

IS71030A: Introduction to Programming for Games and Interactive Graphic(2018-2019)

Student Name: Seyednima (Nima)
Student Surname: Jamalianardakani
Student Number: 33320659
Video report higher quality [link](#)

Assignment 1: 2D Unity game
Project title: BIRDY

Introduction:

My game idea is a combination of Ludom Dare game themes. The game is an endless runner/ Platformer type game. The Platformer aspect of the is similar to the “A Small World”[\[1\]](#) game themes, the character movement is familiar to what we find in most small world/platformer game. In the game you are in control of the character which is a bird. You have to move around and find your way to trough the obstacles. The games also represent the “Running out of space”[\[2\]](#) aspect of Ludom game theme, the feelings you get when you need to find your way trough the obstacles is ver similar to running out of space games. You need to make the best use of space you have, in order to mover forward. In the game, the player moves forward on the X axis and you need to dodge the obstacles that appear in front of you, by going up and down. If you press the UP arrow key you go up and if you press DOWN arrow key you go down. The bird movement speed stays the same. The goal of the game is to get as far as you can. Each time you pass an obstacle you will get one point. The player start the game with 3 hearts. Each time the player collides with another obstacles you will lose one life. If your number of hearts drop to zero, the game is over. In the game you can not get extra hearts. The goal of the game is to bring fun and entertaining experience to the players, with a simple yet addictive gameplay. I also added animation for player, sound effects and camera animations to make the game more alive and engaging for the players. Two of the game which I was inspired by are Flappy Bird[\[3\]](#) and Temple Run[\[4\]](#). Both of these games have simple games play yet they are engaging and addicting.

Design:

I also did research on popular endless runner games, for my games to get a better understanding on what are factors that make a good endless runner game. The first factors is the simple game player. All of current top endless runners games have simple gameplays, you either have to jump over obstacles or moves around the obstacles(simple left, right, up and down movements)[\[5\]](#). The second engaging factor in the endless runner game, is the smoothness of the game and how challenging are the obstacles that you need to surpass. Smooth animation ,character movement and challenges of the game, are what makes an endless runner game especial.

For my game at first, my idea was that, the character will move automatically and so the player need to only jump. I had platform automatically generated with some gape between them, the gaol of the game was the player need to jump over the gapes in order to score points .



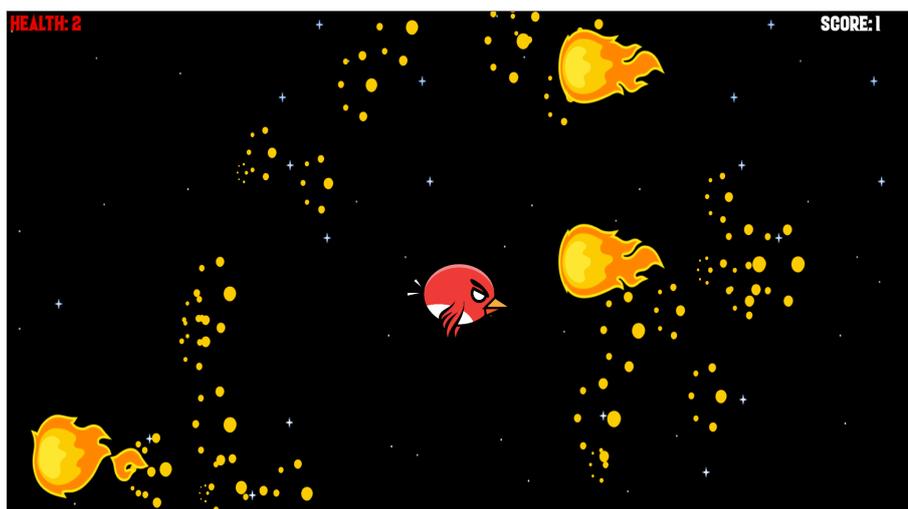
But I had issues with the game balancing, the game was either too difficult, or too simple. You needed to hit the space bar at exact time otherwise you would miss the platform. Or on the other hand I could have changed some of the game setting but then the game would become too simple and is was not much of a challenge. My main issues was platforms spawn system, it was not possible to balance the game properly.

