

# Unit 70: Computer Interface Design Principles

Justin Zammit Zammit

2014



Creating an interactive application

# Table of Contents

<b>General Description.....</b>	<b>2</b>
<b>Task 1 .....</b>	<b>2</b>
<b>Task 2 .....</b>	<b>4</b>
<b>Task 3 .....</b>	<b>5</b>
<b>Task 4 .....</b>	<b>6</b>
<b>Task 5 .....</b>	<b>7</b>
<b>Task 6 .....</b>	<b>9</b>
<b>Task 7 .....</b>	<b>11</b>
<b>Task 8 .....</b>	<b>13</b>
<b>Task 9 .....</b>	<b>24</b>
<b>Task 10.....</b>	<b>25</b>
<b>Task 11.....</b>	<b>27</b>
<b>Task 12.....</b>	<b>28</b>
<b>Task 13.....</b>	<b>30</b>
<b>Task 14.....</b>	<b>31</b>
<b>Task 15.....</b>	<b>32</b>
<b>Bibliography .....</b>	<b>33</b>

### General Description

You have been commissioned to create a game as part of a marketing campaign for your company. The game may have any theme you choose, however it must implement the following features:

- Player object with score
- Player must implement 3 actions
- 3 different GAME scenes with different background images
- A fully featured menu with a new game/exit button
- 3 Power ups
- 3 Weapons / player modifiers
- Functional screen boundaries based on a fixed square resolution of 800x600
- Enemies which could cause damage to the player in at least 2 different ways

## Task 1

### **(P1.1) Identify interactive media systems by answering the following question:**

Which are the most important criteria when it comes to choosing a game engine? List five criteria and justify your choice.

**Tools and Features:** are important when it comes to choose a game engine. The developer must know whether the game engine supports 2D or 3D games or any other features that might be of a benefit to develop the game. For instance we can say that using Cry Engine one can develop an unlimited sized world while with Unity the developer will be limited if he needs to create a large gameplay. Another example is in Unity we can find an integrated store while in other game engines it is not present. Profiling system, live preview on target platform and standalone world builder are features that one must take into consideration.

**Scripting Language Support:** is another criterion that the developer must keep in his mind. Not all the game engines supports the same scripting languages, therefore the developer need to choose which type of coding he is going to choose. C++ and C# is not present in some game engines.

**Platform Support:** is also important. You must take into consideration the device which you are willing to deliver the game, whether it is on mobile applications, on the computers or any other game consoles such as Xbox or PlayStation. Some game engines needs a plugin to develop a game for the web. (Charbonneau, 2013)

**Cost and Complexity:** is another criterion that is significant. If the game needs to be developed in a high end engine, then the cost will be more expensive. Training, plugins and maintenance fees should also be considered. (DeLoura, 2009)

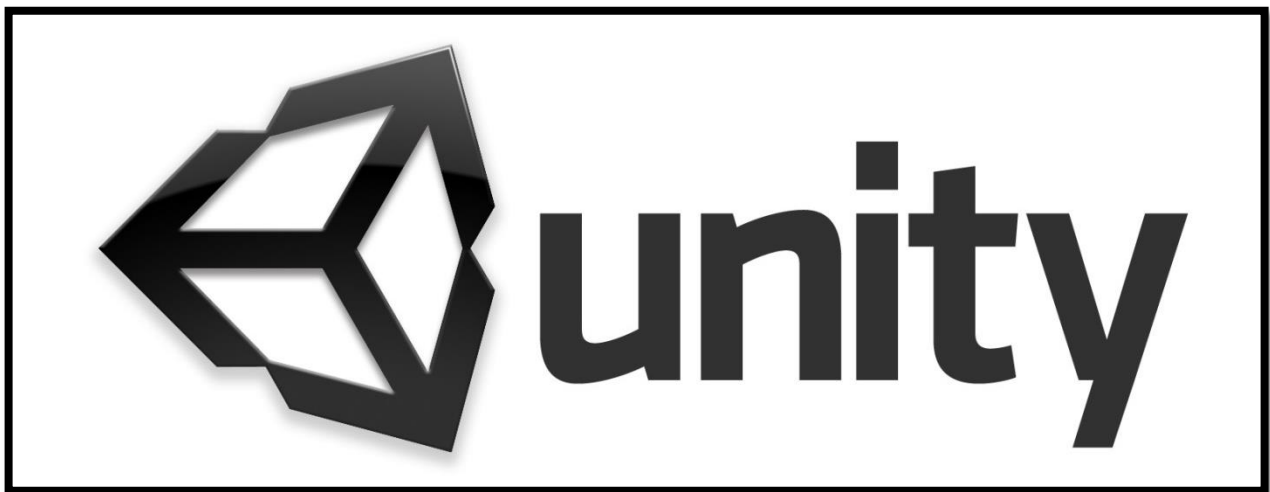
**Functionality:** Before starting to build the game, one must to check which engine systems are the most important. Animation system, multi-threading system, rendering and streaming system are functions that build the game.

Below is a comparison chart of different game engines.

(Esenthel)

Game Engine Comparison				
Engine	Esenthel Engine	Unity	Unreal Engine	CryEngine
Price	Free 14.99 USD/Month 149.99 USD/Year (12.50/Month) 499.99 USD/Year	Free 1500–5000 USD 75–245 USD/Month	19 USD/Month + 5% royalties	9.90 USD/Month
Platforms				
Windows	Yes	Yes	Yes	Yes
Mac	Yes	Yes	Yes	No
Linux	Yes	Yes	No	No
Android	Yes	Yes	Yes	No
iOS	Yes	Yes	Yes	No
Web	Yes	plugin required	No	No
Features				
Collaborative Development	Yes	No	No	No
Unlimited Sized Worlds	Yes	No	No	Yes
C++ Interface	Yes (C++ and Code Editor)	No (C# and JavaScript)	Yes	Yes
Gui Editor	Yes	No	No	No
Integrated Store	Yes	Yes	No	No

Since I am still students and my game doesn't need to be a high end platform game, I am going to use the Unity version 4 free edition game engine. It supports JavaScript and C# but we are going to implement the game using JavaScript. With this program I think I will manage well and will be able to create the game as required.

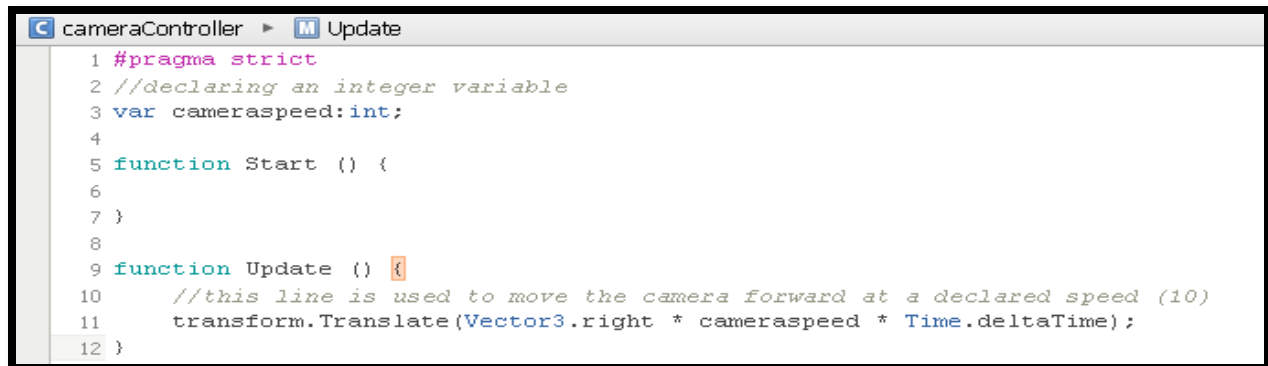


## Task 2

### **(P1.2) Evaluate constraints affecting interface design by completing the following task:**

Show, with reference to your code, how you implemented fixed screen borders so that player objects could not leave the game scene. For this task you need to write a short description and show one screenshot of the code that keeps the player on screen.

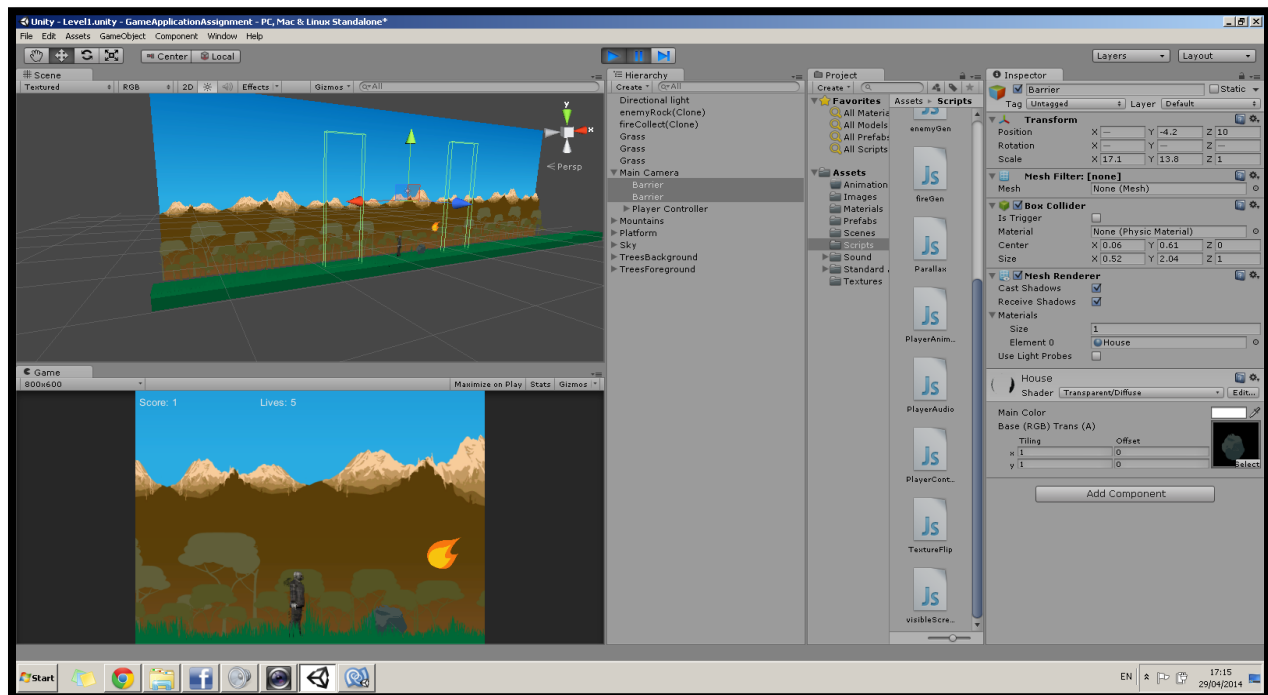
My gameplay consist of a side scrolling game, where my camera should move with a constant speed to the right. Thus my player shouldn't be able to leave the camera scene. In order to do so, I have created 2 invisible cubes as indicated with colour green in the image on the next page and added a Box collider component to the cubes so that they will be like a barrier. I parent them with the main camera so that they will move along with it. Consequently the player will not be able to leave the scene.



```

1 #pragma strict
2 //declaring an integer variable
3 var cameraspeed:int;
4
5 function Start () {
6
7 }
8
9 function Update () {
10     //this line is used to move the camera forward at a declared speed (10)
11     transform.Translate(Vector3.right * cameraspeed * Time.deltaTime);
12 }
  
