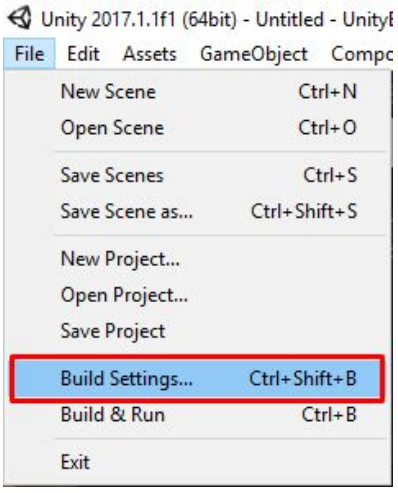
Turnbase RPG Template (Ver.1.01)
Documentation

Getting started 1
Levelling Formula 2
Game Database 3
Item 5
Item Tier 6
Item Attributes 9
Character 9
Prepare Character Entity 9
Character Animator Controller in Depth 15
Character Item 16
Equipment Item 19
Skill 22
Attack Animation 25
Skill Cast Animation 28
Environment Data 29
Stage 32
Loot Box

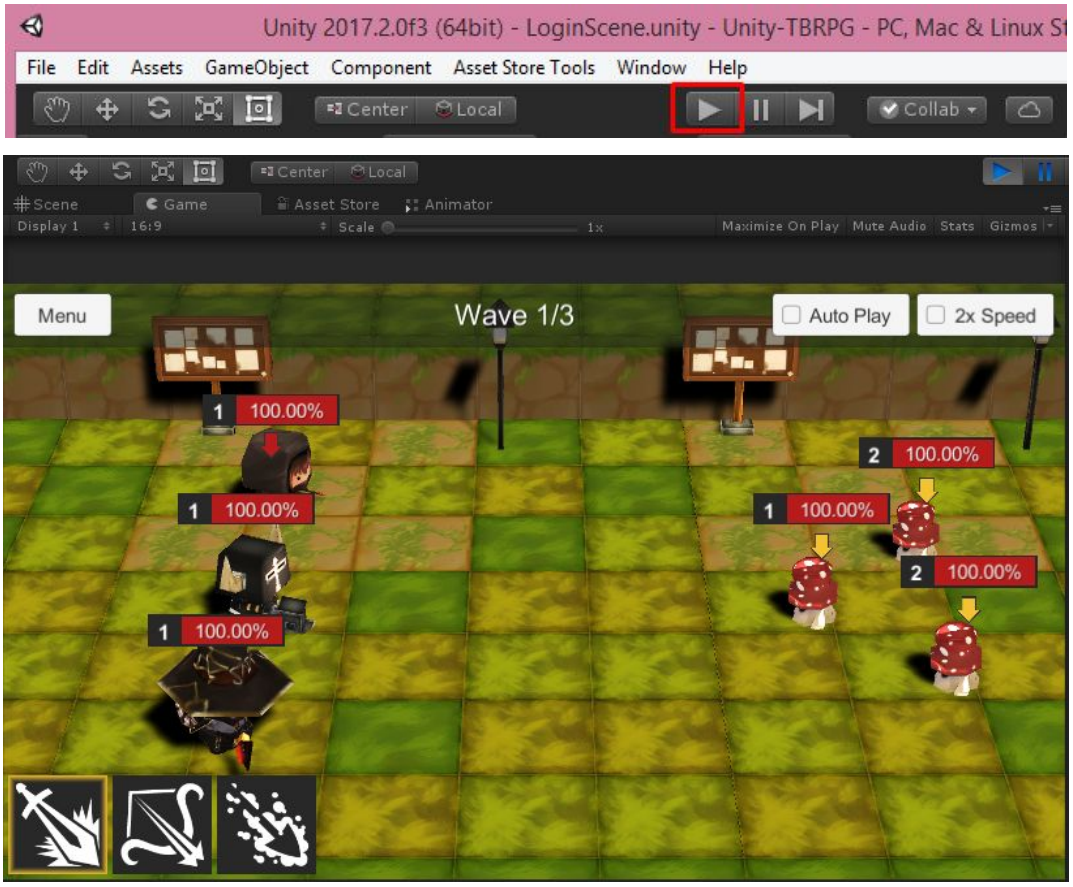
Getting Started

After imported the package, then we're going to test the demo, click menu **File -> Build Settings**



Then drag **LoginScene**, **ManageScene** and **BattleScene** from **/Assets/RPGGame/Demo/Scenes** to add them in **Scenes In Build** section

Then you can try by press play button in the editor



Levelling Formula

Before we start, I'd like to tell you about the formula that I've use to calculate character stats, required exp, price and so on for this project, The formula is

```
Current_Stat = Min_Stat + (Max_Stat - Min_Stat) *
{[(Current_Level - 1) / (Max_Level - 1)] ^ Growth_Factor}
```

While Min_Stat is stat at level 1, Max_Stat is stat at max level and Growth_Factor is value determines how fast this stat growth, for example when Growth_Factor equals to

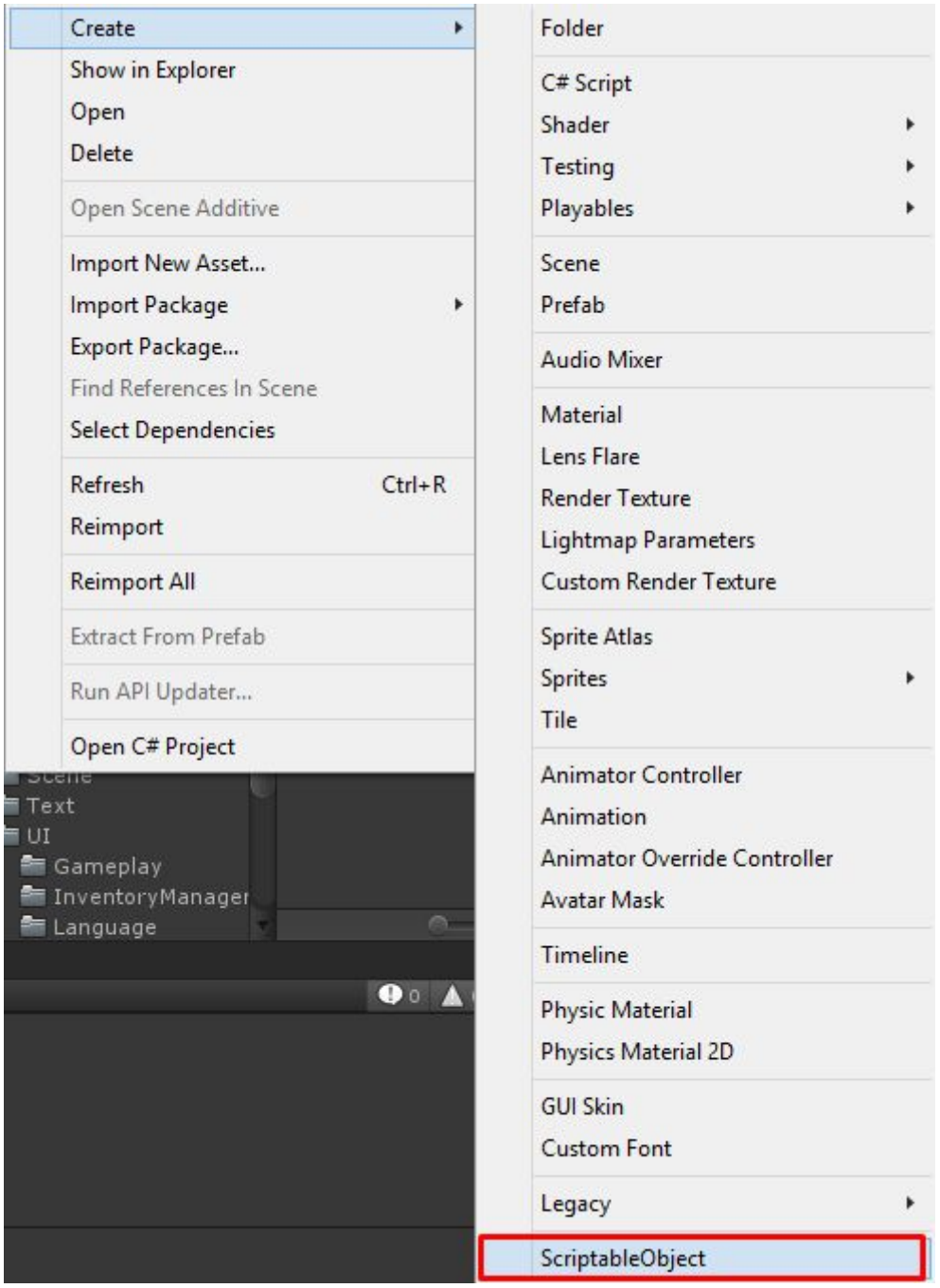
- 0.7 The stat will grows faster at early levels then becomes much slower at later levels.
- 1 The stat will grows at a fixed value per each level.
- 1.5 The stat will grows slower at early levels then becomes much faster at later levels.

So every stats will have fields to set their are Min_Stat, Max_Stat and Growth_Factor, Max_Level will be set at difference place, so you don't have to set stats for every levels just set min value, max value and growth factor

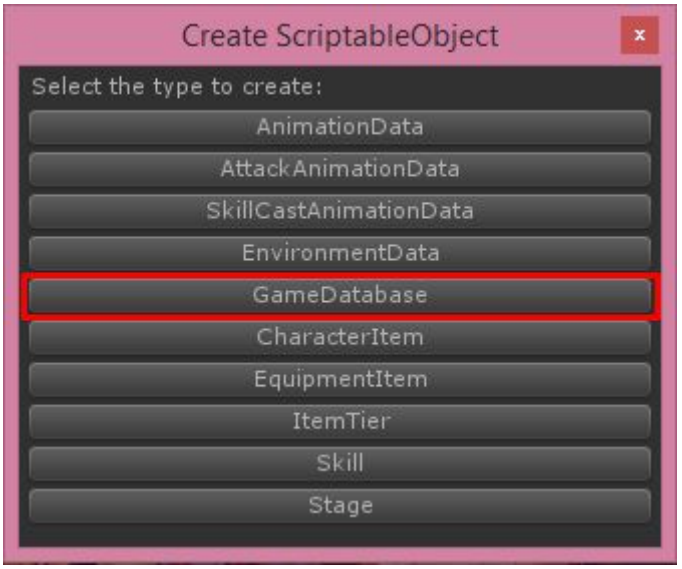
Game Database

Game database is important part, we'll use it to collect an game data such as characters, items, stages, rewards

I've prepared the demo game database so might not want to create new game database but if you want, You can create new game database by right click on **Project** tab then choose **Create -> ScriptableObject**

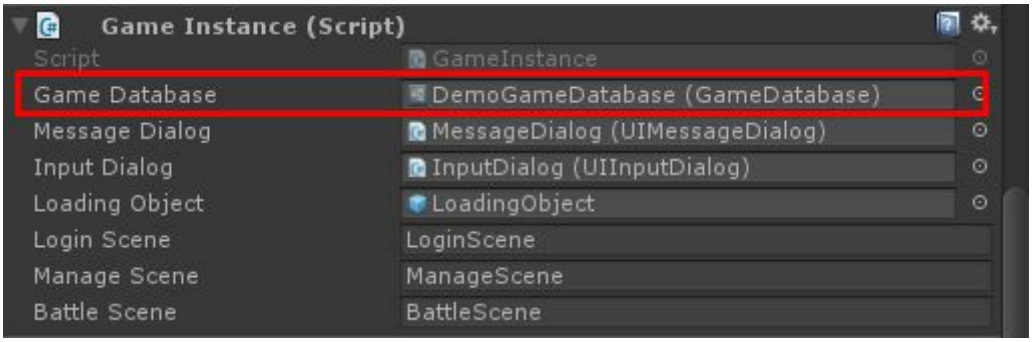


Then choose **GameDatabase**



Then new game database will be created

Then set game database in **GameInstance** which located in **LoginScene**



In game database you can set following configs:

- **Player Max Level** this is max level of player (not the characters or items)
- **Player Exp Table** require exp for level up from level 1 to **Player Max Level** with growth factor
- **Soft Currency** this is currency which we will use it to Upgrade, Evolve items can receive by clear stage, sell items, you can set its **Icon** and **Start Amount** (Amount when start game)
- **Hard Currency** this is currency which we will use it to random item from loot box, revive characters, you can set its **Icon** and **Start Amount** (Amount when start game)
- **Stage Stamina** this is required stamina to start battle, you can set its **Icon**, max amount from level 1 to **Player**

Max Level with growth factor in **Max Amount Table**, **Recover Unit** and **Recover Duration**

- **Stage Formations** this is list of formation and its info
- **Items** this is list of all items, see more info at **Item** section
- **Stages** this is list of all stages, see more info at **Stage** section
- **Start Items** this is list of items that player will receives at beginning of the game
- **Start Characters** this is list of characters that player will receives at beginning of the game and will be set to the formation
- **Unlock Stages** this is list of stages that will be unlocked at beginning of the game, you should first stage here
- **Character Base Attributes**, this is base attributes for all characters, this will be sum with calculated character's attributes
- **Revive Price**, required hard currency to revive all characters
- **Min Atk Vary Rate**, **Max Atk Vary Rate** are Rate that will be random by min and max value to multiply with attack damage
- **Reset Item After Evolve**, if this is True item level will be reset to 1 after evolve