So I changes my game idea a little bit in order to overcome these issue, the first change that I made was in character movements like before you move forward but you can't jump anymore instead you go up and down. And I changed the player character as well. Since the character movement was similar to flying I choose to have a bird instead of a cat as main character.



The next thing that I changed was obstacles design, instead of having platforms that you need to jump over, I added some obstacles that flies toward the player. In the game, the player need to find its paths trough these obstacles and go forward. Changing how the obstacles system works, helped me with the game difficulty balancing a lot. And made the much more enjoyable experience.

I also added elements that makes the game more alive and engaging. I implemented animation for both, the character and obstacles. I also added explosion particle effect and sound for my obstacles. So when the player collides with one of the obstacles, you see and hear the explosion. These elements makes the game more engaging and fun.



Implementation:

They are three different game scenes in my game, First one for the menu which appears when you start the game, the next scene is the game(Everything that is related to the game is stored here) and the final scene is how to play section (In the how to play section, I briefly talk game idea and how to game works).

In the game, two of the programme main section are player movement and obstacles generation. In the player movement, we set how much and how fast our player should move, the player script(Bird) also checks the player inputs and move the character accordingly, and also checks the character movements so the player will not able to go off of the screen. For the obstacles generation, I have defined three different generation patterns, and a generation point, so when the player passes the generation point, the programme will pick one of the obstacle patterns randomly and displays it. Having different obstacles patterns helped me balance my game a lot better compare to before.

One thing to note is that in the game the Bird actually dose not move forward, it only goes up and down, it's the obstacles and background that are moving toward the bird. This gives the effect that the bird is moving forward, however in reality the bird only moves on Y axis.



Execution:

In this part will be covering all of the scripts in my game, In total I have 10 scripts in my game which I am going to cover what they do one by one.

1- Main Menu:

- It is a script for the GameManger object
- Functions for loading game scenes
- Used for our menu buttons in order to load different game scenes

```
Main_Menu ▶ No selection
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5 public class Main_Menu : MonoBehaviour
6 {
7     // Use this for initialization
8     void Start()
9     {
10
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16
17     }
18     //Loading diffrent game scenes
19     public void PlayGame()
20     {
21         SceneManager.LoadScene("Game");
22     }
23     public void HowToPlay()
24     {
25         SceneManager.LoadScene("HowToPlay");
26     }
27     public void menu()
28     {
29         SceneManager.LoadScene("Menu");
30     }
31
32 }
```

2- Background:

- Script of our Background(Quad) game object
- Creating endless background image

```
Background ▶ No selection
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Background : MonoBehaviour
6 {
7     //defining for variables
8     public float Speed;
9     Material material;
10    Vector2 offset;
11    public int Velocity_X;
12
13    private void Awake()
14    {
15        //getting access to the material which is attached to the Background(Quad)
16        material = GetComponent<Renderer>().material;
17    }
18
19    // Use this for initialization
20    void Start()
21    {
22        //defining the offset
23        offset = new Vector2(Velocity_X * Speed, 0);
24    }
25
26    // Update is called once per frame
27    void Update()
28    {
29        //adding offset value to main texture offset
30        material.mainTextureOffset += offset * Time.deltaTime;
31    }
32
33 }
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

3- Bird:

- Script for Bird game object
- Bird control script (UP & DOWN)
- Checks Bird position so it won't go off screen
- Displays Bird Health
- Checks if the game is over (health \leq 0)

