

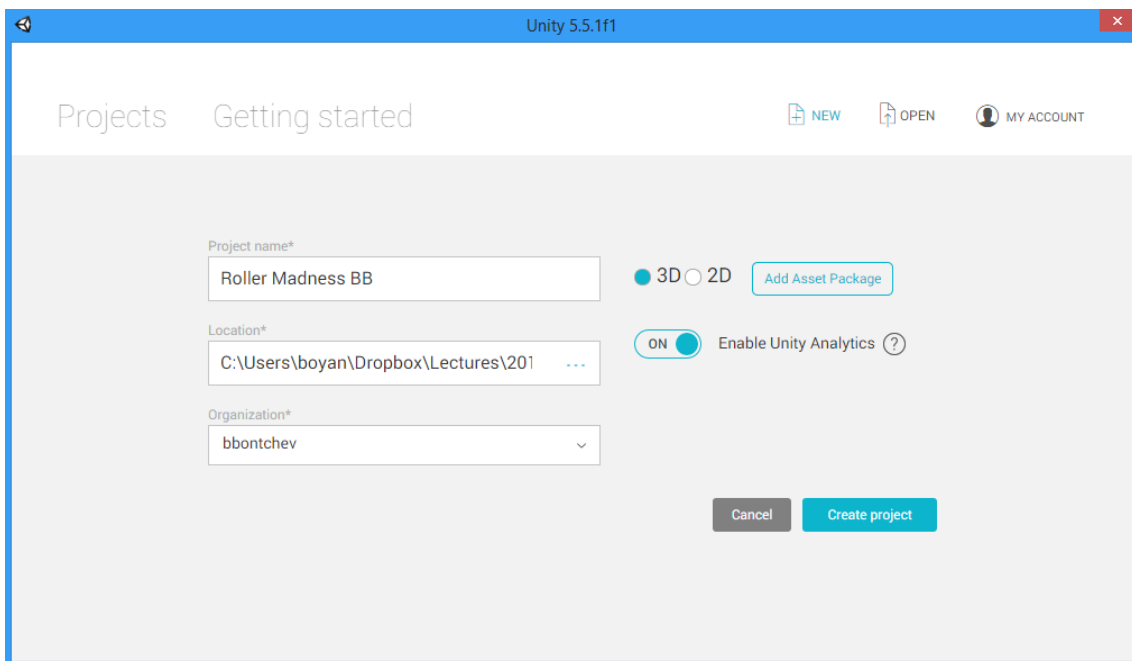
## Лабораторно упражнение N2

### Инструкции към игра "Roller Madness"

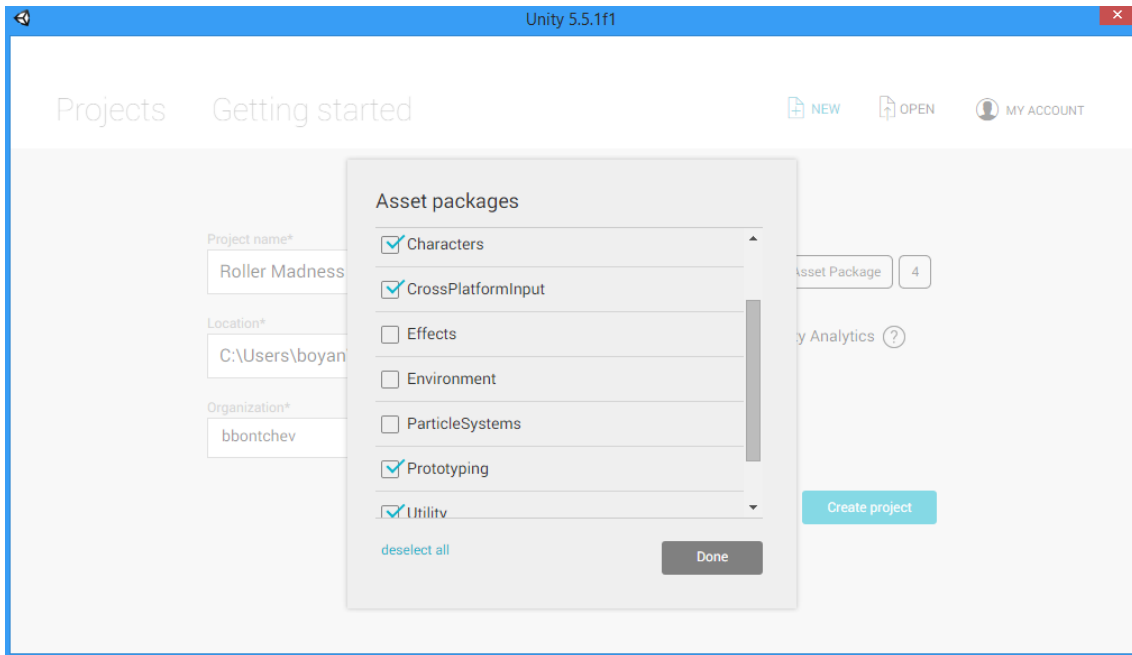
(с материали на Браян Уин, Мичигански държавен университет)

#### А. Настройки и импорт

**Стъпка 1** – Отворете Unity и създайте нов проект с име Roller Madness от тип 3D. Изберете папка за проектните файлове, след което добавете стандартни вградени Unity асети (активи) с цел по-бърза разработка на играта (би могло да ги добавите и после) – посредством бутона Add Asset Package.

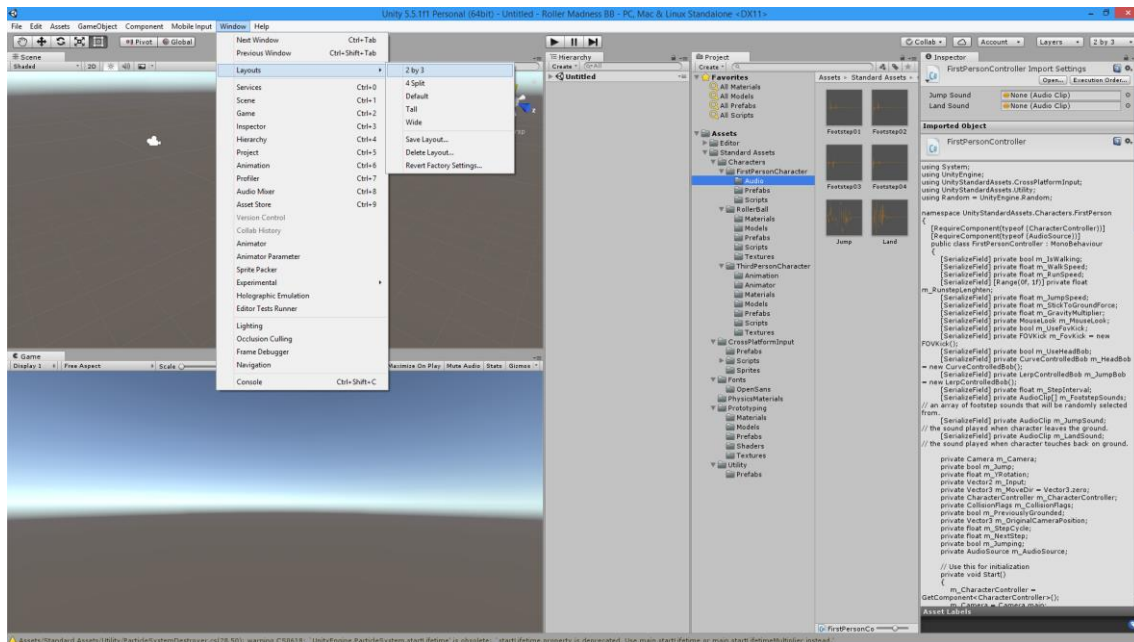


Изберете асетите Characters, CrossPlatformInput, Prototyping, и Utility.