```

Camera script to move forward



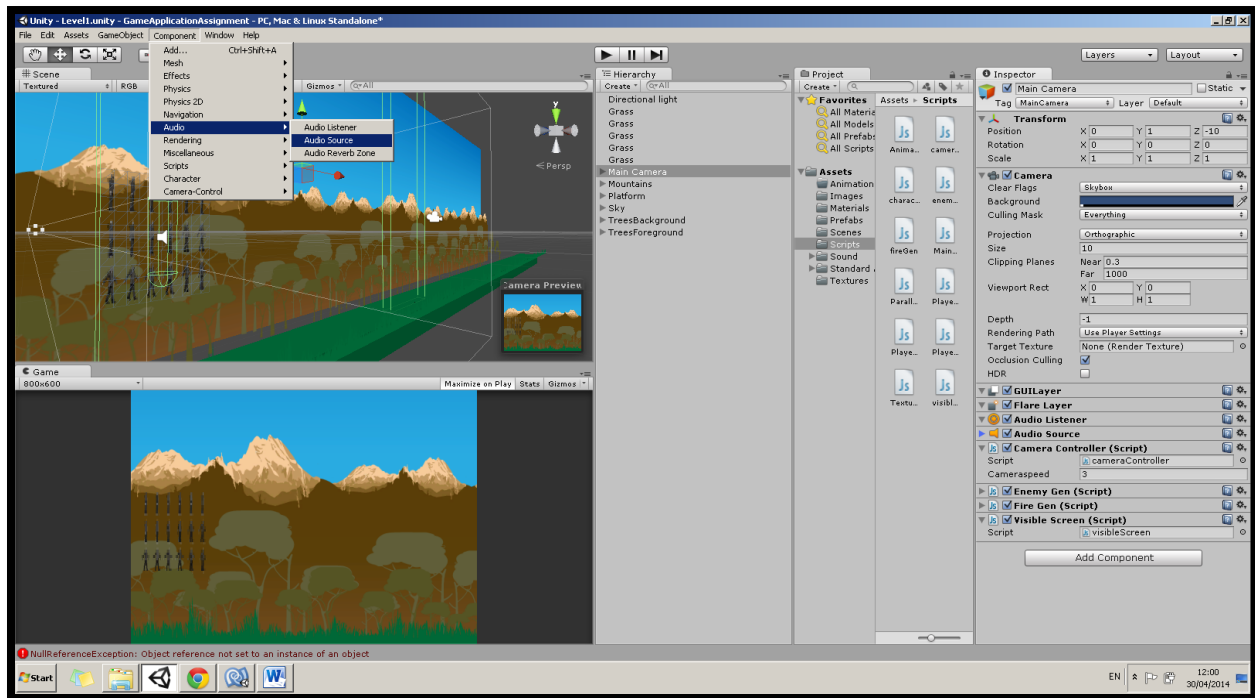
2 cubes with a box collider component that will act as a barrier.

## Task 3

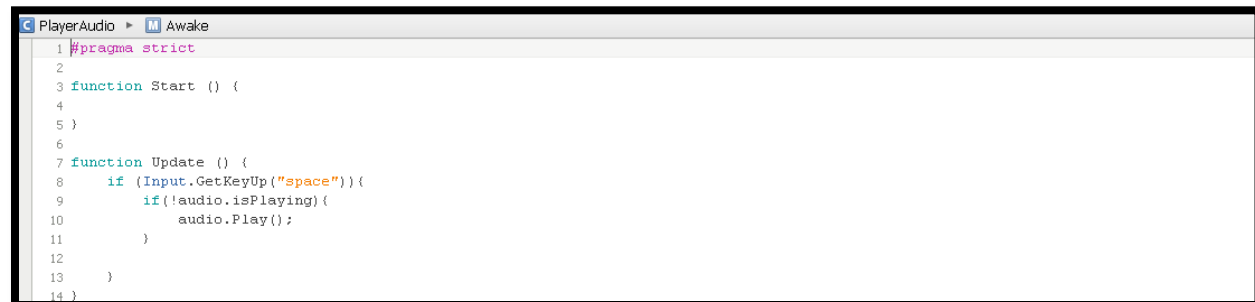
### **(P2.1) Show, with reference to your code, how you implemented a sound effect in your game.**

For this task you will need to write a short description of how you implemented the sound effect and show one screenshot of the code that implements that sound effect.

Firstly, I began by implementing the background sound for the game. In order to do this, I had added an audio source component to the main camera. Then I located the background audio from the sound folder as can be seen below.



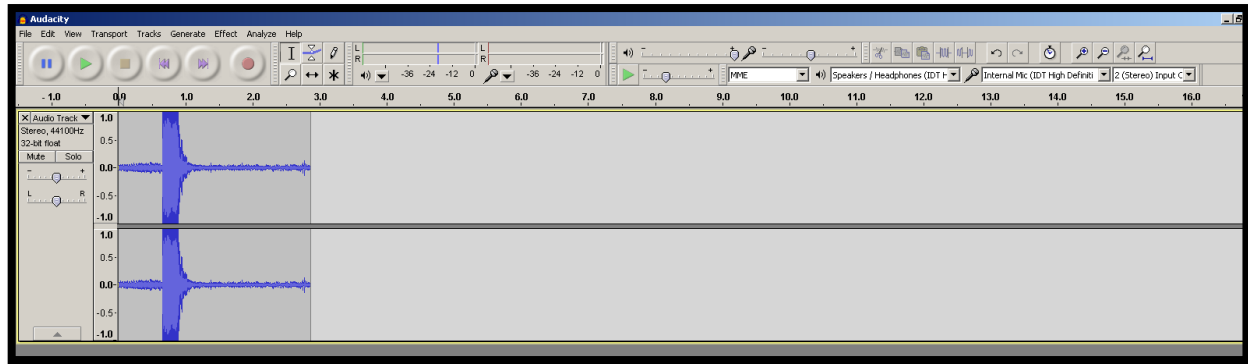
After that I recorded some audio to have some sound effects to the player. I did the same as I did for the background music, added an audio source to the player and located the edited audio. Then I had created a JavaScript to create this sound effect on the player when hitting the space bar to jump. Below we can note the script which was dragged on to the player object. Script was dragged on the Player object.



## Task 4

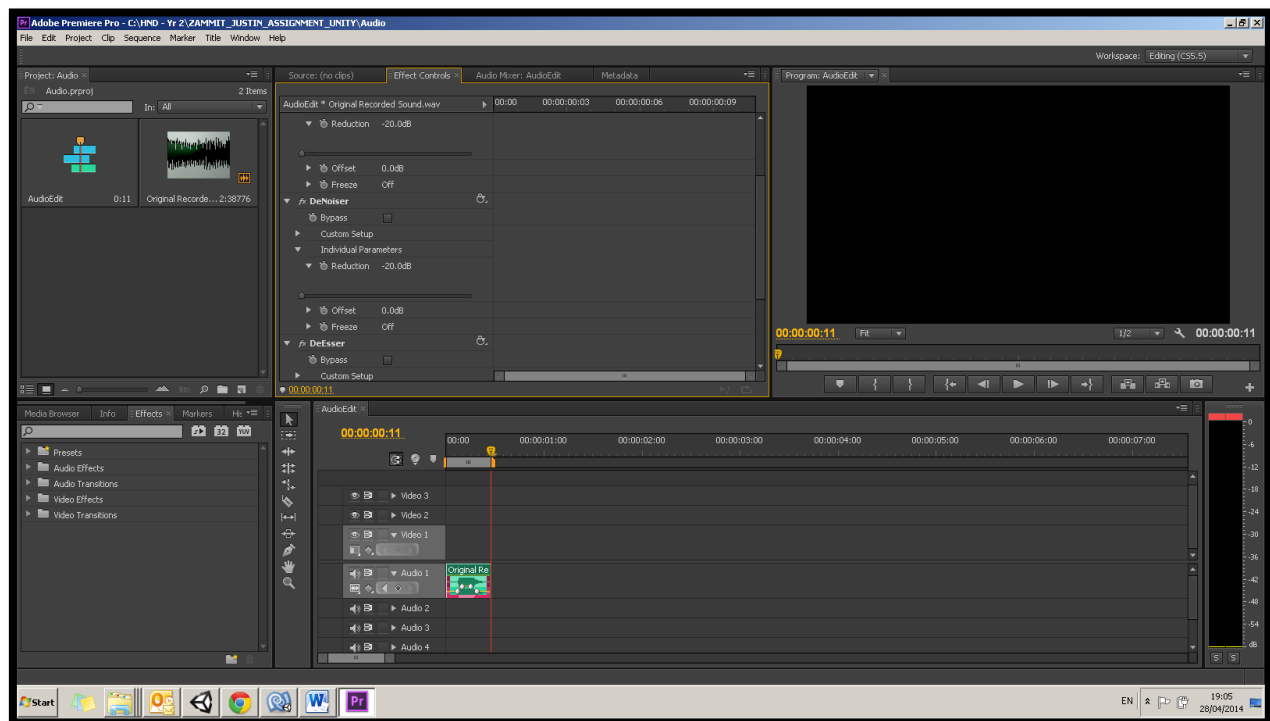
**(P2.2) Record and generate an effective sound for your game. To do this you need to present the following three .mp3 format sounds:**

- The original sound recorded.



Sound was recorded using Audacity program.

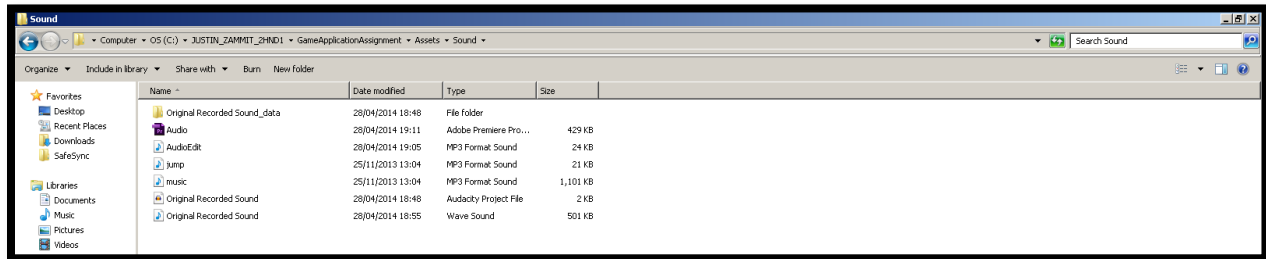
- A description of the effects applied to the sound.



Here we can note the effects that were applied to the sound for our gameplay. This sound will be going to implemented once the user press on the space bar so that the player jumps. I used the DeNoiser effect twice in Adobe Premiere with an individual parametere reduction of -20.00db. I also used DeEsser and speed up the audio.

- **The final modified sound.**

To achieve this task, please write a small reference to the folder where you put these files in your documentation under Task 4.

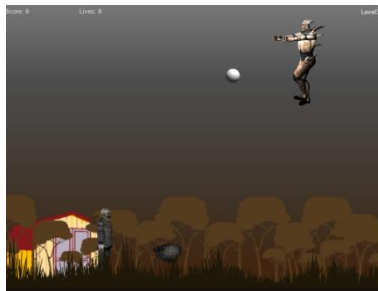


Here we can locate the all 3 files to which I edited and produced the final sound for the jump button gameplay. The final sound is named AudioEdit.mp3.