```
Bird ▶ Update()
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5 using UnityEngine.UI;
6
7 public class Bird : MonoBehaviour
8 {
9     // defining for variables ...
10
11     private Vector2 New_Position;
12     public float UP_Key_Power;
13     //Bird movment speed
14     public float Speed;
15     //For setting the size of screen
16     public float Screen_Max_Hight;
17     public float Screen_Min_Hight;
18     //Health display and counting
19     public int Health;
20     public Text Health_Display;
21     public GameObject gameover;
22     // Use this for initialization
23     void Start()
24     {
25
26     }
27
28     // Update is called once per frame
29     void Update()
30     {
31         //Displaying our text
32         Health_Display.text = "Health: " + Health.ToString();
33         if (Health <= 0)
34         {
35             gameover.SetActive(true);
36             Destroy(gameObject);
37             //stopping the game
38             Time.timeScale = 0;
39         }
40         //Bird changing position
41         // * Time.deltaTime -> The player moves with same speed in different computers(Strong or weak)
42         transform.position = Vector2.MoveTowards(transform.position, New_Position, Speed * Time.deltaTime);
43         //check if the player has pressed UP arrow key
44         //Also check that the Bird doen't go off screen
45         if (Input.GetKey(KeyCode.UpArrow) && transform.position.y < Screen_Max_Hight)
46         {
47             //Changing only Y position of the bird
48             //Position X stays the same cause we only want to move our,character up
49             New_Position = new Vector2(transform.position.x, transform.position.y + UP_Key_Power);
50             //transform.position = New_Position;
51         }
52         //check if the player has pressed DOWN arrow key
53         //Also check that the Bird doen't go off screen
54         else if (Input.GetKey(KeyCode.DownArrow) && transform.position.y > Screen_Min_Hight)
55         {
56             //Same fuction as UP key the only difference is that we changed to pluse to minus
57             //transform.position.y - UP_Key_Power
58             //Position X stays the same
59             New_Position = new Vector2(transform.position.x, transform.position.y - UP_Key_Power);
60             //transform.position = New_Position;
61         }
62     }
63 }
64
```

4- Fire ball:

- Script for fire ball prefab
- Setting fire ball speed
- Fire ball Collider checker
- Display explosions effect
- Camera shake animation

```
Fire_Ball ▶ OnTriggerEnter2D(Collider2D collision)
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Fire_Ball : MonoBehaviour
6 {
7     //How much health you lose
8     public int Damage;
9     //How fast the fire ball is moving
10    public float Speed;
11    //Praticle effect for when Bird colides with fire ball
12    public GameObject Particle_Effect;
13    private Camera_Shake shake;
14    public GameObject Sound_Effect;
15    // Use this for initialization
16    void Start()
17    {
18        //getting component for camera shake
19        shake = GameObject.FindGameObjectWithTag("Camera_Screen_Shake").GetComponent<Camera_Shake>();
20    }
21
22    // Update is called once per frame
23    void Update()
24    {
25        //Fire ball move function
26        transform.Translate(Vector2.left * Speed * Time.deltaTime);
27    }
28
29    //Function get's called whenever fire ball collided with another object
30    private void OnTriggerEnter2D(Collider2D collision)
31    {
32        //check to see if the fire ball has collided with bird
33        if (collision.CompareTag("Bird"))
34        {
35            //Didn't work
36            //There is bug for explosion sound effect
37            //When you start the game, you can hear one explosion, even tough the Bird has not collided with fir ball
38            //I trited to fix it, with time delay funcnacion.But couldn't get it to work
39            //Invoke(Instantiate(Sound_Effect, transform.position, Quaternion.identity),1);
40            //sound effect
41            Instantiate(Sound_Effect, transform.position, Quaternion.identity);
42            //Display particele effect when fire ball collided with bird
43            Instantiate(Particle_Effect, transform.position, Quaternion.identity);
44            //Player loses health
45            collision.GetComponent<Bird>().Health -= Damage;
46            //camera shake
47            shake.CameraShake();
48            //Destroying game object
49            Destroy(gameObject);
50        }
51    }
52 }
53 }
```

5- Fire Ball Generator:

- Script for Fire Ball Generator game object
- Generates fire ball based on time and position
- Randomises generation patterns