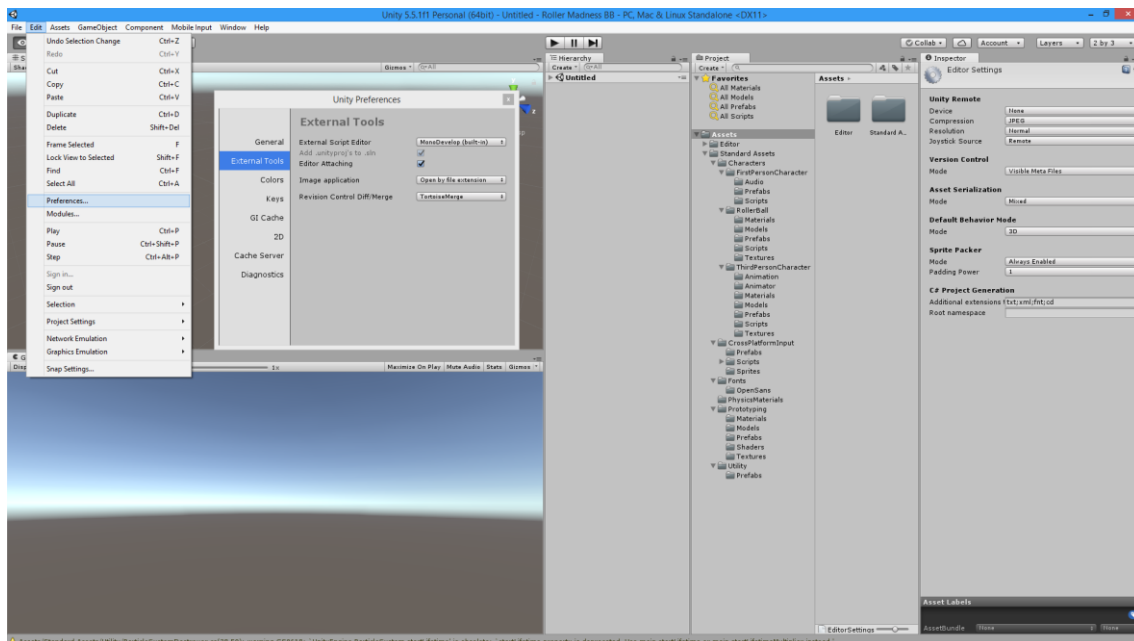
След това щракнете на бутона Done и после на Create project.

**Стъпка 2** – изберете layout 2 by 3, след което разгледайте импортираните асети във всяка папка, както е показано на фигурата по-долу.

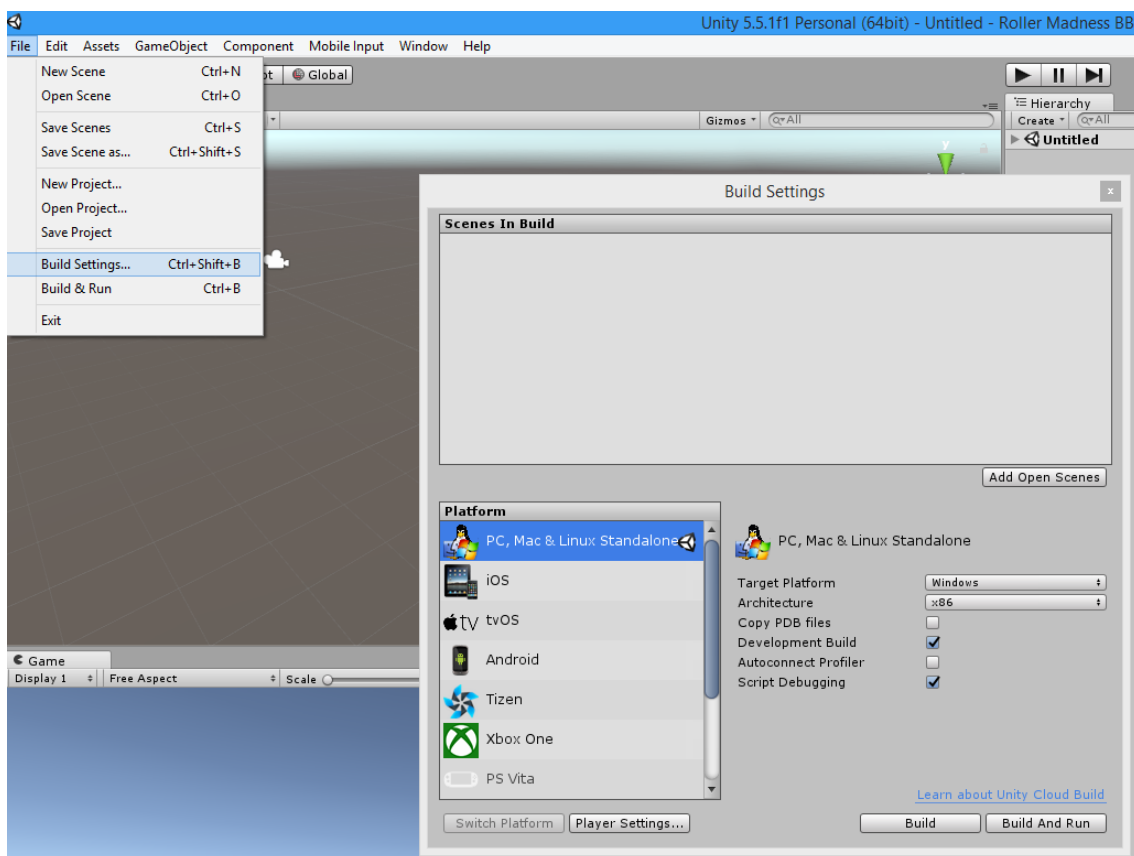


**Стъпка 3** (опционална) – ако желаете да ползвате средата MonoDevelop и не сте я избрали при инсталацията, направете следното:

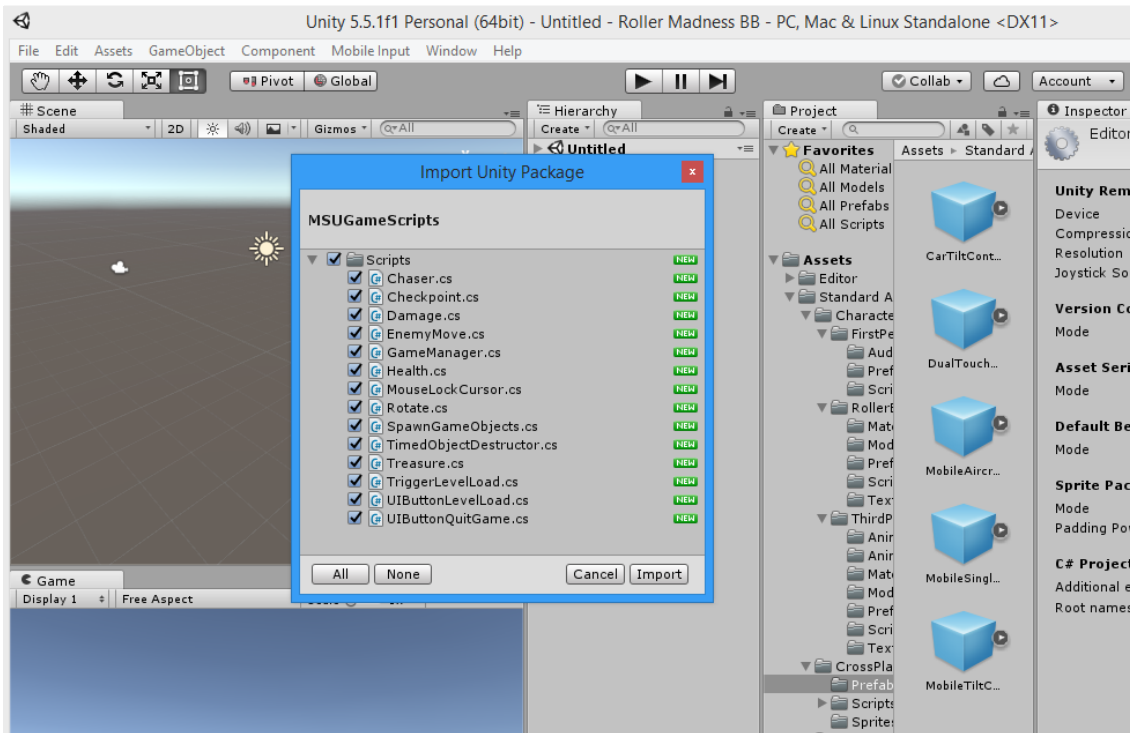
1. В меню Edit > Preferences в панела External Tools изберете за External Script Editor средата MonoDevelop.
2. След това, Unity проектът ви ще се отвори/синхронизира в/с MonoDevelop средата чрез меню Assets > Open C# Project.