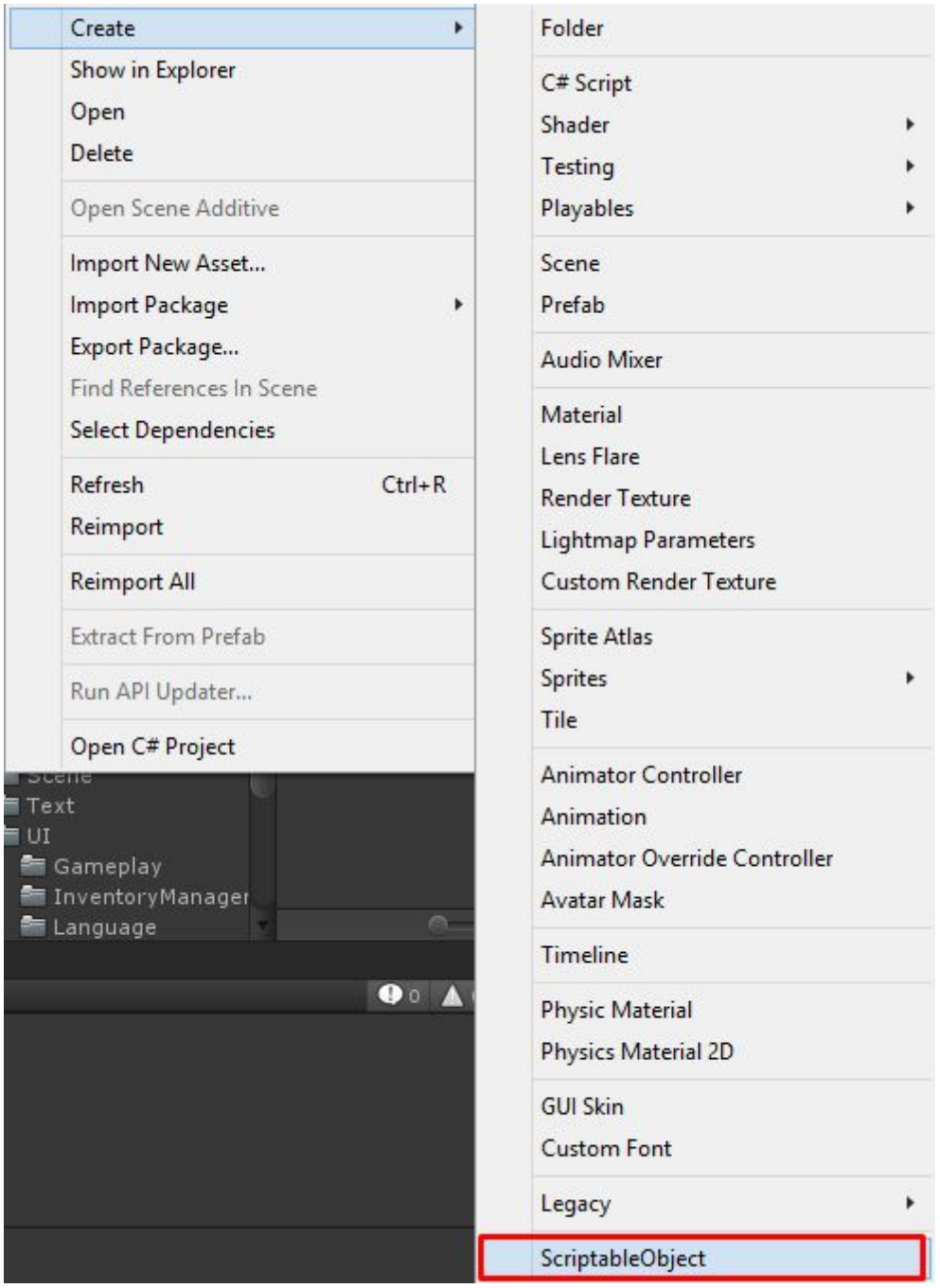
Item

There are many type of items such as Character, Equipment, Junk but for every items there are following configs

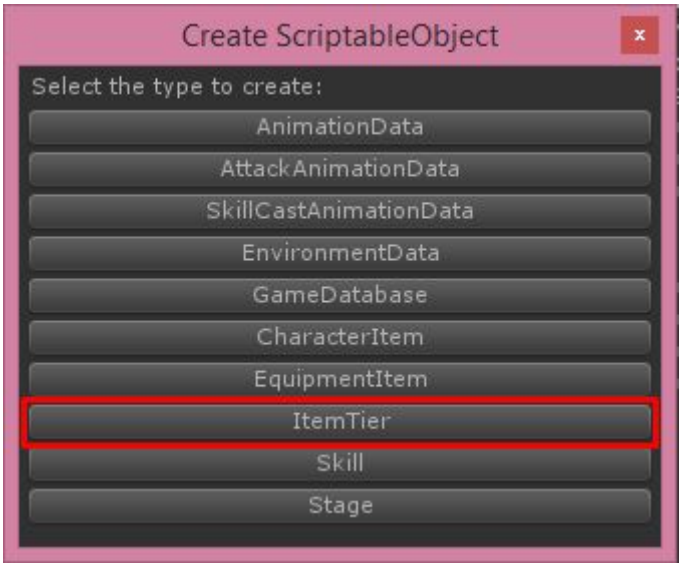
- **Title**, item's title/name
- **Description**, item's description
- **Icon**, item's icon
- **Item Tier**, item's tier see more at **Item Tier** section
- **Max Stack**, item's max stack in inventory

Item Tier

All items data required item tier, You can create new item tier by right click on **Project** tab then choose **Create -> ScriptableObject**

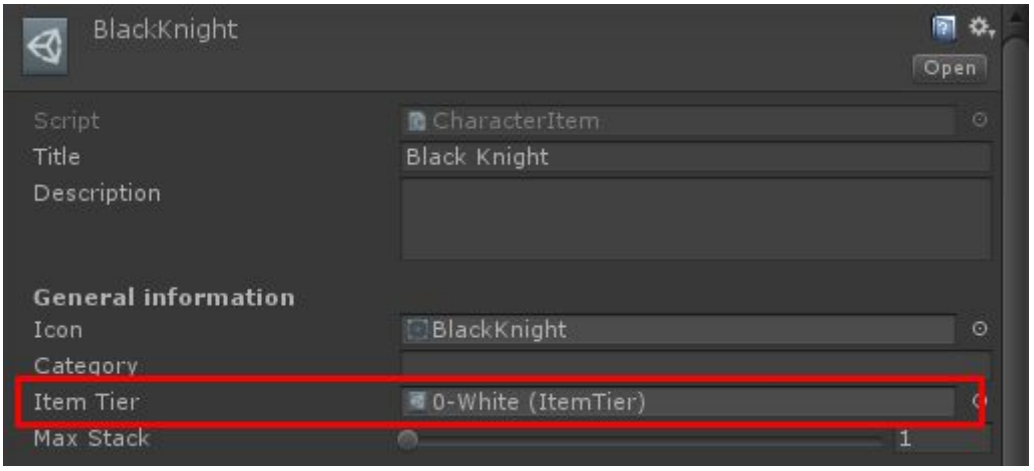


Then choose **Item Tier**



Then new item tier will be created, you must set its name to be unique (Not be the same as another item tier)

Then after you set data for item tier, you can set item tier to your item as you wish



Each item tier there are following configs

- **Title**, tier's title/name
- **Description**, tier's description
- **Icon**, tier's icon
- **Max Level**, item's max level
- **Exp Table**, requires exp for level up from level 1 to **Max Level** with growth factor
- **Sell Price Table**, sell price from level 1 to **Max Level** with growth factor

- **Level Up Price Table**, level up price from level 1 to **Max Level** with growth factor
- **Reward Exp Table**, amount of exp that will be given to level up item when use this item as level up material
- **Evolve Price**, price to evolve

Item Attributes

For an Character and Equipment their are an attributes that will be calculated for character battle purposes there are

- **Hp**, hit point/health point, character will dies when this value <= 0
- **PAtk**, Physical Attack, this will minus to **pDef** to calculate damage
- **PDef**, Physical Defend, **pAtk** will minus to this to calculate damage
- **MAtk**, Magical Attack, this will minus to **mDef** to calculate damage
- **MDef**, Magical Defend, **mAtk** will minus to this to calculate damage
- **Spd**, Speed, character with higher speed will have more chance to attack
- **Eva**, Evasion, character with higher evasion will have more chance to avoid damage from character with lower accuracy
- **Acc**, Accuracy, Character with higher accuracy will have more chance to take damage to character with lower evasion

When character attacks, damage will calculated by

- pDamage = a.pAtk - b.pDef, if this is less than 0 it will equals to 0
- mDamage = a.mAtk - b.mDef, if this is less than 0 it will equals to 0
- totalDamage = (pDamage + mDamage) * (Random between minAtkVaryRate and maxAtkVaryRate)
- If critical occurs, totalDamage = totalDamage * critDamageRate
- If block occurs, totalDamage = totalDamage / blockDamageRate

- Evade chance = (Acc - Eva) / Acc, if character can evade character won't receive damages

Character

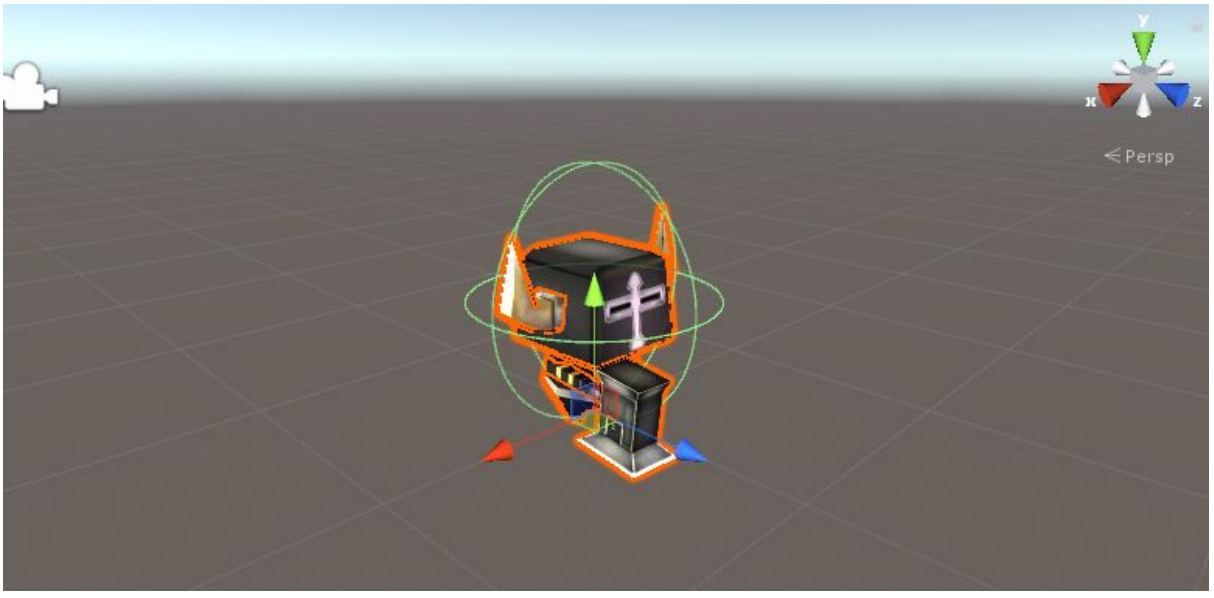
Character is one of item, to create new character you have to do following steps

- Prepare **Character Entity** with its animations, **Character Entity** is will be instantiated in manage and battle scene to show up how character's appearance is and its movement
- Prepare **Character Item** with its data (Item Tier, Attributes, Skills, Evolve Requirements, Character Entity), You can set Skills, Evolve Requirements for **Character Item** later but it's required **Character Entity** so that is why we should prepare **Character Entity** first
- Add **Character Item** to **Items** field in **Game Database**