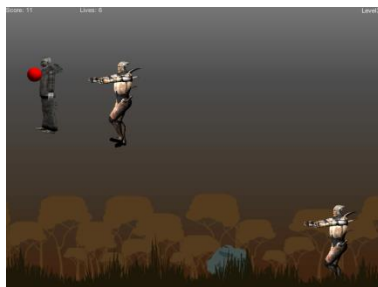
## Task 5

### **(P3.1) Explain how to change materials of different objects based on in-game events, with reference to the OnTriggerEnter and OnTriggerExit functions.**

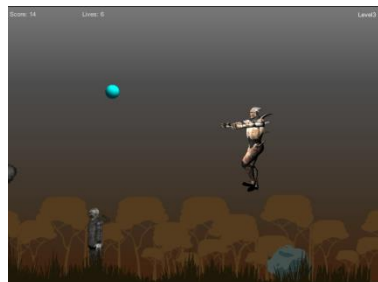
To achieve this task, include one screenshot of the game event which occurs when the material is changed (for example, when the laser hits the enemy and the enemy turns red), and include another screenshot showing the code that runs when this happens.



In Level 3 we can find circles as power ups. The circles will increase the score to +3 every time the player passes through the circle. As we can see in this print screen, the circle power up bonus will be initiated with a white material colour.

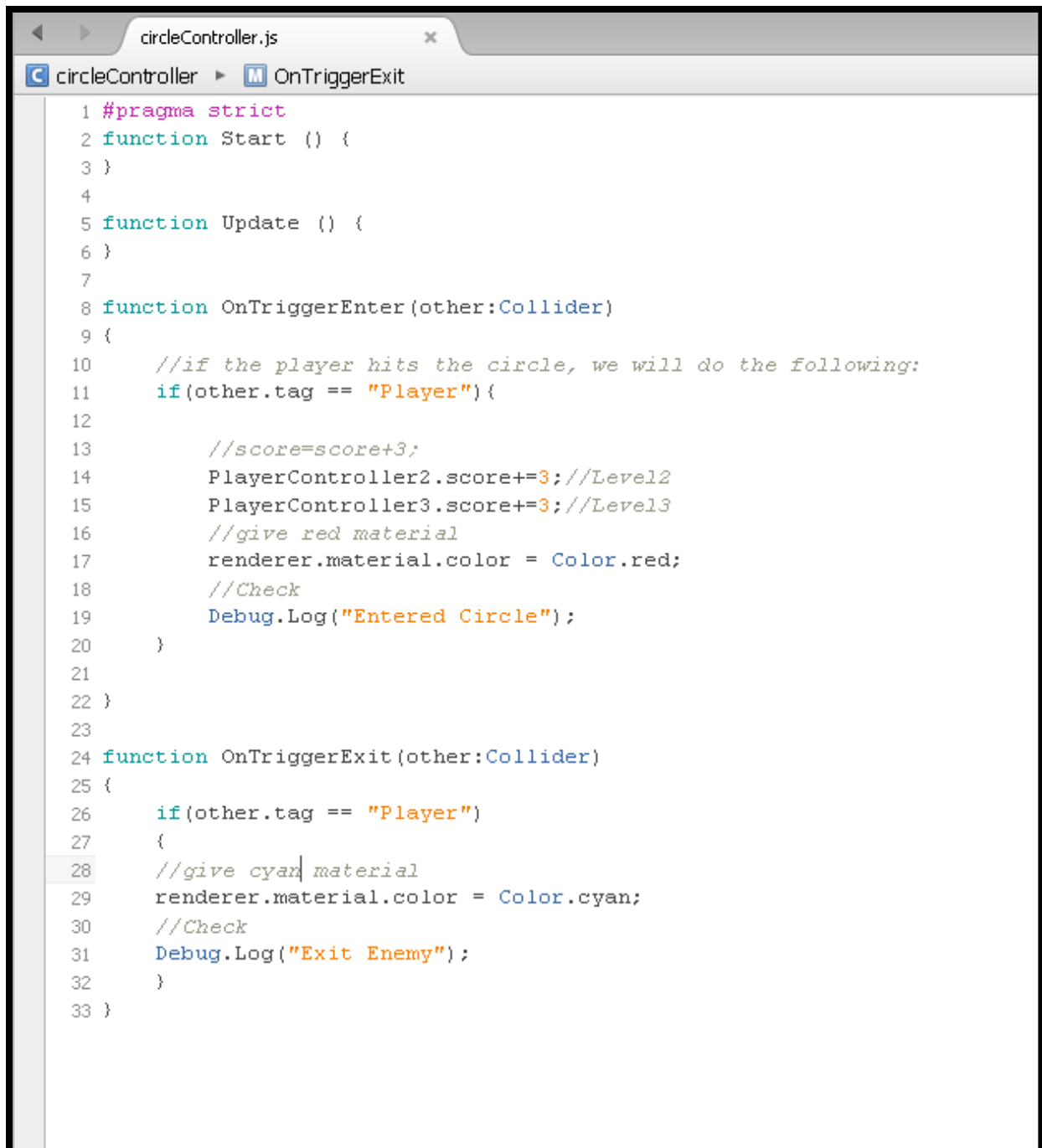


The player when enters through the circle, OnTriggerEnter the circle will change its material colour to red as can be seen in the print screen on the left.



When the player exits the circle, OnTrigger Exit the circle will change its material from red to cyan colour. This is shown in the print screen which was taken place during the event.





```
1 #pragma strict
2 function Start () {
3 }
4
5 function Update () {
6 }
7
8 function OnTriggerEnter(other:Collider)
9 {
10     //if the player hits the circle, we will do the following:
11     if(other.tag == "Player"){
12
13         //score=score+3;
14         PlayerController2.score+=3;//Level2
15         PlayerController3.score+=3;//Level3
16         //give red material
17         renderer.material.color = Color.red;
18         //Check
19         Debug.Log("Entered Circle");
20     }
21
22 }
23
24 function OnTriggerExit(other:Collider)
25 {
26     if(other.tag == "Player")
27     {
28         //give cyan material
29         renderer.material.color = Color.cyan;
30         //Check
31         Debug.Log("Exit Enemy");
32     }
33 }
```

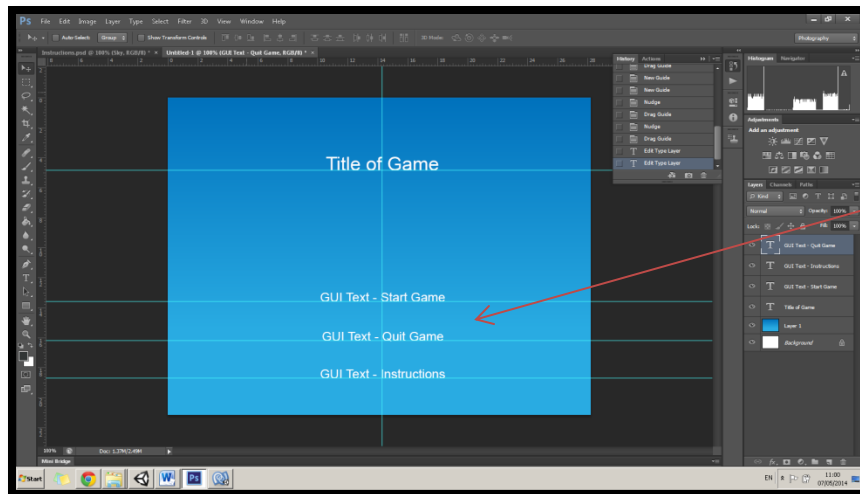
This is a print screen showing the coding which takes place during the event of OnTriggerEnter and OnTriggerExit. This JavaScript which is named “circleController.js” was dragged on the circle material prefab so that it could generate these circles during the gameplay. Then I created another JavaScript named “circleGen.js” and dragged it on the Main Camera. In order to generate the circles in the game, I combined the Prefab of the circle with this script.

## Task 6

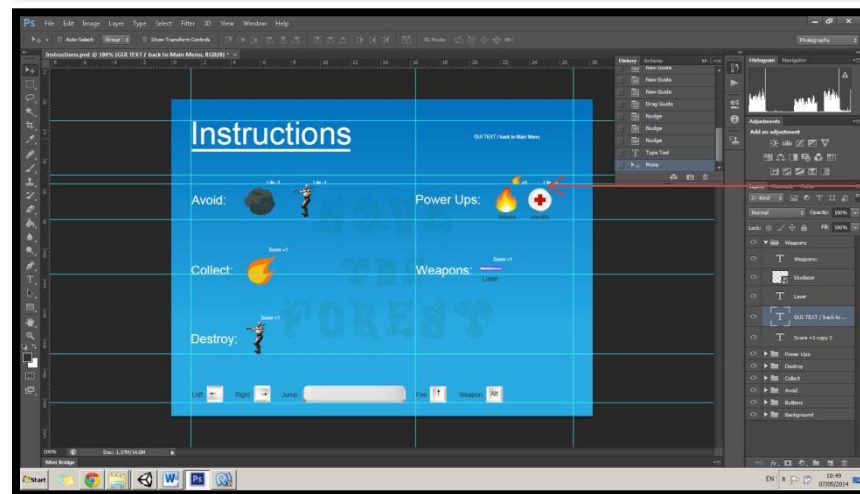
### (P3.2) Draw a sketch of two screens of your game:

The sketches will show the positions of your GUI elements. You must include 2 sketches which may be built as simple wireframes in Photoshop.

- Main menu



This is a sketch of the Main Menu where I am going to implement 3 GUI Text elements: Start Game, Quit Game and Instructions. I built this in Photoshop to have a clear idea.