3. Проверете дали опциите Development Build и Script Debugging са избрани в Build Settings за избраната от вас платформа (меню: File > Build Settings).



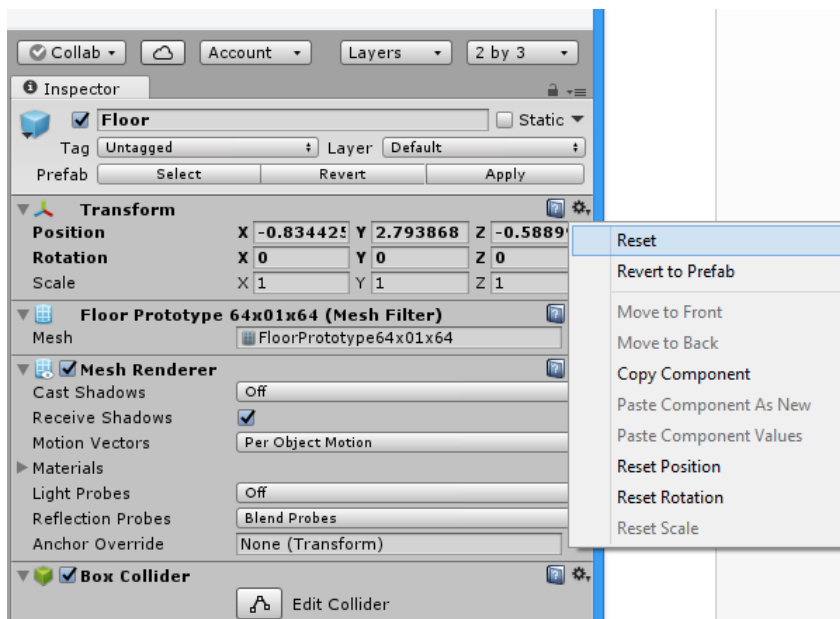
**Стъпка 4** – импортирайте пакета със скрипт асети (на Браян Уин, Мичигански държавен университет) с име MSUGameScripts.unitypackage (от архива RollerMadnessAssets.zip, наличен в сайта на курса), с влачене на файла в папка Assets в Project панела и последващо натискане на бутона Import. Проверете, че скриптовете са импортирани в папка Assets > Scripts.



**Стъпка 5** – импортирайте аудио асетите от архива RollerMadnessAssets.zip с влачене на папката Audio в Assets в Project панела. Проверете, че аудио асетите са импортирани в папка Assets > Audio. Разгледайте ги и ги прослушайте в Инспектор панела.

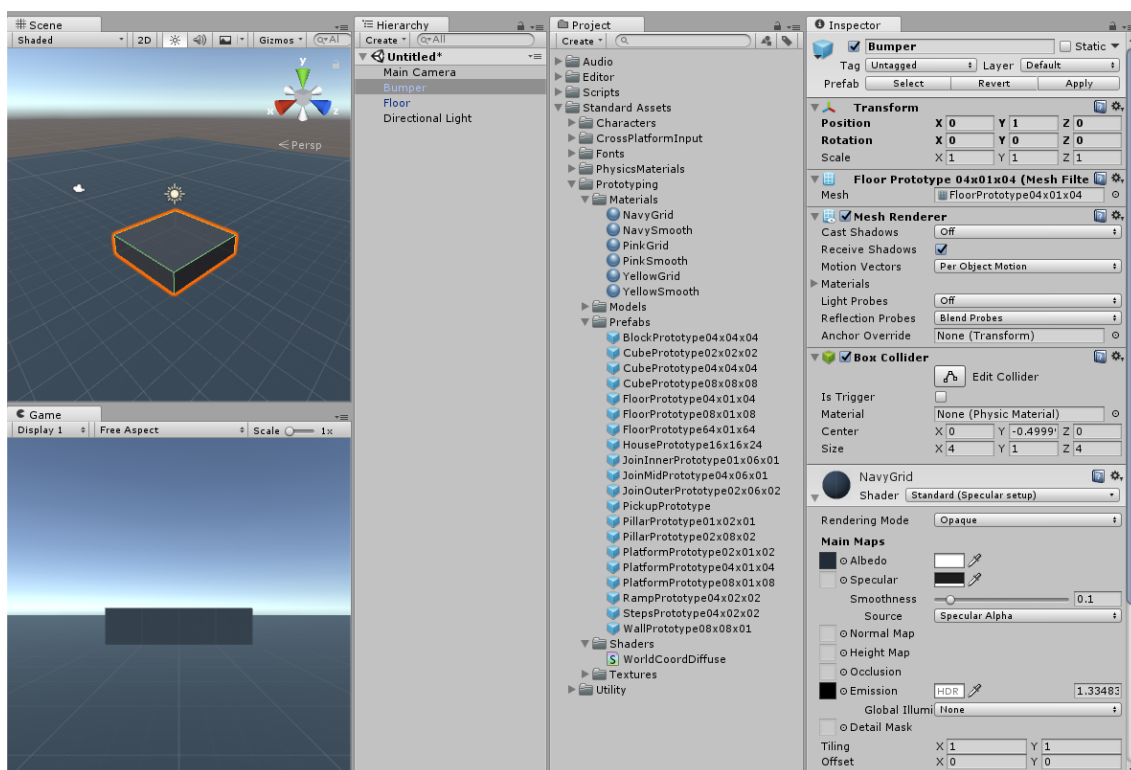
## Б. Базов дизайн на първа сцена

**Стъпка 6** – от папка Prefabs, изберете FloorPrototype64x01x64 и го завлечете в дизайн прозореца. Преименувайте го на Floor и направете нулиране на координатите му – или поединично, или чрез Reset.



**Стъпка 7** – скалирайте го (чрез Scale в Inspector) по X и по Z със стойност 0.5.

**Стъпка 8** – използвайте FloorPrototype04x01x04 по начина, описан по-горе, за да създадете препятствие в играта, като го преименувате на Bumper. Забележете, че неговият материал е също NavyGrid, което пречи различаването му от Floor.



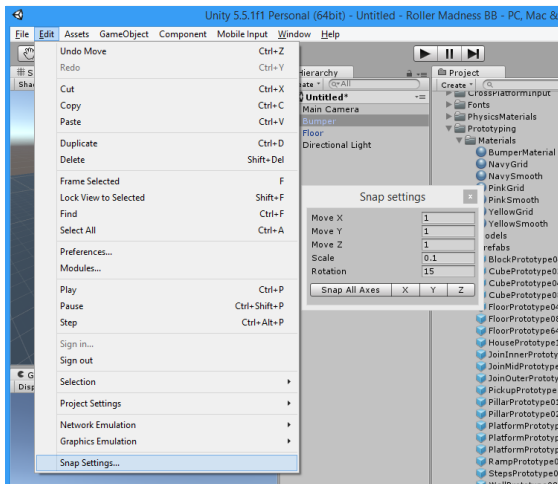
**Стъпка 9** – създайте нов материал с име BumberMaterial, като дублицирате материала NavyGrid чрез Edit > Dublicate или чрез CTRL+D и смените името на BumberMaterial. Сменете Albedo на новия материал с избрана по ваш вкус текстура от Textures (напр. SwatchOrangeAlbedo) и после приложете с влачене и пускане новия материал спрямо Bumber.

**Стъпка 10** – упражнете се, като местите по трите координати Bumber:

- с влачене на осите
- с влачене на осите + CTRL (при което местите със snar ефект).

