

# RPG Battle System

Contact :

[pondomaniac@outlook.com](mailto:pondomaniac@outlook.com)

Contact :

<http://pondomaniac.wix.com/pondomaniac>

## Table des matières

|  |                             |
|--|-----------------------------|
| Introduction:.....                         | 3                           |
| Unity 5 and 2d: .....                      | 3                           |
| Unity 3d Coordinate system and units:..... | 4                           |
| The organization of the project: .....     | 5                           |
| How the scenes are organized:.....         | 6                           |
| Behind the scenes:.....                    | 9                           |
| UI (User Interface): .....                 | 11                          |
| Sounds: .....                              | 11                          |
| Datas:.....                                | 13                          |
| Settings: .....                            | 13                          |
| Next step:.....                            | 13                          |
| Credits:.....                              | Erreur ! Signet non défini. |

# Introduction:

---

This is a complete RPG project. It's using the latest Unity features like Sprites, 2D features and more.

In this guide we will show you how to create your own RPG with your own assets and also we will expose some key features to help you personalize and enhance the game experience.

## Unity 5 and 2d:

---

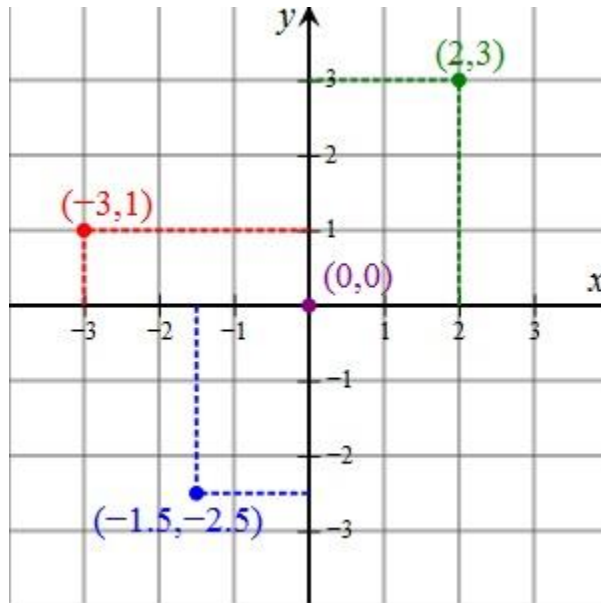
Unity 5 introduces some new 2d game creation tools, the main idea behind this is that creating a 2d game inside a 3d tools is not that simple, so the unity team introduced some new features to make it more easy: 2d view, Sprites, 2d collision, 2d physics...

For more information please refer to the following link :

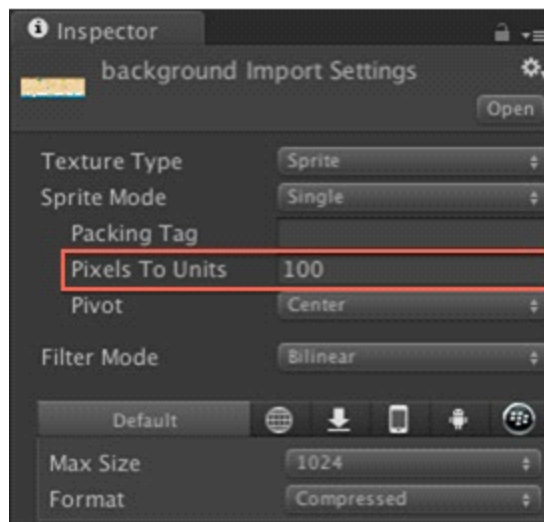
[unity3d.com/pages/2d-power](http://unity3d.com/pages/2d-power)

# Unity 3d Coordinate system and units:

The Unity coordinate system represents how Unity locate objects in the world, since we have make the choice of a 2d project we will focus just on the X and Y axis ,The ( 0,0 ) represent the center of the scene, each square represent a Unit.



