**Turn Based RPG with Scriptable Objects and Simple AI**

Ryan Tan

1800832F

P04

Table of Contents

**Cover Page1**

**Table Contents2**

**Main Menu 3**

**Camera Follow 5**

**PlayerController 7**

**PlayerData 11**

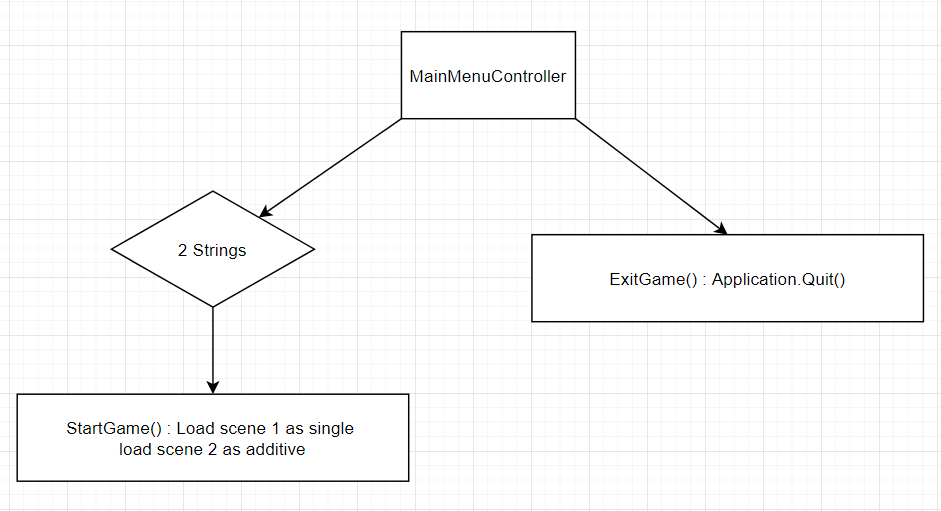
**UIManager 14**

MAIN MENU CONTROLLER

This script provides 2 methods for the 2 buttons on the start of the game. The Start Button and Exit Button. The Start button has a function attached the StartGame().

This method simple loads 2 scenes according the 2 strings that is assigned. 1 of the scene is responsible for the data, the second being a level. It loads the second scene using LoadSceneMode.Additive.

The Exit Button has a method called ExitGame(), which quits the application.



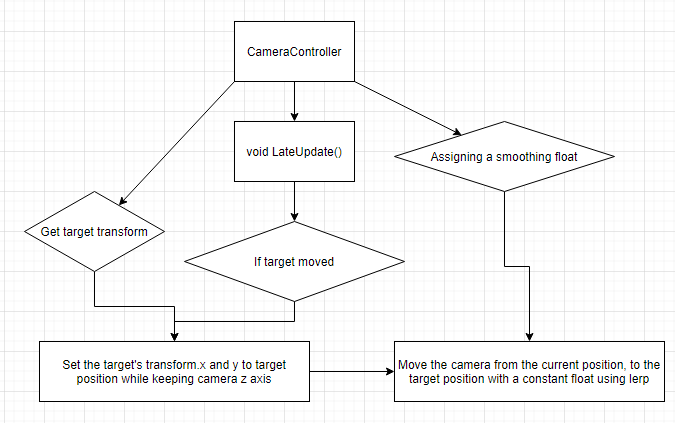
|  |  |
| --- | --- |
| Game object | Start Button |
| Script | MainMenuController() |
| Class | public class MainMenuController : MonoBehaviour |
| Method | public void StartGame() |
| Description | This method loads 2 scenes, the first scene is loaded using the LoadSceneMode.Single, the second is loaded as LoadSceneMode.Additive to add on top of the first scene. This is to help maintain a DataManagement scene that helps to maintain all the information for other scenes. |

|  |  |
| --- | --- |
| Game object | Exit Button |
| Script | MainMenuController() |
| Class | public class MainMenuController : MonoBehaviour |
| Method | public void ExitGame() |
| Description | This method is attached to the Exit Button. This method simple quits the application using Application.Quit(). |

CAMERA FOLLOWER

This method is in the CameraController.cs script, which is attached to the camera object. moves the camera to where the target referenced in the inspector moves, in this case the player. It gets the target position which is the XY axis of the target but for the Z axis, it uses the camera’s camera as moving it in the Z axis affects how the camera sees the game.

