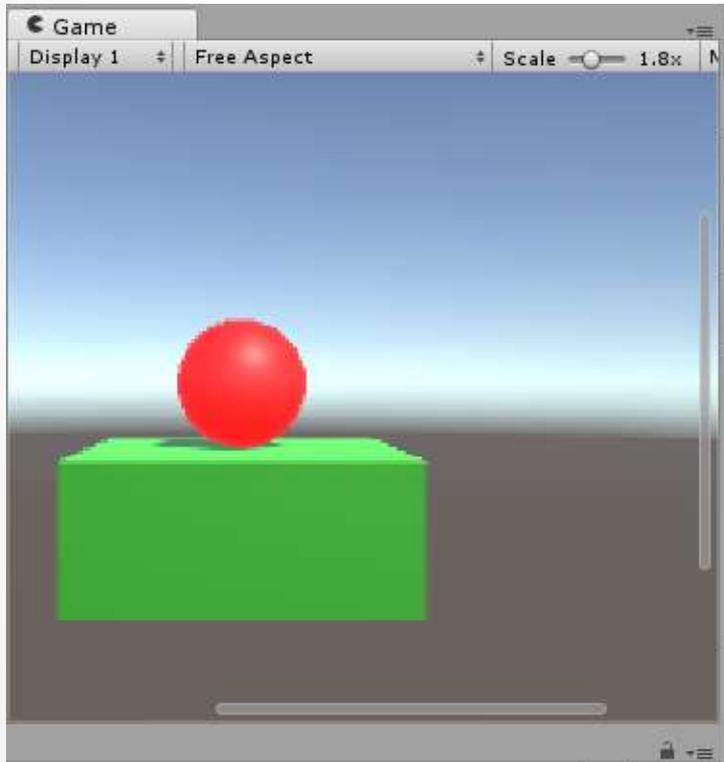
# FSM — Finite-state machine

- В основе использования PlayMaker лежит FSM (конечный автомат). Не нужно пугаться этого понятия.
- FSM - это сценарий поведения, который применяется к игровым объектам. FSM не могут существовать сами по себе, они являются неотъемлемой частью объекта.
- FSM предоставляет нам метод визуального проектирования сценария. Существуют различные методы, с помощью которых инженеры программного обеспечения могут создавать системы или программы, используя диаграммы. FSM являются одним из таких инструментов.

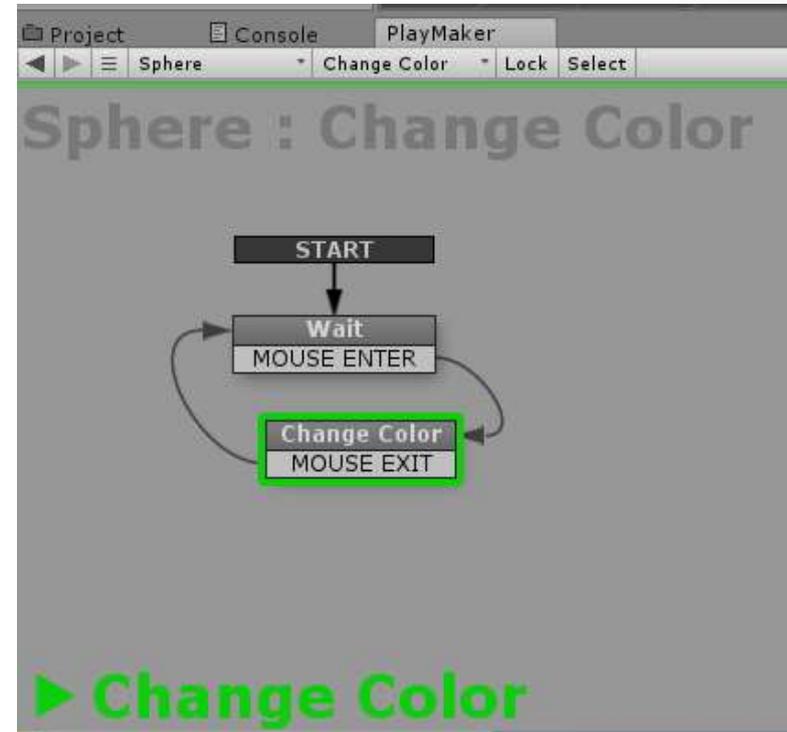


# FSM изменения цвета объекта

Рассмотрим пример. Сфера меняет цвет при нажатии на неё курсором мыши.



Создадим схему FSM в редакторе Play Maker



Но прежде чем мы это сделаем, нам нужно разобраться с понятием FSM более подробно.

# Машина состояний или конечный автомат

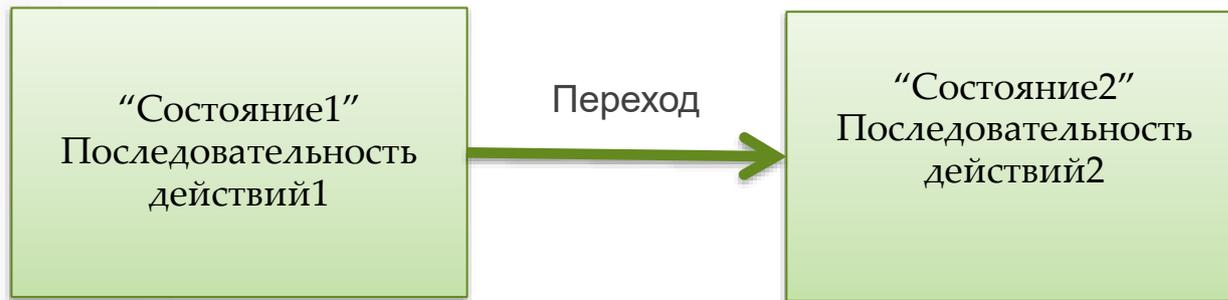
FSM состоят из трех ключевых элементов:

*Состояния*

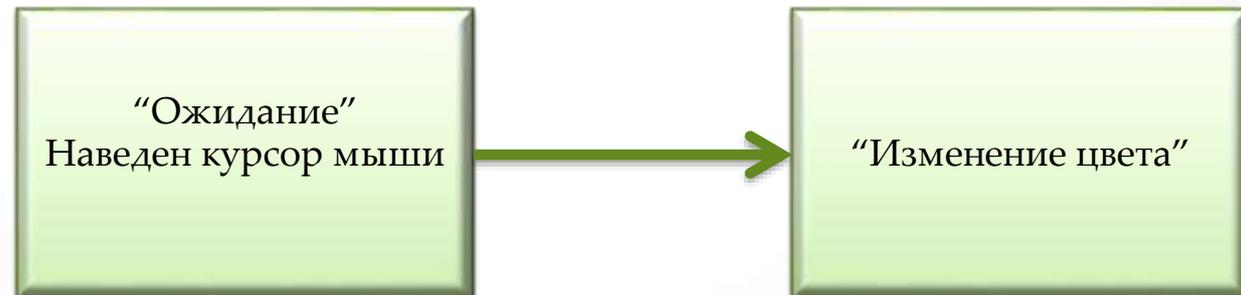
*Последовательности действий*

*Переходы*

Состояние показано на рисунке в виде белых прямоугольников. У нас есть два состояния в этой машине. Одно обозначается как «Состояние1» и другое «Состояние2».



Обозначение состояния является очень важным, поскольку позволяет понимать, какую функцию оно может выполнять.



# Составление сценария

Когда мы щелкаем по объекту Сфера могут произойти различные события. Мы начнем с того, что изменим цвет сферы, когда над ней курсор. Затем мы добавим возможность изменять цвета других объектов, кликнув по сфере. Для начала нам нужно представить наш сценарий в виде маркированного списка, чтобы убедиться, что мы понимаем, что он делает:

- Установить цвет по умолчанию
- Когда пользователь перемещает указатель мыши над объектом, изменить цвет на новый.
- Когда указатель мыши отходит от объекта, вернуть цвет по умолчанию и повторить.
- Если пользователь нажимает на сферу, установить цвет блока как у сферы.

# Построение графического сценария

Установить цвет сферы по умолчанию

Когда **пользователь перемещает указатель мыши над сферой**,  
изменить цвет на **новый**.

Когда **указатель мыши отходит от сферы**,  
вернуть цвет по умолчанию и повторить.

Если **пользователь нажимает на сферу**,  
установить цвет блока как у сферы.

# Построение FSM

Установить цвет сферы по умолчанию

Когда указатель мыши над сферой,  
изменить цвет сферы на новый.

Когда указатель мыши уходит от сферы,  
вернуть цвет по умолчанию .

Если кликнуть по сфере,  
установить цвет блока как у сферы.

- Схема FSM



# Построение графического сценария

- **Задание:**

Придумайте и опишите сценарий игровой ситуации или из реальной жизни в виде схемы FSM или нумерованного списка.

Пример: Как приготовить чай во время Кофе-брейк.

1. Взять стаканчик.
2. Включить чайник.
3. Когда чайник.....

# Получение пакета Unity PlayMaker

- Существует два различных метода установки PlayMaker.
- Первый вариант - использовать Unity Asset Store.
- Второй вариант - непосредственно с сайта разработчика PlayMaker: <http://hutonggames.com/>.
- Как только мы купим пакет, он теперь будет доступен для добавления к проектам через инструменты импорта пакетов в Unity.