Можете да променят snar настройките на така, както е показано на фигурата по-долу.

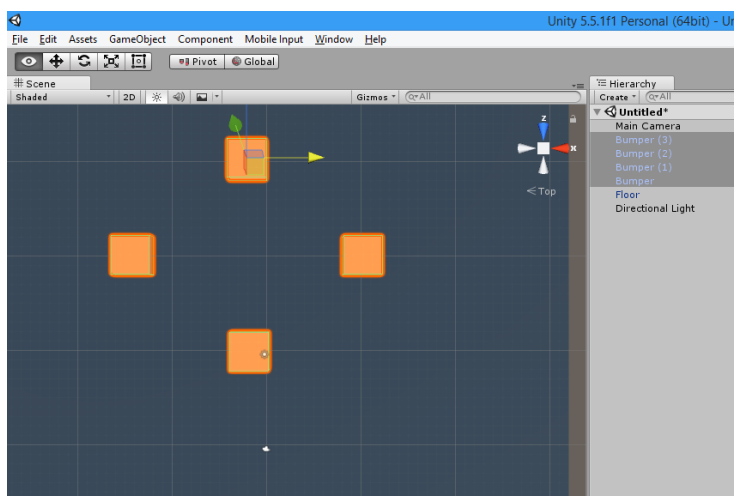
Накрая нулирайте координатите и повдигнете Bumber с 1 нагоре.



**Стъпка 11** – създайте два Prefab обекта по два различни начина:

- създайте празен Prefab обект и завлечете Bumber върху него
- завлечете Floor в йерархията от асети

**Стъпка 12** – използвайки Prefab с име Bumber, създайте три нови обекта на височина 1, като завлечете префаба на желаните места. После изравнете местоположенията им, като влачите избраната ос (X или Z) с натиснат клавиш CTRL. Сменете перспективата да е отгоре, селектирайте и четирите Bumber обекти и ги позиционирайте в центъра на пода.



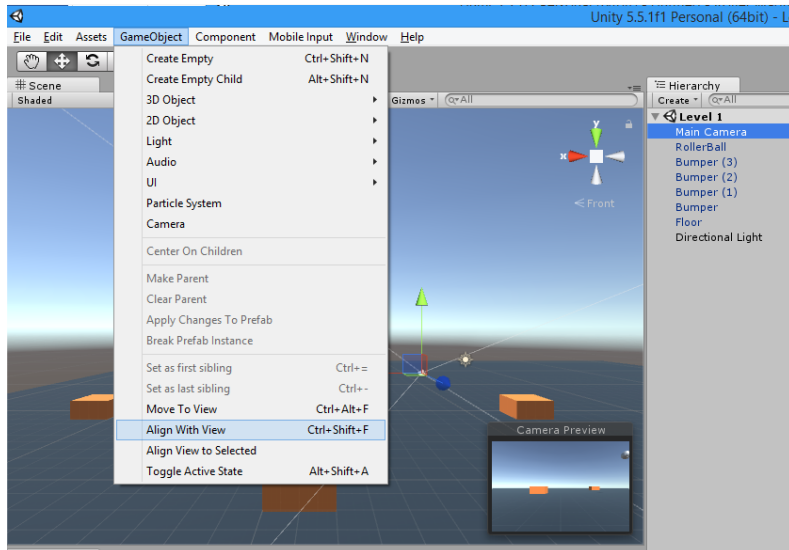
После скалирайте четирите Bumber обекти с промяна на префаба така, че по X и Z скалирането да е 0.5 от оригиналния размер.

**Стъпка 13** – Изберете RollerBall префаб за героя в играта от Standard Assets > Characters > RollerBall > PreFabs (забележете, че имате достъп до FirstPersonCharacter и ThirdPersonCharacter) и го завлечете над пода, за да създадете нов обект с име RollerBall. Позиционирайте го 5 единици над пода и разгледайте скриптовете му в инспектора. Уверете се, че сте в лицева перспектива и опитайте да го движите в режим на игра в четирите посоки чрез стрелките или с клавиши WASD.

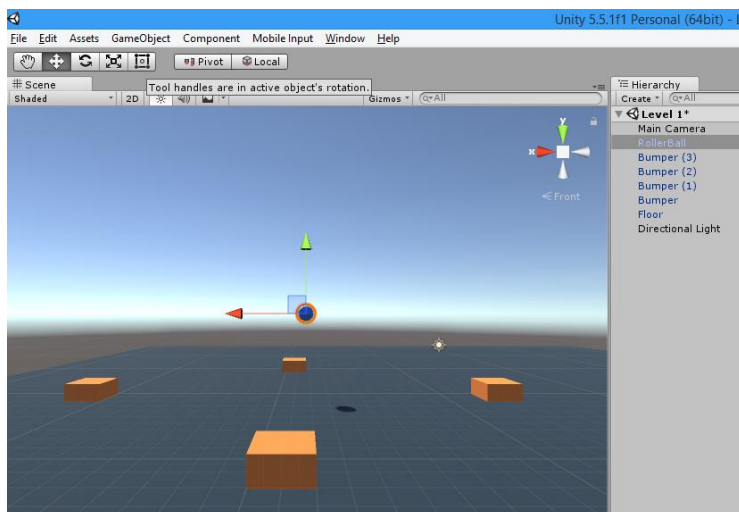
**Стъпка 14** – запазете създадената дотук сцена под име Level 1.

## **В. Позициониране на камерата и организация на проекта**

**Стъпка 15** – в дизайн прозореца, изберете такъв изглед, че да се вижда топката и бъмперите. Селектирайте главната камера и позиционирайте по дизайн изгледа камерата в игровия прозорец, чрез GameObject > Align With View. Забележете промяната в игровия изглед. Играйте и вижте използвания статичен режим на камерата и въздушна перспектива.



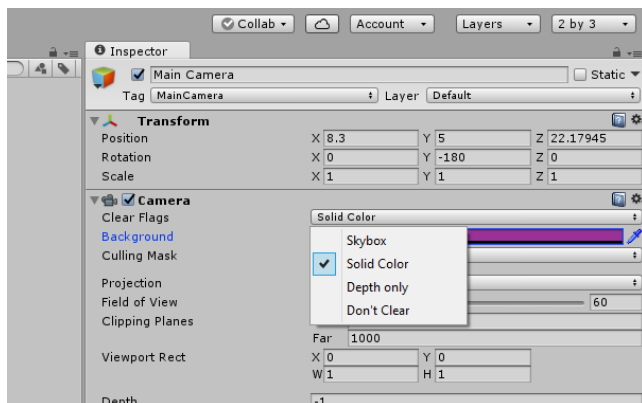
**Стъпка 16** – опитайте да движите динамично камерата следейки RollerBall, като за целта я направите част от йерархията на RollerBall (с преместване и влагане под RollerBall). Какъв е резултатът в игрови режим и защо? Сменете показването на посоките от Global режим в Local (при избран RollerBall обект), за да обясните промяната на положението на камерата. Ако не сте доволни, върнете камерата в главната йерархия от обекти.



**Стъпка 17** – за да накараме камерата да следи RollerBall, завлечете в инспектора й скрипта Standard Assets > Utility > Prefabs > SmoothFollow. За Target на този скрипт

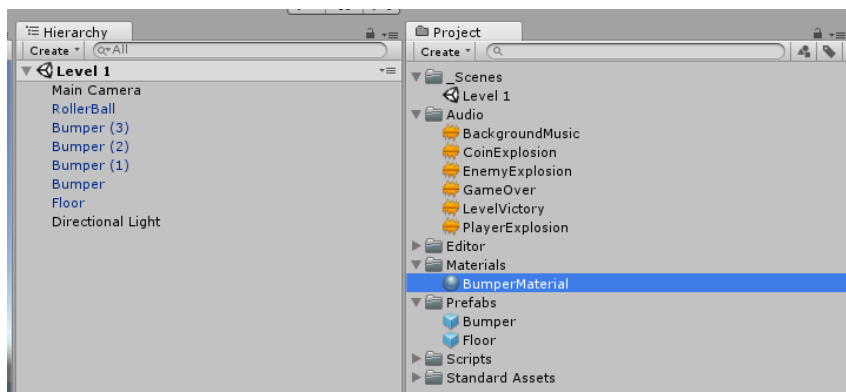
задайте (пак с влачене) RollerBall и установете параметрите Distance в 6 и Height в 5. Разгледайте и изпробвайте скрипта. Съхранявайте периодично сцената.