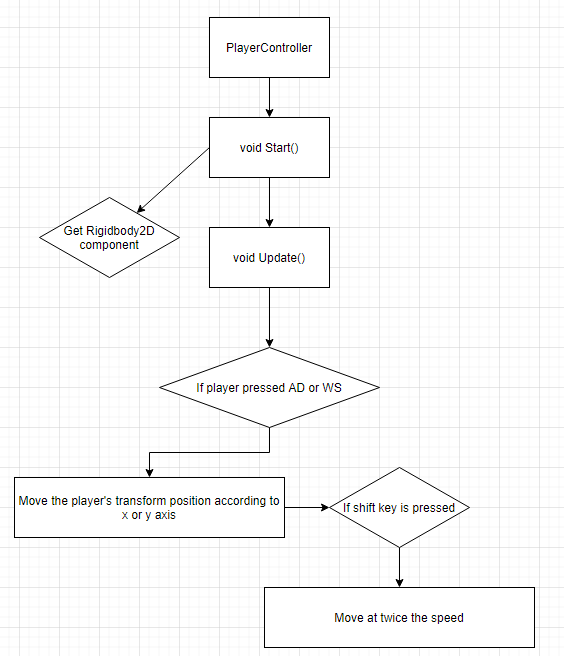
It then lerps the position of the camera and the target according to a smoothing float, to give it a smooth follow. Lerping adjusts the camera towards the target direction slowly, according to the float value in the parameter.



|  |  |
| --- | --- |
| Game object | Player and Camera |
| Script | CamerController.cs |
| Class | public class CameraController : MonoBehaviour |
| Method | void LateUpdate() |
| Description | The method gets the X and Y position of the target, in this case the player. It sets the target position to be the XY but the Z position will be the camera. It then lerps the position between the transform of the camera to the target’s position by a speed of the smoothing float. |

PLAYER CONTROLLER

The player movement is made a script called PlayerController.cs. This player controller is attached to the player object. It gets the Rigidbody2D of the player at the start. After which, it checks every frame if the player has pressed any horizontal and/or vertical keys (AS and/or WD). If it is, it moves the transform of the player accordingly in the XY axis. If further checks if the player held down the shift key. If it is, it moves the transform at twice the speed.



|  |  |
| --- | --- |
| Game object | Player |
| Script | PlayerController.cs |
| Class | public class PlayerController : MonoBehaviour |
| Method | void Start() |
| Description | This class is responsible for primarily moving the character around smoothly overtime via moving the rigidbody.  The speed, Rigidbody2D, a Vector3 called change and a private bool inventory is referenced before the start. The speed moves the rigidbody a set direction. The Vector3 change calls the movement whenever the player moves beyond the change Vector. The bool inventory is to give the player an inventory using the Inventory script later on in the documentation.  The Start method helps the script get the Rigidbody2D by getting the Rigidbody2D component attached to the player object. |

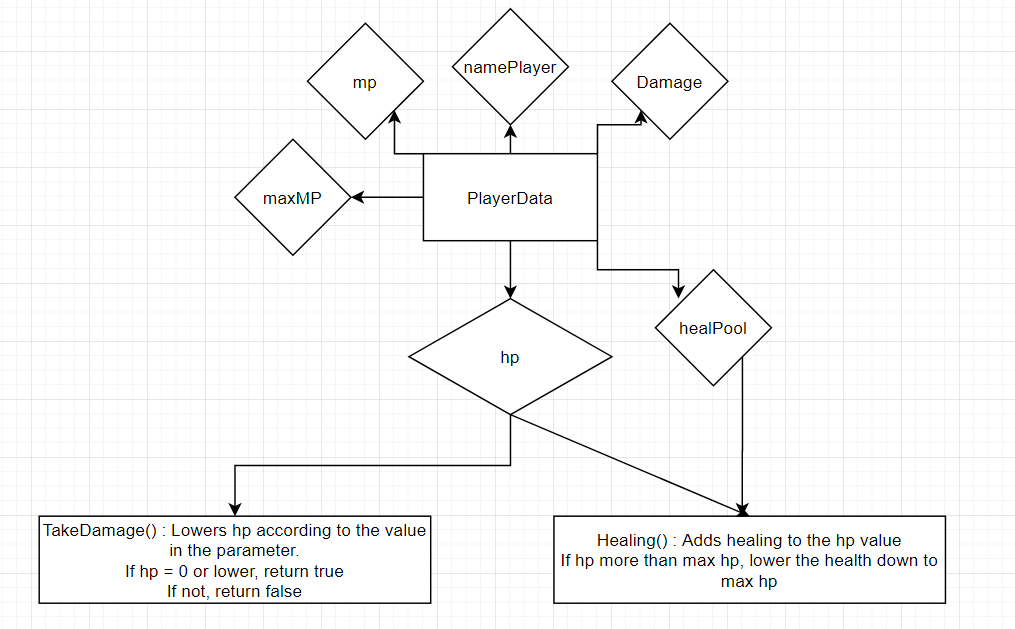
|  |  |
| --- | --- |
| Game object | Player |
| Script | PlayerController.cs |
| Class | public class PlayerController : MonoBehaviour |
| Method | void Update() |
| Description | This method helps to check if the player has pressed any input keys, horizontal (A and D) and/or vertical (W and D). It also checks if the change in position is zero or not. If it is not zero, it calls for the MoveCharacter method. |

|  |  |
| --- | --- |
| Game object | Player |
| Script | PlayerController.cs |
| Class | public class PlayerController : MonoBehaviour |
| Method | void MoveCharacter() |
| Description | This method moves the transform position of the player by any change, the change being the change in x or y axis depending on the input key, by a set distance and speed over time.  If also checks if the player is pressing shift. If the player is holding down the shift key, it moves at double the speed. |

PLAYER DATA

This script contains all the data the game needs to process for battle and Heads Up Display(HUD). It contains the maxHP, maxMP, hp, mp, namePlayer, damage and healPool.