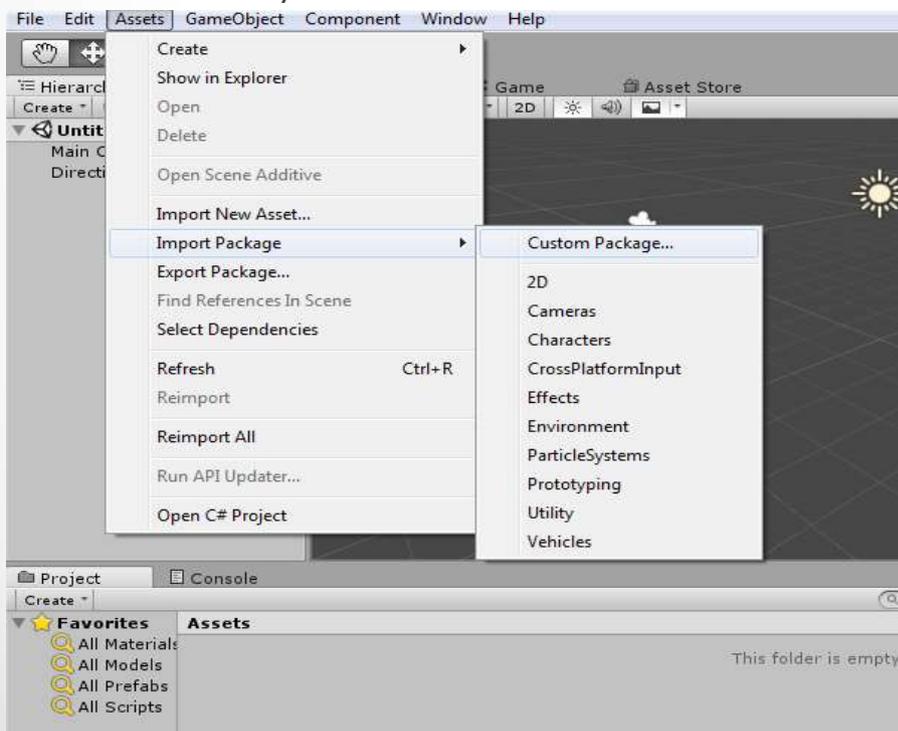


# Установка Unity PlayMaker

Процесс «установки» PlayMaker является процессом импорта пакета PlayMaker в проект, над которым мы работаем.

Мы можем использовать для импорта пакета PlayMaker в наш текущий проект два метода:

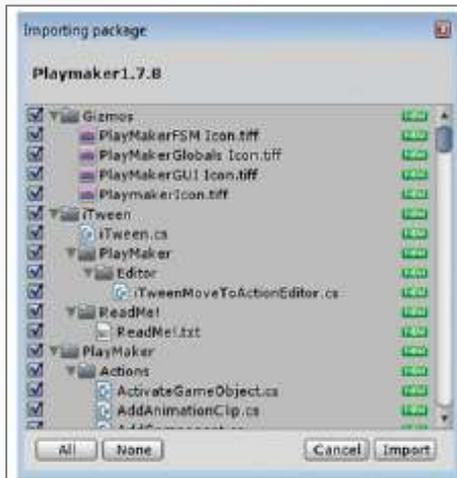
- PlayMaker приобретенный через Asset Store Unity будет добавлен в стандартные Packages, которые доступны для импорта в наш проект.
  - ➔ После загрузки проекта выберите «Assets» в верхнем меню, а затем «Import Package». Всплывающий список выбора должен включать PlayMaker.



- ➔ Если пакет PlayMaker отсутствует в списке, выберите вариант «Custom Package» и перейдите в папку, где пакет PlayMaker хранится на вашем компьютере.

- Другой метод импорта пакета в проект Unity - это перетащить пакет (файл .lip) из папки, в которой он находится в настоящий момент, и поместить его на панель проекта.

# Установка Unity PlayMaker



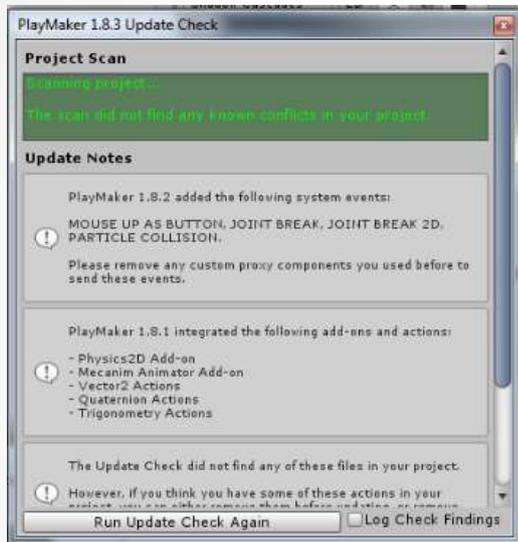
После этого появится диалог для выбора частей пакета для импорта. Мы хотим импортировать все части пакета и нажимаем «Import». Во время разработки проекта будут моменты, когда мы захотим импортировать только небольшую часть пакета, конкретную модель или аудиофайл.



Как видно появилась папка «Playmaker» в «Assets». Но наберитесь терпения еще несколько шагов.

В появившемся окне выбираем «Install Playmaker».

# Установка Unity PlayMaker



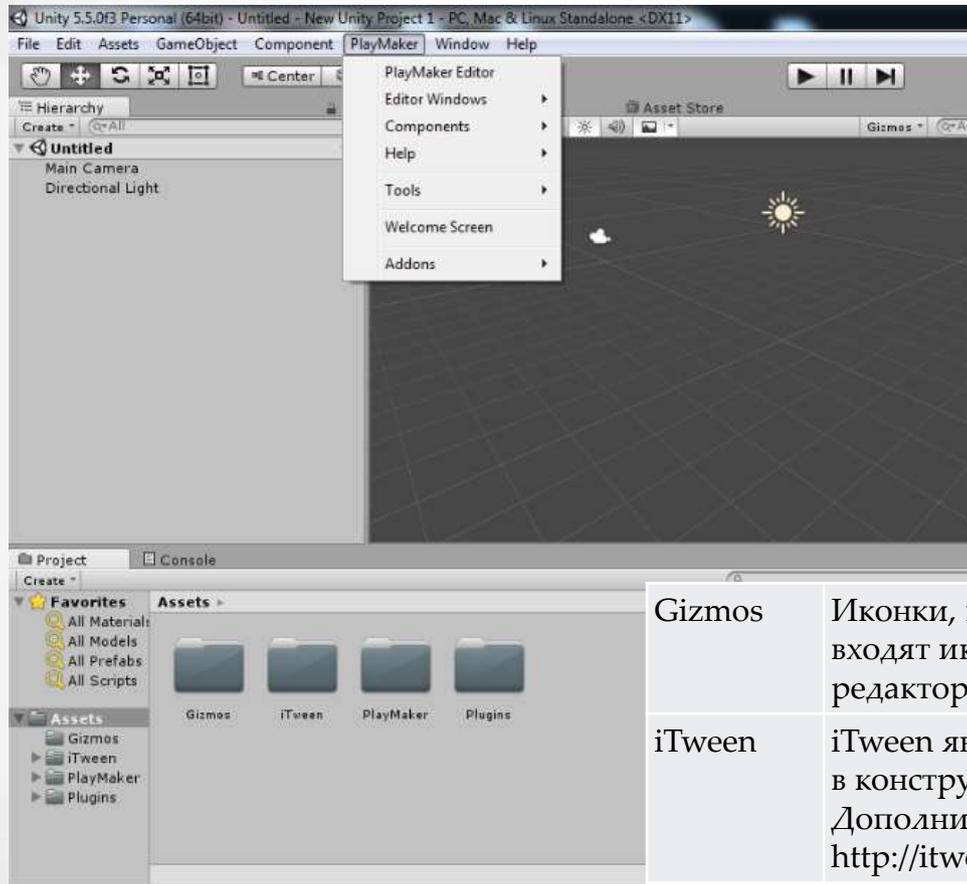
- В этом окне выводится сообщение о возможности продолжения установки. Проект просканирован. Конфликтов нет-продолжаем установку. Закрываем окно “Update Check” и снова выбираем «Install Playmaker».



Нам предлагают сделать Backup проекта перед тем как продолжить установку.

- Нажимаем “I Made a Backup. Go Ahead!”
- Затем снова «Import» .
- Нажимая “I Made a Backup” еще раз мы соглашаемся с изменения в проекте, которые вносит Play Maker.

# Структура проекта



В результате установки должен появиться заголовок “Play Maker” в главном меню. В папку “Assets” добавлены новые папки. Папка “PlayMaker” является основной, если мы ее удалим, мы не сможем использовать PlayMaker.