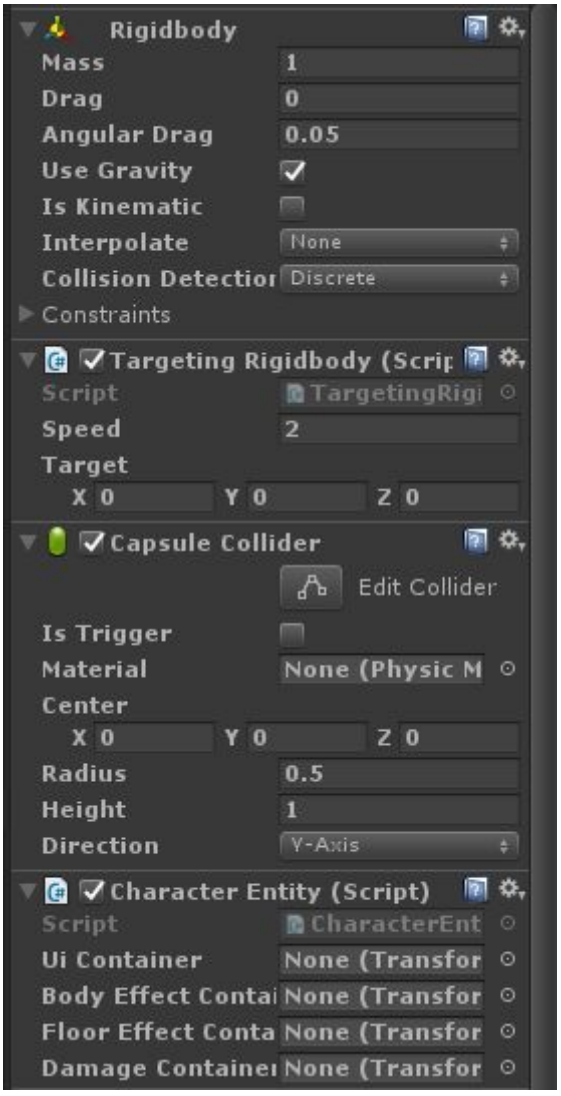
Prepare Character Entity

To prepare character entity you should create new empty scene (from menu **File -> New Scene**) for easily to manage character model and its animation

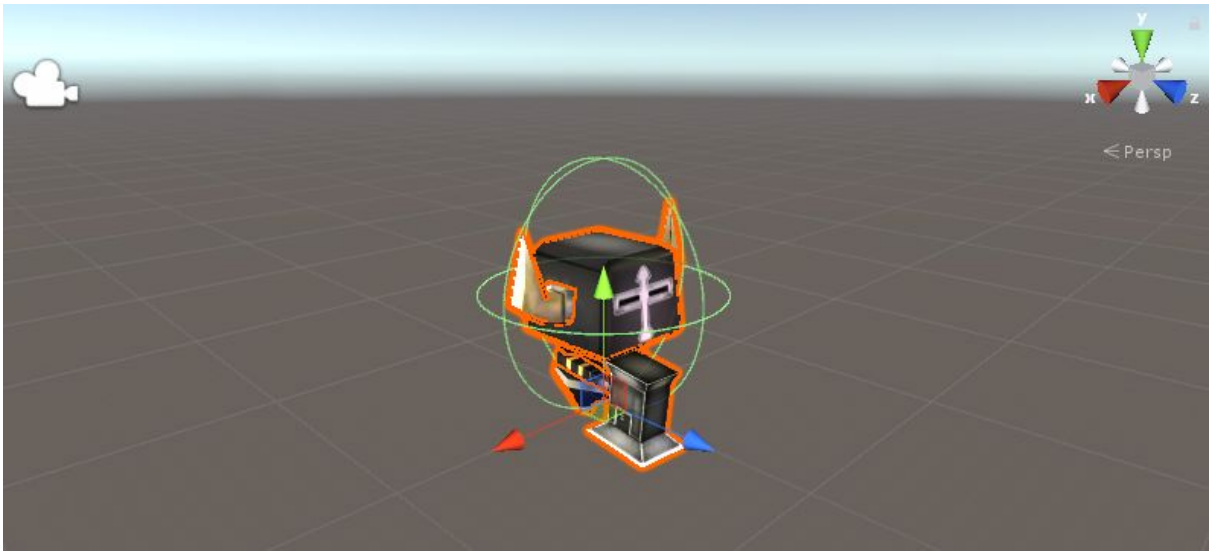
Then drag you character mesh from **Project** tab to the scene



Then add **Character Entity** component

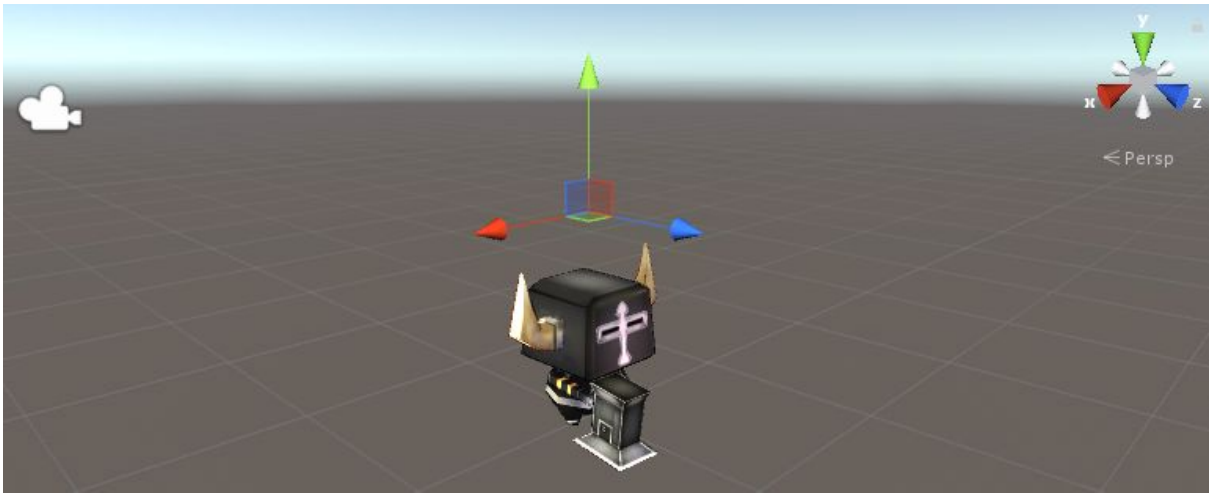


Then adjust your Capsule Collider, it should fit to character model

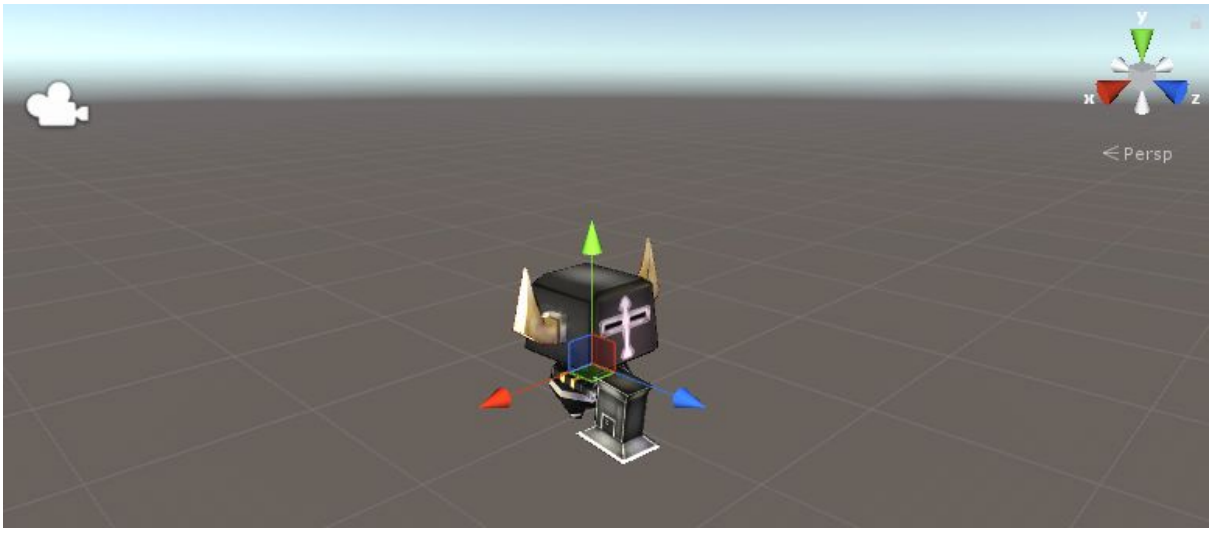


Then set an element container there are:

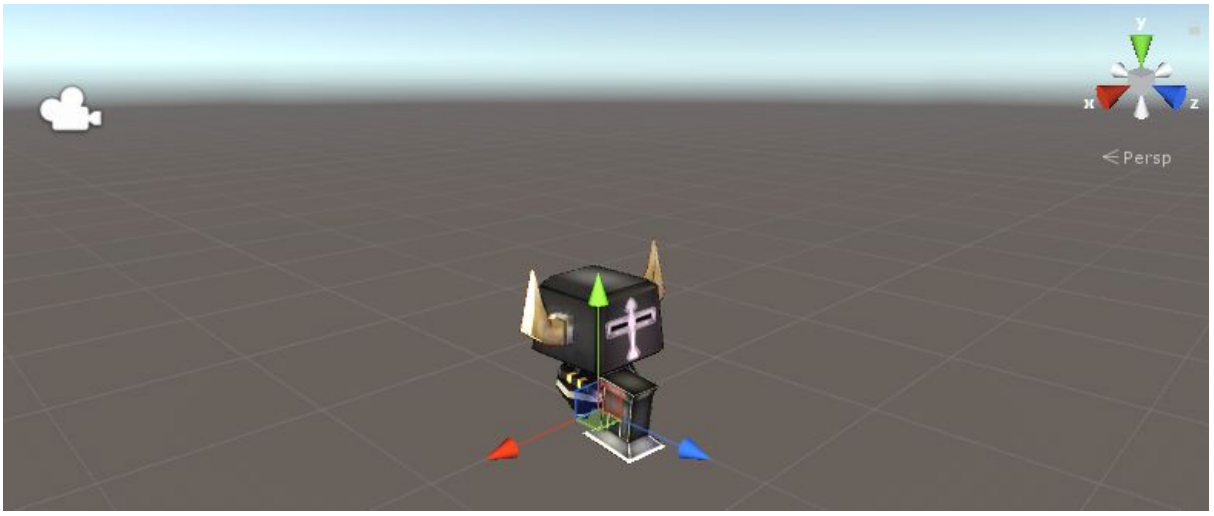
- **Ui Container**, the transform which will instantiates an UI element into it if you want to show an Hp, Buff icons above your character you may set transform to be above the character



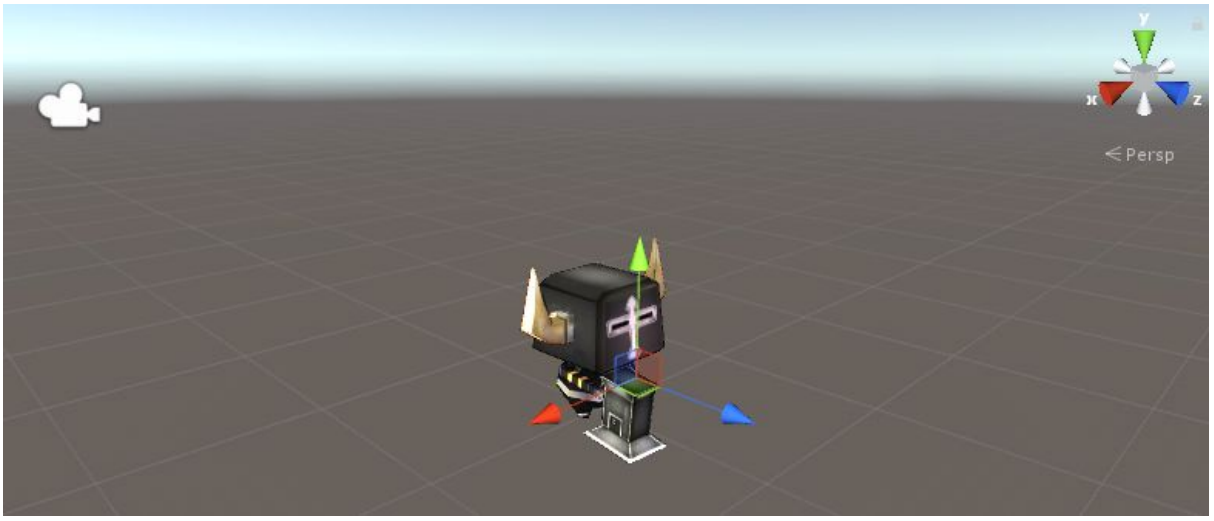
- **Body Effect Container**, the transform which will instantiates an effects such as hit effects, cast effect that will be appeared on the character



- **Floor Effect Container**, the transform which will instantiates an effects such as hit effects, cast effect that will be appeared below the character

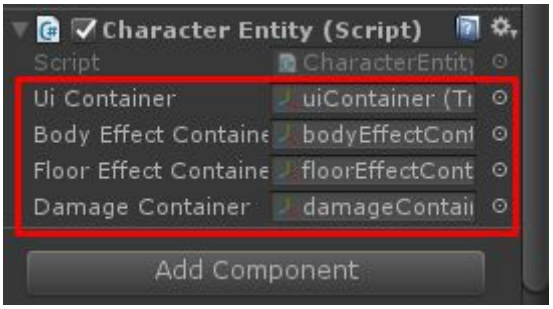


- **Damage Container,** the transform which will instantiates an damages such as arrow, fireball it should be on front of character