There are also 2 methods, TakeDamage and healing, they are primarily used in battle. TakeDamage reduces the hp value of the player. It further checks if the player’s hp is 0. If it is, it returns true for another bool in another script. If not, it returns false. For healing, it adds the hp according to the heal pool and checks if the hp is more than the maxHP. If it is more, it lowers the hp back down to the maxHP.

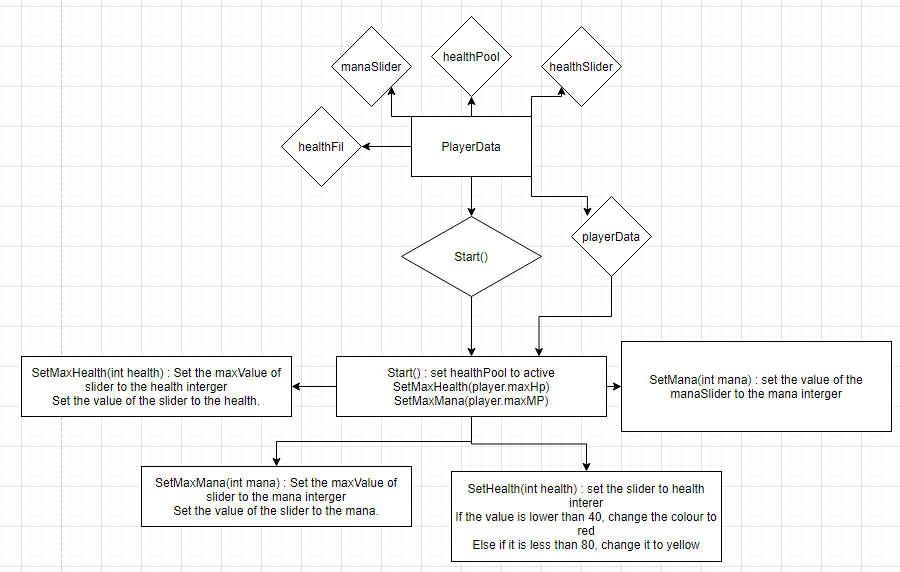


|  |  |
| --- | --- |
| Game object | Player |
| Script | PlayerData.cs |
| Class | public class PlayerData : MonoBehaviour |
| Method | public bool TakeDamage(int dmg) |
| Description | This method lowers the hp of the player according to the damage integer in the parameter. It then checks if the player’s hp is 0. If it is 0, it returns true. If not, it returns false  This bool will be used in another script for battle |

|  |  |
| --- | --- |
| Game object | Player |
| Script | PlayerData.cs |
| Class | public class PlayerData : MonoBehaviour |
| Method | public bool healing(int dmg) |
| Description | This method adds the healPool integer of the player to the hp. It then checks if the hp is more than the maxHP. If it is, it reduces the hp back down to maxHP. |

UI MANAGER

This script is in charge of managing the UI when the player is in the overworld. The UI comprises of several elements, the player’s name, hp and mp bars. The hp and mp bars are made with 2 images each, 1 for the border and 1 for filling up the border. The UI manager adjusts the fill image for the hp and mp, as well as the name, according to the player data. It adjusts all these data by referencing it in the scripts, it also references the player data to follow accordingly.



|  |  |
| --- | --- |
| Game object | UIM |
| Script | UIManager.cs |
| Class | public class UIManager : MonoBehaviour |
| Method | public void Start() |
| Description | This method activates the panel for the name, health and mana bar, followed by setting the value of the health and mana bar using a SetMaxHealth and SetMaxMana method. |

|  |  |
| --- | --- |
| Game object | UIM |
| Script | UIManager.cs |
| Class | public class UIManager : MonoBehaviour |
| Method | public void SetMaxHealth() |
| Description | This method sets the maximum value and current value of the slider of the health bar to be the maxHp of the player according to the player data |

|  |  |
| --- | --- |
| Game object | UIM |
| Script | UIManager.cs |
| Class | public class UIManager : MonoBehaviour |
| Method | public void SetMaxMana() |
| Description | Similar to SetMaxHealth(), this method changes the maximum value and current value of the mana bar to be the maxMP of the player data. |

|  |  |
| --- | --- |
| Game object | UIM |
| Script | UIManager.cs |
| Class | public class UIManager : MonoBehaviour |
| Method | public void SetHealth(int health) |
| Description | This method is called whenever there is a change in health. It is not placed in Update() as it will take up some performance doing so. This method sets the value of the slider to the current hp value in the parameter. It then checks if the health is below 40. If it is, it changes the image of the fill image to red. Else it will check if the health is lesser than 80. If it is, change the image to orange. |

|  |  |
| --- | --- |
| Game object | UIM |
| Script | UIManager.cs |
| Class | public class UIManager : MonoBehaviour |
| Method | public void SetMana(int mana) |
| Description | This method changes the value of the mp slider to the mana integer in the parameter. |