## Структура папок пакета PlayMaker

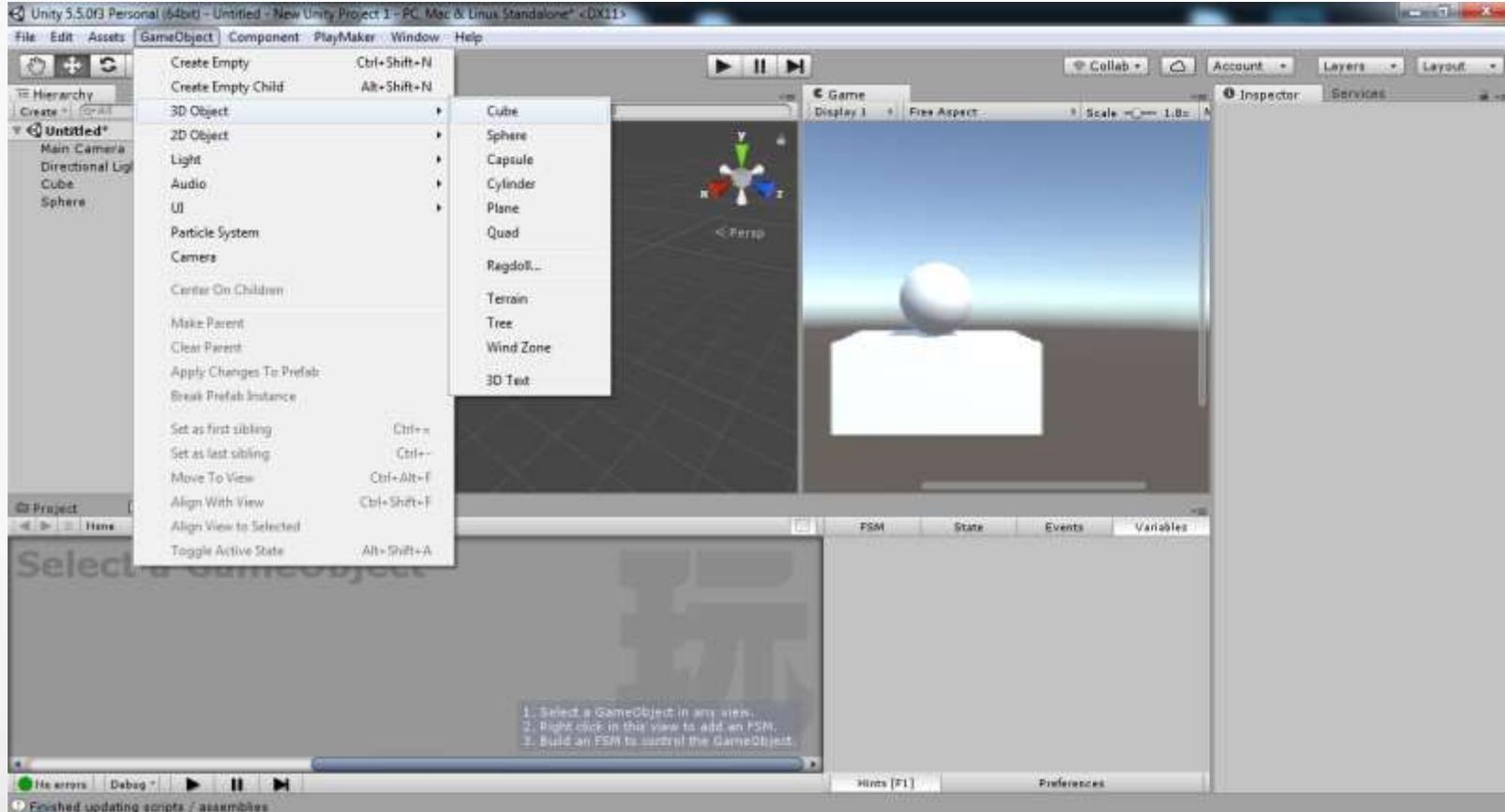
Gizmos	Иконки, которые появятся в редакторе Unity при необходимости. Сюда входят иконки, которые будут отображаться непосредственно в редакторе сцен и в «Inspector Pane»
iTween	iTween является расширением для редактора Unity, который добавляет в конструкции анимации использование интерполяции. Дополнительная информация доступна по адресу: <a href="http://itween.pixelplacement.com/index.php">http://itween.pixelplacement.com/index.php</a>
Plugins	Эта папка содержит плагины, которые созданы или будут созданы для PlayMaker. Например плагины для развертывания WebGL и Windows Phone 8.
PlayMaker	Эта папка содержит необходимые сценарии C#, которые являются основой PlayMaker.

# Интерфейс

Преобразуем интерфейс проекта Unity для удобства работы с Play Maker

- Откроем “Play Maker Editor” и перетащим в панель “Project”
- Окно “Game” поместим рядом со сценой, держа мышью за заголовок.

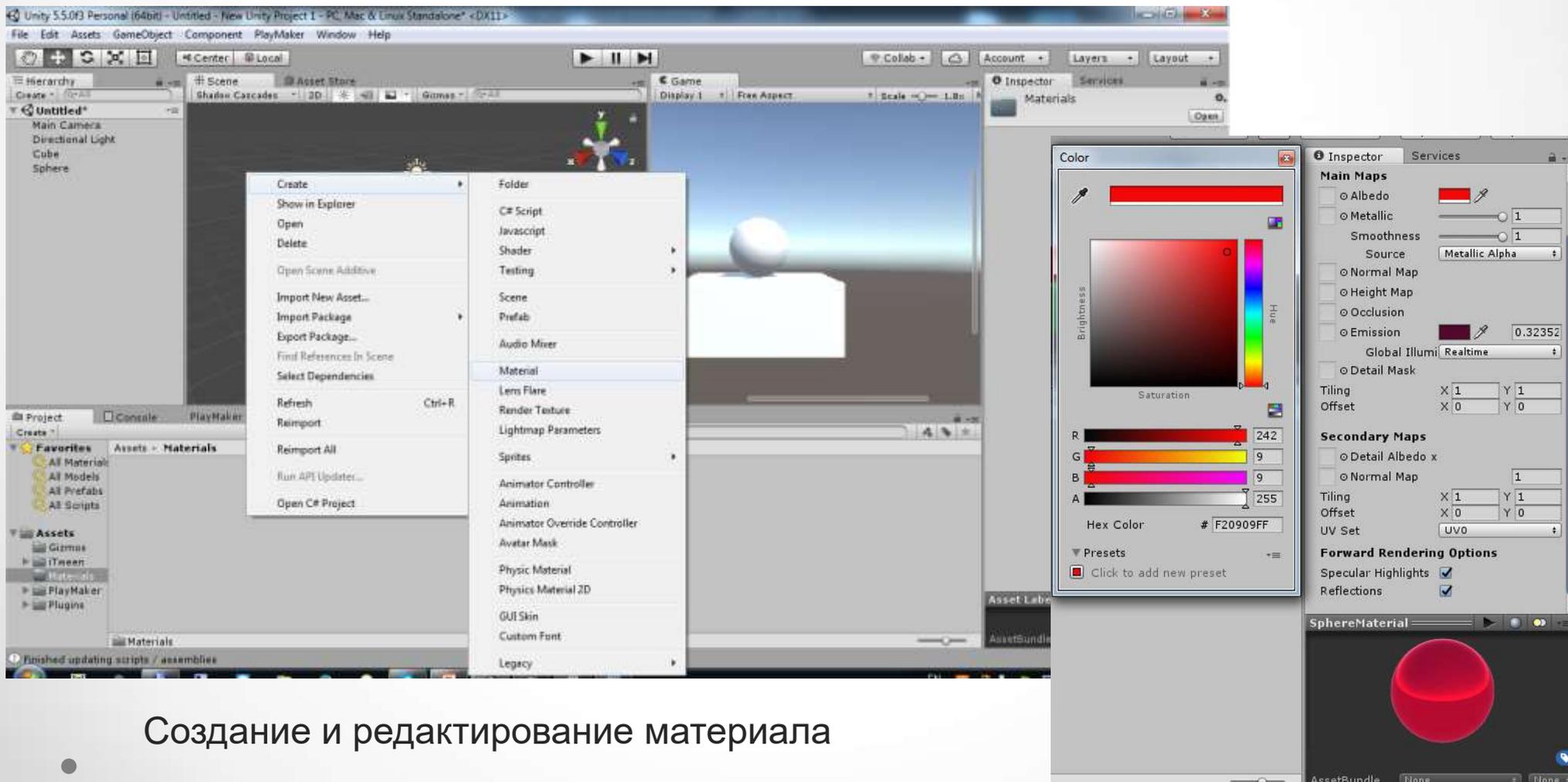
Выберем объекты “Sphere” и “Cube” и разместим в сцене, выбрав их в меню “Game Object ” -- “3D Object”



# Создание материала

Для того, чтобы изменить цвет объекту ему нужно назначить материал

- Создадим папку "Materials" в "Assets"
- В создадим два новых материала для блока и сферы. Можно использовать контекстное меню находясь в папке, выбирая "Create" -- "Material". Затем в "Inspector" назначим материалу цвет

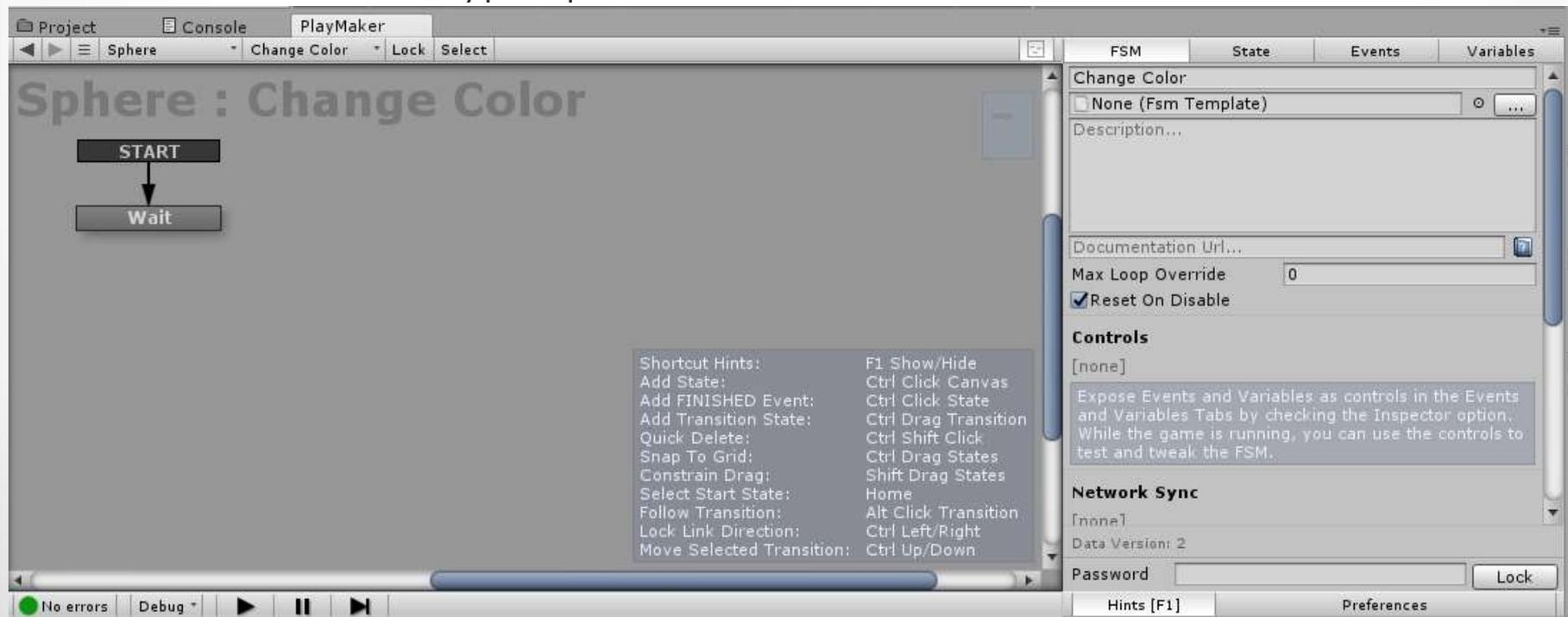


Создание и редактирование материала

# Play Maker Editor

- Выделяем объект “Sphere”
- Переходим в редактор PlayMaker.
- В контекстном меню выбираем “Add FSM”
- Справа во вкладке FSM меняем название на “Change Color”
- В контекстном меню выбираем “Add State”
- Во вкладке State меняем название на “Waiting”