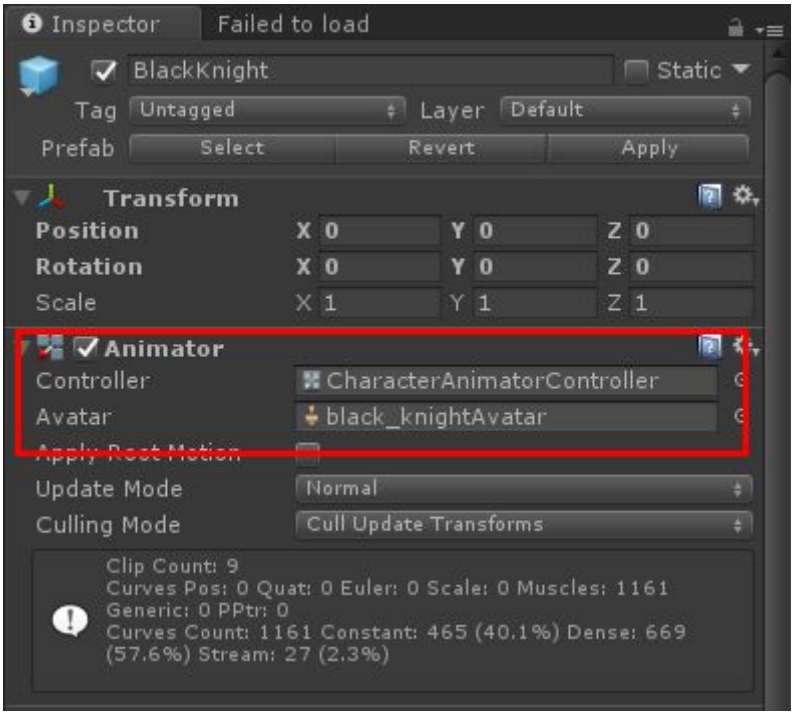


All container should be the child of the character entity so after you setup all container transform you should place them inside the character entity

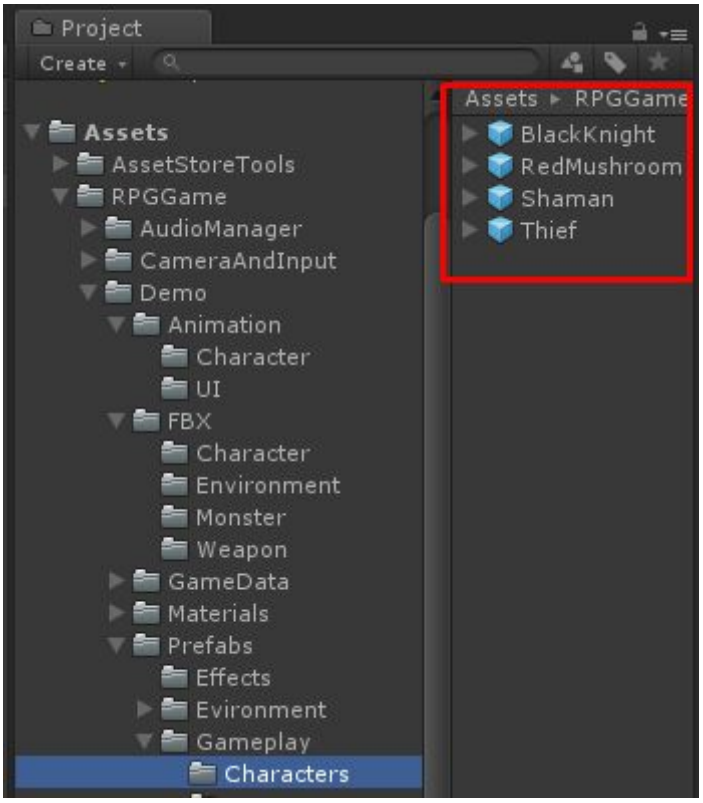
Then set them to the Character Entity component



After finish an containers setup then you have to set animator controller, I've prepared the Demo's animator controller which located at **/Assets/RPGGame/Demo/Animation/Character** its name is **CharacterAnimatorController** you may duplicate it as your own (Press Ctrl+d) then drag it to **Animator** component which attached to your **Character Entity**



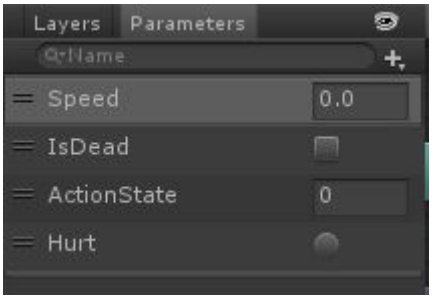
Then make your **Character Entity** as **Prefab** by drag it to **Project** tab



Character Animator Controller in Depth

An Animator Controller for the Character Entity is required following parameters:

- **Speed (Float),** parameter for move speed if character is moving speed will be greater than 0 so you can make transition by speed condition so you should set **Animation Clip** to the **State** which will play move animation with transition condition: **Speed** greater than 0
- **IsDead (Boolean),** parameter to play dead animation when character dead so you should set **Animation Clip** to the **State** which will play dead animation with transition condition: **IsDead** is **True**
- **ActionState (Integer),** parameter to play attack, cast skills animations which receives from **Animation Data**'s field named **Animation Action State** so you should set **Animation Clip** to the **State** which will play attack, cast skills animations with transition condition: **ActionState** equals to the number that set at **Animation Data -> Animation Action State** see more at **Attack Animation** and **Skill Cast Animation** section
- **Hurt (Trigger),** parameter to play hurt animation when character attacked



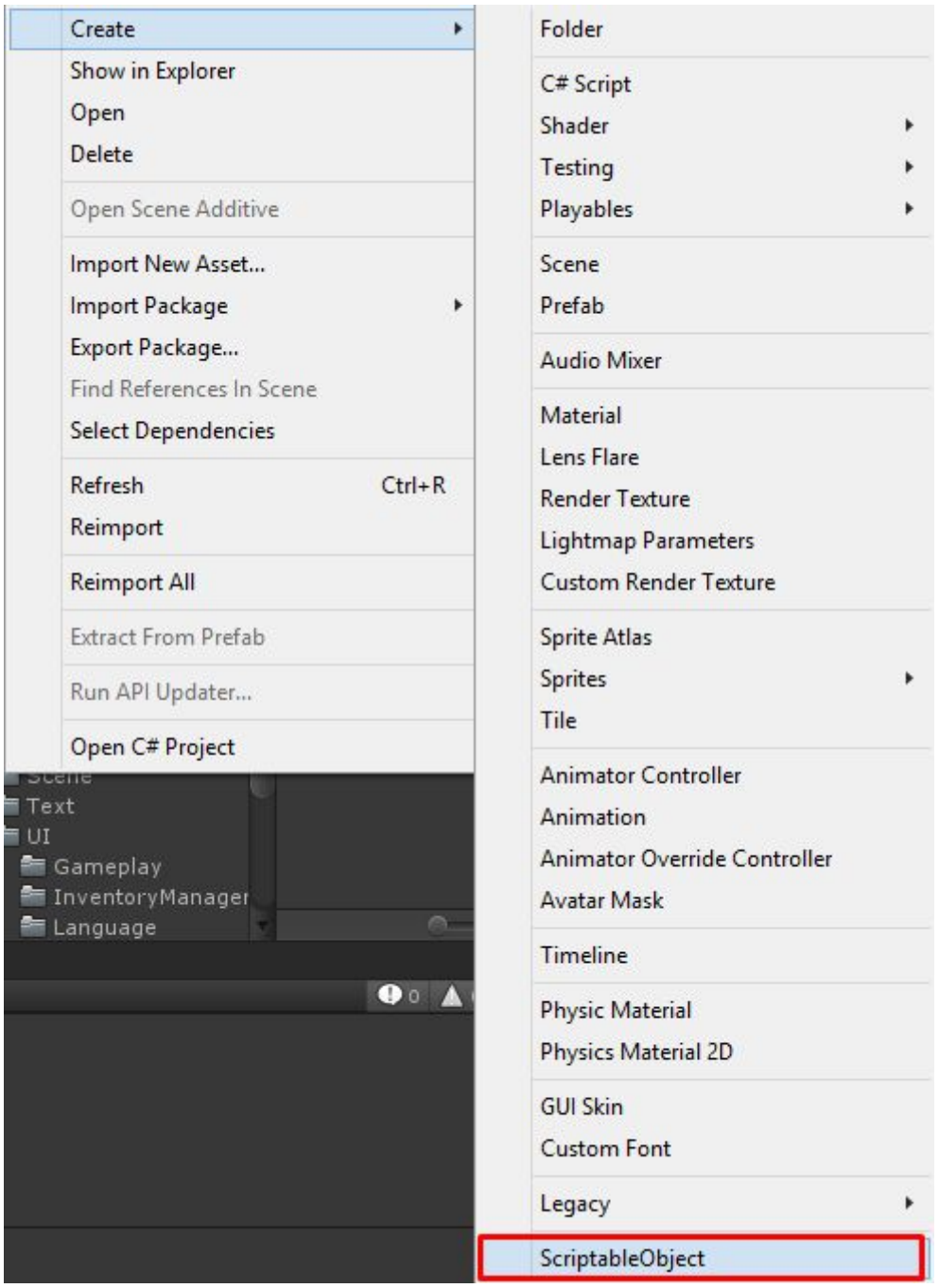
You can see Demo's **CharacterAnimatorController** which located at **/Assets/RPGGame/Demo/Animation/Character**

Character Item

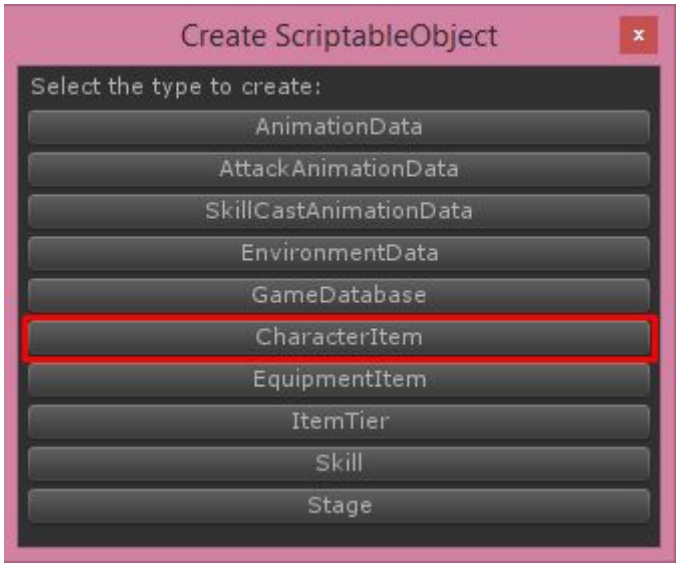
Character Item is extended from Item with more configs there are

- **Attack Animation,** animation data to play attack animation see more at **Attack Animation** section
- **Skills,** skills that can use by the character
- **Model,** this will placed by **Character Entity,** which is will be instantiated in manage and battle scene to show up how character's appearance is and its movement
- **Evolve Info,** an required evolve materials and target item which will be evolving to will be set here

You can create new character item by right click on **Project** tab then choose **Create -> ScriptableObject**

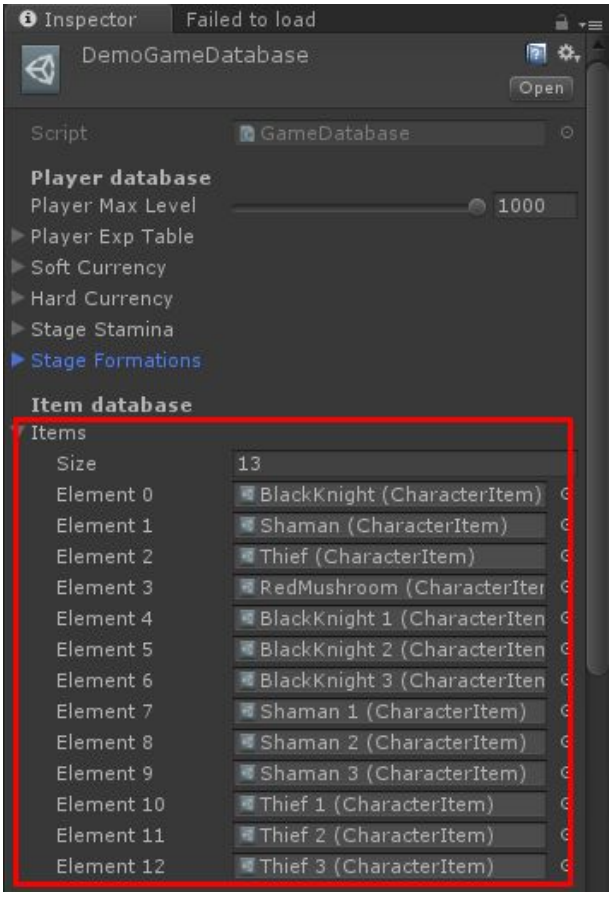

Create
Show in Explorer
Open
Delete
Open Scene Additive
Import New Asset...
Import Package
Export Package...
Find References In Scene
Select Dependencies
Refresh Ctrl+R
Reimport
Reimport All
Extract From Prefab
Run API Updater...
Open C# Project
Text
UI
Gameplay
InventoryManager
Language
Folder
C# Script
Shader
Testing
Playables
Scene
Prefab
Audio Mixer
Material
Lens Flare
Render Texture
Lightmap Parameters
Custom Render Texture
Sprite Atlas
Sprites
Tile
Animator Controller
Animation
Animator Override Controller
Avatar Mask
Timeline
Physic Material
Physics Material 2D
GUI Skin
Custom Font
Legacy
ScriptableObject