**Стъпка 18** – сменете в настройките на камерата в инспектора небесния изглед с плътен цвят по избор, както е показано по-долу.



**Стъпка 19** – добавете музика към самата камера (ако добавите музика към друг обект в играта, звука ще затихва при отдалечаване от камерата), чрез Add Component > Audio > Audio Source. Изберете за AudioClip записа BackgroundMusic. Маркирайте Loop и задайте за Volume стойността 0.25 (25% от оригиналната сила). Съхранете сцената и я проиграйте.

**Стъпка 20** – за създадените от нас асети, създайте на май-високо ниво в йерархията три нови папки: \_Scenes, Prefabs и Materials, и преместете в тях съответно сцената Level 1, двата създадени до момента префаба, и материала BumperMaterial.



**Стъпка 21** – в Hierarchy, създайте празен игрови обект чрез Create > Create Empty, с цел структуриране на йерархия от обекти. Преименувайте го Environment и преместете в йерархията му бъмперите и пода. Сега с промяна на местоположението на Environment променяме цялата група на йерархията му.

## Г. Физика и контрол



**Стъпка 22** – селектирайте колайдър компонента на топката и променете радиуса му от 0.5 на 1.5. Проиграйте и обяснете разликата. Върнете старата стойност на радиуса. Променете колайдъра и на бъмперите.

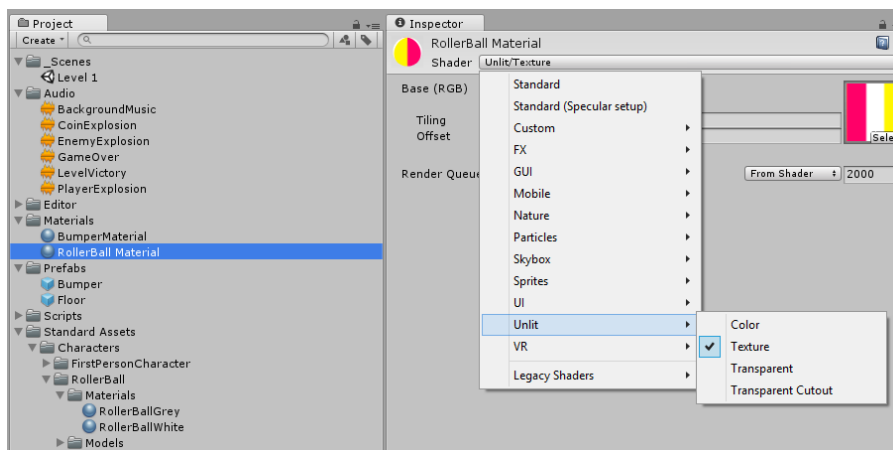
**Стъпка 23** – демаркирайте Mesh Renderer на един от бъмперите и проверете дали той става невидим. Проиграйте сцената така, че да блъснете топката в невидимия бъмпер. Защо тя отскача? Върнете старата стойност и сега демаркирайте Box Collider на този бъмпер. Проиграйте и обяснете преминаването през него. Върнете старата стойност и съхранете сцената.

**Стъпка 24** – пробвайте как зависи отскачането на топката от масата ѝ и от физичния материал, използван в Rigid Body. Заменете физичния материал Rubber с None, а после с Bouncy, и почувствайте разликата. Възстановете Rubber като физичен материал, използван в Rigid Body.

*По желание: Ако имате време, създайте и изпробвайте свои физични материали.*

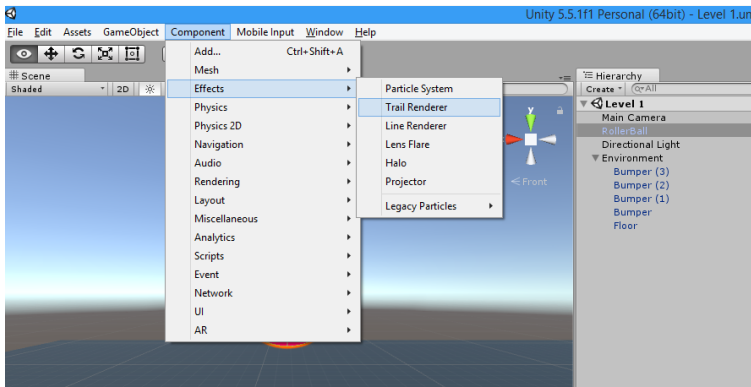
**Стъпка 25** – задайте за физичен материал за префаба на бъмперите Bouncy и почувствайте разликата.

**Стъпка 26** – локализирайте материала на топката, като в нейния Mesh Renderer изберете материала му. Дублирайте този материал, преименувайте го и го преместете в папката с вашите материали. Сменете му цвета. Задайте му текстура чрез Shader > Unlit > Texture и избор на предпочитана текстура. Ако тя е разноцветна, в режим на игра ясно ще проличи хлъзгането на топката (можете да го промените параметрично).



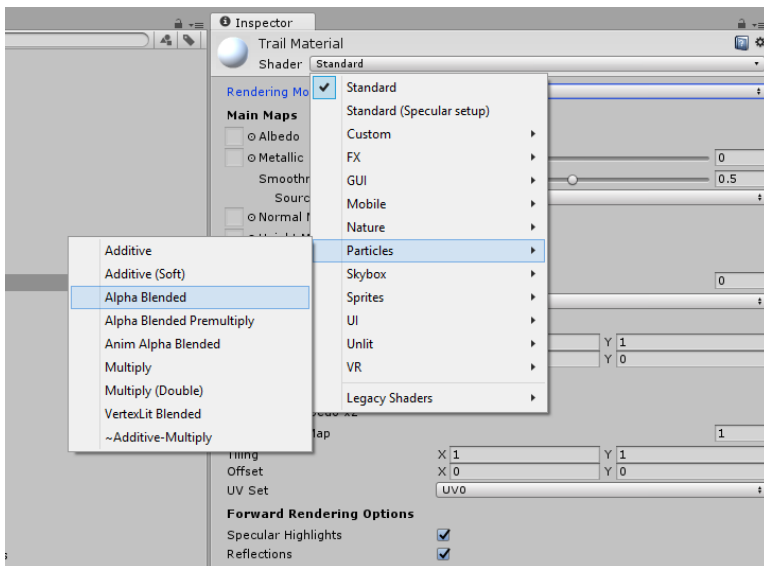
**Стъпка 27** – изберете топката и добавете към нея компонент Trail Renderer по два начина:

- Чрез добавяне на компонент в инспектора
- Чрез меню Components > Effects > Trail Renderer

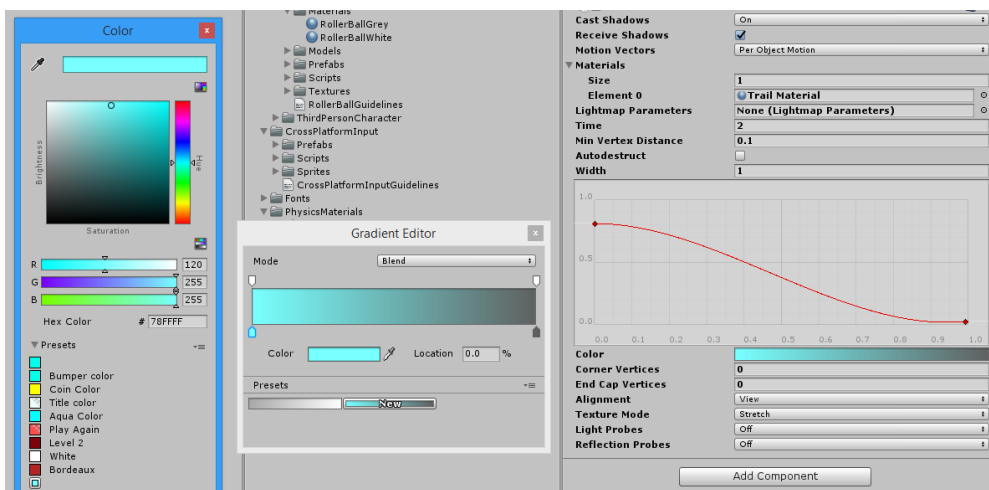


Проиграйте.