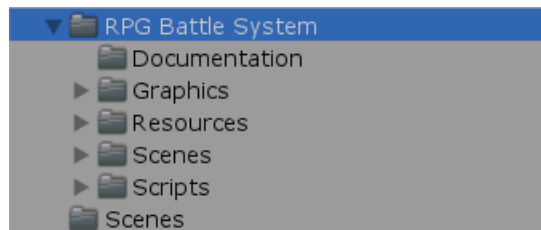
When you import a 2d asset you can define in the asset inspector how many pixels correspond to a unit, in the example bellow 100 pixels represent 1 unit:



# The organization of the project:

---

You can find below a print screen of the project Tree:



**Animations:** Contains all the animations used in the project.

**Documentation:** Contains the documentation of the project.

**GameObjects :** All the game objects used in the project.

**Musics :** All the static music used in the project ( there is also some sounds presents in the Resources folder ).

**Ressources:** It contains all the “Prefabs”, “Sounds” and “Sprites” used dynamically in the project.

**Scenes:** All the scene of the project.

**Scripts:** The project’s code.

**Sprites:** It contains all the project’s sprites.

# How the scenes are organized ?:

---

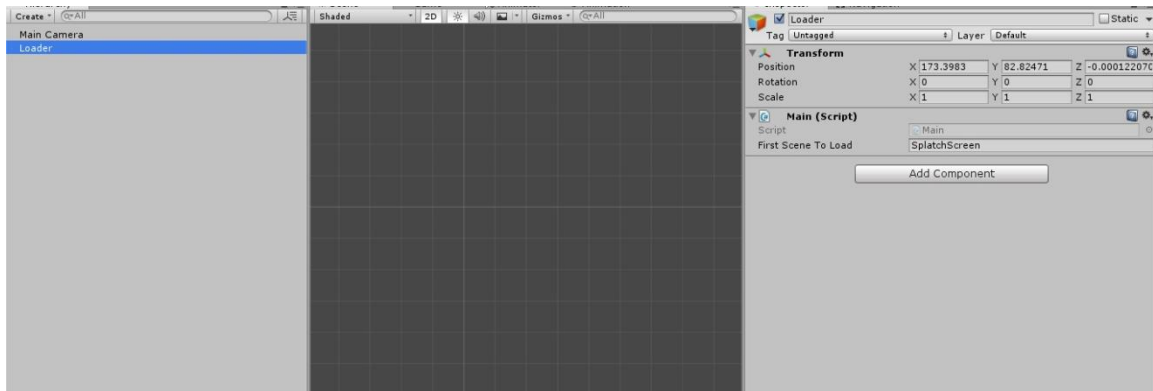
There are 6 types of scenes:

- Loader scene
- Splatch screen scene
- Main Menu Scene
- Character Name scene
- Exploration scenes
- Battle Scenes

We will describe each type of scenes and its main components:

## *Loader scene:*

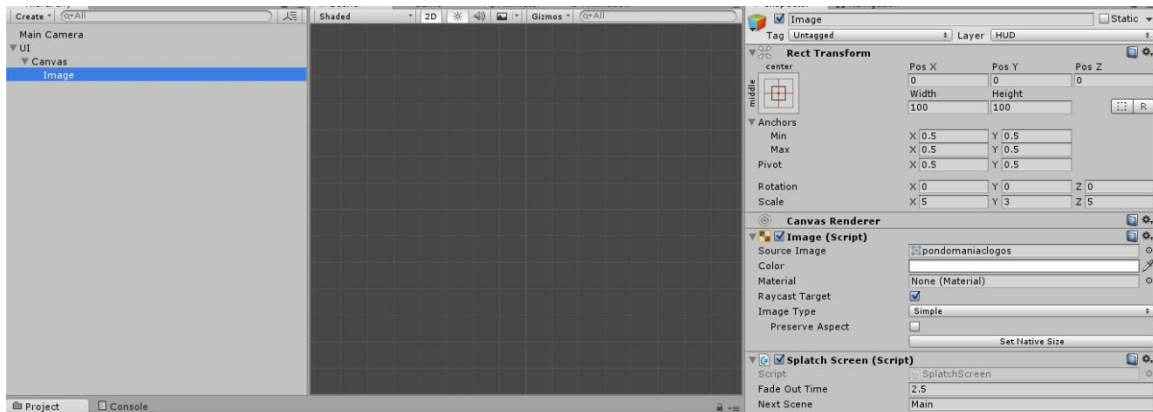
This is the first scene to load, it's an empty scene and it's used to load all the static objects that will be used and shared by the Game scenes. There is a "Main" script attached to the "Loader" game object, the "Main.cs" script contains for example all current characters, objects, current scene, and previous scene and more, you can open the script to have a better idea.