Then choose **Character Item**



Then new character item will be created, you must set its name to be unique (Not be the same as another items includes another types items)

Then add it to your **Game Database**

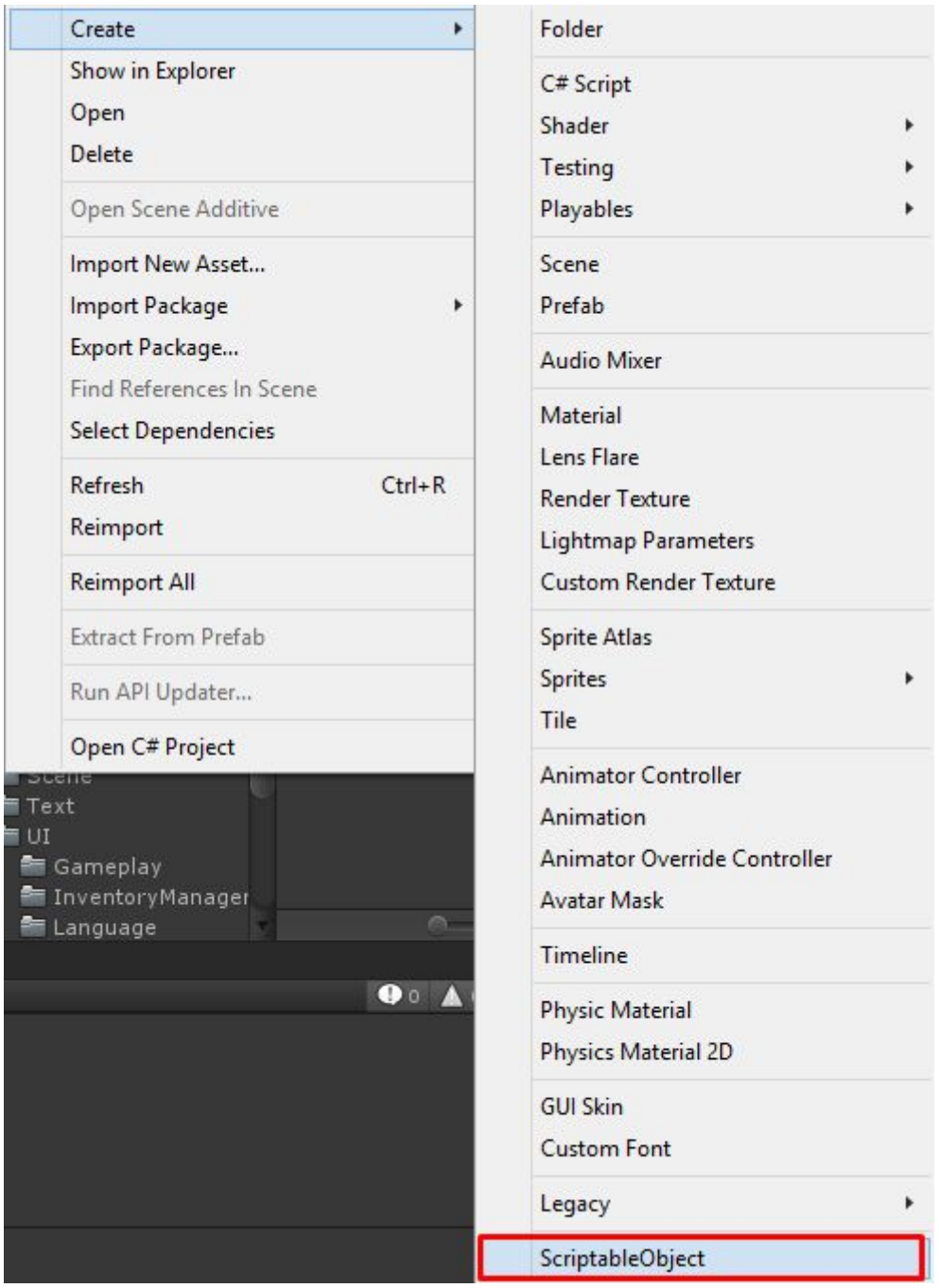


Equipment Item

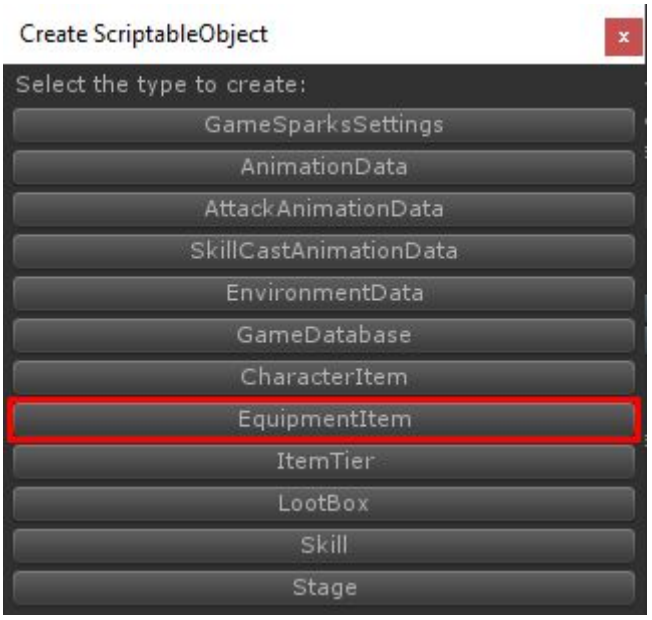
Equipment Item is extended from Item with more configs there are

- **Extra Attributes,** these extra attributes will not increase when item level up
- **Equippable Positions,** positions that this item can be equipped
- **Evolve Info,** an required evolve materials and target item which will be evolving to will be set here

You can create new equipment item by right click on **Project** tab then choose **Create -> ScriptableObject**

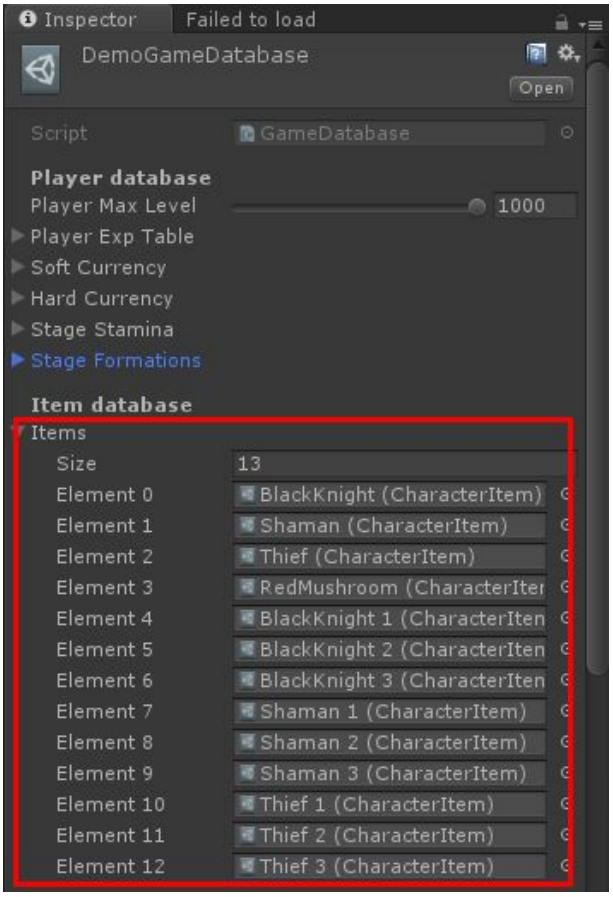

Create
Show in Explorer
Open
Delete
Open Scene Additive
Import New Asset...
Import Package
Export Package...
Find References In Scene
Select Dependencies
Refresh Ctrl+R
Reimport
Reimport All
Extract From Prefab
Run API Updater...
Open C# Project
Text
UI
Gameplay
InventoryManager
Language
Folder
C# Script
Shader
Testing
Playables
Scene
Prefab
Audio Mixer
Material
Lens Flare
Render Texture
Lightmap Parameters
Custom Render Texture
Sprite Atlas
Sprites
Tile
Animator Controller
Animation
Animator Override Controller
Avatar Mask
Timeline
Physic Material
Physics Material 2D
GUI Skin
Custom Font
Legacy
ScriptableObject

Then choose **Equipment Item**



Then new equipment item will be created, you must set its name to be unique (Not be the same as another items includes another types items)

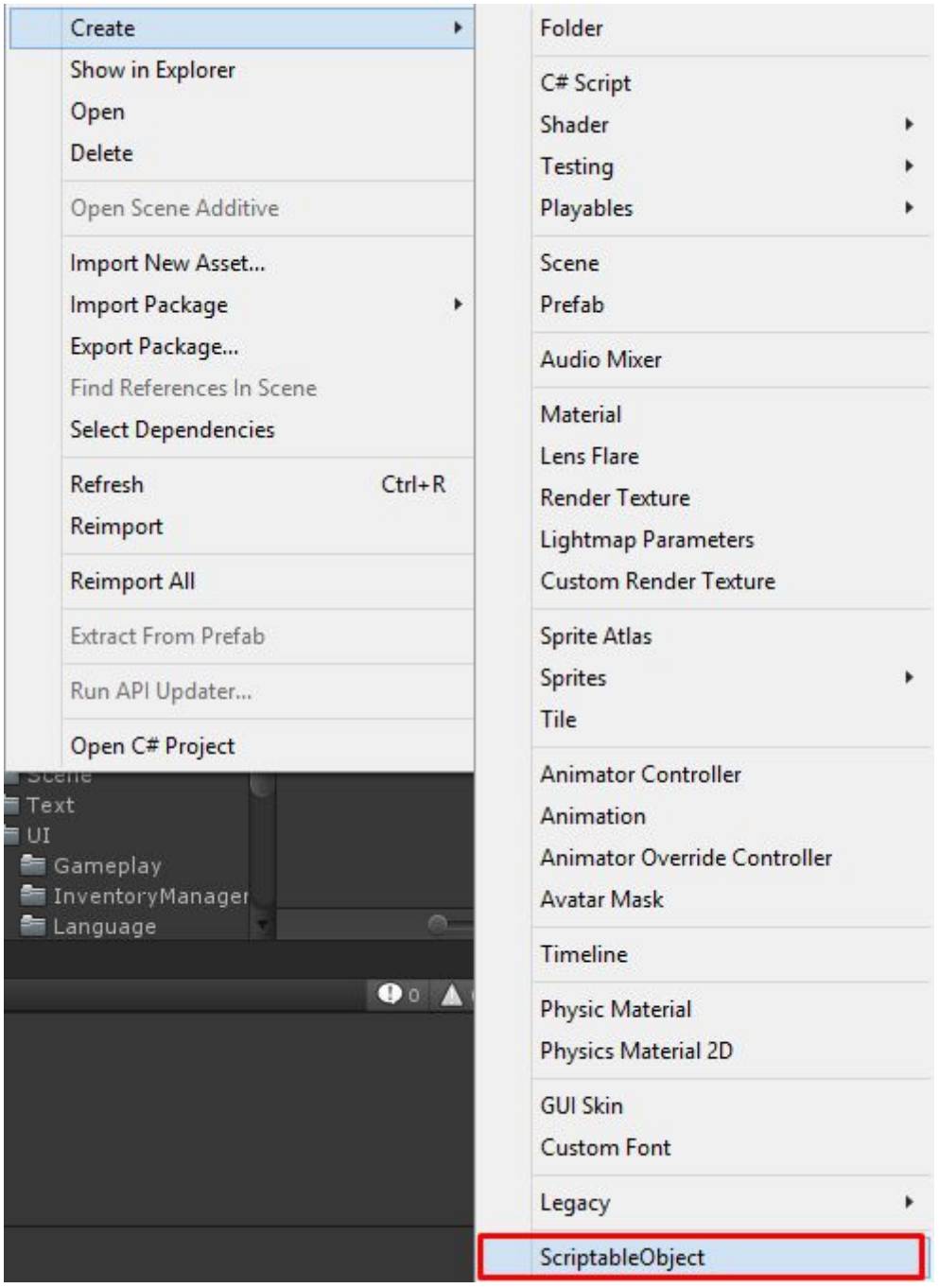
Then add it to your **Game Database**



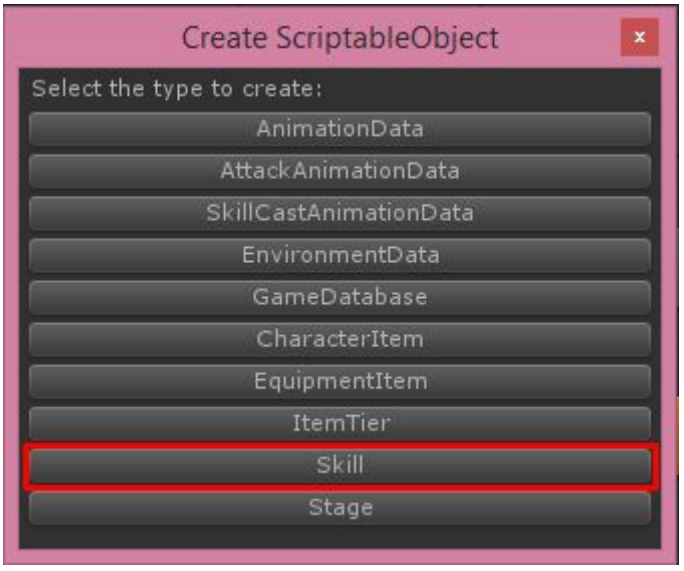
Skill

An characters can have skills, skill can be buff skill, nerf skill, attack skill or may mix all of them as you wish