**Стъпка 28** – създайте нов ваш материал, наречен Trail Material, в папката Materials. В инспектора му, за Shader задайте Particles > Alpha Blended. Задайте му бял цвят със прозрачност (Алфа ниво) около 30%. Приложете го като Material към Trail Material.



Променете параметрите Time, Starts With и Ends With съответно на 2 секунди, 0.8 и 0 за по-стари версии, или задайте ширина на опашката като функция, както е показано по-долу. Променете и цвета със задаване на градиент (в по-старите версии се задаваха 5 цвята с промяна на Алфа нивото).



**Стъпка 29** – до момента при излизане извън пода топката пропада безкрайно. За да контролираме това, добавяме към топката скрипта Health. Оставете без промяна параметрите на скрипта с имена Health Points, Respawn Health Points, и Number of Lives. За параметъра On Lives Gone изберете Load Level When Dead, а за това кой Level to Load да се зареди задайте Level 1.

**Стъпка 30** – с цел установяване кога топката изпада от пода, създайте дубликат на пода и го преименувайте DeathZone. Скалирайте пода хоризонтално на 0.7, а DeathZone – на 2. Свалете DeathZone на около 2 метра под пода на играта. Създайте нов префаб DeathZone с влачене на обекта в префаб папката и сменете материала на DeathZone с новосъздаден от вас материал.

**Стъпка 31** – добавете към обекта DeathZone скрипта Damage. Маркирайте в неговия Box Collider чек-бокса Is Trigger, за да създаде събитие при колизия, вместо физическо взаимодействие. Проиграйте сценария с изпадане извън пода на играта – скрипта Damage нулира Health Points, при което за Level to Load се зарежда отначало Level 1.

#### **Д. Събиране на монети**

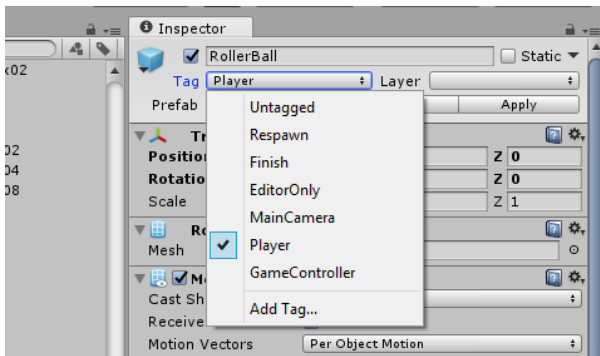
**Стъпка 32** – създайте обект-монета, като използвате префаба PickupPrototype01x01x01 от Standard Assets > Prototyping > Models. Преименувайте обекта на Coin и го издигнете над пода. Скалирайте X, Y и Z по 0.4. Създайте (чрез дублиране) нов Coin Material и в него приложете текстурата SwatchYellowAlbedo с влачене в Albedo на материала. Приложете този нов материал към Coin.

**Стъпка 33** – добавете към Coin сферичен колайдър компонент от раздела Physics. Маркирайте полето му Is Trigger, за да се ползва от скрипта Rotate, и задайте за Radius 1.2.

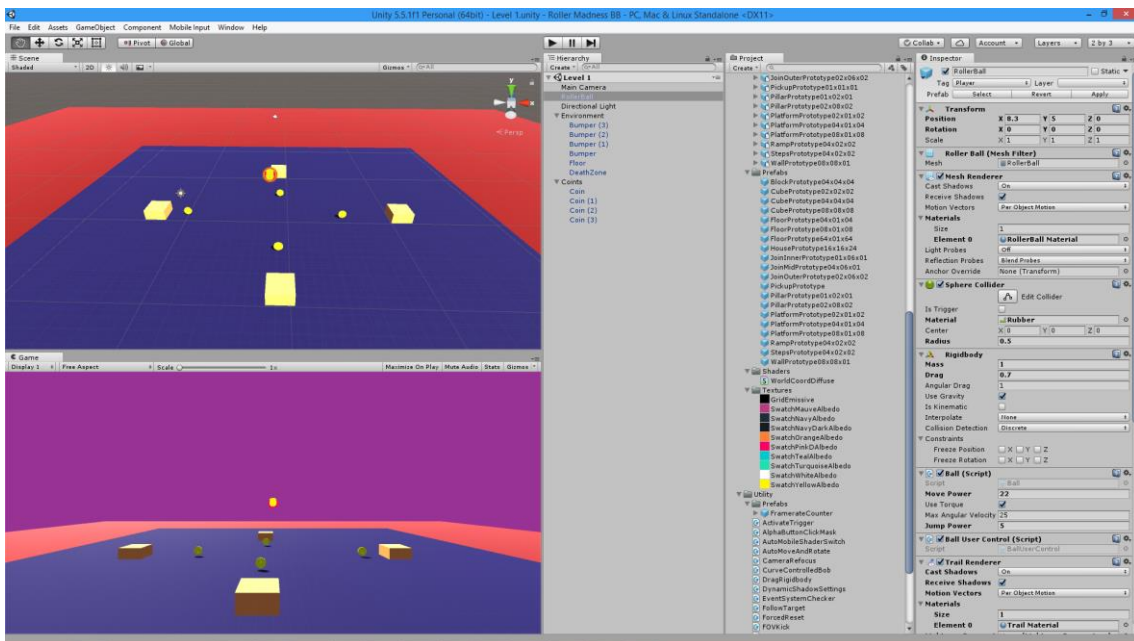
**Стъпка 34** – добавете към Coin скрипта Rotate, като за параметъра Speed зададете 180 градуса за секунда, а за Way зададете Around Y (т.е. да се върти около оста Y). Проиграйте, за да се уверите във въртенето.

**Стъпка 35** – добавете към Coin компонента Rigid Body. Демаркирайте полето Use Gravity и маркирайте полето Is Kinematic, защото Coin ще се управлява не от гравитацията, а от скрипт.

**Стъпка 36** – добавете към Coin скрипта Treasure и задайте за Value стойност 1. Когато RollerBall влезе в колизия с Coin, то резултатът ще се увеличи с 1 точка и монетата ще изчезне. Скриптът Treasure разбира за колизията с героя на играта RollerBall по неговия таг Player и така я различава от други колизии.