```
Fire_Ball_Generator ▶ No selection
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Fire_Ball_Generator : MonoBehaviour
6  {
7      //Array of generation patterns for Fire ball
8      public GameObject[] Fire_Ball_Patterns;
9      private float Generation_Time;
10     //Fire ball generation time
11     public float Start_Time_Generation;
12     public float Decrease_Time;
13     //Minimum time for to generate
14     public float Minimum_Time;
15     // Use this for initialization
16     void Start()
17     {
18
19     }
20
21     // Update is called once per frame
22     void Update()
23     {
24         if (Generation_Time <= 0)
25         {
26             //randomizing fire ball patterns
27             int random = Random.Range(0, Fire_Ball_Patterns.Length);
28             //fire ball generation
29             //randomizing the pattern
30             Instantiate(Fire_Ball_Patterns[random], transform.position, Quaternion.identity);
31             //adding delaying the auto generation
32             Generation_Time = Start_Time_Generation;
33             if (Start_Time_Generation > Minimum_Time)
34             {
35                 Start_Time_Generation -= Decrease_Time;
36             }
37         }
38         else
39         {
40             Generation_Time -= Time.deltaTime;
41         }
42     }
43 }
44
```

6- Fire Ball Destroyer:

- Script of Fire Ball prefabs
- Destroys Fire Balls after certain amounts of time (8 second)

```
Fire_Ball_Destroyer ▶ No selection
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Fire_Ball_Destroyer : MonoBehaviour
6 {
7     public float Destroy_Time;
8     // Use this for initialization
9     void Start()
10    {
11
12    }
13
14    // Update is called once per frame
15    void Update()
16    {
17        //Destroys the game object which this script is attacht to after it reaches Destroy_Time
18        Destroy(gameObject, Destroy_Time);
19    }
20 }
21
```

7- Generation Point

- Script for Generation Patterns prefabs
- set the generation position

```
Generation_Point ▶ No selection
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Generation_Point : MonoBehaviour {
6     public GameObject Fire_Ball;
7     // Use this for initialization
8     void Start () {
9         //generation pos for fire ball
10        Instantiate(Fire_Ball, transform.position, Quaternion.identity);
11    }
12
13    // Update is called once per frame
14    void Update () {
15
16    }
17 }
18
```

8- Camera Shake:

- Script for Main Camera game object
- function for triggering camera shake animation

```
Camera_Shake ▶ No selection
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Camera_Shake : MonoBehaviour
6  {
7      public Animator Camera_Animation;
8      // Use this for initialization
9      void Start()
10     {
11     }
12
13
14     // Update is called once per frame
15     void Update()
16     {
17     }
18
19
20     public void CameraShake()
21     {
22         //triggers camera shake animation
23         Camera_Animation.SetTrigger("Shake");
24     }
25 }
```

9- Score;

- Script for Score Manager game object
- Displays Score
- Check if the Fire Ball has passed the Bird (if yes, it increase the score)

```
Score ▶ Update()
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class Score : MonoBehaviour
7  {
8      public int score;
9      //public int highscore;
10     public Text scoreDisplay;
11     //public Text highscoreDisplay;
12     // Use this for initialization
13     void Start()
14     {
15     }
16
17
18     // Update is called once per frame
19     void Update()
20     {
21         //setting the highscore
22         //if (score > highscore)
23         //{
24             //highscore = score;
25         //}
26         //changing score to string and display it
27         scoreDisplay.text = "Score: " + score.ToString();
28         //highscoreDisplay.text = "High Score: " + highscore.ToString();
29     }
30     //check to see if the bird has passed the obstacle
31     //I put a box at the end of screen
32     //It can detect the fireballs that are going out of screen
33     //that way I calculate the score
34     private void OnTriggerEnter2D(Collider2D collision)
35     {
36         //check to see if the fire ball has collided with bird
37         if (collision.CompareTag("FireBall"))
38         {
39             //increase the score by 1
40             score++;
41         }
42     }
43 }
44
```

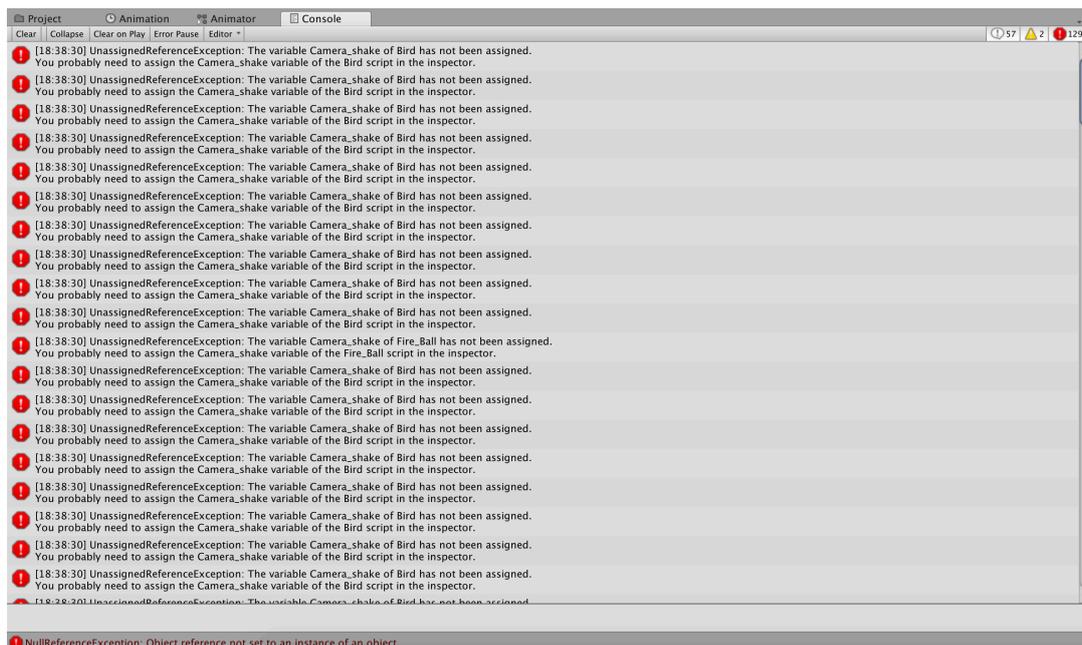
10- Gameover;

- Script of Gameover game object
- If player pressed space , Reload the game scene again
- and set the time to 1 again