You can create new skill by right click on **Project** tab then choose **Create -> ScriptableObject**



Then choose **Skill**



Then new skill will be created, you must set its name to be unique (Not be the same as another skills)

Each skill there are following configs

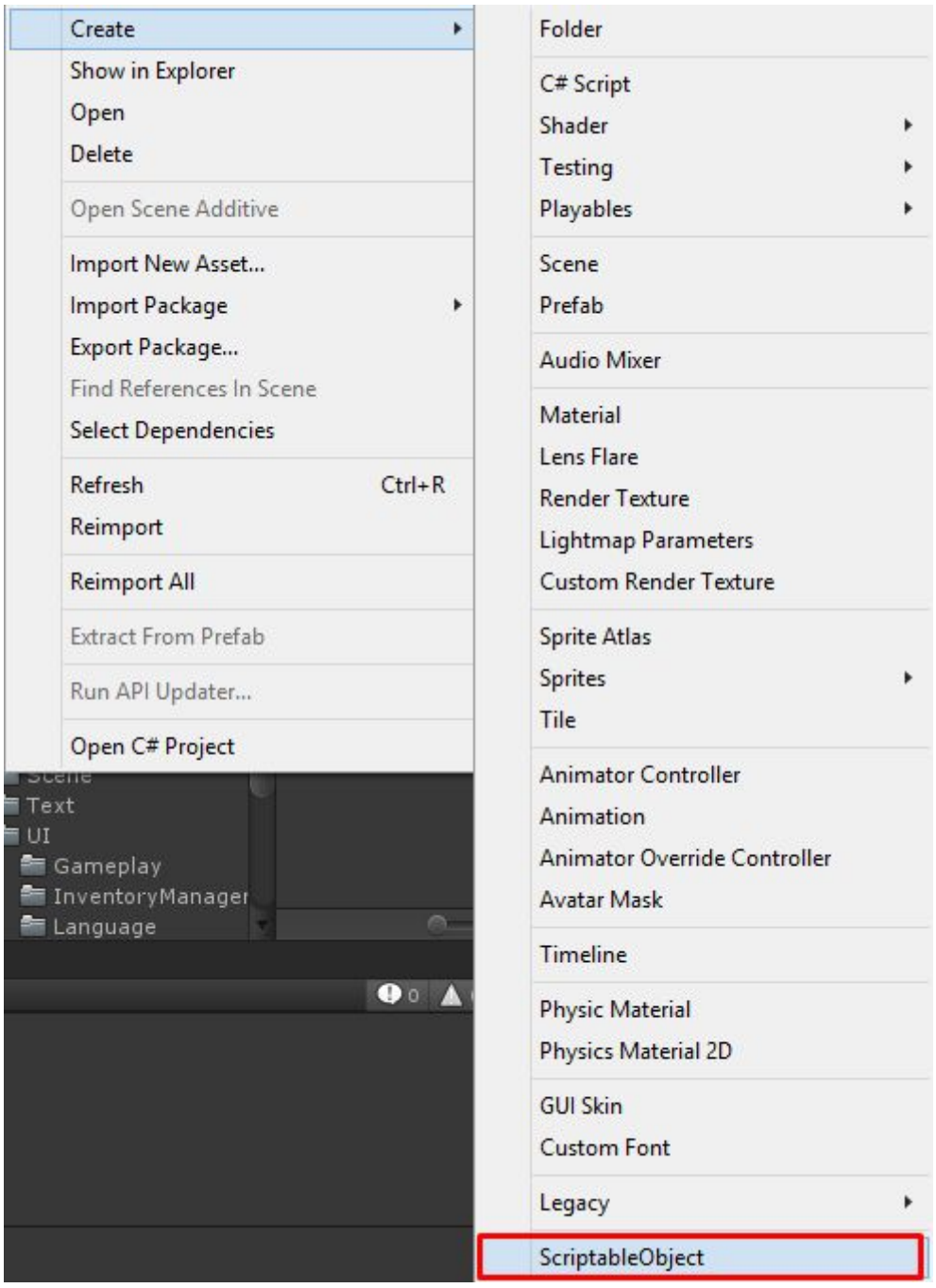
- **Title**, skill's title/name
- **Description**, skill's description
- **Icon**, skill's icon
- **Usage Scope**, target scope that play can choose to apply the skill
- **Cool Down Turns**, amount of turns that skill will be ready again after used
- **Attacks**, an attack data each attack you can set **Attack Scope**, scope of target that will receive damage. **Attack Animation** animation data when attacking see more at **Attack Animation** section. **Attack Damage** the damage when use skill will equals to `a.fixDamage` + ((`a.pAtkDamageRate` * `a.pAtk`) - `b.pDef`) + ((`a.mAtkDamageRate` * `a.mAtk`) - `b.mDef`) when a is attacker and b is the guy whom receive damage.
- **Buffs**, an buff data each buff you can set **Buff Scope**, scope of target that will receive buff. **Icon**, icon which will appears when buff applied. **Buff Effects**, effects that will instantiates when buff applied. **Type**, type of buff. **Apply Chance**, chance that buff will be applied. **Apply Turns**, how many turns that buff will be applied. **Clear Buffs**, amount of buffs that will be cleared randomly. **Clear Nerfs**, amount of nerfs that will be

cleared randomly. **Is Stun** if this is **True** character whom this buff applied to will stun. **Attributes** buff attributes for nerfs it should be lower than zero. **P Atk Heal Rate** amount of skill user's P Atk multiplier to heal to character which buff applied to. **M Atk Heal Rate** amount of skill user's M Atk multiplier to heal to character which buff applied to.

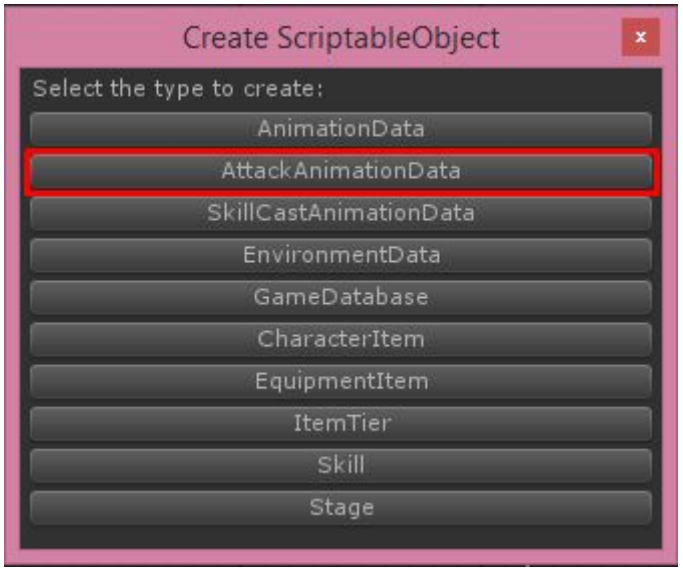
- **Cast Animation,** animation data when casting skill see more at **Skill Cast Animation** section

Attack Animation

An characters required Attack Animation, for an skills if the skill can attack it must set Attack Animation, You can create new attack animation by right click on **Project** tab then choose **Create -> ScriptableObject**



Then choose **Attack Animation Data**



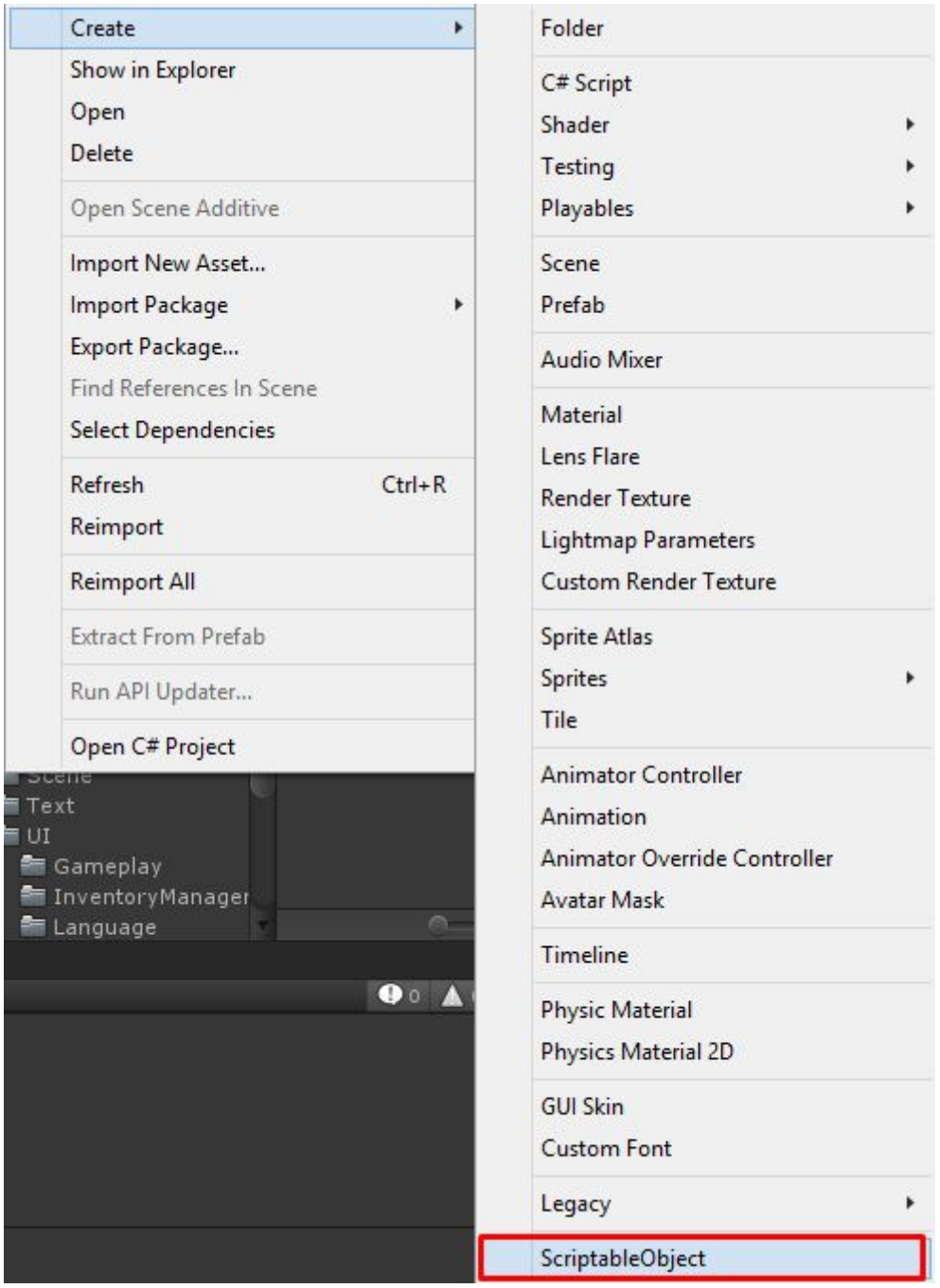
Then new attack animation will be created

Each attack animation there are following configs

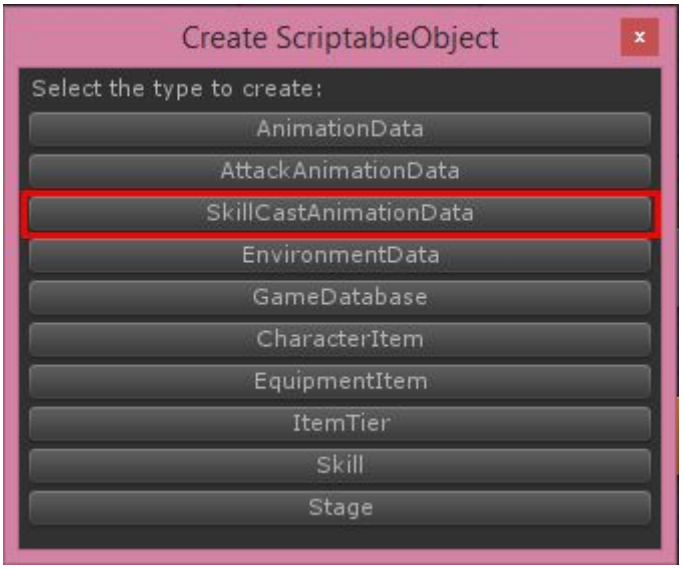
- **Animation Action State,** the number for Character Animator Controller's **Action State** condition
- **Animation Duration,** total duration of animation
- **Hit Duration,** duration to instantiates damages while playing animation
- **Is Range Attack,** if this is **True** character will not move to front of target to attack
- **Damage,** damage effects with how it appear/move in the game see more at Damage section