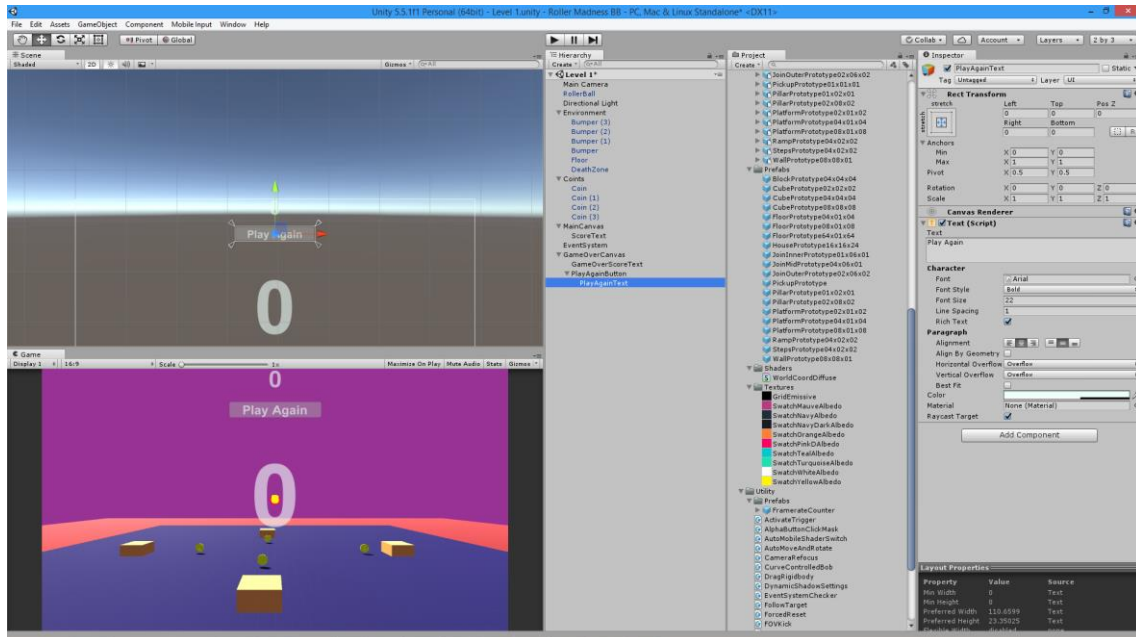
**Стъпка 37** – създайте нов таг (маркер) с име Pickup и го добавете към Coin. Създайте три нови монети и ги разположете подобно на фигурата по-долу. Създайте празен обект, преименувайте го на Coins и в неговата йерархия добавете четирите монети. Проиграйте в вземане на монетите.



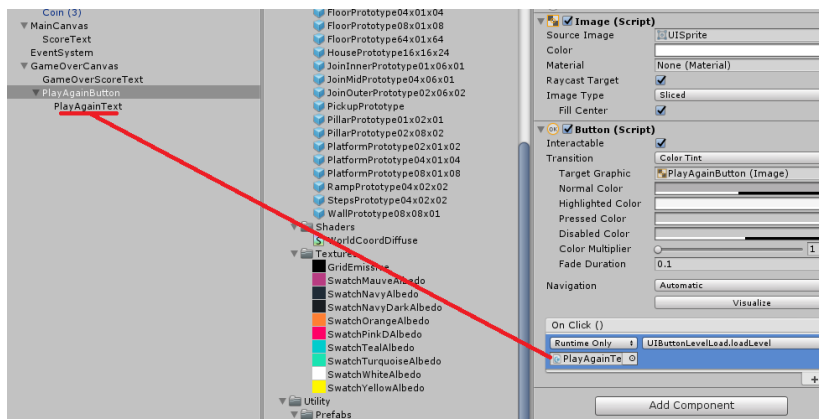
**Стъпка 38** – създайте текстово поле за показване на резултата чрез GameObject > UI > Text. Преименувайте Canvas на MainCanvas, а в неговия инспектор за Scaler задайте UI Scale Mode да е равно на Scale with Screen Size, така че като променим пропорциите на игровия прозорец на 16:9 (втория таб над прозореца), текстът да се скалира. Преименувайте Text на ScoreText и го вижте в режим 2D (избира се над дизайн прозореца). Центрирайте текста, направете го удебелен, с размер 40, стойност 0 и бял цвят.

**Стъпка 39** – създайте GameOverCanvas с GameOverScoreText. Копирайте компонента ScoreText и поставете стойностите му в GameOverScoreText. Увеличете размера на GameOverScoreText.

**Стъпка 40** – създайте бутон PlayAgainButton с текст PlayAgainText, който да има стойност Play Again. Копирайте компонента ScoreText и поставете стойностите му в PlayAgainText. Настройте го така, че текстът да се побира в бутона.

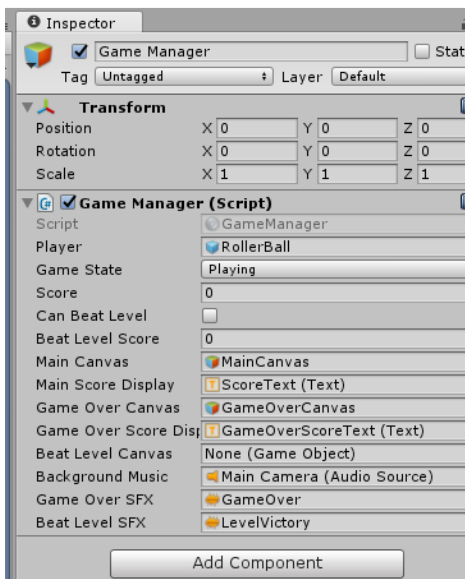


**Стъпка 41** – за текста на бутона PlayAgainText добавете скрипта с име UI Button Level Load. За Level to Load изберете Level 1. Селектирайте PlayAgainButton и в инспектора за събитието On Click() добавете настройките, показани на фигурата по-долу.



**Стъпка 42** – деактивирайте целия GameOverCanvas с деселектирането му в инспектора (първият най-горен чек-бокс).

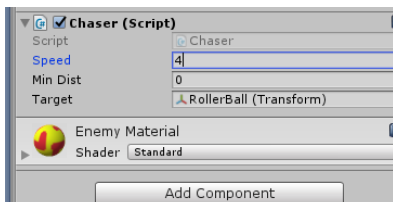
**Стъпка 43** – на най-високо ниво в йерархията създайте празен обект и го преименувайте на Game Manager. Нулирайте го. Добавете към него скриптовия компонент с име Game Manager и му задайте стойности на параметрите така, както е показано на фигурата по-долу.



**Стъпка 44** – за играча RollerBall, в компонента Health задайте за параметъра On Lives Gone стойността Do nothing When Dead, понеже скриптовият компонент с име Game Manager ще управлява играта вместо да се зарежда автоматично дадено ниво.

## Е. Врагове и преследване; ефекти

**Стъпка 45** – превключете отново на 3D визуализация. Създайте враг в играта под формата на куб с име Enemy, със собствен материал EnemyMaterial, напр. използващ текстурата Enemy. За обекта добавете скрипт с име Chaser и го настройте по фигурата по-долу. Тествайте преследването.