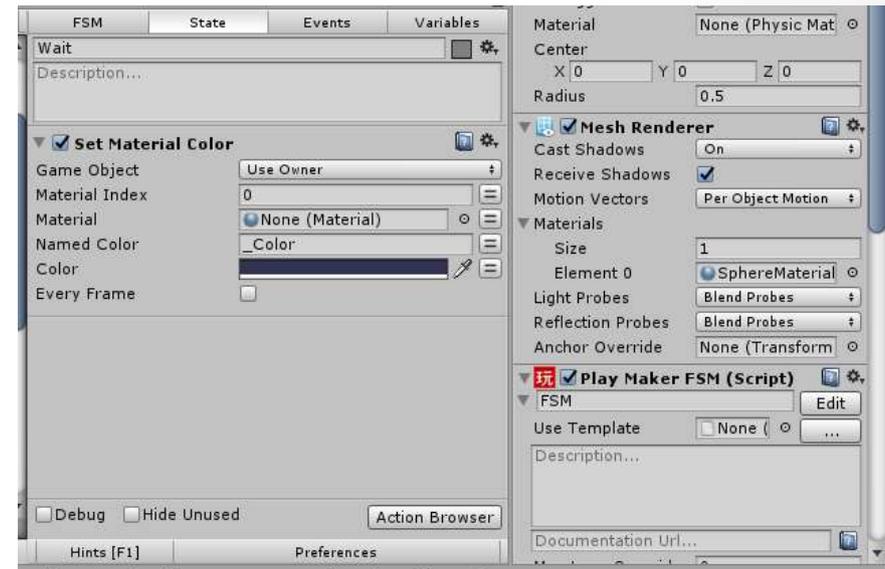
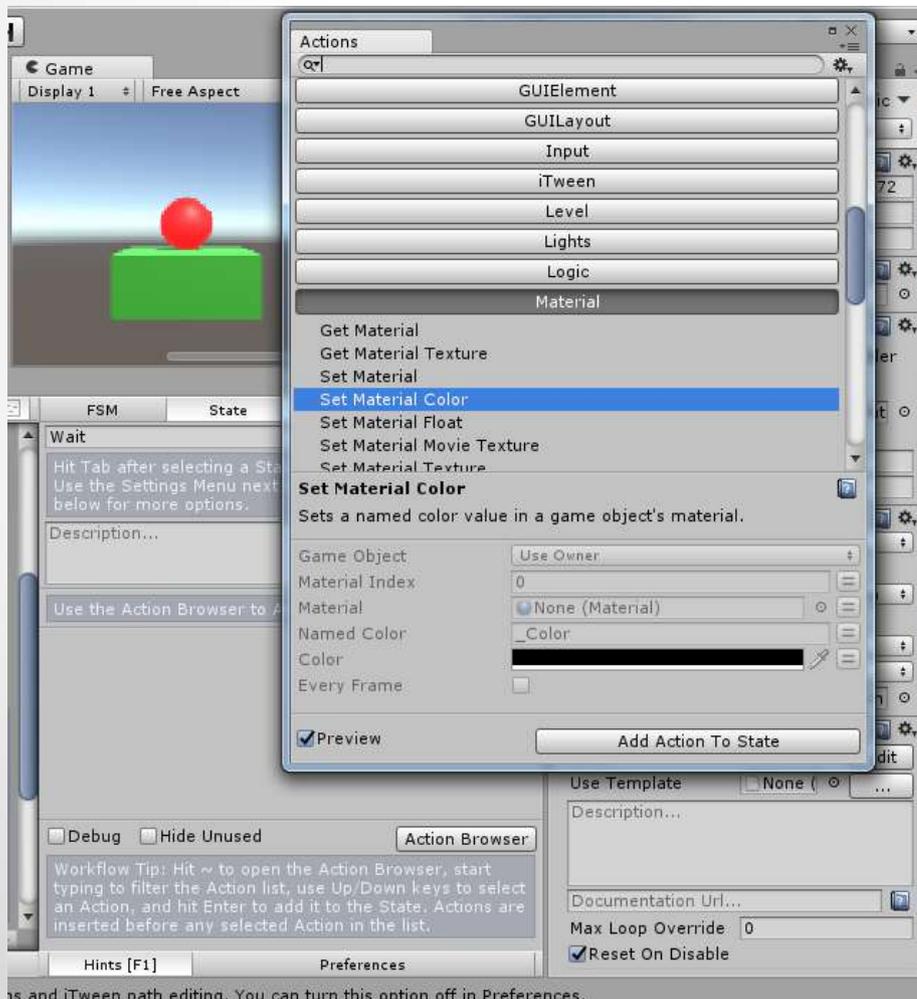
Мы создали первую схему FSM для сферы и ее первое состояние Ожидание действий курсора мыши.



# Создание действий

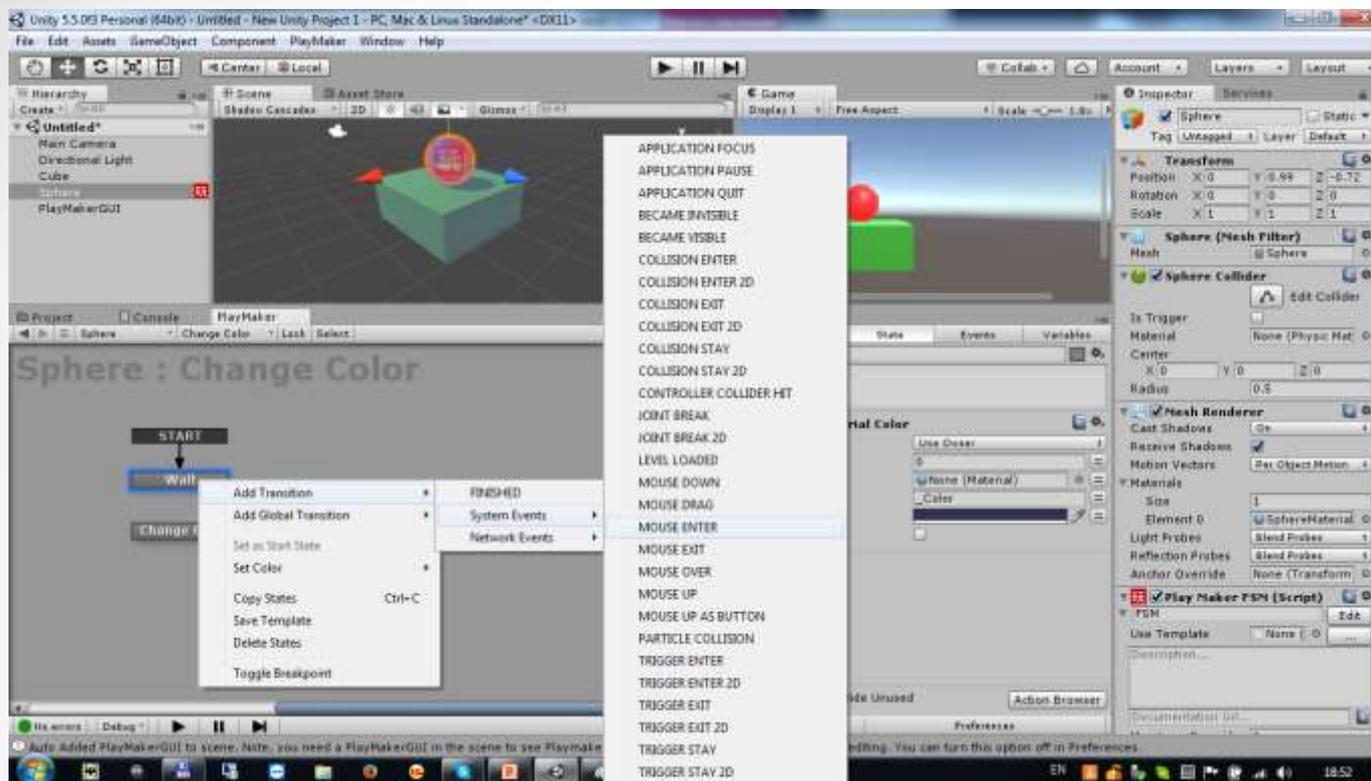
Мы должны установить цвет по умолчанию, к которому мы будем возвращаться, когда курсор будет уходить. Для этого мы создадим наше первое действие:

- Нажимаем кнопку “Action Browser”.
- В появившемся списке выбираем “Material” – “Set Material Color”
- Нажимаем “Add Action To State”



- Game Object – Use Owner, т.е. наша сфера.
- Mater Index – 0. Номер материала объекта. У нас материал один. Нумерация с “0”
- Material оставляем “None”. Если выбрать материал, то измениться цвет материала.
- Затем с помощью пипетки копируем цвет сферы.

# События мыши



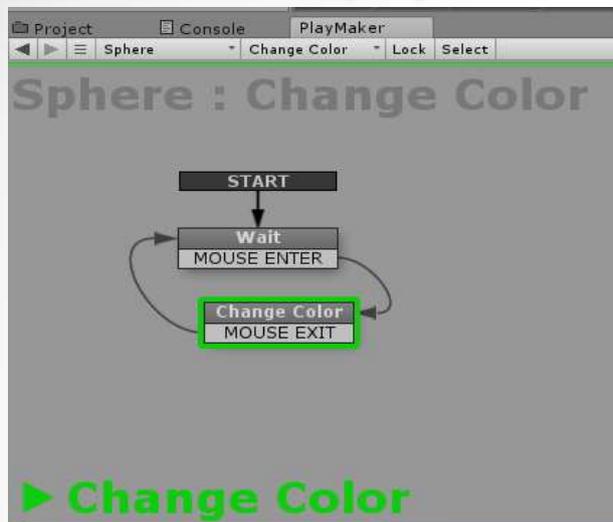
Создаем еще одно состояние “ChangeSphereColor”. Назначаем в нем цвет сферы отличный от цвета по умолчанию.