Skill Cast Animation

An skills required Cast Animation, You can create new attack animation by right click on **Project** tab then choose **Create -> ScriptableObject**



Then choose **Skill Cast Animation Data**



Then new skill cast animation will be created

Each skill cast animation there are following configs

- **Animation Action State,** the number for Character Animator Controller's **Action State** condition
- **Animation Duration,** total duration of animation
- **Cast At Map Center,** if this is **True** character will move to map center to cast skill
- **Cast Effects,** effects that will instantiates when casting skill

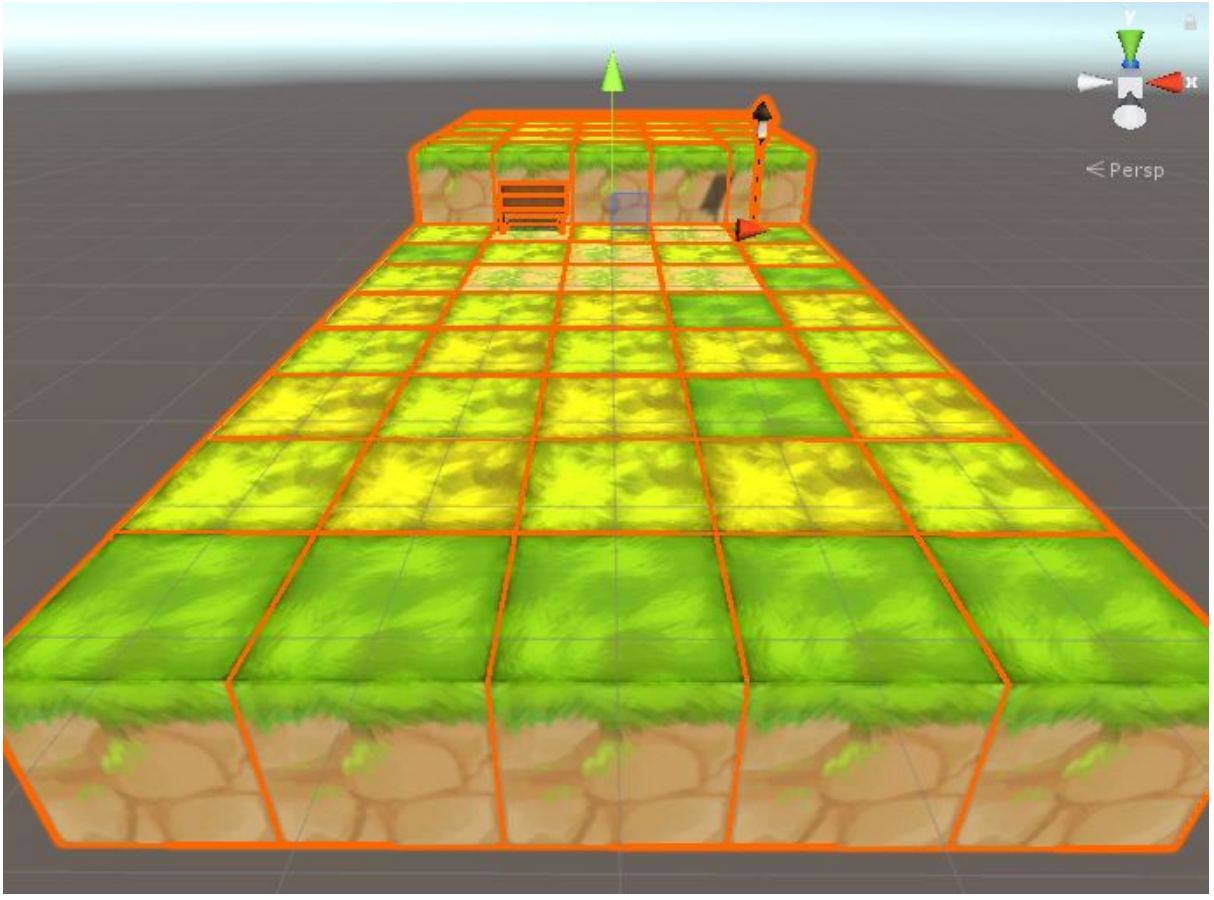
Environment Data

Environment data will store an **Environment Object** which will appears in Battle scene as background environment it will be set to **Stage** to determine how stage's environment will be

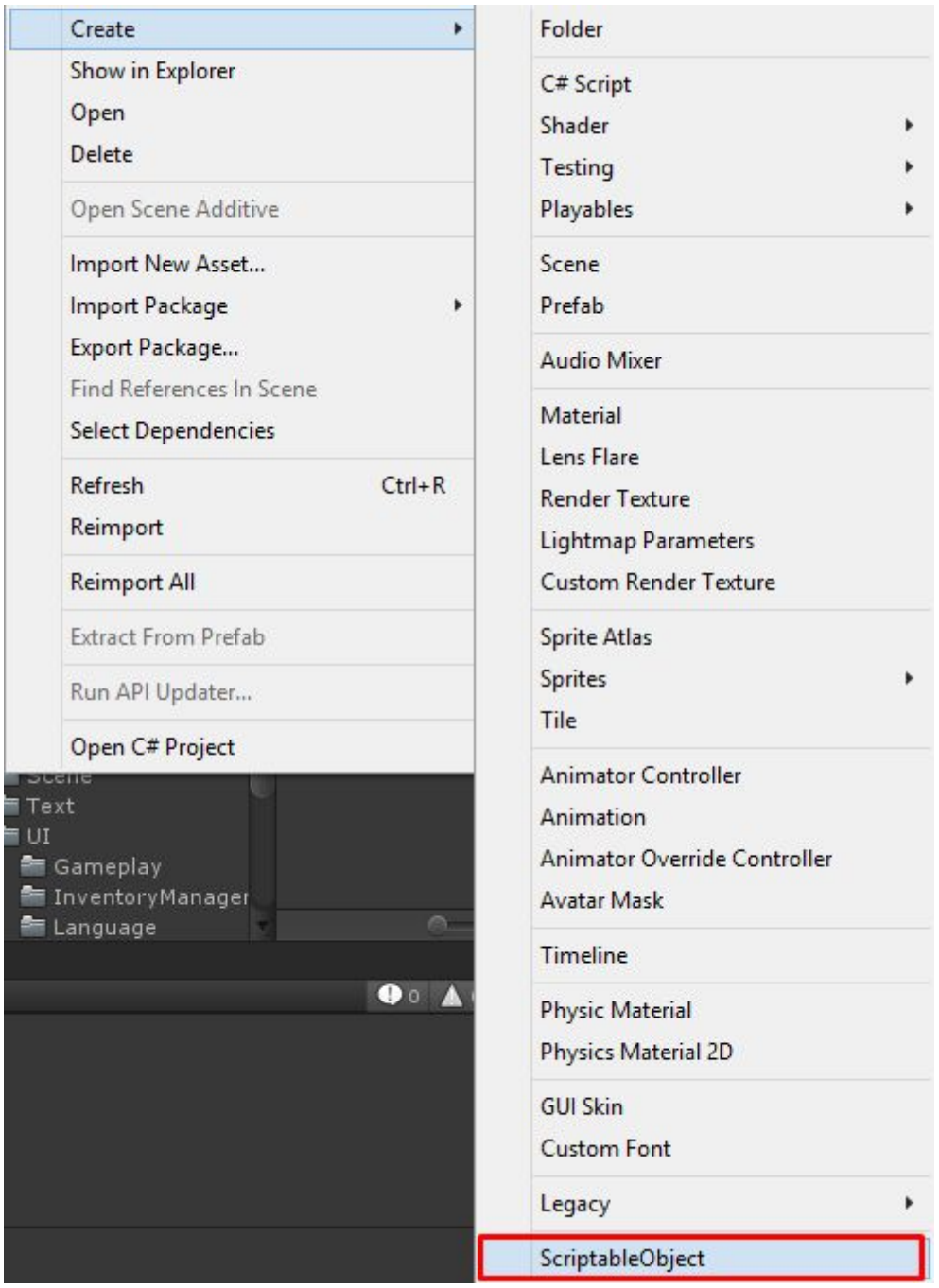
Before create environment data you should prepare an environment objects

To create **Environment Object** you just create empty gameobject then add **Environment Object** component

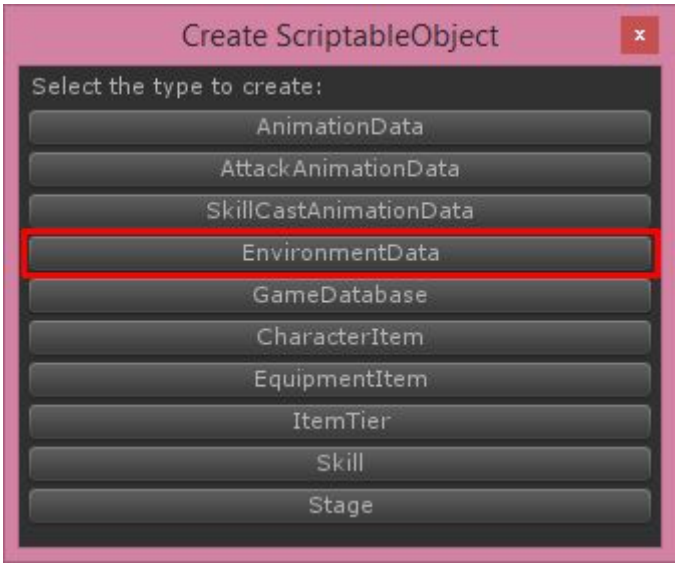
You have to create the environment then add it as children of **Environment Object** then adjust collider to fit with the environment and make it as prefab



Then create environment data, You can create environment data by right click on **Project** tab then choose **Create -> ScriptableObject**

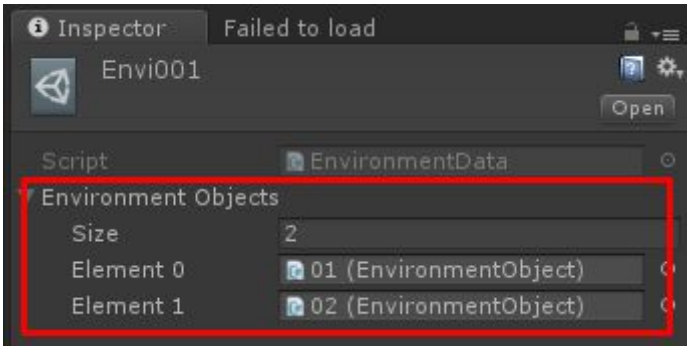

Create
Show in Explorer
Open
Delete
Open Scene Additive
Import New Asset...
Import Package
Export Package...
Find References In Scene
Select Dependencies
Refresh Ctrl+R
Reimport
Reimport All
Extract From Prefab
Run API Updater...
Open C# Project
Text
UI
Gameplay
InventoryManager
Language
Folder
C# Script
Shader
Testing
Playables
Scene
Prefab
Audio Mixer
Material
Lens Flare
Render Texture
Lightmap Parameters
Custom Render Texture
Sprite Atlas
Sprites
Tile
Animator Controller
Animation
Animator Override Controller
Avatar Mask
Timeline
Physic Material
Physics Material 2D
GUI Skin
Custom Font
Legacy
ScriptableObject