**First Scene to Load:** The scene that will be loaded when the game launch.

### *Splash screen scene:*

This scene contains the splatscreen logo; you can replace the image by your own logo by changing “Source Image” on the Image component.

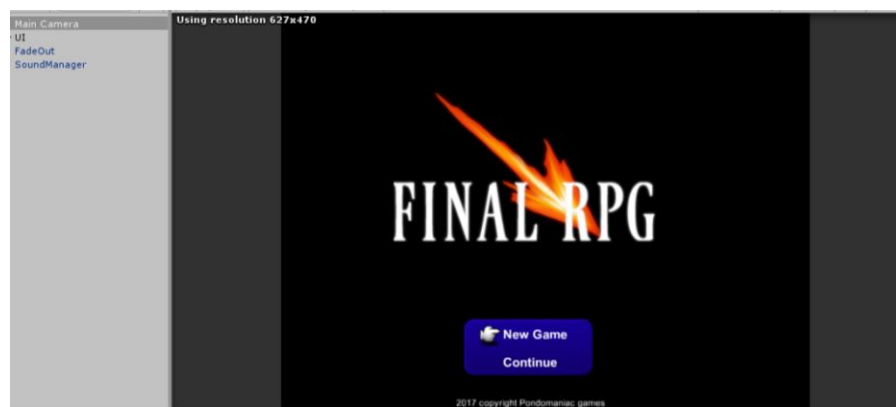


**Next Scene:** The next scene to load.

**Fade Out Time:** The duration of the animation.

### *Main Menu screen scene:*

This scene contains the main menu UI where you choose “New Game” or “Continue”, Here is a description of the objects the scene:



**Main Camera:** The orthogonal camera used by the scene.

**UI:** The buttons that redirect to different scene you can take a look at the script that will just load a new game or Load the last saved file.

**Sound Manager:** This game object plays the music of the scene and also contains several static functions that play different sound effects.

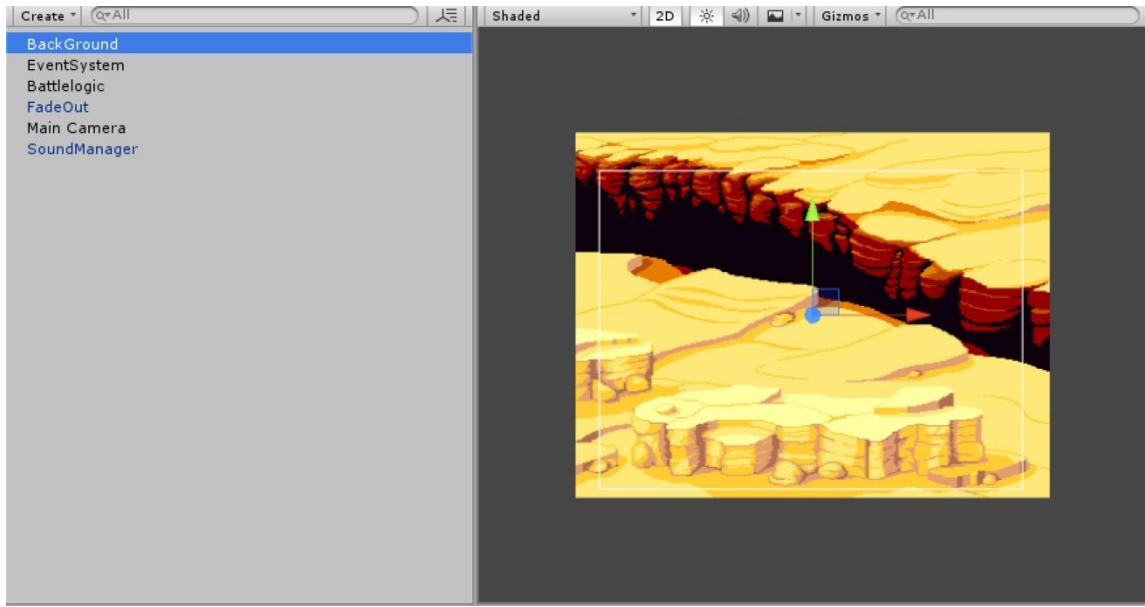
**Fade Out:** Play a fade out animation you can parameter the duration.