This is the Instructions of how to play the game. The GUI Text element shown with the red arrow, once the user click on it, he will be redirected to the Main Menu.

```
#pragma strict
var isQuit = false;

//change text color to green on mouse roll over
function OnMouseEnter(){
    guiText.material.color= Color.cyan;
}

//change text color to white
function OnMouseExit(){
    guiText.material.color= Color.white;
}
```

```
function OnMouseUp(){
    if (isQuit==true){
        Application.Quit();

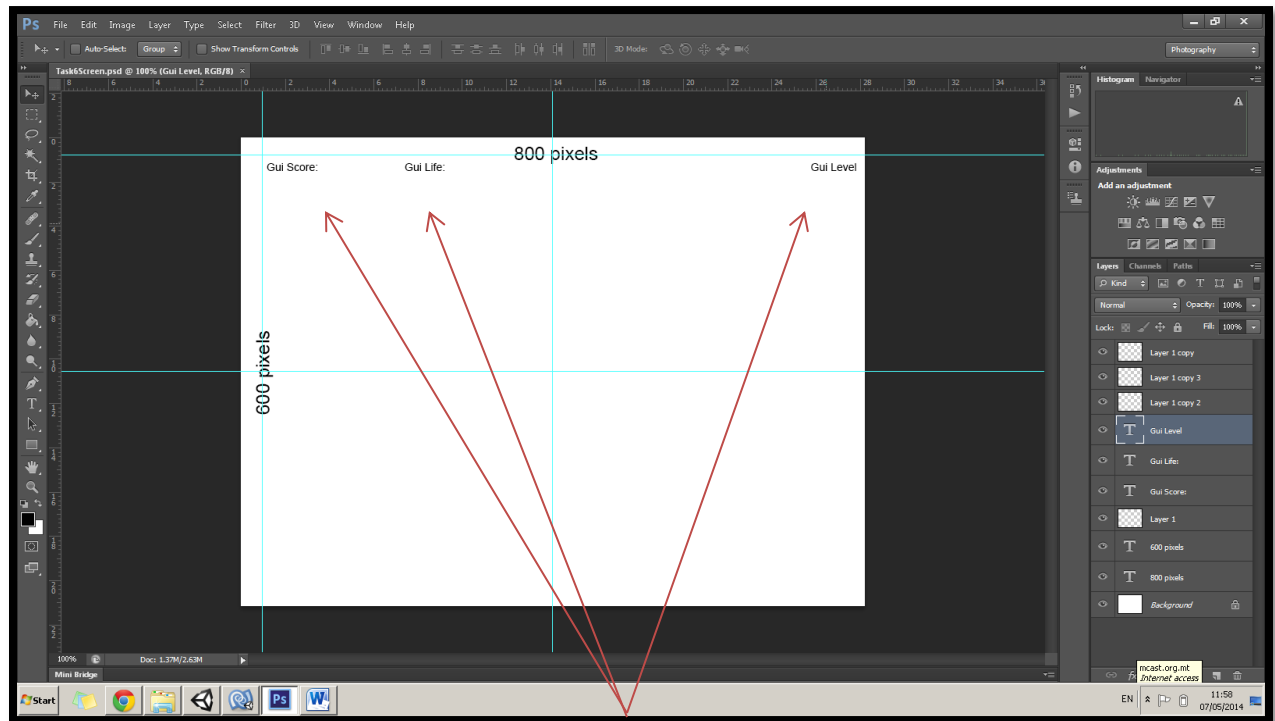
        //quit the game
    }

    else {

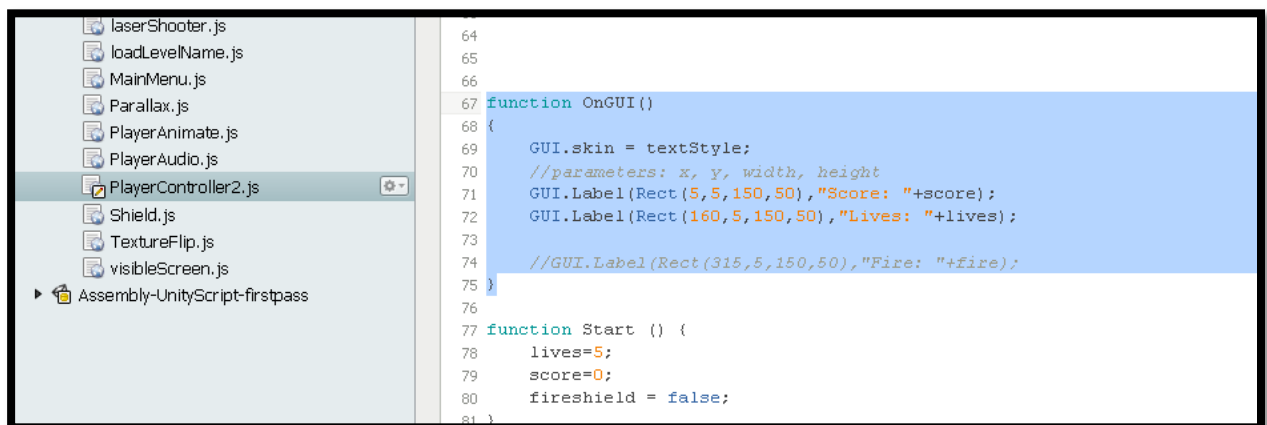
        Application.LoadLevel("Level1");
        // load level 1
    }
}
```

- Game screen

In Photoshop using the wireframe, I had sketched an 800 by 600 pixels Screen. This will be set through the gameplay. At the top margin I will implement the Score, Life and Level as GUI text elements. Below we can see some coding of how I had implemented these.



Screen Sketch



Coding for the GUI Elements on Screen

## Task 7

### **(P4.1) Conceptualize an interactive system by explaining the reason for the following 3 methods:**

To achieve this criterion you need to write a paragraph for each method describing the role of that method and when it is called. You must then explain, in that paragraph, at which point of the game such a method could be called for each of your prefabs.

- Start(),

Start is called on the frame when a script is enabled just before any of the Update methods is called the first time. Start is called exactly once in the lifetime of the script. However, Awake is called when the script object is initialized, regardless of whether or not the script is enabled. Start may not be called on the same frame as Awake if the script is not enabled at initialization time. (MonoBehaviour.Start(), 2014)

```
56 //when the game starts
57 function Start ()
58 {
59     lives=5;
60     score=0;
61 }
```

This is an example when the user can implement the Start() function. When the game starts, lives are equal to 5 while the score will be 0.

### **Prefabs in game**

```
25
26 function Start () {
27     //parameter 1: wait 1.0 seconds
28     //parameter 2: every 1.2 seconds
29     InvokeRepeating("createEnemy",2.0,3.0);
30 }
```

In the JavaScript named enemyGenCharacter.js which is in conjunction to the prefab "enemyCharacter" one can note that when the game starts, the prefab object will wait 2.0

seconds and will repeat the same object at random position every 3.0 seconds. Other similar prefabs have the same coding.

- Update(),

Update is called in every frame, if the MonoBehaviour is enabled. Update is the most commonly used function to implement any kind of game behaviour. (MonoBehaviour.Update(), 2014)

```
63 //updating score and lives in gameplay
64 function Update ()
65 {
66     checklives();
67     checkscore();
68 }
```

This is an example when the game is continuously updating and checking the score and lives during the game play.

## Prefabs in game

```

28
29 function Update () {
30     //move the laser FORWARD
31     transform.Translate(Vector3.right *
32         laserSpeed * Time.deltaTime);
33

```

In the prefab “blueLaser” I added a JavaScript named “laserController.js” where one can note that the camera is updating by moving forward to the right. This is being called continuously.

- OnTriggerEnter()

This is called when the Collider other enters the trigger. This message is sent to the trigger collider and the rigidbody (if any) that the trigger collider belongs to, and to the rigidbody (or the collider if there is no rigidbody) that touches the trigger. Note that trigger events are only sent if one of the colliders also has a rigidbody attached.

(Collider.OnTriggerEnter(Collider), 2014)

## Prefabs in game

```

8 function OnTriggerEnter (other: Collider)
9 {
10 //when the player hits the life, the lives will increase +1
11     if(other.tag == "life")
12     {
13         Destroy(other.gameObject);
14         lives=lives+1;
15     }
16 //when the player hits the life, the lives will increase +1
17     if(other.gameObject.tag=="enemy")
18     {
19 //decrease lives by 1
20         lives--;
21 //destroy the enemy
22         Destroy(other.gameObject);
23     }
24 }

```

This is an example when the function takes place. We can note that when the Player collides with the object tagged “life”, the object will be destroyed while increases the lives to +1. Within the same function when the Player hits an object tagged as “enemy”, the Player decreases 1 life.

```

8 //function that is triggered when laser hits the enemy
9 function OnTriggerEnter (other:Collider)
10 {
11     //if the laser hits an enemy, we will do the following two things:
12     //Destroy the enemy
13     //Destroy the laser
14
15     if(other.tag == "enemy"){
16         Destroy(other.gameObject);
17         //score=score+1;
18         PlayerController3.score++;
19     }
20     //destroy the enemy
21     Destroy(other.gameObject);
22     //destroy the laser
23     Destroy(this.gameObject);
24
25 }

```

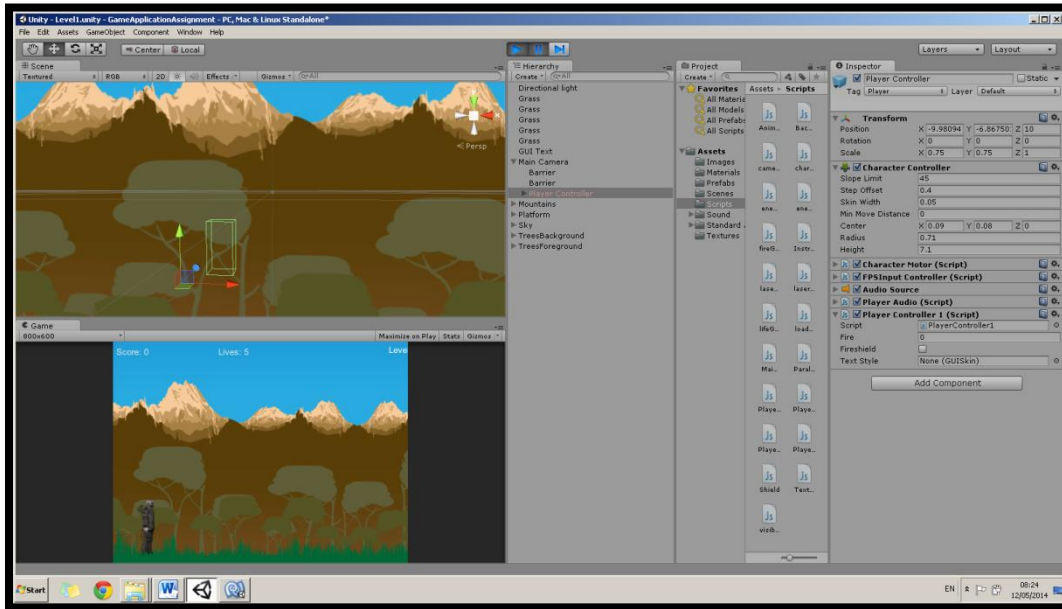
In the “laseController.js” which is in conjunction with the “blueLaser” prefab one can note when the function is called.

## Task 8

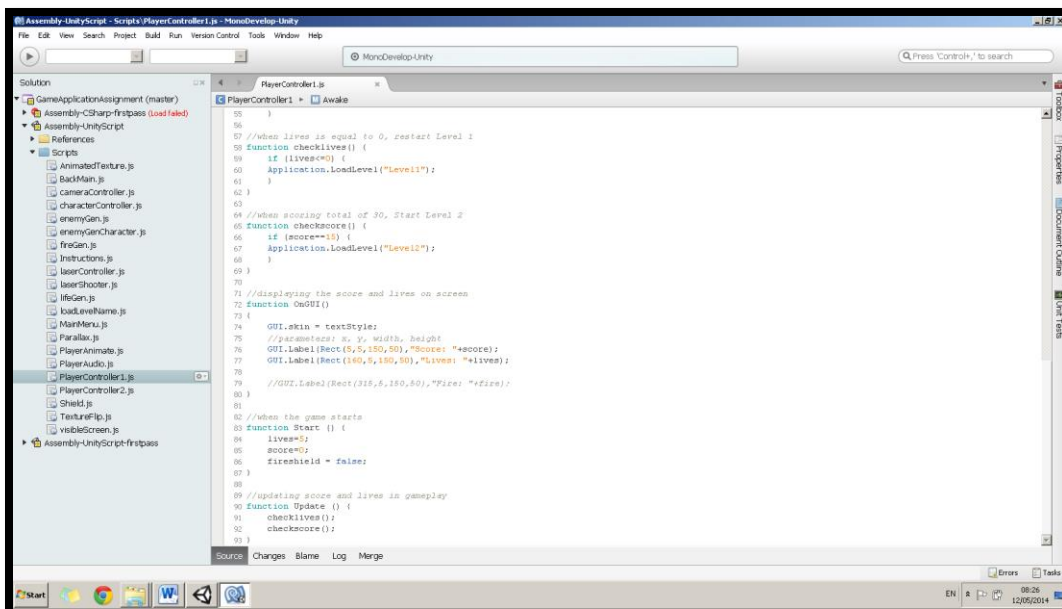
### **(P4.2) Present your game. The game must implement the following functionalities:**

Your game must be presented as an executable file. To achieve this task, please write a short paragraph on how to play the game, including a FULL key map and include a note in your documentation explaining the location of the executable file.