Выбираем состояние “Waiting” в контекстном меню “Add Transition” – “System Events” – “MOUSE OVER” или “MOUSE ENTER”.

Список событий мыши по умолчанию, созданных Unity

MOUSE DOWN	Левая кнопка мыши была нажата над объектом-владельцем.
MOUSE DRAG	Игрок нажал на объект-владелец и все еще удерживает кнопку при перемещении курсора.
MOUSE ENTER	Курсор мыши попал в область коллайдера владельца.
MOUSE EXIT	Курсор мыши покинул область коллайдера владельца.
MOUSE OVER	Вызывается каждый кадр, когда указатель мыши над коллайдером владельца
MOUSE UP	Игрок отпустил кнопку мыши.

# Создание FSM в Unity



Аналогично для состояния “ChangeSphere Color”. Добавим “MOUSE EXIT”.

Соединяем стрелками переходов состояния, как перетаскивая стрелку от одного состояния к другому.

Запустим игру. Сфера меняет цвет при наведении курсора. Понаблюдаем за переходами между состояниями.

```

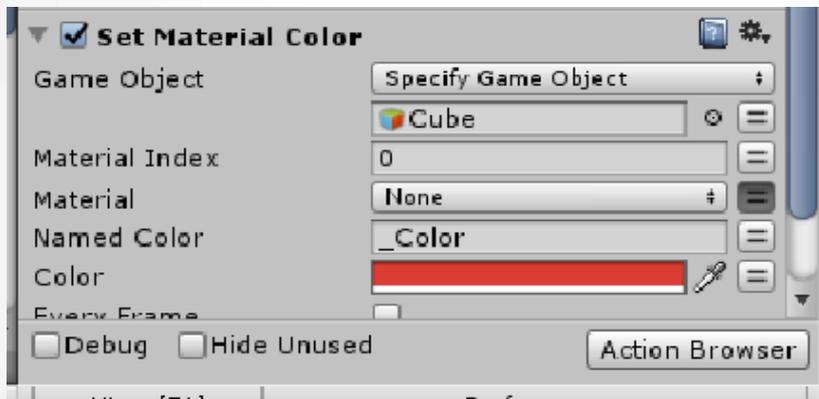
    graph TD
      START[START] --> Waiting[Waiting  
MOUSE OVER]
      Waiting --> ChangeCubeColor[ChangeCubeColor  
FINISHED]
      ChangeCubeColor --> ChangeSphereColor[ChangeSphereColor  
MOUSE EXIT  
MOUSE DOWN]
      ChangeSphereColor --> Waiting
  
```

Теперь заставим блок менять цвет. Для состояния “ChangeSphereColor”. Добавим “MOUSE DOWN”. Далее добавим состояние “ChangeCubeColor”. Для него с помощью контекстного меню выберем “Add Transition” – “FINISHED”

# Создание FSM в Unity

“FINISHED” – это специфическое действие в Play Maker, означающее, что все действия завершены и можем переходить к следующему состоянию. Оно может быть использовано как заглушка - нет событий или они пока не определены для перехода в следующее состояние.

В новом состоянии снова определим “Set Material Color”. Назначим новый цвет для блока. При этом свойство Game Object меняем на “Specify Game Object” и перетаскиваем Cube в появившееся поле.



Правильно соединим все стрелки переходов и запустим игру. Первая игра с использованием FSM готова. Перейдем к более интересному сценарию игры.

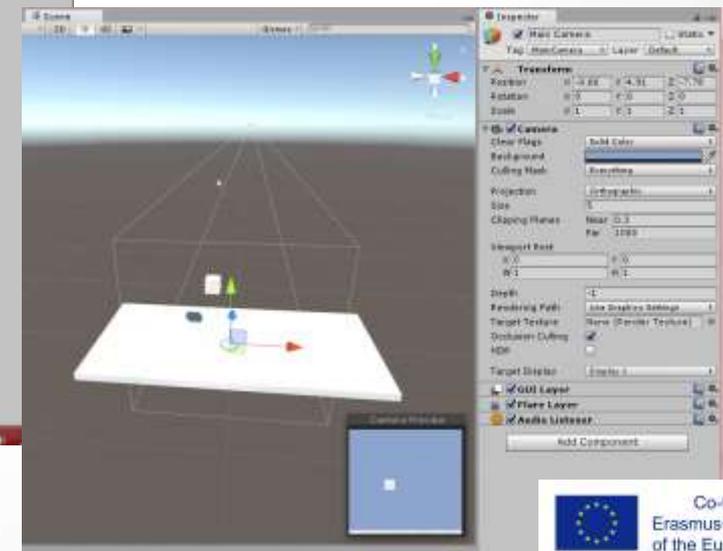
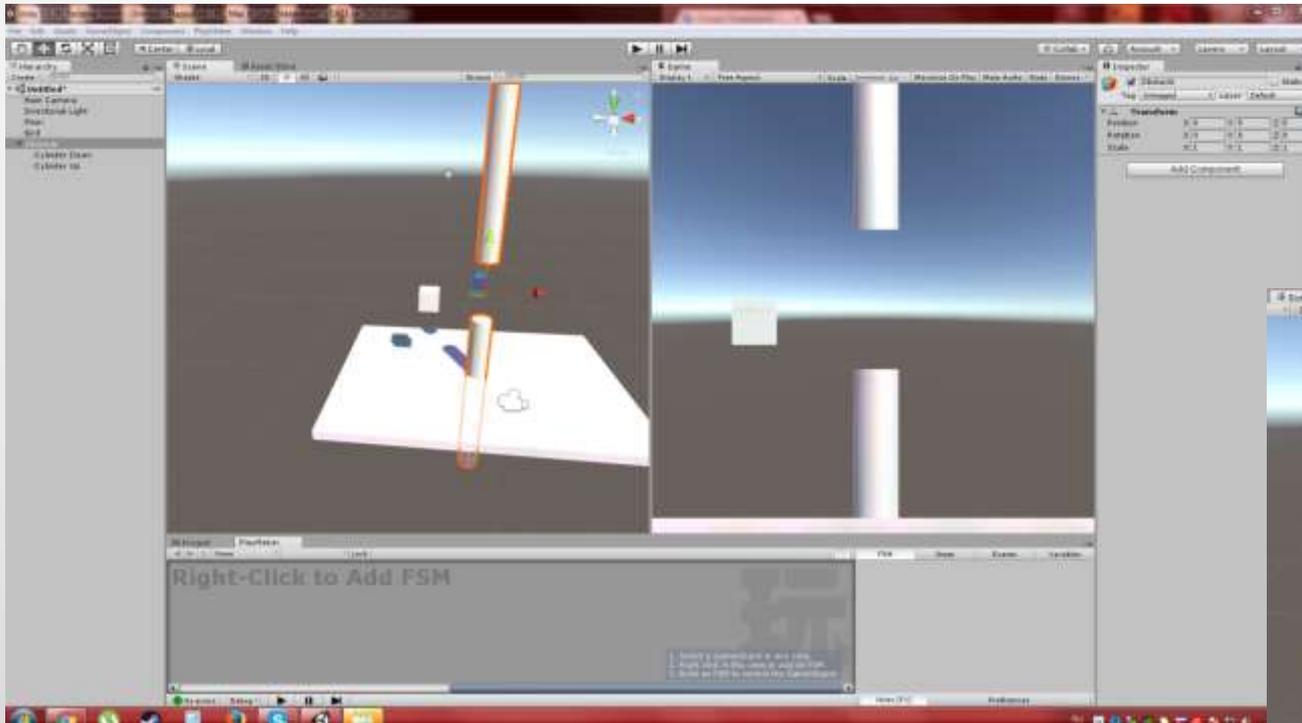
# Прототип игры Flappy Bird

- Создадим новую игровую сцену.
- Поместим на нее Cube. Определим размеры 10; 0,5; 10. Position по умолчанию. Изменим имя на Floor.
- Затем еще один Cube, Поместив его над Floor немного влево от центра. Дадим ему имя Bird. Position по Z должна быть "0"
- Затем 2 одинаковых Cylinder, разместив их как показано на рисунке. Position и Scale по Z , X должны быть "0". Дадим им имена CylinderUp и CylinderDown
- Создаем пустой объект Game Object. Разместим его между цилиндрами. Position и Scale по Z , X -- "0"

## Настройки камеры

Камеру разместим перед игровым полем.

