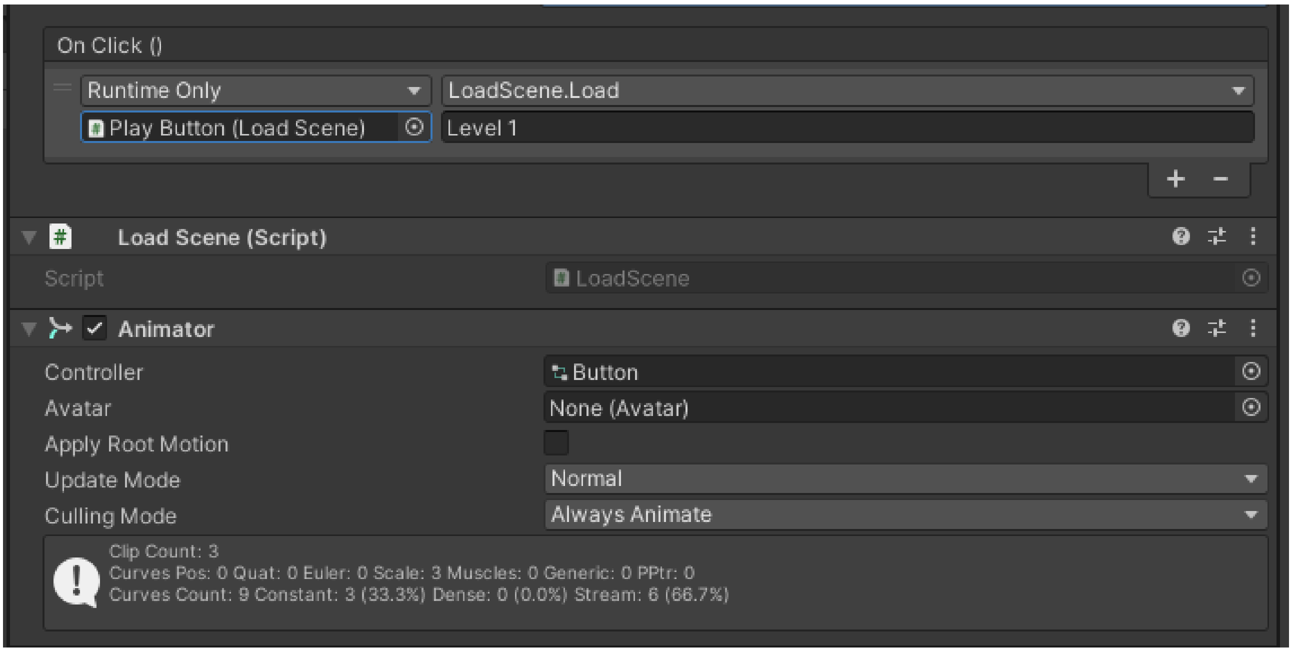
# Lost My Keys

## Main Menu



On Click ()

Runtime Only | LoadScene.Load

Play Button (Load Scene) | Level 1

Load Scene (Script)

Script: LoadScene

☑ Animator

Controller: Button

Avatar: None (Avatar)

Apply Root Motion: □

Update Mode: Normal

Culling Mode: Always Animate

Clip Count: 3
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 3 Muscles: 0 Generic: 0 PPtr: 0
Curves Count: 9 Constant: 3 (33.3%) Dense: 0 (0.0%) Stream: 6 (66.7%)

Attached to each main menu button you can find a LoadScene script and an animator attached.