- Player object with score



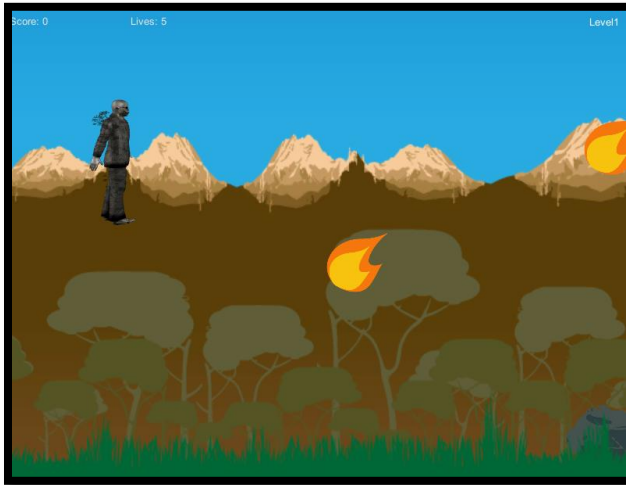
Player Object Controller



Coding to display score when Player collides with other object on trigger

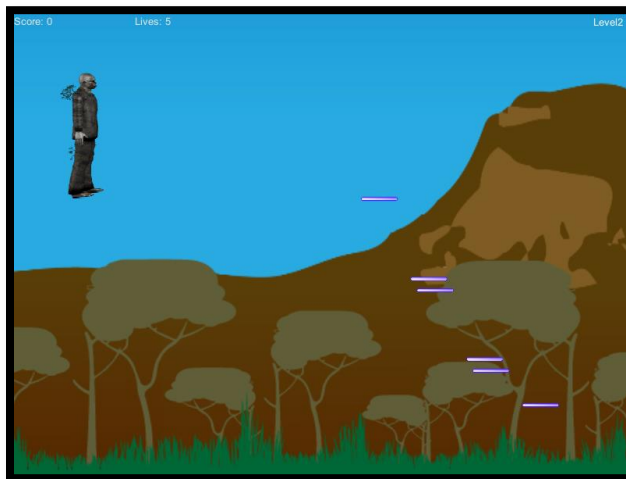
- Player must implement 3 actions

### Jumps with Sound Effects



The player can jump in all the Levels by pressing the space bar button. When hitting on the space button to jump, the player makes a sound.

### Shoots



The player can shoot the laser when hitting the up button in Level 2 and 3. When the laser hits an enemy, the enemy will be destroyed and the player will score points.

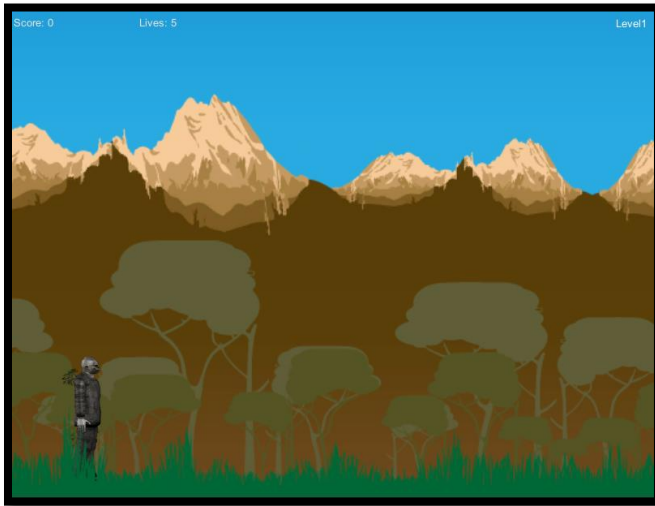
### Move Left & Right with Animation



The player can move either to the left or to the right but it isn't able to move out of the game border. When Player moves, it will be animated so that the Player look realistic when walking.

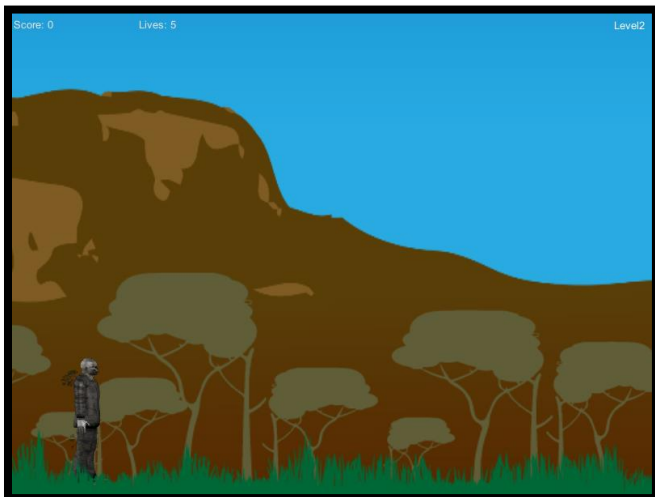


- 3 different GAME scenes with different background images



Level 1

This is the first background image that I have implemented in Level 1. I used different layers images set as Sprite. In order to generate the background when the camera scrolls to the right I have inserted some JavaScript on the backgrounds objects.



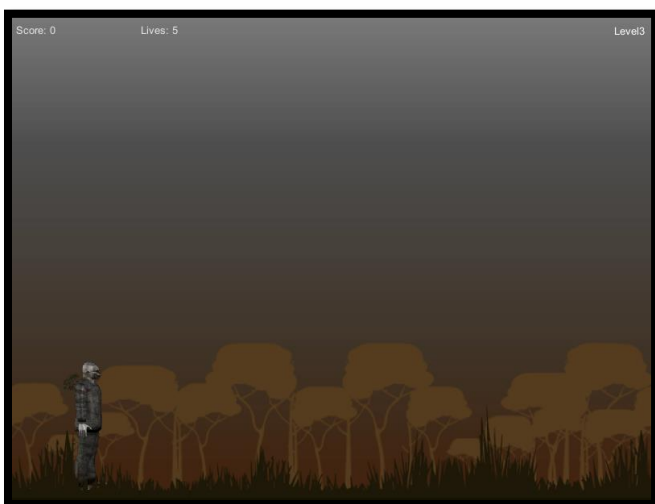
Level 2

### VisibleScreen.js

```
function OnBecameInvisible()
{
    //when the background leaves the screen,
    place the background on the horizontal

    //edge of the screen

    transform.position.x =
    Camera.main.ScreenToWorldPoint(Vector3(Screen.
    width,0,0)).x;
}
```



Level 3



- A fully featured menu with a new game/exit button



Main Menu



Instructions

- 3 Power ups



**Shield**

```
function OnTriggerEnter (other: Collider) {
    //count how many fires collected
    if(other.tag == "fire"){
        fire++;
    }

    //when collecting fire, destroy fire and
    score will increase +1
    if(other.tag == "fire")
    {
        Destroy(other.gameObject);
        score=score+1;
    }

    //when fire collected is equal to 5, the
    shield will power up the player. Only once can be used.
    if (fire==5)
    {
        fireshield = true;

        GameObject.FindGameObjectWithTag("shield").t
ransform.position = Vector3(transform.position.x+1,
transform.position.y, 0);
        fire++;
    }
}
```

In Level 1 I have implemented a Shield Power Up. When the Player collects 5 fires, the Player will have a shield as a power up. This can only be present once so if Player hits an enemy, the shield will be destroyed.



**Life**

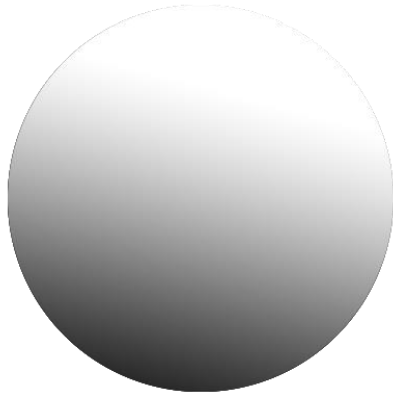
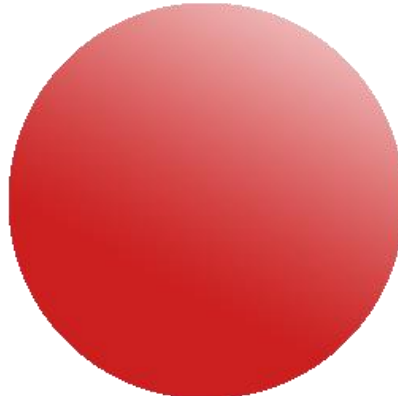
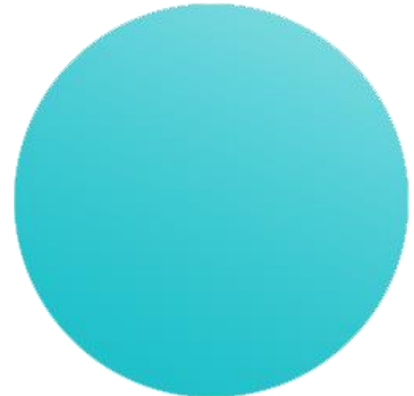
```
function OnTriggerEnter (other: Collider) {

    //when the player hits the life, the lives
    will increase +1
    if(other.tag == "life")
    {
        Destroy(other.gameObject);
        lives=lives+1;
    }

    //when the game starts
    function Start () {
        lives=5;
        score=0;
        fireshield = false;
    }

    //updating score and lives in gameplay
    function Update () {
        checklives();
        checkscore();
    }
}
```

In Level 2 I have implemented a Life Power Up. When the Player collects Life, the Player will increase its life.