### *Battle scenes:*

This type of scene is used for battles, all battle scenes names begin with “Battle” prefix  
You can parameter the battle scene to generate a certain type of monsters and also the number of the generated monsters.







**BackGround:** The background of the battle scene.

**Event System:** This game object catches all the events of the scene.

**Battlelogic:** The game logic of the battle scene.

**Fade Out:** Play a fade out animation, you can parameter the duration.

**Main Camera:** The orthogonal camera used by the scene.

**Sound Manager:** This game object plays the music of the scene and also contains several static functions that play different sound effects.

## Behind the scenes:

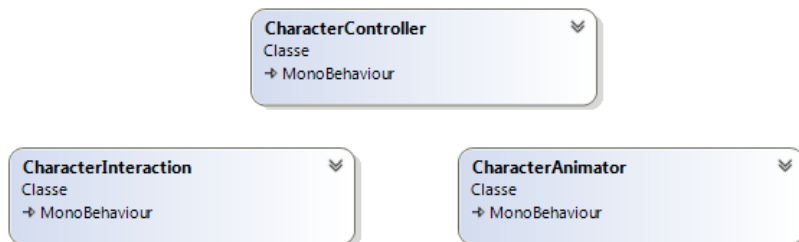
---

### *Player:*

We will describe in this section the player Game Object and how it moves, gets animated and interacts with other Game Objects, it's the same concept for NPC character that we will cover later :



Tree components are used to move and animate the character, we invite you to take a look at the parameters of each script:



**CharacterController** : it coordinate between character movement and animation

**CharacterAnimator** : Contains a dictionary of all characters movements, and play the corresponding animation base on the Animator component that is attached to the game object. Here is tutorial to make your own animation from your tileset (image that contains all 2d animations):

<https://unity3d.com/learn/tutorials/projects/2d-roguelike-tutorial/player-and-enemy-animations>

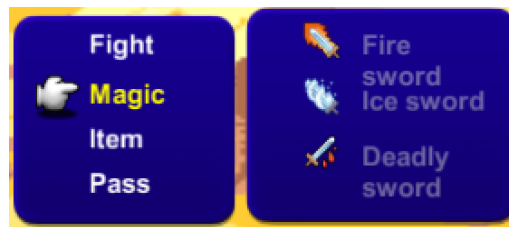
**CharacterInteraction** : Contains movement, Chat and market interaction, you can change the movement speed and also the smoothness

# UI (User Interface):

---

## *Battle Menu:*

The battle menu that is called “BattleUI” is like the GameMenu but contains different information and action like Weapon to use, Spell to launch, item to use in battle you can take a look at the script corresponding in Scripts/UI/BattleUI.



# Sounds:

---

The sound is controlled by the SoundManager present on every scene, if the music is the same as the previous scene the music will continue to play even if the scene change, if you want to restart the music check “IgnorePreviousSceneMusic”. The SoundManager class contains also different useful static functions to play different sounds, please take a look if you want to understand more.



# Datas:

---

The Datas in the game are static, it means that we don't use a database the datas are stored inside a file "Datas.cs" with fixed values, the main reason behind this is to make the game more compatible with different platforms, for example in the file you have the following:

```
var spell = new SpellsData();
spell.Name = "Fire ball";
spell.Description = "Send a fireball on your ennemies";
spell.AllowedCharacterType = EnumCharacterType.Wizard;
spell.PicturesName = "S_Fire03";
spell.ManaAmount = 5;
spell.Attack = 10;
spell.ParticleEffect = "FireBall";
spell.SoundEffect = "foom_0";
SpellsData[1]= spell;
```

As you can see the datas are written inside the code, for ParticleEffect, SoundEffect,picturesName... You can store the following datas :

**SpellsData** : For spells and magic

**ItemsData** : For equipment and potions

**CharactersData** : For the characters in the team

# Settings:

---

By setting we mean the general setting of the game used by the code, for example you will find the path to different folders,musics, speed of dialogs, dialog max characters, directional keys to move the player, the name of the saved file ... we invite you to take a look at "Settings.cs" file to see all the general settings.

# Next step:

---

This tutorial shows you just the main mechanism of the game, we invite you to open the project change the sprite and assets with your own assets and make your own game.

Don't forget to contact us at: [pondomaniac@outlook.com](mailto:pondomaniac@outlook.com)

If you need some help, think that something is missing or need a new feature.

Thank you!