

RPG Map Framework

Creative Spore

Support: creativespore@gmail.com

Web: <http://www.creativespore.com>

Unity Forums: user CreativeSpore

Index

1 Introduction.....	3
2 Character Animation Editor.....	3
3 Prefabs.....	7
3.1 Player.....	7
3.2 PlayerTouch.....	7
3.3 Player Camera.....	7
3.4 Boat & Ballon.....	7
3.5 EnemyGen.....	7
3.6 ZombiePlay.....	7
3.7 Teleporter.....	7
4 Autonomous Agents.....	8
5 Path Finding.....	8
6 RPG Map Helper.....	8

1 Introduction

This is the user manual for the Unity asset RPG Map Editor.

This manual will teach you how to use all the framework included with the asset.

This will also include some of the script found in the "Samples\Scripts" folder.

2 Character Animation Editor

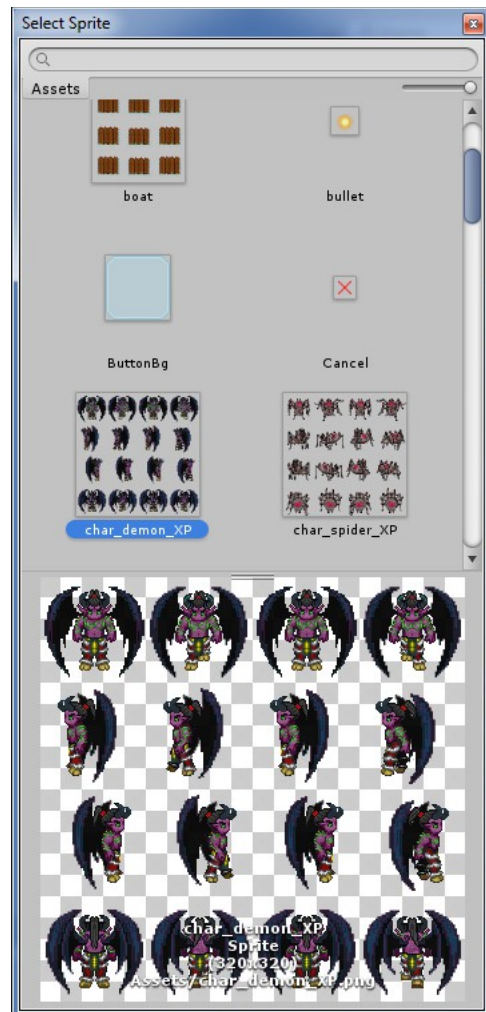
The **Character Animation Editor** is an editor window opened through the inspector view of a **CharAnimationController**.

You will be able to edit and create the character animation using an Editor Window to make it easy to change parameters like animation speed or the sprite pivot and create enemies, NPCs or the player faster and easier.

The new inspector of the **CharAnimationController** looks like this:



From the inspector, you can change some parameters of the animation like the **SpriteCharSet**. It has to be an sprite with all frames of the animation and should be a character sprite sheet compatible with RPG Maker XP or VX characters like these ones:



If you change the SpriteCharSet remember to change also the **Charset Type** to the right type of charset.

The **Target Sprite Renderer** will be the Sprite Renderer where the animation will be rendered.

Anim Speed is the speed of the animation in frames per second.

Ping-Pong Anim change the order of animation frames. There are always 3 frames per walking animation: frame 0, 1 & 2. If Ping-Pong is activated, the animation will be like 0,1,2,1,0 in a loop. Otherwise it will be

0,1,2,0,1,2. Some animations are thought to be played in ping-pong, some are not.

Facing Dir will change the character facing direction.

You can changed most of the parameters here, in the inspector window, but if you want to access more advanced parameters, you can access the Character Animation Window by pressing the **Open Editor...** button.





Here, you can change the **View Mode** to see the character animated in different configurations.

One of the view modes, **Edit Pivot**, allow you to change the pivot of the character for each direction. Most of the cases are good using the default pivot (0.5, 0) but in some cases it's useful to change the pivot.

You can change the **grid options** as well and **drag the mouse** over the grid to move the character and use the **mouse wheel to zoom** in and out.

3 Prefabs

3.1 Player

Manage the control of the main character, allowing moving it in all directions and shooting in 4 directions.

3.2 PlayerTouch

Manage the control of the main character but moving it by clicking over the ground. If the clicked position is reached, it will find a way to move to the destination.

This is used for controlling the player in a touch device.

3.3 Player Camera

Manage the control of the pixel perfect camera following the player.

3.4 Boat & Ballon

Manage the different vehicles included in the asset.

The player can drive any of them standing close and pressing Enter key.

3.5 EnemyGen

Manage an object generator to create a number of entities and create more when they are destroyed.

This prefab is used to spawn enemies.

3.6 ZombiePlay

Manage an enemy with a simple AI using path finding to reach the player.

Once reached, it's moving around him.

3.7 Teleporter

Manage a transition area to move the player to a destination scene and target teleporter or other game object set by name.

4 Autonomous Agents

Manage the AI moving behaviour like steering, seeking, following a path, etc,

Check "**Scripts\AutonomousAgents\MovingBehaviour**".

5 Path Finding

The path finding solution included with the asset can be found in the "**Script\MapPathFinding**" folder and contains the following scripts:

- PathFinding: contains the basics of the path finding implementation derived from the other scripts.
- MapPathFinding: contains the tile map specific implementation of the path finding used by find a path in the tilemap having into account the tile collisions.
- MapPathFindingBehaviour: is the Monobehaviour used by the entities to find a path to a target tile or position and it's used as a part of the enemy AI to reach the player.

6 RPG Map Helper

If you need help working with the tilemap by code, this will help you to solve some of the common problems.

In the "**Script\AutoTileMap**" folder you can find an script file **RpgMapHelper.cs**.

This class contains methods to help you getting or setting a tile from the tilemap, throwing an special ray to check if a tile with collisions is in the middle or modifying the fog of war, for example.