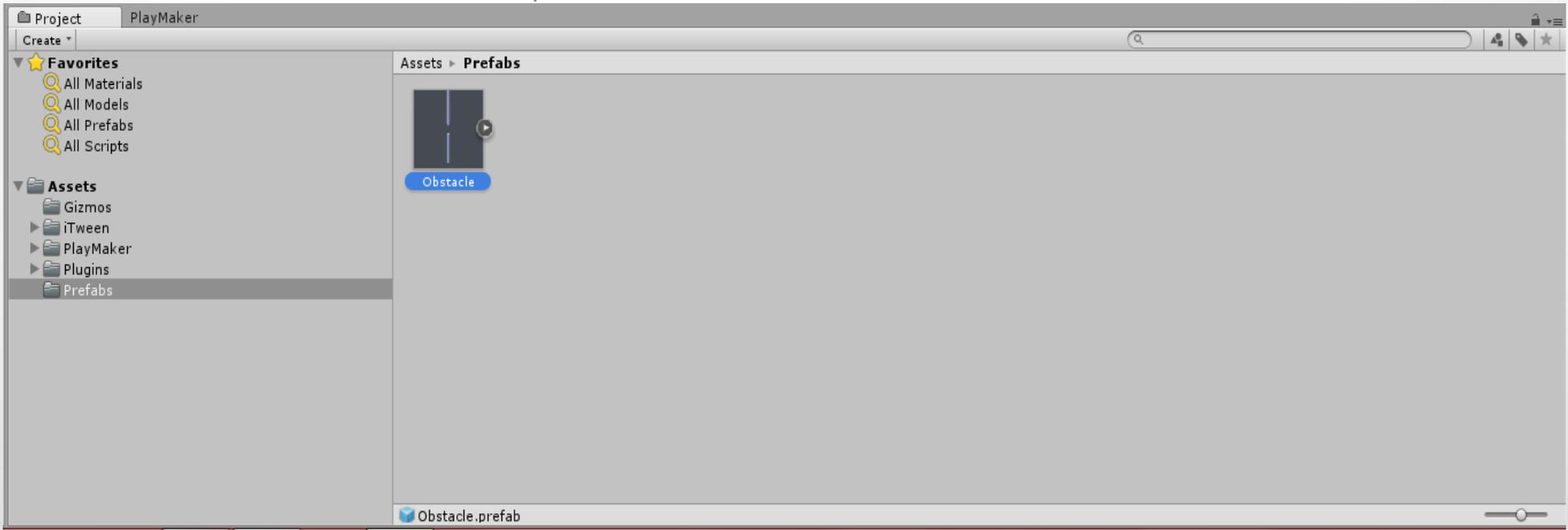
Определим свойства  
Projection = Orthographic;  
Size=5;



# Шаблоны Prefabs

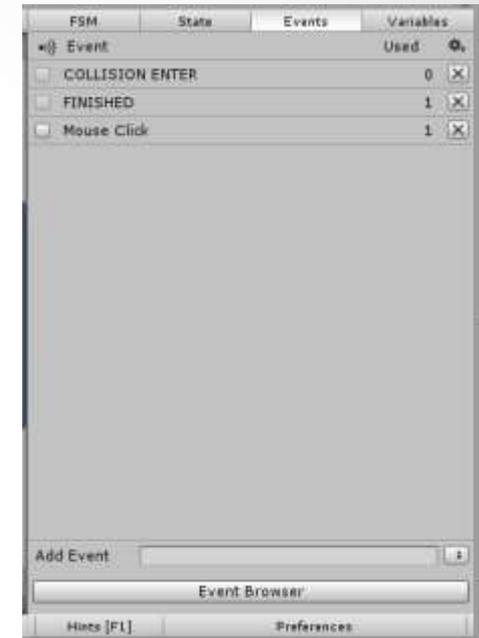


- Назовем наш Game Object – Obstacle (препятствие)
- Мышкой перетащим в него цилиндры, как показано на картинке. Obstacle теперь родительский элемент для цилиндров, объединяет их в один .
- Создаем в Assets папку Prefabs с помощью контекстного меню
- В папку Prefabs перетаскиваем объект Obstacle. Он стал шаблоном и мы можем удалить его из игровой сцены



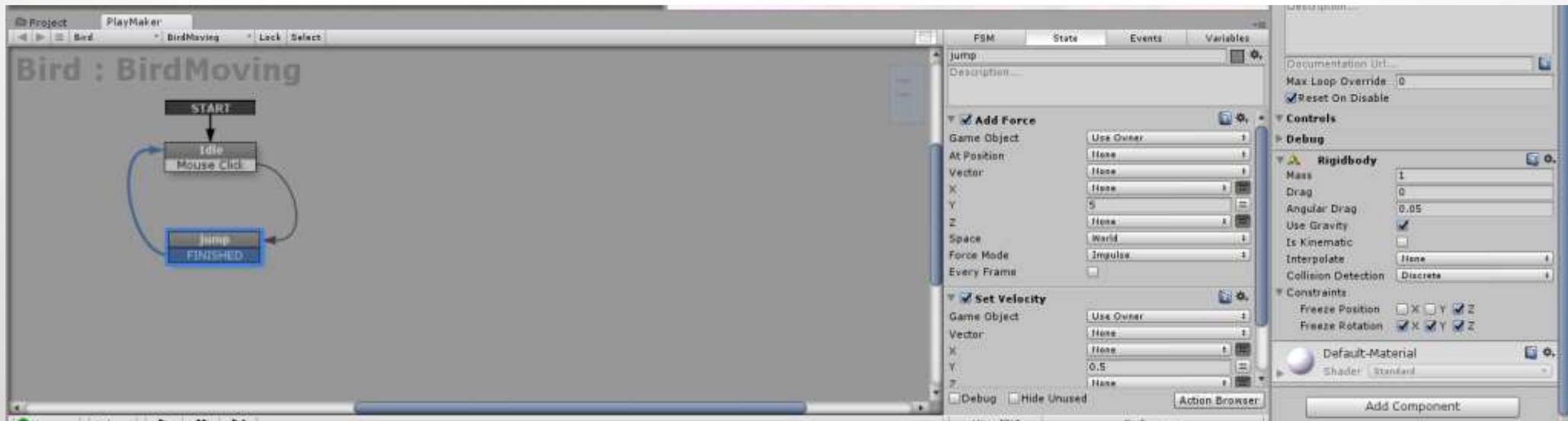
# Flappy Bird. FSM

- Выбираем объект Bird.
- Зададим нашему объекту физические свойства, а именно гравитацию. Нажимаем кнопку “Add Component” – Physics 3D – Rigidbody. Use Gravity должно быть отмечено.
- Создаем FSM – Bird Moving
- Добавим два состояния – idle (Бездействие) и jump
- Выделяем idle. Добавляем новое действие. Нажимаем Action Browser – Input – GetMouse Button Down.
- Во вкладке Events создаем в Add Event новое событие – Mouse Click
- В Get Mouse Button Down, в Send Event выбираем MouseClick



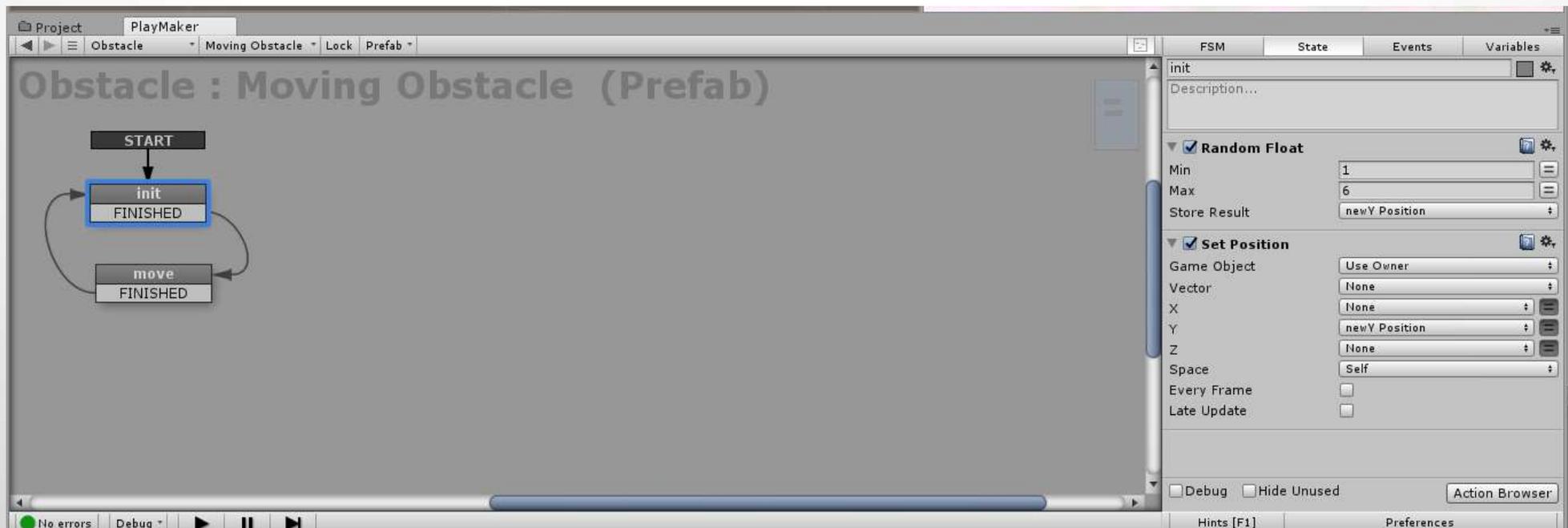
# Flappy Bird. FSM

- Для объекта Bird в Rigidbody отмечаем – Constraints – Freeze Position Z; Freeze Rotation X;Y;Z
- Аналогично выбираем действия для jump  
Physics – Add Force Y = 5  
Physics – Set Velocity Y = 0,5
- Условие перехода для jump “FINISHED”
- Добавляем стрелки переходов. Проверим работу FSM.
- Запустим игру. Bird движется вверх по нажатию левой кнопки мышки и медленно опускается, под действием силы тяжести, когда кнопка отпущена.



# Flappy Bird. FSM

- Выбираем объект Obstacle в Prefabs.
- Создаем FSM – Obstacle Moving
- Добавим два состояния : init и move
- Создаем переменную new Y Position типа float во вкладке Variables.
- Выделяем init. Добавляем новые действия.  
Math – Random Float min = 5; max = 6; Store Result = new Y Position  
Transform – Set Position Y = new Y Position
- Условие перехода “FINISHED”



# Flappy Bird. FSM

- Аналогично выбираем действие для move  
Transform -- Translate X = -2. Отмечаем Per Second, Every Frame.
- Условие перехода для move "FINISHED"
- Добавляем стрелки переходов

The screenshot shows the Unity PlayMaker FSM editor for a 'Moving Obstacle' prefab. The FSM diagram on the left illustrates a cycle: a 'START' state leads to an 'init' state (with a 'FINISHED' condition), which then leads to a 'move' state (with a 'FINISHED' condition), which loops back to 'init'. The 'move' state is configured with a 'Translate' action, set to move X by -2 units, every frame.