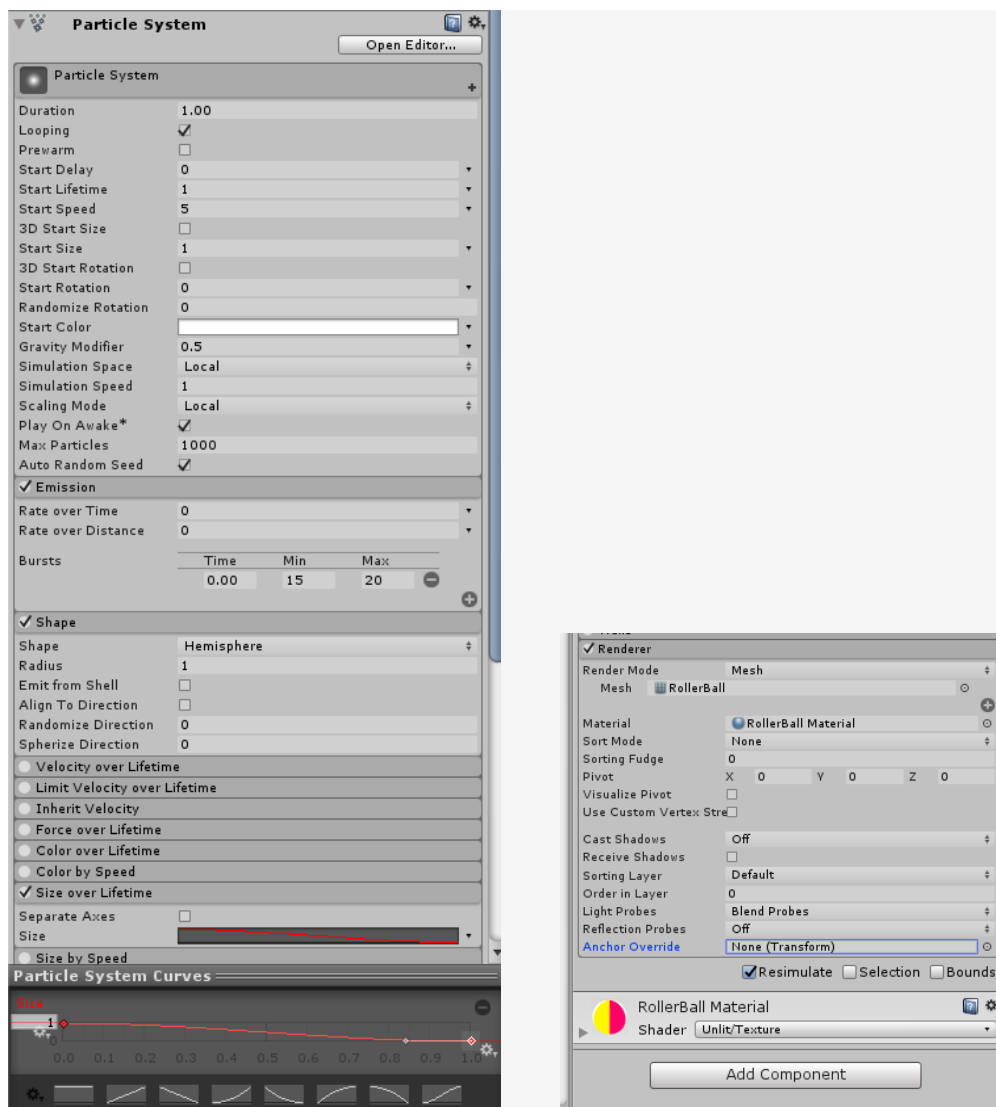


**Стъпка 46** – добавете към врага компонент Rigid Body, без да променяте настройките. Задайте за материал в Box Collider. Отдалечете го от топката и го вдигнете на 3 метра.

**Стъпка 47** – добавете към врага компонент-скрипт с име Damage. Селектирайте Damage on Collision и деселектирайте Damage on Trigger. добавете към врага компонент-скрипт с име Health. Така при удар (било с героя, било с друг враг) врагът също ще загива, и то веднага.

**Стъпка 48** – създайте префаб на врага и чрез него – още един враг. Разположете го срещуположно на първия. Организирайте враговете в йерархия подобно на монетите, която да е на най-високо ниво. Проиграйте сценария, при който двата врага се убиват взаимно при маневри при преследването. Ако е необходимо, намалете скоростта на врага.

**Стъпка 49** – създайте система от частици чрез менюто GameObject > Particle System. Преименувайте я на Explode Player Particle.

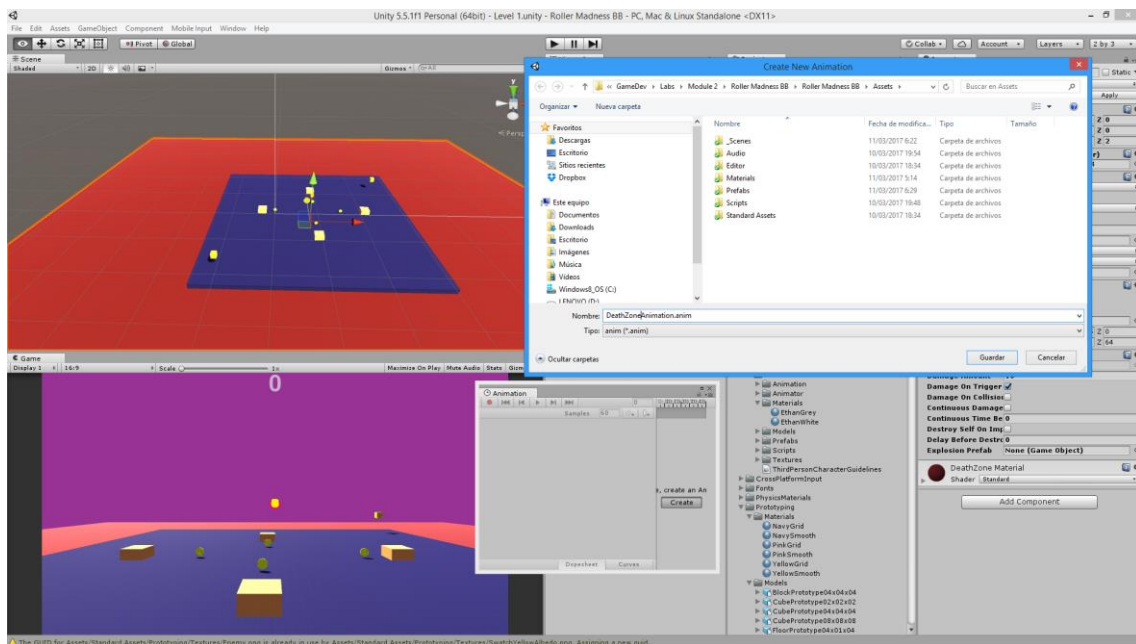


Добавете компонент Audio Source с аудио клип Player Explosion. Добавете и компонент Timed Object Destructor с цел разрушаването на системата от частици след 2 сек, като премахнете цикъла.

**Стъпка 50** – създайте в Prefabs префаб на Particle System и след това изтрийте обектите за този префаб. После в компонента Health на RollerBall, задайте този параметър като стойност на Explosion Prefab. Тествайте играта.

**Стъпка 51** – създайте в Prefabs префаб-копия на Particle System с име EnemyExplosion и CoinExplosion. В съответните компоненти Health на Enemy и Coin, задайте ги като стойност на параметъра Explosion Prefab. Запазете сцената и проекта и тествайте играта.

**Стъпка 52** – нека анимираме DeathZone - зоната на смъртта, за да покажем, че е опасна. Избираме DeathZone и задаваме Window > Animation (Strl+6). В новия прозорец на анимационния редактор избираме бутона Create и създаваме компонент-анимация. Можем да пхнем този прозорец в Таб след прозореца за игрови режим.



**Стъпка 53** – в режим на запис в анимационния прозорец, на 3-4 места задаваме през 1.5 секунди промяна на цвета на мъртвата зона и на височината ѝ. Добавяме компонент Rigid Body с кинематичен контрол заради движението на зоната. Проиграваме анимацията.

**Стъпка 54** – добавяме точки на клониране (spawners) с цел по-вълнуваща игра. Създаваме празен обект с име EnemySpawner и за него добавяме скрипта SpawnGameObject. Като параметри на скрипта задаваме Spawn Prefab=Enemy, Min Seconds Between Spawning=4 и Max Seconds Between Spawning=8 (времето за размножаване е между тези две стойности). За цел на преследването задаваме топката. Позиционираме EnemySpawner на 10 метра височина над един от крайщата на пода и създаваме негов префаб, с който генерираме още три EnemySpawner обекта. Изтриваме съществуващите врагове (те не са нужни) и слагаме четирите EnemySpawner обекти в йерархия на генератори на врагове. Запазваме сцената и тестваме.

**Стъпка 55** – нека създадем точки на клониране (spawners) и за монетите, които обаче ще падат отгоре и ще отскачат, т.е. ще са от материал Bouncy. За целта дублираме префаба на Coin и го кръщаваме BouncyCoin. Извършваме следните промени:

- Rigid Body вече управляваме само гравитационно, но не и кинематично
- Понеже в съществуващия колайдър Is Trigger трябва да остане маркирано, създаваме нов Box Collider с празно поле Is Trigger и с физичен материал Bouncy
- Премахваме скрипта Rotate

С този нов BouncyCoin правим тестване с обект, създаден на 5м височина.



**Стъпка 56** – копираме компонента Trance Renderer RollerBall в BouncyCoin. Добавяме компонент Time Object Distractor с Time Out равен на 12 сек.

**Стъпка 57** – Тествайте играта в различни сценарии.

**Стъпка 58** – За домашна работа – направете играта в две сцени – една за ниво 1 и друга за ниво 2. Осигурете лесно преминаване от ниво 1 към ниво 2, където разположете повече препятствия и настройте враговете да се движат по-бързо. Успех!