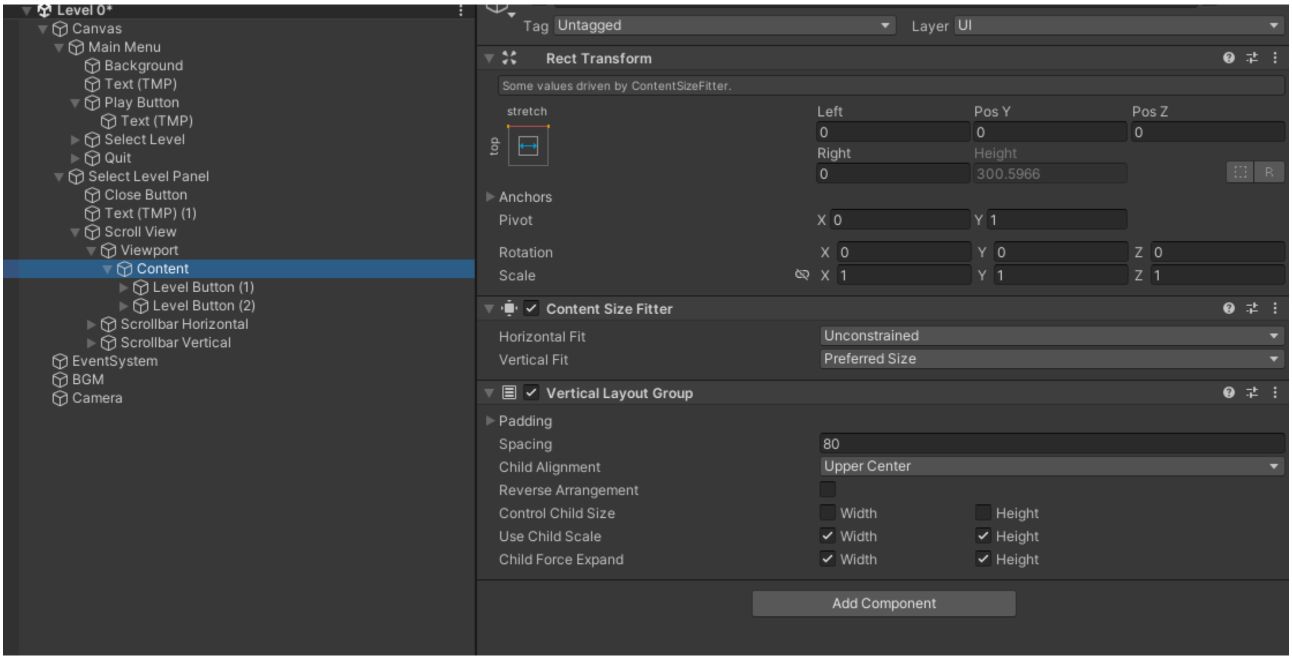
The animator is used to animate the button when hover with the mouse and when clicking.

The LoadScene Button contains few methods to load, reload and quit the game that are called from the OnClick event.

The OnClick event is also playing a sound.

You can find the tutorial for similar code in the Lesson 5.3 Game Over in the unity course.

## Level Select



Tag Untagged Layer UI

Rect Transform

Some values driven by ContentSizeFitter.