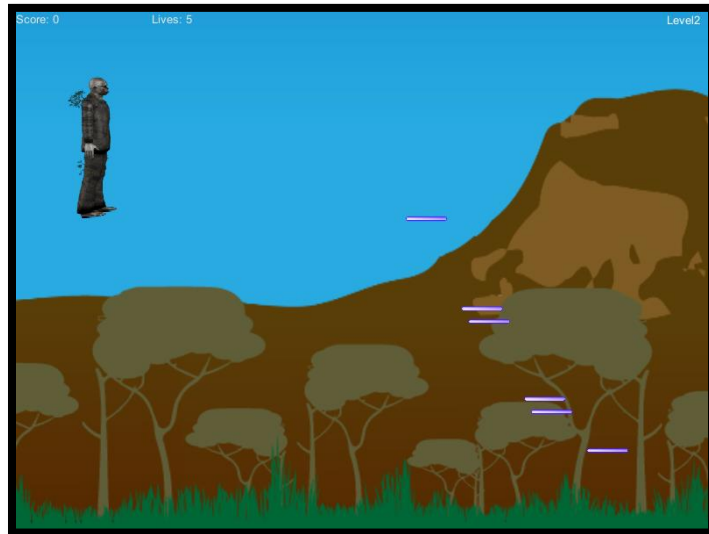
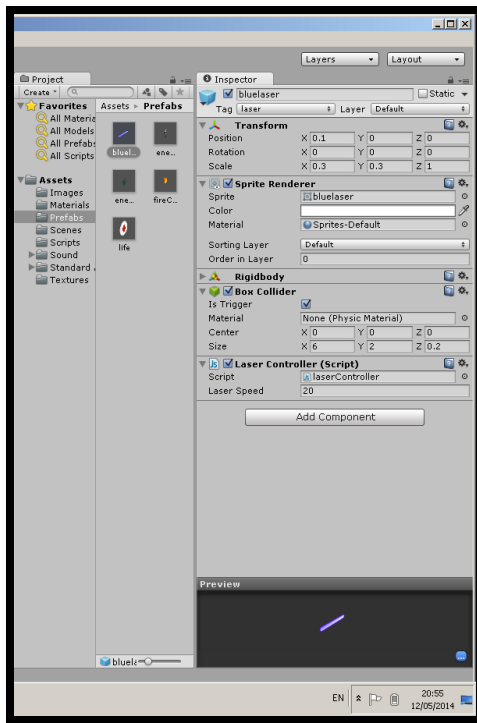
**White Material****Red Material****Cyan Material**

I implemented a sphere object with a white colour material. This circle will give the player bonus points when passing through it. Every time the player enters the circle, the circle will generate +3 to the score. In order to know that this power up is working, I employed different colours to the sphere so that the user will know what he/she is doing. When the player enters through the circle, it will change colour to red. When the player leaves through the circle, it will change from red to can colour. This will keep generating ever time the player passes through the circle.

```
function OnTriggerEnter(other:Collider)
{
    //if the player hits the circle, we will do the following:
    if(other.tag == "Player")
    {
        //score=score+3;
        PlayerController2.score+=3; //Level2
        PlayerController3.score+=3; //Level3
        //give red material
        renderer.material.color = Color.red;
        //Check
        Debug.Log("Entered Circle");
    }
}
```

- 3 Weapons / player modifiers

### Blue Laser

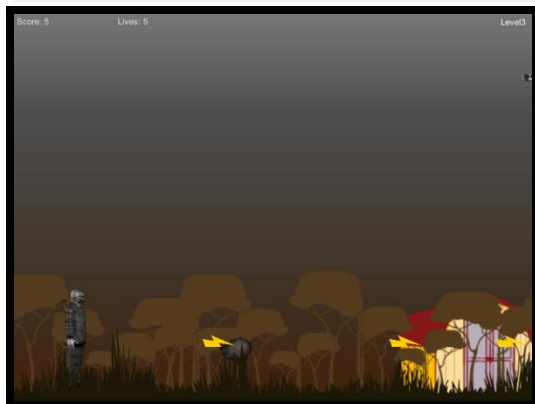


When the user reaches Level 2, the Player will have a weapon present that shoots lasers by pressing the up button.



### Bone Shooter

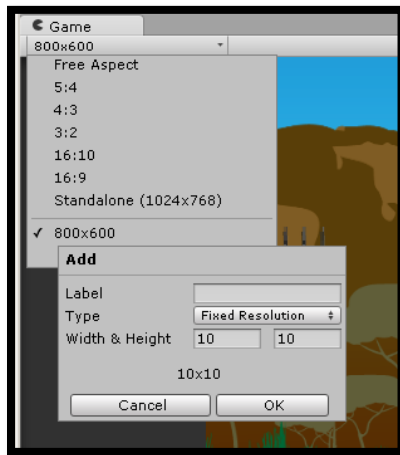
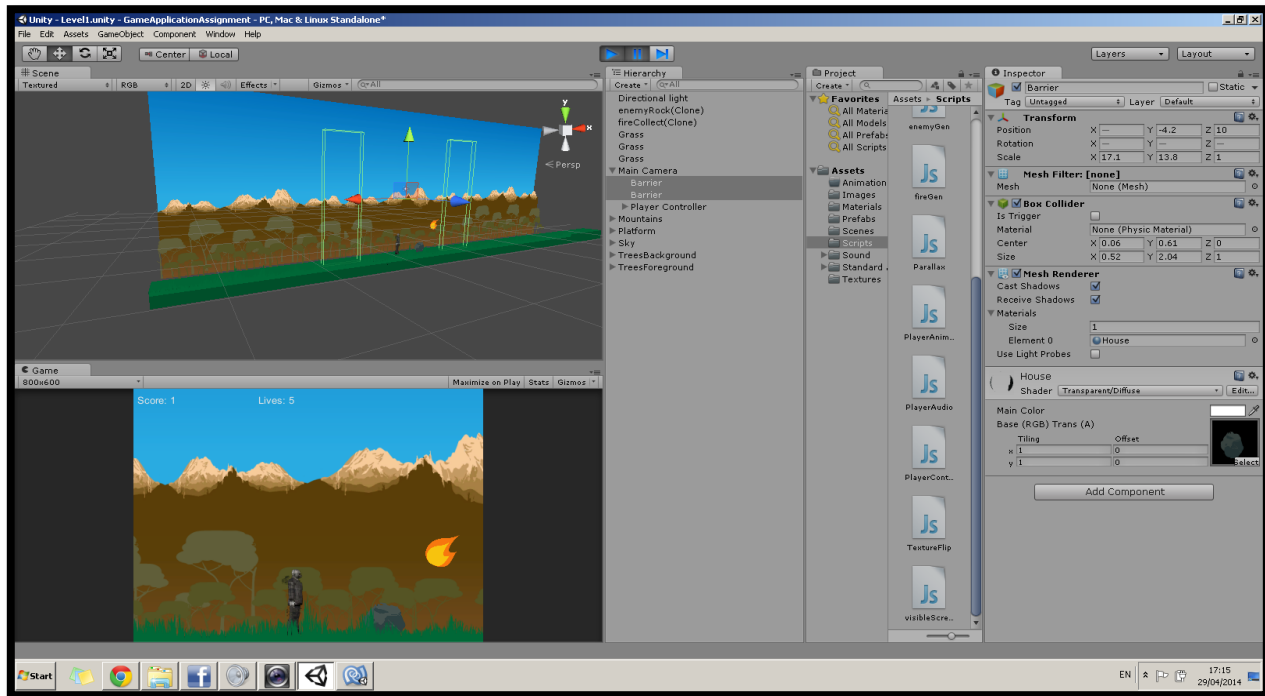
On pressing the S button in Level 2, the user will make use of a different weapon, shooting bones towards the enemies. Eventually it will increase the score by 2.



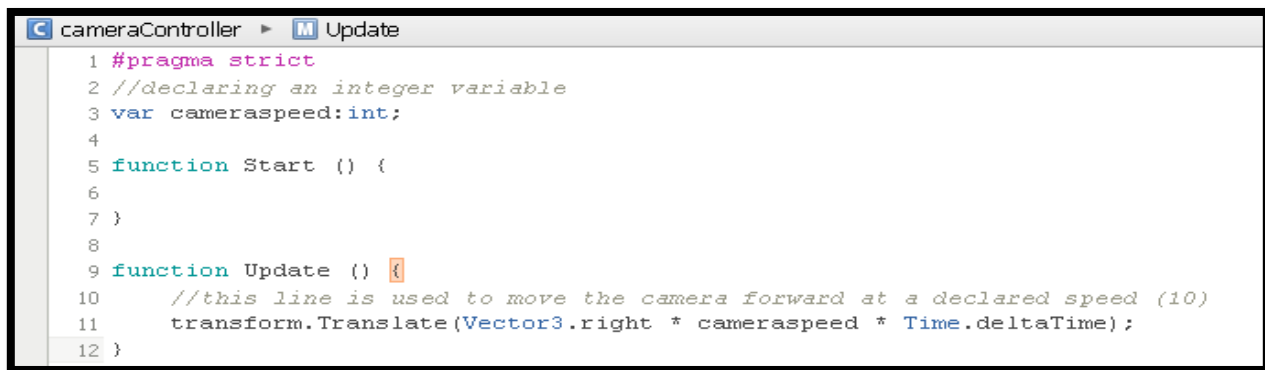
### Thunder Shooter

On pressing the F button in Level 3, the user will make use of another different weapon, shooting thunders towards the enemies. Eventually it will increase the score by 5.

- Functional screen boundaries based on a fixed square resolution of 800x600



In order to have a fixed square resolution I had to set the desired resolution as shown in the left print screen. I had inserted 800x600 res fixed size. Apart from that I had implemented 2 cube objects on the left and right of the screen and added the box collider component to them. Thus, they will act as the screen boundaries so that the Player when moving left or right will not be able to leave the screen. The Player together with these 2 barriers were parented with the Camera.



Camera moving to the right

- Enemies which could cause damage to the player in at least 2 different ways

### On Collision



When the enemy objects collide with the Player, the enemy will cause damage to the Player by reducing 1 life from the Player's lives.

```
//when the player hits the enemy, the lives will decrease -1
if(other.gameObject.tag=="enemy")
{
    //decrease lives by 1
    lives--;
    //destroy the enemy
    Destroy(other.gameObject);
}
```

### Laser Shooter



When the enemy objects will be generated, they will shoot red lasers. If the red lasers hit the Player, they will cause damage to the Player by reducing 1 life from the Player's lives.

I created an empty object and parent it with the main camera. A red laser prefab was created and was dragged in the "EnemyCharacterController.js" script. When the character will be generated, the red laser will instantiate at the enemy's character position.

```
var enemyCharacterlaser:Rigidbody;

function shootLaser()
{
    //shoot the laser
    Instantiate(enemyCharacterlaser,transform.position,transform.rotation);
}

function Start () {
    //enemyCharacter will shoot half a second after they spawn and every second after
    InvokeRepeating("shootLaser",0.5,1.0);
}

function Update () {
}

}
```





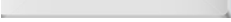
- How to Play the Game







To Play the game online one must follow this link:

<http://ict.mcast.edu.mt/ictweb/justzamm/SaveTheForest/SaveTheForest.html>





### Level 1

In level 1 the player will start with 5 lives and 0 score. The player can move to the right  or to the left . The player needs to avoid the rock enemies as seen above. In case the user will collide with the rock, the player loses 1 live. When the lives will decrease to 0, the player will have to restart the level again. To jump, the user must press on the spacebar . To increase the score by 1, the user needs to collect the fire shown above. When the player scores 5, a shield will be applied to the player as a power up. If the player hits an enemy when the shield is present, it will be destroyed and it cannot be applied again in that level so the user need to be careful not to dispose it. The shield power up will be present only in level 1. Eventually when the player scores 15 by collecting the fire, the user passes level 1 and will go to level 2.

**Level 2**

In level 2 the player will start with 5 lives and 0 score, irrelevant to the previous score and lives in level 1. The player can move to the right  or to the left . The player needs to avoid the rock and other character enemies as seen above. In case the user will collide with the enemies, the player loses 1 live. When the lives will decrease to 0, the player will have to restart the level again. To jump, the user must press on the spacebar . The other enemy character will shoot lasers. If the player hits one of these, lives will decrease by 1. In this level the player can shoot blue lasers toward the enemies by pressing the up button . The minute the blue laser hits an enemy, a small explosion will occur. This eventually will increase the score by 1. Apart from that, when the player presses the S button, the user will shoot bones which will destroy the enemy and increase the score by 2. This weapon is only present in level 2. If the player collects the life icon power up as shown above, the lives will increase by 1. Ultimately when the player scores 35, the user passes level 2 and will go to level 3.