```
Gameover ▶ Start()
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class Gameover : MonoBehaviour
7 {
8     //public GameObject high_score;
9     // Use this for initialization
10    void Start()
11    {
12
13    }
14
15    // Update is called once per frame
16    void Update()
17    {
18        if (Input.GetKeyDown(KeyCode.Space))
19        {
20            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
21            //HighScore, BUG
22            //HigheScore, is not working, I disabled it
23            //DontDestroyOnLoad(high_score);
24            //running the game again
25            Time.timeScale = 1;
26        }
27    }
28 }
29
```

Note that I have written the source of all of the game resources that I have used at the end of the report, and all of the references for code is attached to scripts, you can find it at the end of each script.

Problems:



I ran in to number of problems while making this game. I managed to fixed most of them, the game runs without any errors or warnings. However they were two part which I had problem with, and couldn't figure how to fixed it in time.

First one is creating a high score system, right now when the game restarts, I reload the game scene again. I could not figure out how to save my high score when reloading a game scene. I tried to fix this issue using DontDestroyOnLoad function for my high score but it did not worked.

The other issue is explosion sound bug which I did not have enough time to fix it. The explosion sound should only gets played when the Bird(player) collides with our Fire ball(obstacle). And it works all the time as intended except at the beginning of the game when you start the game you can hear the explosion sound, even though the Bird did not collide with Fireball. I believe the problem is from the original Fire Ball prefab position. I tried to fix this issue using Unity time passed function, for example run this code after 1 second of game start. However I could not get it to work.

Conclusion:

One of the main areas which I can still improve the game is, obstacle generation system. I think I can improve the game and make it more challenging by making more obstacle generation patterns. Overall in the process of making this game I have learned lots of new material about unity.

Bibliography:

- 1- [A Small world game](#)
- 2- [Running out of space](#)
- 3- [Flappy Bird](#)
- 4- [Temple Run](#)
- 5- Joe Hindy, ["10 best endless runner games for Android"](#)

Resources:

Bird(character) sprite sheets: <https://opengameart.org/content/game-characters-flying-birds-attack-sprite-sheets>

Background Picture: <http://m.bizhitupian.com/wall/36926.html>

Fire Ball sprite sheets: <https://opengameart.org/content/animated-traps-and-obstacles>

Explosion sound effect: <https://www.freesoundeffects.com/free-sounds/explosion-10070/>

Background Music: <https://www.proudmusiclibrary.com/en/tag/8-bit>

Font(ONE SLICE): <https://www.dafont.com>