The right panel shows the configuration for the 'move' state:

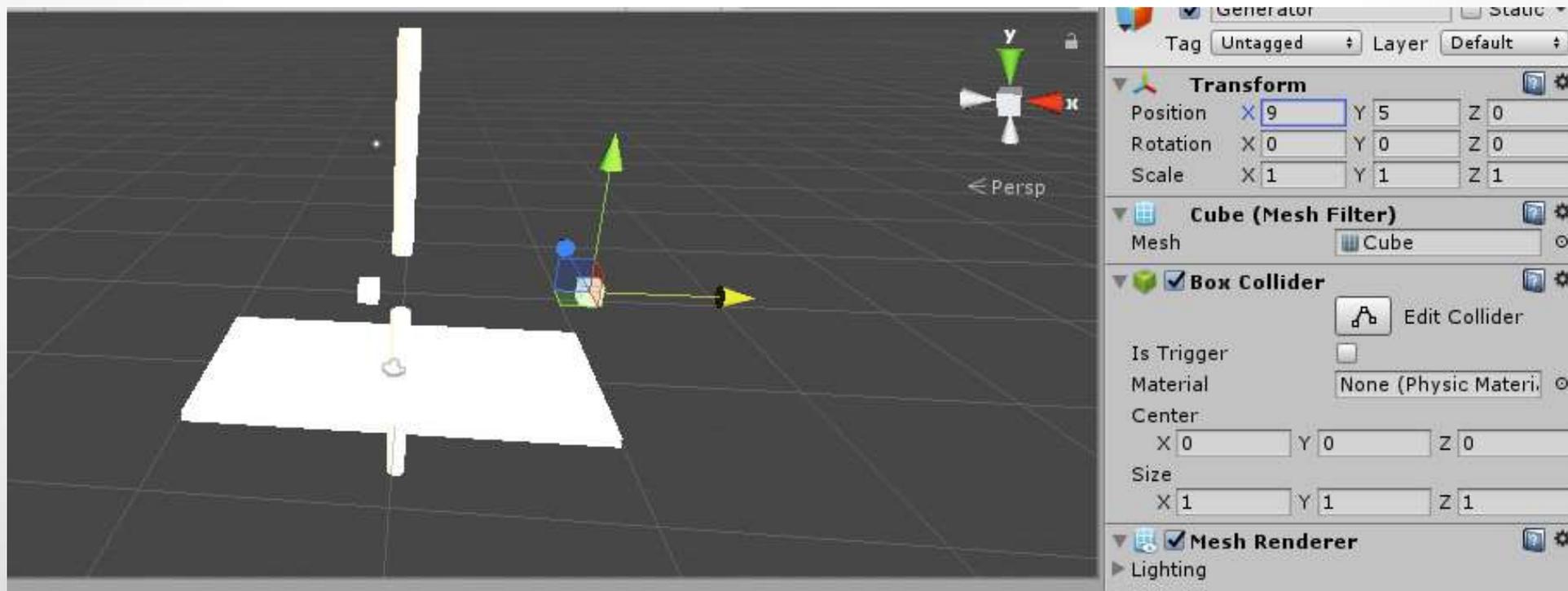
- FSM** tab selected.
- move** state selected.
- Translate** action checked.
- Game Object**: Use Owner
- Vector**: X: -2, Y: None, Z: None
- Space**: World
- Per Second**:
- Every Frame**:
- Late Update**:
- Fixed Update**:

# FSM и Spawn Point

- Когда нужно создать FSM, не привязываясь к объекту, можно создать пустой объект, он играет роль точки возникновения коллекции объектов.
- Создадим новый объект Cube. Назовем Generator. Разместим как показано на рисунке. Этот объект будет точкой генерации препятствий.

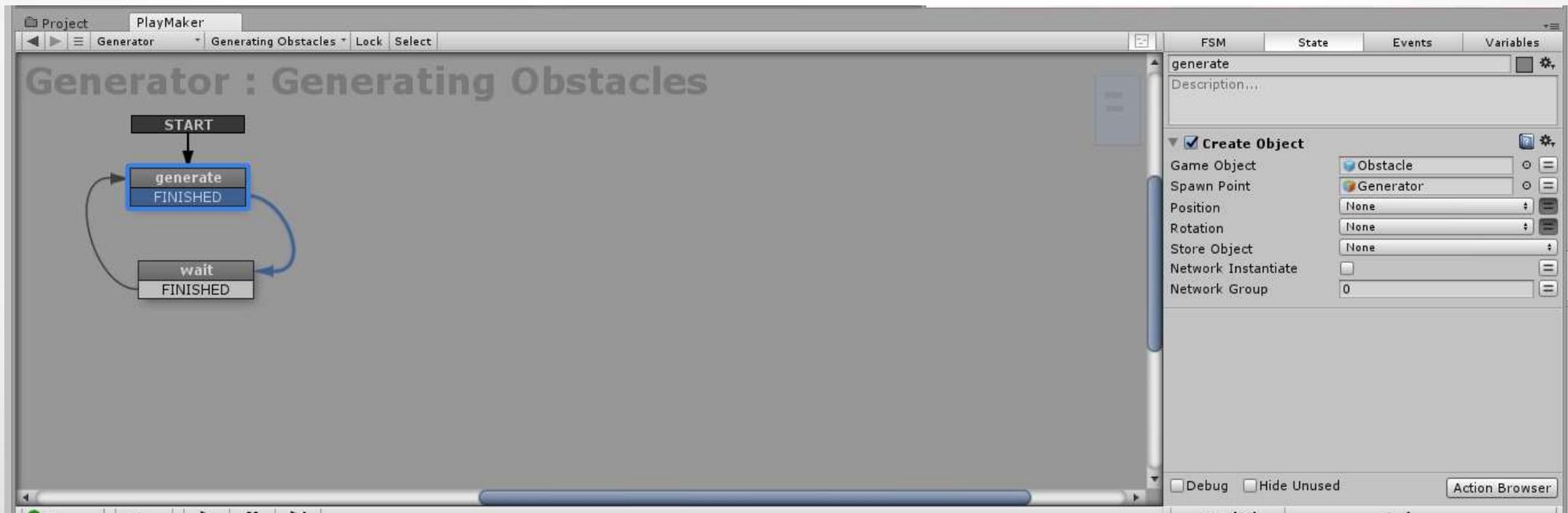
Position Z = 0; Y = 5; X = 9

- Создаем для объекта Generator новую FSM – Generating Obstacles



# Прототип игры Flappy Bird

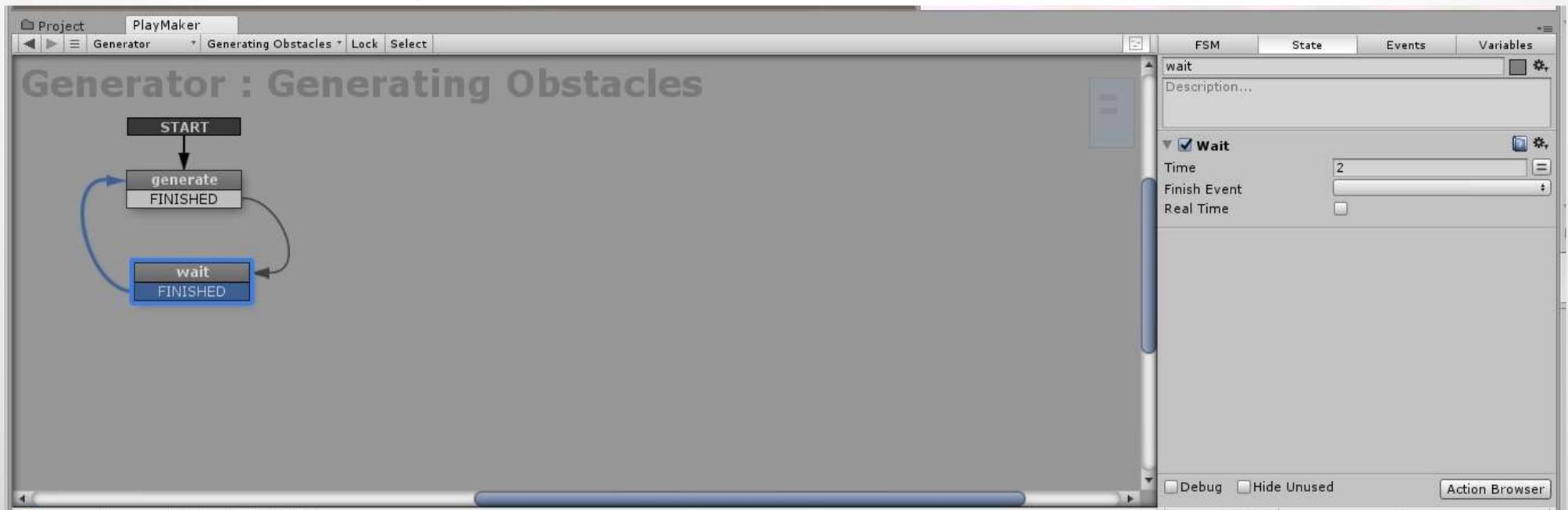
- Добавим два состояния – generate и wait
- Выделяем generate. Добавляем новое действие. Нажимаем Game Object – Create Object. В Game Object перетаскиваем Obstacle В Spawn Point – Generator
- Условие перехода “FINISHED”



# Прототип игры Flappy Bird

- Аналогично выбираем действие для wait  
Time - Wait Time = 2.
- Условие перехода "FINISHED"
- Добавляем стрелки переходов
- Объект можно сделать невидимым, убрав выделение с Mesh Renderer в окне свойств объекта