stretch top

Left 0 Pos Y 0 Pos Z 0

Right 0 Height 300.5966

Anchors

Pivot X 0 Y 1

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Content Size Fitter

Horizontal Fit Unconstrained

Vertical Fit Preferred Size

☑ Vertical Layout Group

Padding

Spacing 80

Child Alignment Upper Center

Reverse Arrangement □

Control Child Size □ Width □ Height

Use Child Scale ☑ Width ☑ Height

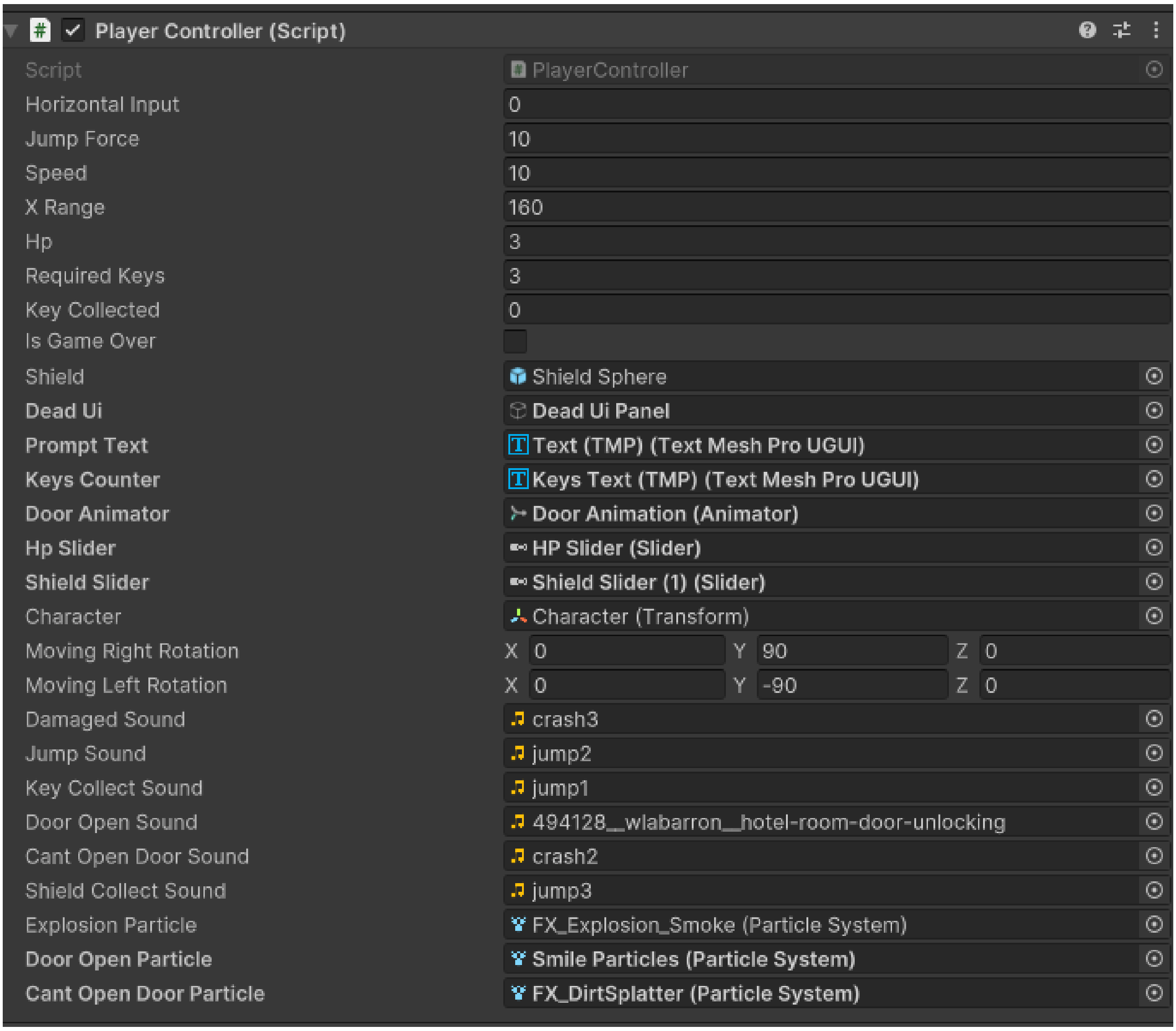
Child Force Expand ☑ Width ☑ Height

In the level select I’ve added a scroll view to fulfill the assignment requirements of having a “slider on a panel”.

The scroll view is a basic unity component, I didn’t find it used anywhere in the course, you can find more info here [Scroll Rect | Unity UI | 2.0.0 (unity3d.com)]

In the content on the scroll rect attached there is the Control Size Fitter component, used to change the scroll rect size based on the element inside, and the vertical layout group that is used to align the elements vertically.

# Player Controller



Player Controller (Script)

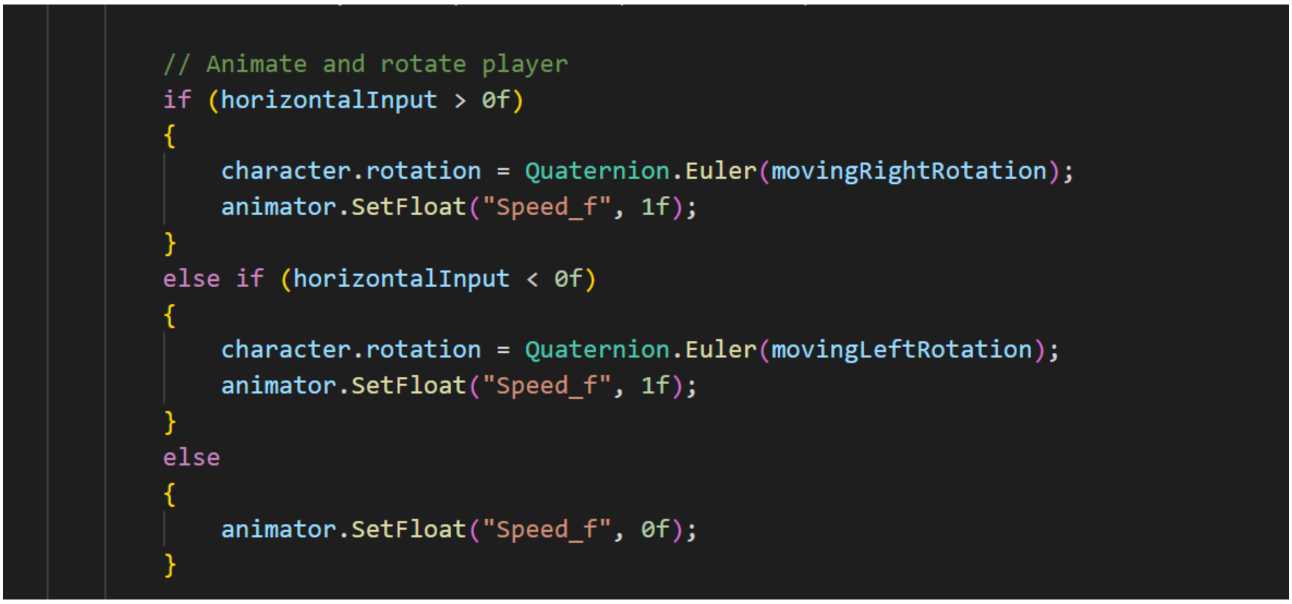
| | |
|---|---|
| Script | PlayerController |
| Horizontal Input | 0 |
| Jump Force | 10 |
| Speed | 10 |
| X Range | 160 |
| Hp | 3 |
| Required Keys | 3 |
| Key Collected | 0 |
| Is Game Over | ☐ |
| Shield | Shield Sphere |
| Dead Ui | Dead Ui Panel |
| Prompt Text | Text (TMP) (Text Mesh Pro UGUI) |
| Keys Counter | Keys Text (TMP) (Text Mesh Pro UGUI) |
| Door Animator | Door Animation (Animator) |
| Hp Slider | HP Slider (Slider) |
| Shield Slider | Shield Slider (1) (Slider) |
| Character | Character (Transform) |
| Moving Right Rotation | X 0 Y 90 Z 0 |
| Moving Left Rotation | X 0 Y -90 Z 0 |
| Damaged Sound | crash3 |
| Jump Sound | jump2 |
| Key Collect Sound | jump1 |
| Door Open Sound | 494128__wlabarron__hotel-room-door-unlocking |
| Cant Open Door Sound | crash2 |
| Shield Collect Sound | jump3 |
| Explosion Particle | FX_Explosion_Smoke (Particle System) |
| Door Open Particle | Smile Particles (Particle System) |
| Cant Open Door Particle | FX_DirtSplatter (Particle System) |