**Level 3**

In level 3 the player will also start with 5 lives and 0 score, irrelevant to the previous score and lives gained in level 2. The player can move to the right  or to the left . The player needs to avoid the rock and other character enemies as seen above. In case the user will collide with the enemies, the player loses 1 live. When the lives will decrease to 0, the player will have to restart the level again. To jump, the user must press on the spacebar . The other enemy character will shoot lasers. If the player hits one of these, lives will decrease by 1. In this level the player can shoot blue lasers toward the enemies by pressing the up button . The minute the blue laser hits an enemy, a small explosion will occur. This eventually will increase the score by 1. If the player collects the life icon power up as shown above, the lives will increase by 1. As a bonus power up, when the player passes through the white sphere, on entering the sphere, it will change its colour to red and on exiting it will change to cyan colour and increasing the score by 3. This will keep active every time the player passes through the sphere. Moreover when pressing the F button, the player will shoot thunders which will destroy the enemy and increase the score by 5. This weapon is only present in level 3. Ultimately when the player scores 60, the user passes level 3 and will go to Level 4.

**Level 4 – Boss Challenge**

In level 4 the player will also start with 5 lives and 0 score, irrelevant to the previous score and lives gained in level 3. Together with these, we can find the lives of the boss, starting with 15 lives. In order to jump and move the player will use the same buttons as used in the previous levels. The boss enemy will shoot lasers. If the player hits one of these, its live will decrease by 1. In this level the player can swap weapons by using the Alt Key while shooting blades or bullets with the AltGr, according to what weapon the user have chosen. The minute these weapons hits the enemy boss, a small explosion will occur. This eventually will increase the score by 5. Ultimately when the player shoots the enemy boss and the enemy boss will decrease its lives to 0, the user passes level 4 and will win the game.



## Task 9

**(P4.3) Write a short paragraph explaining what improvements you would have implemented in the game if you had more time.**

I would have elaborated the background images more for the gameplay since I only changed the colour so that they will look differently. Since I have implemented the background on different layers I tried to make use of the parallax effect but when I wanted to generate the background for the side scrolling game, the parallax coding was not working and so I decided to leave the images still. Also, I would have animated the character more correctly so that it will look more realistically. Some images are pixilated; therefore I could have created some vector form illustrations.

The sound effect could have been more appealing since the one that is present makes you feel irritated. Also the background music could have been better since the one that I had implemented does not reflect the sound of the daylight in the forest.

The Levels aren't continuous. Once the user passes a Level and goes to the next, the score and lives will begin from scratch. For instance, if I had 3 lives left in Level 1 and I passed to Level 2, when the user begins Level 2, the lives will not be 3 but will be set to 5 according to how it was set at the start of the game.

Considering the time I had to create this game, in my opinion I did quite well but there is always room for improvement.

## Task 10

**(M1.1) Show that complex problems with more than one variable have been explored in the context of presenting an interactive application by adding the following functionality to your game:**

- Additional scene with boss challenge



```
var topMargin:int;
var bottomMargin:int;
var moveUp:boolean;
var moveDown:boolean;
```

```
function Start ()
```

```
{
    topMargin = 8;
    bottomMargin = -6;
    moveUp = true;
    moveDown = false;
}
```

```
function Update ()
```

```
{
    if (moveUp == true)
    {
        //move the boss up
        transform.Translate(Vector3.up * 8 *
        Time.deltaTime);
    }
```

```
if(transform.position.y >= topMargin)
```

```
{
    moveUp = false;
    moveDown = true;
}
```

```
if (moveDown == true)
```

```
{
    //move the boss up
    transform.Translate(Vector3.down * 12 *
    Time.deltaTime);
}
```

```
bottomMargin)
```

```
if(transform.position.y <=
```

```
{
    moveUp = true;
    moveDown = false;
}
```

```
}
```

```
}
```

- Additional 2 weapon types using alt keys

**Left Alt** to change weapon **AltGr** to fire weapon



Blade



Bullet

```

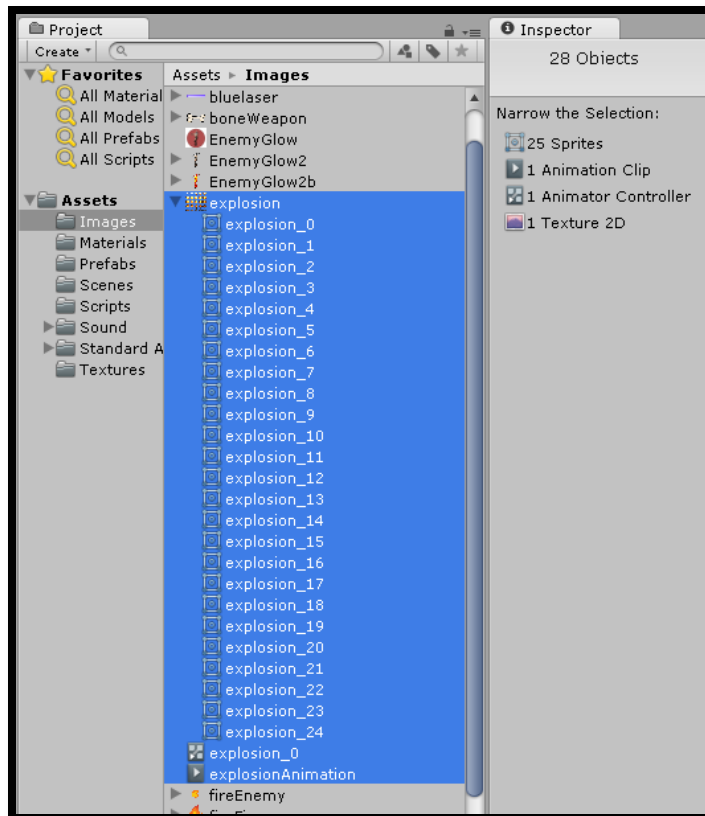
C WeaponsShooter ▶ M Awake
1 #pragma strict
2 //refers to prefabs
3 var BladeToShoot:Rigidbody;
4 var BulletToShoot:Rigidbody;
5 //to set up a counter to choose between the two weapons
6 var weaponChange:int=0;
7
8 function Start () {
9 }
10
11 //when pressing the altGr button, the laser will be shoot either Blades or Bullets
12 function Update () {
13
14     if (Input.GetKeyDown(KeyCode.AltGr))
15     {
16         if (weaponChange == 0)//set as weapon zero
17         {
18             Instantiate(BladeToShoot,transform.position,Quaternion.identity);
19         }
20         if (weaponChange == 1)//set as weapon one
21         {
22             Instantiate(BulletToShoot,transform.position,Quaternion.identity);
23         }
24     }
25
26 //pressing left alt key to change between the two weapons
27 if(Input.GetKeyDown(KeyCode.LeftAlt)){
28     weaponChange += 1; //go up by one
29 }
30
31 if(weaponChange >= 2){//change between the 2 weapons
32     weaponChange = 0; //start at weapon zero
33 }
34 }
35

```

This script was dragged onto the player object. Prefabs were dragged onto the script.

## Task 11

**(M3.1) Use the appropriate structure and approach to creating appropriate visual cues in interactive contexts by adding animated explosions to the game.**



Here we can see how I had animated the explosion by using the sprite animation technique in unity. The explosion will be triggered once a weapon shot by the player will hit an enemy. Below we can find some coding of how I had implemented this accordingly.

```
//function that is triggered when thunder hits the enemy
function OnTriggerEnter(other:Collider)
{
    //if the thunder hits an enemy, we will do the following two things:
    //Destroy the enemy
    //Destroy the thunder

    if((other.tag == "enemy") || (other.tag == "enemyCharacter")){

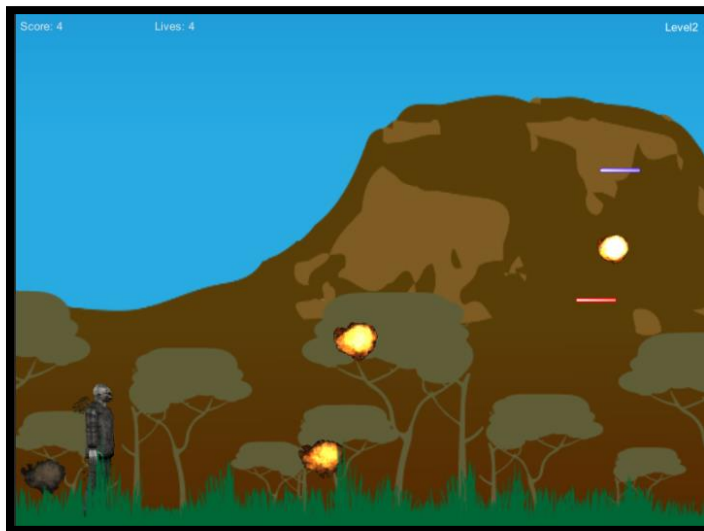
        //score=score+5;
        PlayerController2.score+=5;
        PlayerController3.score+=5;
        //destroy the enemy
        Destroy(other.gameObject);
        //destroy the bone
        Destroy(this.gameObject);
        //create the explosion at the enemies's position
        Instantiate(explosion,transform.position,transform.rotation);
    }
}
```

Instantiate(explosion,transform.position,transform.rotation);

```
Debug.Log("Entered Enemy");
}
```

This script will destroy the explosions once they leave the screen.

```
function OnBecomeInvisible()
{
    Destroy(this.gameObject);
}
```



## Task 12

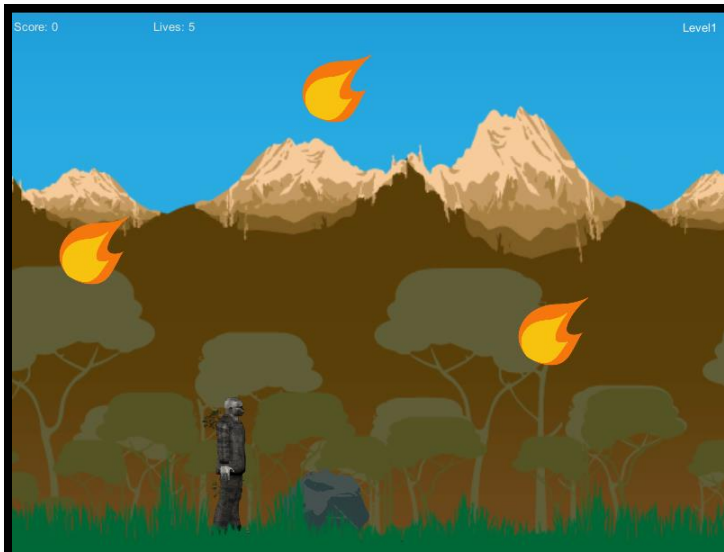
**(M2.1) Apply relevant theories and techniques to creating an interactive interface project by writing a short paragraph explaining specifically what genre of game you have created.**

Include references to similar games which have been created. Show similarities using 2 screenshots. (4 in total, 2 of your game, 2 of the game you used as a reference).

My game will fall under the adventure genre because the story is about a protagonist who journey is to accomplish something. The protagonist has a mission and faces obstacles to get to his destination, trying to save the forest. Since my game consists of a side scrolling game, I was inspired from the old and famous Mario Bros game. I was a big fan of this game when I was a child; therefore I made up my mind to make something similar. Below we can see some screen shots showing the similarity.