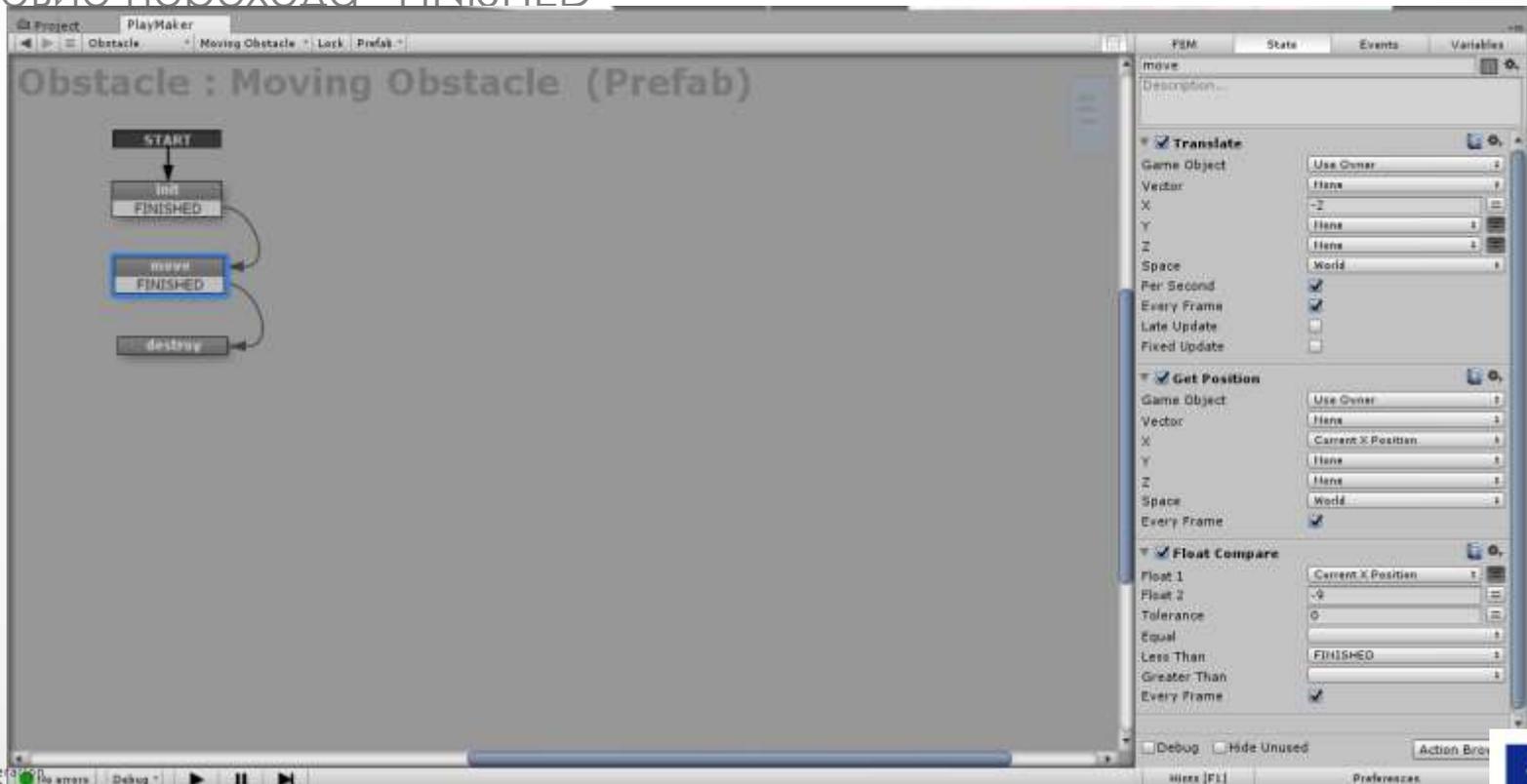
Запустим игру. Препятствия передвигаются по полю, появляясь в начале, но не исчезают в конце него.



# Прототип игры Flappy Bird

Доработаем FSM для препятствий Moving Obstacle.

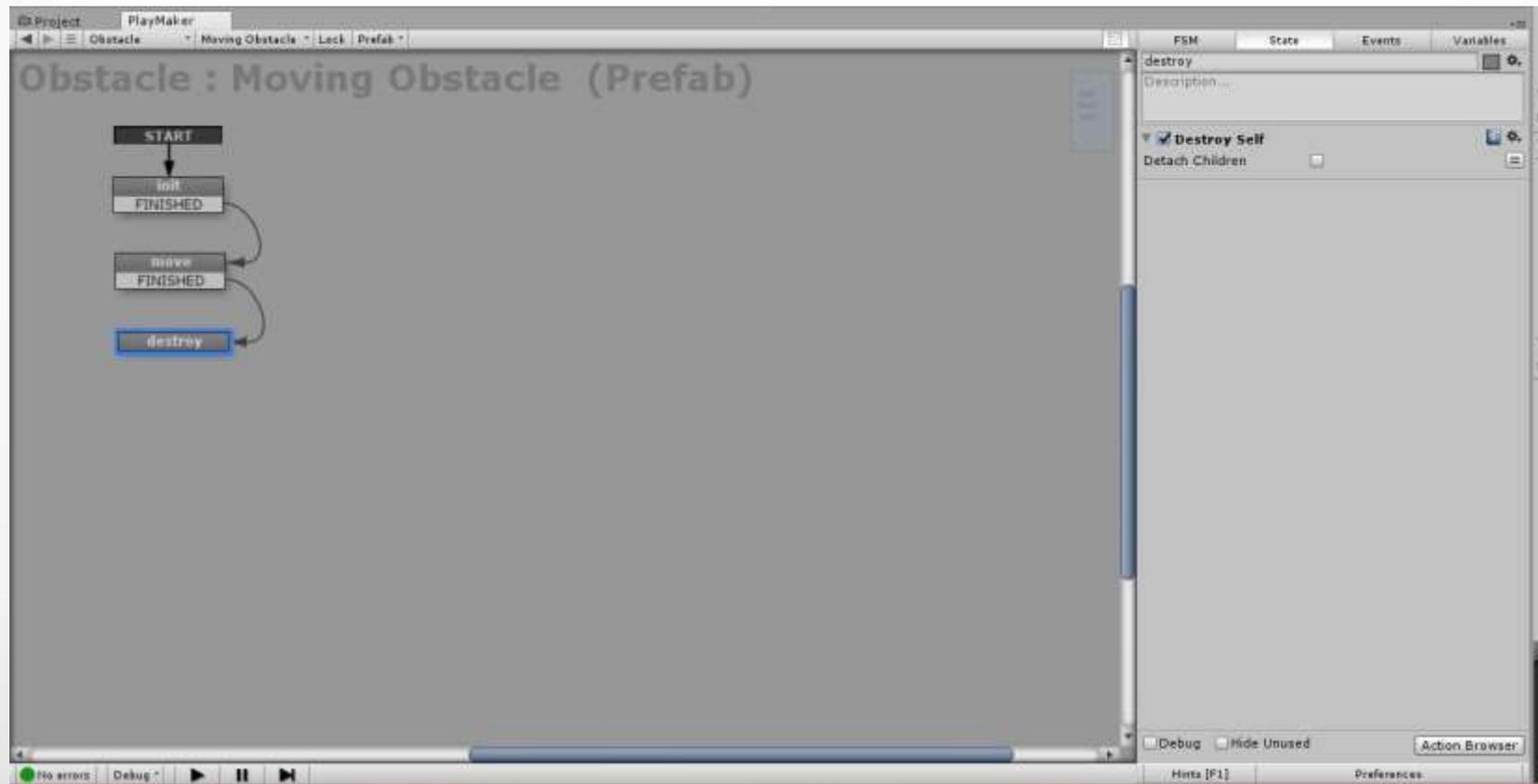
- Создадим переменную Current X Position
- В состояние move добавим Transform - Get Position. X = Current X Position  
Logic -- Float Compare. Float1 = Current X Position; Float2 = -9; Less Than = FINISHED
- Условие перехода "FINISHED"



# Прототип игры Flappy Bird

- Добавляем состояние destroy. Добавим действие Game Object -- Destroy Self
- Условия перехода нет. Это конечное состояние
- Добавляем стрелки переходов

Запустим игру. Препятствия передвигаются по полю, появляясь в начале и исчезая в конце.

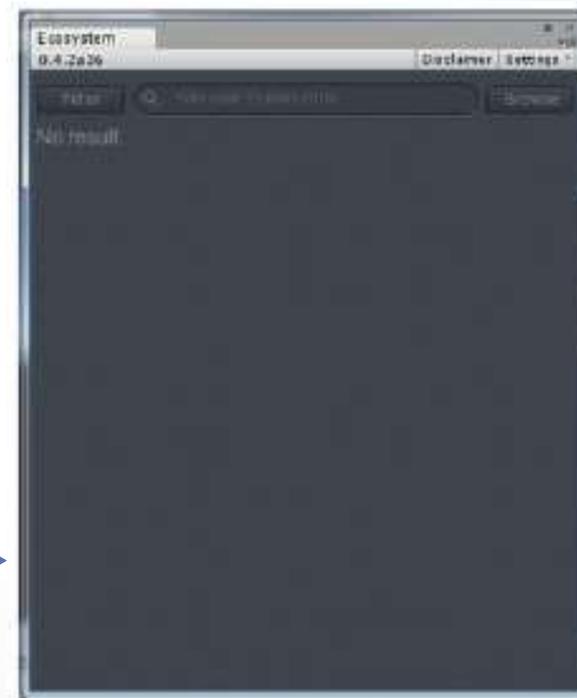


# Пакет Ecosystem

- Ecosystem - пакет и браузер действий для использования в PlayMaker, который мы загружать в наш текущий проект, когда нам понадобятся какие-либо другие действия, которые не включены в настройки по умолчанию.
- Загрузка пакета Ecosystem находится по адресу <https://hutonggames.fogbugz.com/?W1181> Его можно загрузить, нажав на ссылку «EcosystemBrowser Package». После того, как файл пакета был загружен, перетащите этот пакет в область проекта Unity, как показано на рисунке. Как только пакет перетащен на панель проекта, Unity распакует пакет и далее предложит установить так же, как это было с Playmaker



Ecosystem  
браузер и  
интерфейс



# Полезные ресурсы

- На веб-сайте разработчика можно ознакомиться с форумами (<http://hutonggames.com/playmakerforum/index.php>).
- Эти форумы - замечательный ресурс для поиска ответов на вопросы, которые могут возникнуть по мере продвижения в PlayMaker.
- Загрузка пакета Ecosystem находится по адресу
- Уроки Unity PlayMaker  
<https://www.youtube.com/channel/UCII8FyIFgevUyLfd-jhhskA>