Edge Hill University

Games Engine

Coursework 2

Author:
Igor Miranda

December 17, 2014

Edge Hill University

1 Introduction

This assignment consists of a mini project in which the students will utilise the skills developed through the creation of the portfolio to implement a software solution which makes use of a number of features of a games engine. The implemented solution may be a game on an available platform (e.g. PC, console or mobile app), or may be an individual implementation of a mini games engine itself. The outcome of the project will evidence the range of skills which the individual student has developed through the module.

2 Background

In this coursework we produce a simple puzzle game, wad tried to use everything learned in class and another features such as database. The game is very small and simple, nothing that can be sold to the public but its a begin. Was tried to build all the scripts, the project dont have any script of another author but some tutorial blog ,videos and the unity community helped in the construction of some of the scripts. The all models and materials were took from assets store.
As we had little time an architecture was not planed for the game but was tryed to keep it as modular as possible.

3 The Game

The inspiration for this game came from a game called Limbo from Playdead, a puzzle game that dont tell you what is happening you just keep going and let you create your story. Many puzzles was took from Limbo. The plot of our game is an alian tring to escape from the humans but we just tell that in the end of the game, but the plot was not whar we focus, we focus more on the scripts.

3.1 Principal Menu

The Menu is the first thing that the player will see when open the game.
The principal menu has 5 buttons:

- New Game: Start a new game and erase the data from last game;

- Load: Load the player in the last scene he was after exit the game;
- Scenes:A new menu will appear with the scenes that he has passed, he can choose one to start;
- Credits:
- Exit:Quit the game(don't work inside unity);

3.2 Pause Menu

The pause menu is called when the player press Esc.
What the script does is set Time.timeScale = 0,so the game will pause, and show the pause menu ,when the player press the button Resume the PauseMenu script will set Time.timeScale = 1, so the game will run as real time.

3.2.1 Game Manager

It's responsable for:

- Initialize the Menu and disable it when the game start.
- Load the game, instantiating all the scenes ,player and the player camera.
- Reset the current scene that player is playing: when the player dies the GameManager delete and instantiate a new scene of the current scene the player is playing.

3.3 Scenes

The game has 4 scenes prefabs, when the game start all scenes are instantiate and when the player dies the scene which he is will be destroyed and a new one will be instatiate, that is necessary because when the player dies we need to reset all the scene to its initial state.In the begin of each scene has a checpoint that is the point where the player will spawn when die or when load the game or choose a scene.

3.4 Prefabs and Scripts

3.4.1 Player

The player prefab has 2 scripts, PlayerManager and PlayerController.

1. The PlayerManager it's responsable for know when the player is dead, set the position of the player(checkpoint),manage the animation and manage the collisions;

2. The PlayreController inherit PlayerManager and it's responsable for the controller of the player such as walk, run, jump, fall and interact. The player uses the CharacterController, a player with a rigidbody physic will be implemented in a future update;

3.4.2 Gun

The guns of the game has 1 script. The Gun script allow the guns to be fixed or a move gun, each gun has a different time of fire rate, over heating time, cooling time, movement speed if it's a move gun, distance of the movement, a start fire time and life, this is possible because all of this parameter can be set throw unity interface. Inside the gun there is a script for draw the lazer called DrawLine. This script uses a LineRender and a RayCast to know the distance of the LineRender.

3.4.3 Teleport

The teleport script is very simple, has the reference of the other teport so when the player is inside the teleport and press E he will move to the position of the other teleport, the box variable is for move the box with the player when the box in inside of the teleport area too, this variable is necessary because We don't know if it's possible to handle with a vector of collider in the method OnCollisionEnte.

3.4.4 Box

The Box has a tag CanInteract just to tell the PlayerController that is an object that he can interact with.

3.4.5 Check Point

The CheckPoint script is necessary just to save in the database the last checkpoint the player reach, well he load the game he will spanw in that checkpoint. When the player pass throw a chepoint the variable checkPoint-Number will be update with the number of the checkpoit the player just reach, the number is take from the name of the checkpoint(last caracter).

3.4.6 Buttons Systems

1. OnButton System
   For that system we have 2 scripts, one for the button called Button and other for the cube called OnButton, that system was used in the TRap System 1 and 2.

   - Button: Has a reference to OnButton and a virtual method ActionOnTrigger, when the player touch the button will call onButton.ActionOnTrigger.
   - OnButton: Inherit Button and implement ActionOnTrigger to set the cube gravite to true and kinematic to false.

2. Trigger System

The Trigger System was implemented after the OnButton System. We notice that the OnButton System cannot be used in other scripts without changing the script, so we implemented the Trigger System. This script recives a GameObject that will bi the object to activet the trigger, if it's null player will be set as default, when the gameobject it's in the trigger the variable state will be set to true if not will be set to false. To access the value of the variable state ouy need to call the stateTrigger(read only). The script which needs the button just need to add a variable Trigger and the reference of the button to it and check the stateTrigger(read only).

3.4.7 Elevator System

The elevator system has 2 "components" one is the elevator and the other are the contact points. The contact points are necessary to know when the elevator needs to stop and with that collision you can set the variables necessary like which door needs to be open and what is the next direction

the elevator will take when the player press E.
The elevator "component" has the script, when a object with tag CanInteract is inside of the box elevator it is insert into a list because when the player is inside the elevator and press E the player and all objects inside the list will move with the elevator. The elevator will move until detect a collision with a contact point, that contact point has a name related with its position (right, left up, down) a contact with one contact point will set the variables necessary for the next movement, each contact point has a tag to tell wich door needs to be open. The Elevator System has a Trigger variable so the player or other object can call or active the elevator.

3.4.8 Trap System 1

In this trap the player needs to pass without touch the fire, so he needs to press the red button to release the cube in the fire to pass throw the trap.

3.4.9 Trap System 2

That trap uses the same system of the Trap system 1, the Button is the horizontal cylinder and the cube OnButton.

3.4.10 Rope Box Trap

That trap has a Trigger variable that when is active release the box, when the box bit the end gameobject the box is release from the rope and the DieArea is disable.

3.5 Database

The database is very simple and was used the SQLite to manage it.
When the game begin is DatabaseCheck is called to check if the table exits, if not create it, and if the database does not exist create one too.
The database just do 2 commands:

1. Select: select the unique line that exists in the database that contains the last checkpoint;

2. Insert: delete all table and insert the new last checkpoint;

As was told the database is very simple, is just used to save the last checkpoint.

3.6 Tried

- Hope Platfor:Was tried to implement a platfor with a rope thatbalance with the player, a rope was implemented with hingejoints we that find a way to make the player fallow the movement of the platfor.
- Sters: Was tried to make the player climb sters, but we dont know why he dont keep the same position when the user is not pressing any key. So the sters system will be released in a future update.
- Rope System:Was tried to implement a rope system that the plaer could climb.

4 Bugs

1. Sometimes the collisions with the player are not detected.
2. The player movement is not so fluid, that because the PlayerControler script is not so hulled.
3. If the player has gravity he will not fallow the elevator.
4. The has some lag when the is running for a long time, I dont know if it's my computer or if is the game.

5 Conclusion

In this project we tried to use every thing that we learned in class and seach other features not covered in the class. The project submited has a begein and a end but still not a game that can be sold. is just a begening and we will continue the project.
Link of the project on GitHub: Project-Limbo