Then choose **Environment Data**



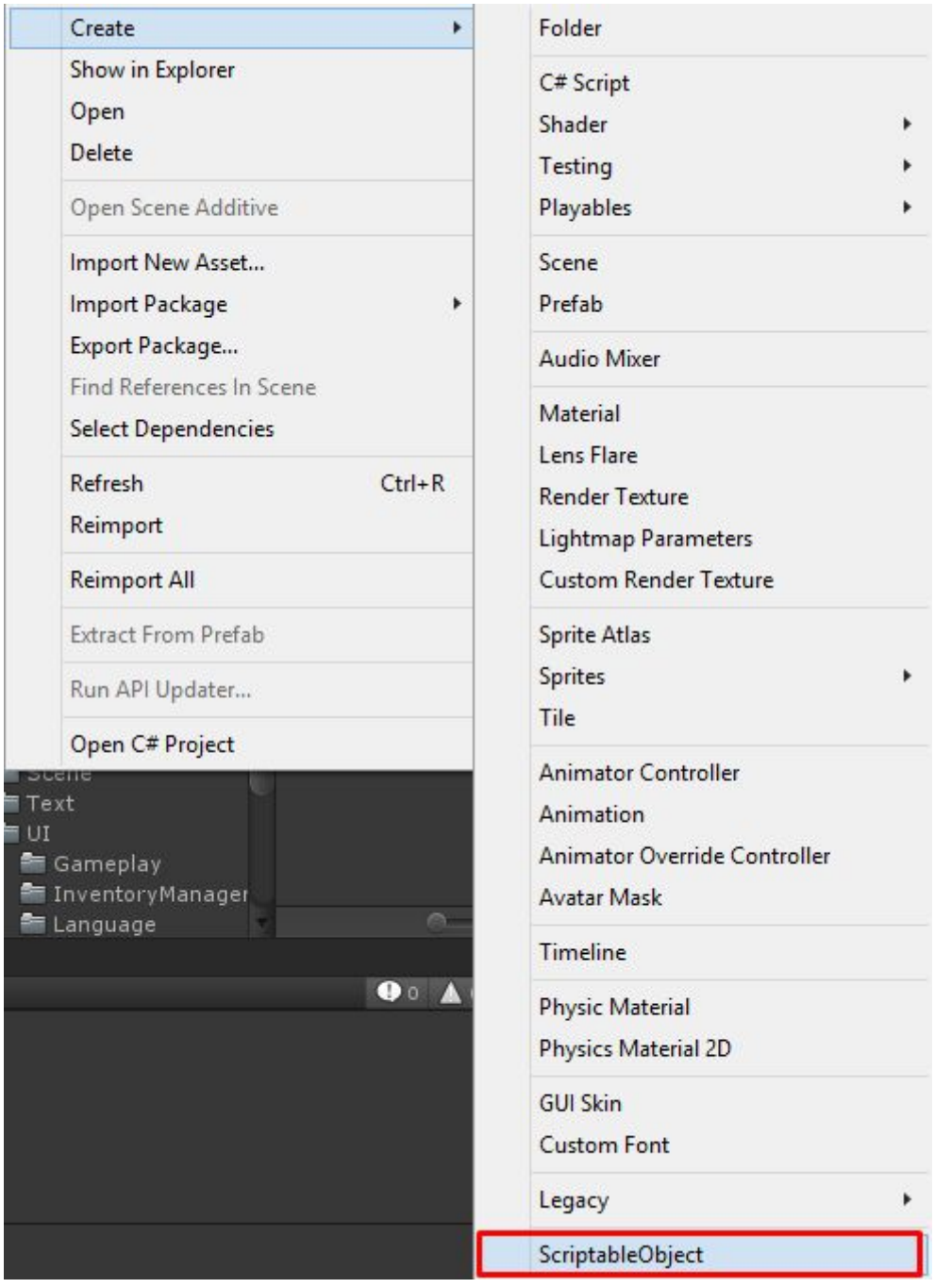
Then new environment data will be created

Then add an environment objects to field **Environment Objects**

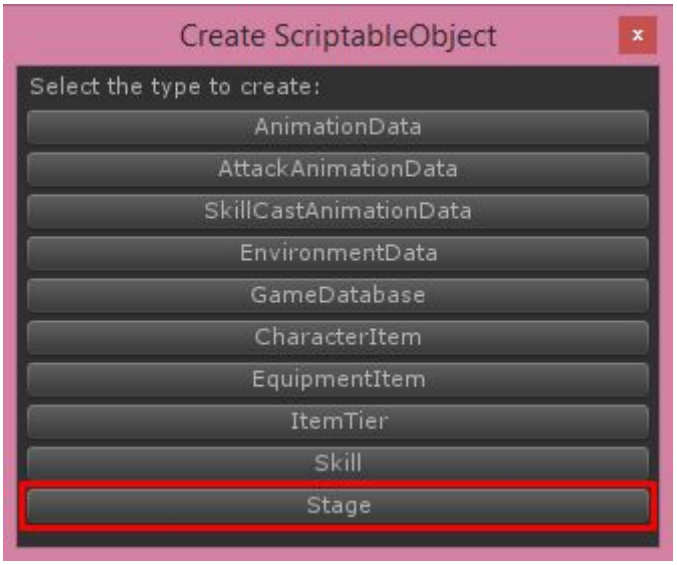


Stage

You can create stage by right click on **Project** tab then choose **Create -> ScriptableObject**



Then choose **Stage**



Then new stage will be created, you must set its name to be unique (Not be the same as another stages)

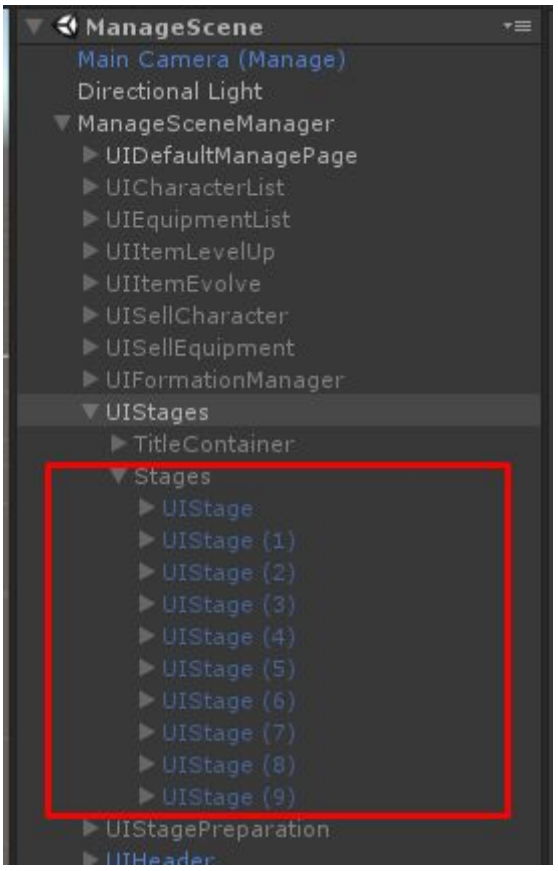
Each stage there are following configs

- **Title**, stage's title/name
- **Description**, stage's description
- **Stage Number**, the display number which you show on UI
- **Icon**, stage's icon
- **Environment**, environment that will be appears when playing the stage
- **Waves**, its length will be amount of waves when playing the stage each wave you can set: **Use Random Foes**, if this is **True** this will use foes data from **Random Foes**. **Foes**, Foe character with its level.
- **Random Foes**, Randoming foes for every waves when the wave is use random foes
- **Require Stamina**, require stamina to start playing the stage
- **Random Soft Currency Min Amount** and **Random Soft Currency Max Amount**, reward soft currency will random by min value to max
- **Reward Player Exp**, Amount of exp that player will receive when clear the stage
- **Reward Character Exp**, Amount of exp that character will receive when clear the stage
- **Reward Items**, item that player can receive when clear the stage

- **Unlock Stages,** stages that will be unlocked after clear the stage

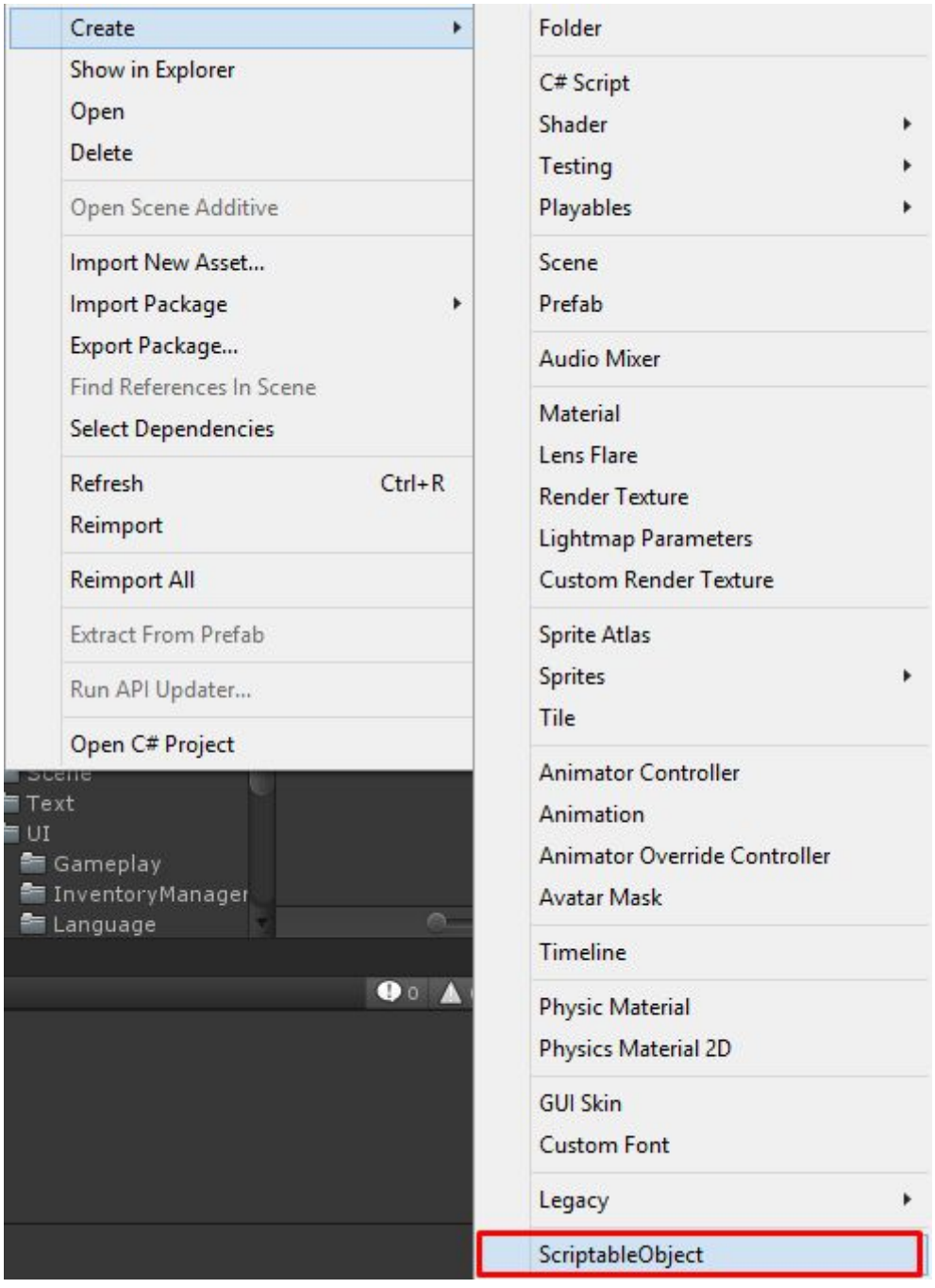
After prepare the stage you have to set it to stage selection UI to make it available to select at **ManageScene** which located at **/Assets/RPGGame/Demo/Scenes**

For the example **ManageScene** you can see how to add stage UI in **ManageSceneManager/UIStages/Stages**

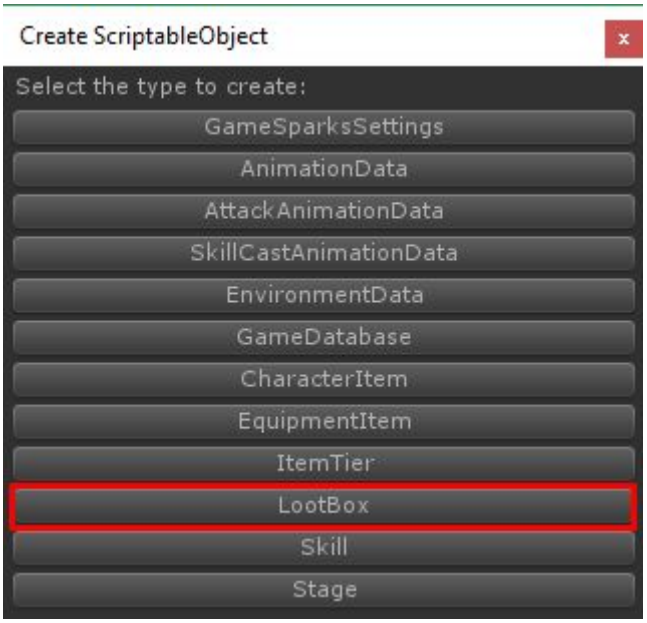


Loot Box

You can create stage by right click on **Project** tab then choose **Create -> ScriptableObject**

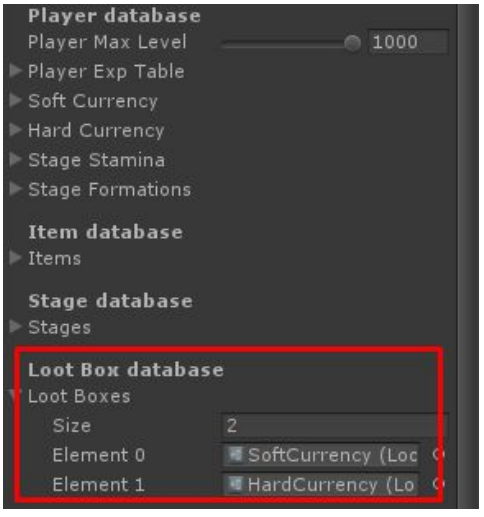


Then choose **LootBox**



Then new loot box will be created, you must set its name to be unique (Not be the same as another loot boxes)

Each loot box there are following configs

- **Title**, loot box's title/name
- **Description**, loot box's description
- **Icon**, loot box's icon
- **Requirement Type**, type of currency
- **Requirement Currency Amount**, amount of currency
- **Lootbox Rewards**, item that player can receive when open the loot box

Then add it to your **Game Database**