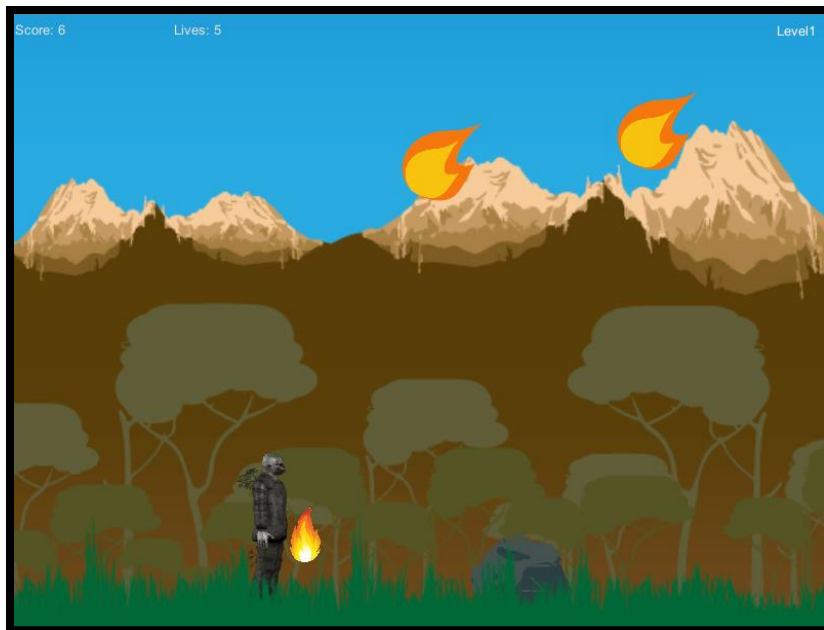
Mario Bros Screenshot showing player with obstacles and enemies.



Similar gameplay but instead of coins the player must collect fire. The setup is identical.



Mario modified with a powerup, changes clothes.



Player with shield power up. The idea was taken from the above, instead of changing the clothes, creating shield in front of the character.

### Game Play:

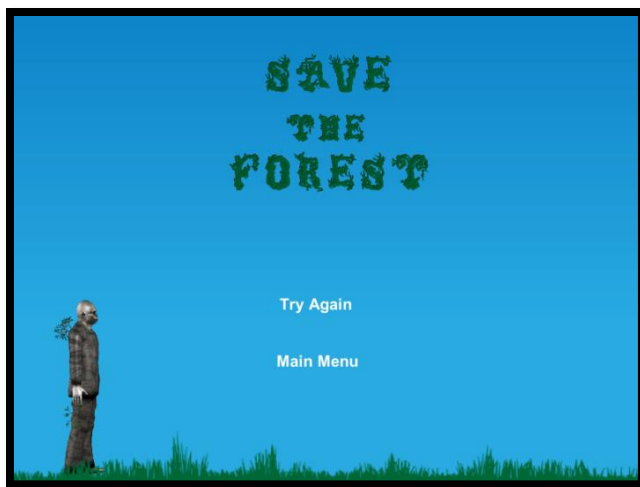
[https://github.com/JustinZammit/JUSTIN\\_ZAMMIT\\_2HND1/commits/master](https://github.com/JustinZammit/JUSTIN_ZAMMIT_2HND1/commits/master)

## Task 13

**(D1.1) Show that realistic improvements have been proposed against defined characteristics for success by explaining how you would update your game to reflect one of the finished titles you used as your inspiration in Task 12.**

What features are you missing and how would you go about implementing them?  
To achieve this task, identify the missing features and implement them in your game.

In every game when a user loses a level in a gameplay, a try again feature is always present. Thus, since my game doesn't have this feature, I am going to implement it. This will allow the user to re try that particular level from scratch.



When the user will pass the whole game, in every game we find sound which compliments the ending. Since my game doesn't have this feature, I made up my mind to implement this. Below we can find a screenshot when the user passes my game where sound will be present.

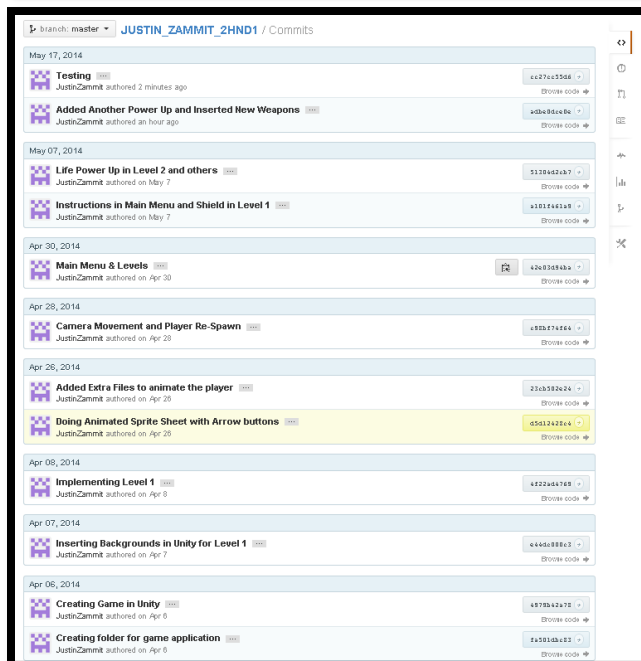
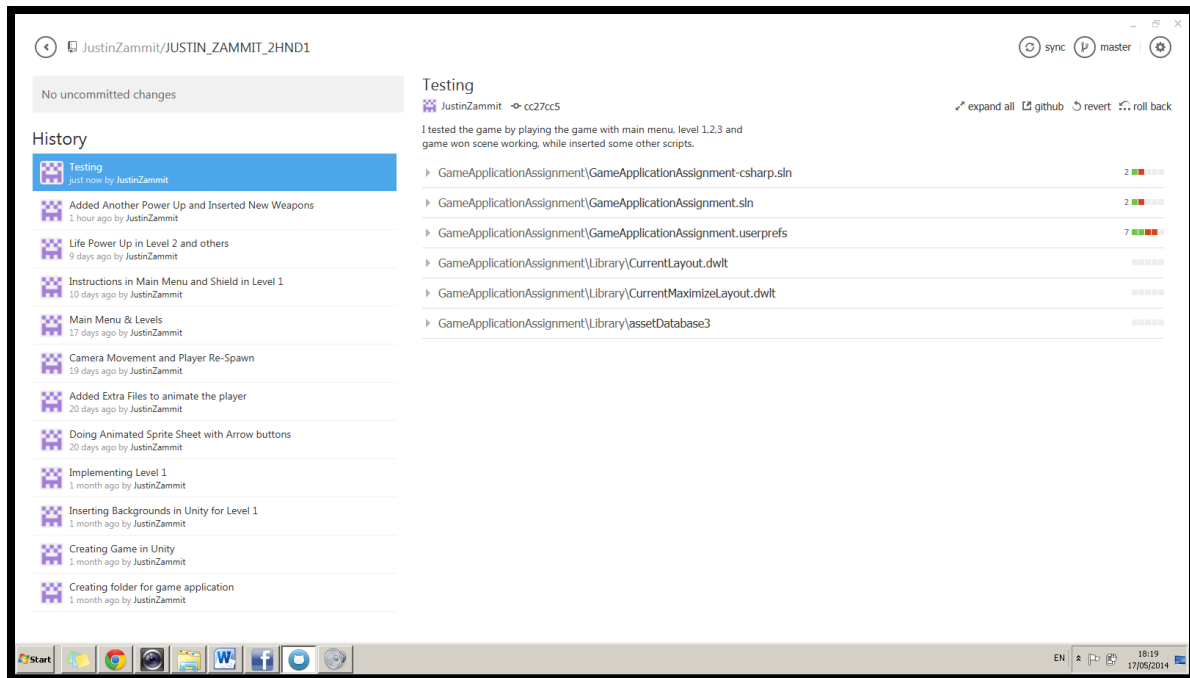




## Task 14

**(D2.1) Show activities have been managed by including a link to your Github page. All commits must be FULLY documented with the exact features that were implemented in that commit. The sequence of commits must CLEARLY show how the game was implemented over a realistic time span.**

To achieve this task, include a link to the git project that you have been using as part of your project under this task heading. You must have at least 8 fully documented commits with a realistic sequence of commit dates to achieve this criterion.



**Link to GitHub:**

[https://github.com/JustinZammit/JUSTIN\\_ZAMMIT\\_2HND1/commits/master](https://github.com/JustinZammit/JUSTIN_ZAMMIT_2HND1/commits/master)



## Task 15

### **(D3.1) Demonstrate convergent/lateral/creative thinking by performing the following task:**

Show how professional game development may be achieved by explaining how a professional team of different specialists would have worked on your game. Identify a list of job descriptions in the computer game development context and map them to specific parts of your game development.

To achieve this task, you must identify 4 different job titles and map them to 4 different elements in your game.

A game can take a while in order to be completed. There are many steps before a game is released, from generating thoughts and characters to encoding and testing. There are various responsibilities involved at each stage, a professional team of different specialists is needed.

**Artist** – will create the game's visual characters, objects and scenery, and producing concept art and drawings (storyboards) at the planning stage. This person will come up with a concept generated from my original knowledge.

**Game Designer** – deciding what my game looks like and how it plays. He would work on my existing concept. Perhaps the game designer should choose my setup for the game accordingly such as background images, power ups, and weapons.

**Animator** – bringing the characters, objects and scenery to life with computer modelling and animation software during the production stage. Perhaps when animating the character, the animator is fundamental since he knows how to implement the key movement in an animation. In my game I also have an enemy boss where it needs to be animated accordingly, so the animator will need to do this.

**Programmer** – creating the code to make the game work. This person could specialise in developing graphics, artificial intelligence (AI), or gameplay software. By using unity for my game, this person will take care all of the java scripts needed to implement my game. Testing will also be fall under the programmer's responsibility.

All of these roles would report to a producer or project manager, who oversees the whole process and makes sure that the finished game is completed on time. This job could be carried by myself in order to check that the gameplay will be completed on time. (Computer Games Developer, 2012)

## Bibliography

- Charbonneau, M. (2013, February 22nd). *Choosing the Perfect Game Engine*. Retrieved 2014, from [http://www.gamasutra.com/blogs/MikoCharbonneau/20130222/187185/Choosing\\_the\\_Perfect\\_Game\\_Engine.php](http://www.gamasutra.com/blogs/MikoCharbonneau/20130222/187185/Choosing_the_Perfect_Game_Engine.php)
- Collider.OnTriggerEnter(Collider)*. (2014). Retrieved 2014, from <https://docs.unity3d.com/Documentation/ScriptReference/Collider.OnTriggerEnter.html>
- Computer Games Developer*. (2012). Retrieved May 18th, 2014, from <https://nationalcareersservice.direct.gov.uk/advice/planning/jobprofiles/Pages/computergamesdeveloper.aspx>
- DeLoura, M. (2009, October 12th). *Choosing the right game engine*. Retrieved 2014, from <http://www.slideshare.net/markdeloura/choosing-a-game-engine>
- Esenthel*. (n.d.). Retrieved 2014, from <http://esenthel.com/?id=compare>
- MonoBehaviour.Start()*. (2014). Retrieved April 25th, 2014, from Unity: <http://docs.unity3d.com/Documentation/ScriptReference/MonoBehaviour.Start.html>
- MonoBehaviour.Update()*. (2014). Retrieved 2014, from <https://docs.unity3d.com/Documentation/ScriptReference/MonoBehaviour.Update.html>