The player controller is responsible for most of the things that happens in the game, most of the code used is from the Unit 3 – Sounds and Effects course of the Unity website: Unit 3 - Sound and Effects - Unity Learn

But there are various additions:

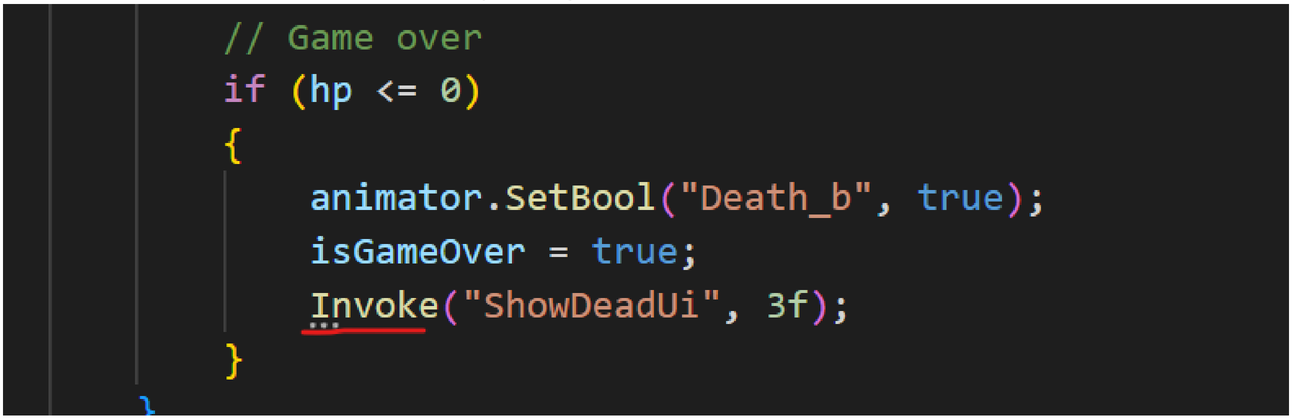
- There is an heal bar (required in the assignment)
- I've also added a shield power up that you can pick up (Extra marks in the assignment)
- In the courses I couldn't find an example of how to rotate the character so I've added a very simple code that rotate the character to a set rotation in inspector (moving

right rotation and moving left rotation fields) based on the horizontal input



```
// Animate and rotate player
if (horizontalInput > 0f)
{
    character.rotation = Quaternion.Euler(movingRightRotation);
    animator.SetFloat("Speed_f", 1f);
}
else if (horizontalInput < 0f)
{
    character.rotation = Quaternion.Euler(movingLeftRotation);
    animator.SetFloat("Speed_f", 1f);
}
else
{
    animator.SetFloat("Speed_f", 0f);
}
```

- Some code is executed after a delay using the Invoke method



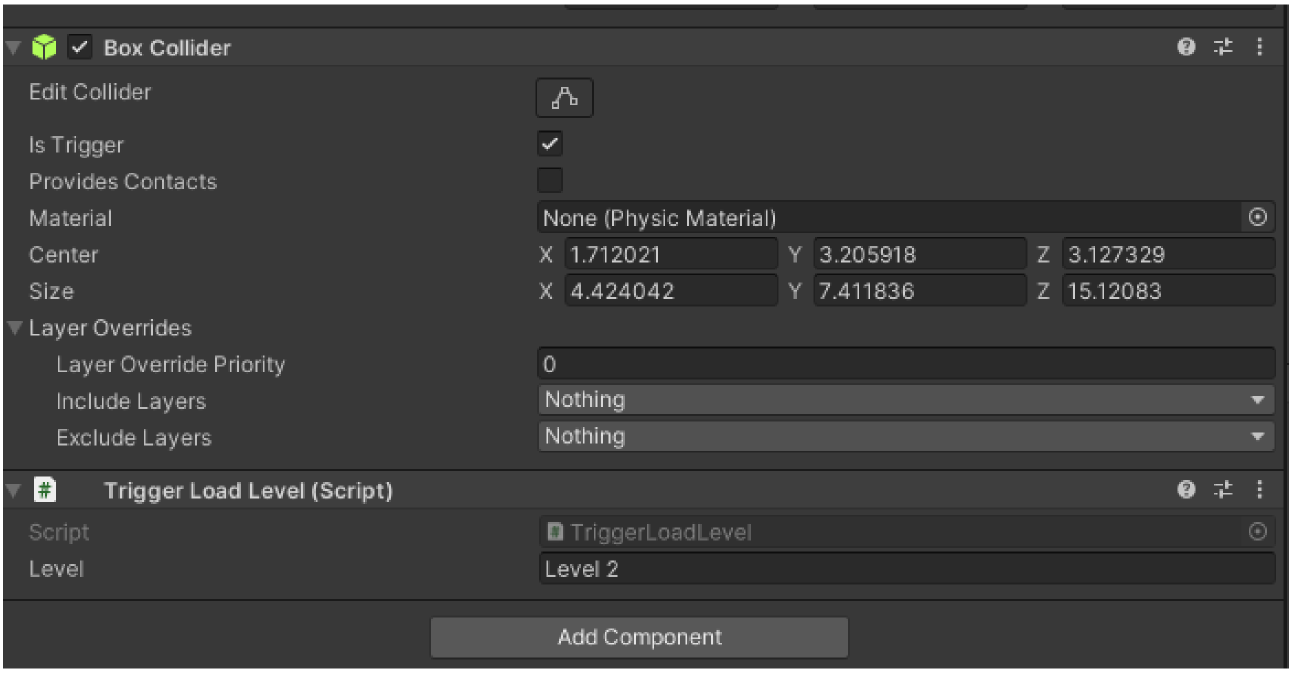
```
    // Game over
    if (hp <= 0)
    {
        animator.SetBool("Death_b", true);
        isGameOver = true;
        Invoke("ShowDeadUi", 3f);
    }
}
```

This method is shown in the unity course to spawn objects, so I've used this when it was needed to delay execution (example after player is dead, the dead ui shows after 3 seconds) because it's probably the only way you should be aware of.

- In the player controller you can set how much hp the player has and how many keys are required to open the door to the level.

The rest of the fields in the script are mostly references to component that are used in different points in the games, example particles, or text to update the amount of keys etc.

# Trigger Load Level



Box Collider

| | |
|---|---|
| Edit Collider | |
| Is Trigger | ☑ |
| Provides Contacts | □ |
| Material | None (Physic Material) |
| Center | X 1.712021 Y 3.205918 Z 3.127329 |
| Size | X 4.424042 Y 7.411836 Z 15.12083 |
| Layer Overrides | |
| Layer Override Priority | 0 |
| Include Layers | Nothing |
| Exclude Layers | Nothing |

Trigger Load Level (Script)

| | |
|---|---|
| Script | TriggerLoadLevel |
| Level | Level 2 |

Inside the house you can find a trigger collider that will load the next scene after the player enters the trigger.

The door to the house is opened with an animator when the player collides with it if it has collected all the required keys.

After the second level the scene will show a win scene with a simple text and animation, if you die and select give up it will instead show the Lose scene.