

OddBall

Το OddBall είναι ένα εντελώς αυτοσχέδιο παιχνίδι το οποίο φτιάχτηκε πλήρως (design και gameplay) μέσα σε μόνο μερικές ώρες. Οι μηχανισμοί του μοιάζουν αρκετά με το Color Switch, αυτό το οποίο το διαφοροποιεί είναι το ότι πρέπει ο παίκτης να αποφύγει τα εμπόδια και να πιάσει τα ScorePoints που εμφανίζονται τυχαία πάνω στην σκηνή.

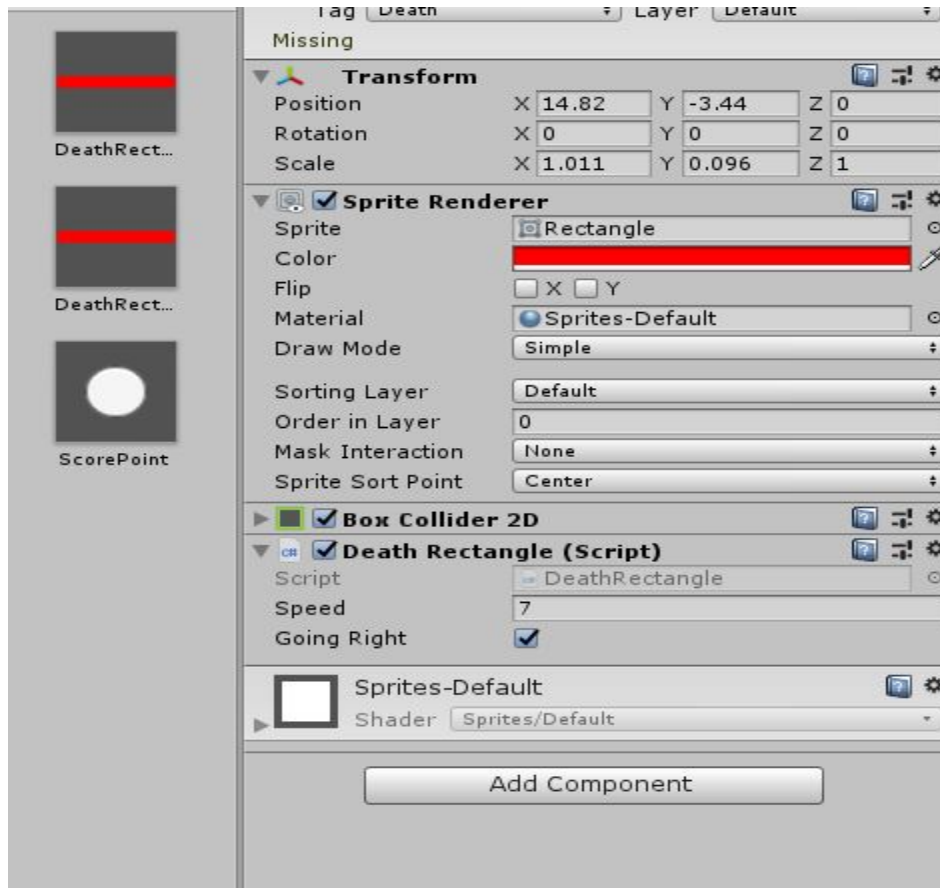
Όπως και με τα προηγούμενα Projects θα χρειαστούμε μια βασική δομή στην Unity πάνω στην οποία θα μπορέσουμε να λειτουργήσουμε. Δημιουργούμε φακέλους για

- Scenes
- Prefabs
- Scripts
- Sprites

Το συγκεκριμένο παιχνίδι δεν χρειάζεται animations ή τίποτα περίπλοκο το οποίο δεν έχουμε ξανακάνει. Δείχνει ωστόσο μερικές τεχνικές οι οποίες μπορούν να δώσουν το ελεύθερο σε οποιονδήποτε να φτιάξει ένα παιχνίδι το οποίο είναι κλιμακώμενο σε δυσκολία. Από τα βασικότερα μαθήματα είναι το **Instantiate** το οποίο μας αφήνει να δημιουργήσουμε GameObjects στην σκηνή μέσα από τον κώδικα. Επίσης θα δούμε και ένα πολύ απλό **Scoring System**

OddBall

Για αρχή, και αφού έχουμε φτιάξει τους φακέλους για να είμαστε σωστά οργανωμένοι, φτιάχνουμε τα πρώτα GameObjects, ειδικά αυτά που θα χρειαστούμε για prefabs.

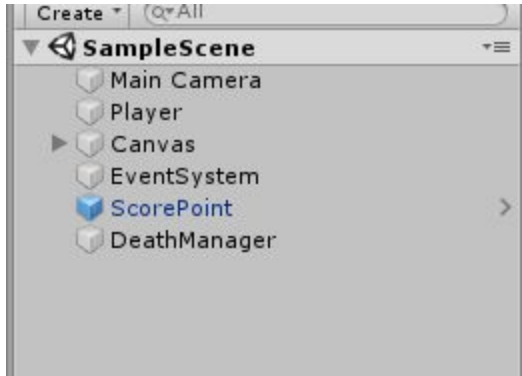


Τα εμπόδια πρέπει να σπάσουν σε 2 κατηγορίες, αυτά τα οποία βγαίνουν από τα αριστερά και αυτά που βγαίνουν από τα δεξιά. Το μόνο που αλλάζει μεταξύ τους είναι το που βρίσκονται αρχικά στον άξονα των X. Και τα δύο χρειάζονται το ίδιο script, Sprite Renderer για να μπορεί να φαίνεται σχηματικά/γραφικά στην οθόνη καθώς και Colliders (**τσεκάρετε το isTrigger στον collider**) για να μπορεί να γίνεται σωστά η κρούση. Επίσης Πρέπει να δημιουργήσουμε ένα Tag το οποίο υποδηλώνει ότι το αντικείμενο αυτό είναι εχθρικό, και άρα αν χτυπήσει με τον παίκτη θα πρέπει ο παίκτης να χάσει.

Επιπλέον θα χρειαστούμε ένα prefab για το ScorePoint για να μπορούμε να το εμφανίζουμε. Αυτό θα χρειαστεί έναν Collider (**τσεκάρετε το isTrigger στον collider**), ένα script, Sprite Renderer και ένα καινούργιο Tag που να υποδηλώνει ότι αυτό το αντικείμενο μπορεί να δώσει πόντους αν ακουμπήσει με τον παίκτη (και να καταστραφεί όταν γίνει αυτό)

Αυτό που θέλουμε είναι να εμφανίζονται εμπόδια στον παίκτη καθώς αυτός προσπαθεί να μαζέψει τα **ScorePoints**. Γι αυτό το λόγο θα χρειαστούμε και ένα GameObject στο hierarchy πέρα από τα προηγούμενα και τον παίκτη μας που να μπορεί να διαχειρίζεται την δημιουργία καινούργιων εχθρών παίρνοντας τα μέσα από τον φάκελο των prefabs.

OddBall



Αυτή την δουλειά θα την κάνει ο DeathManager.

Όσον αφορά τον παίκτη μας θα χρειαστεί ο,τι χρειάστηκαν τα προηγούμενα Objects με την μόνη διαφορά ότι δεν χρειάζεται isTrigger τσεκαρισμένο στον Collider του. Επίσης δεν χρειάζεται να δημιουργήσουμε καινούριο Tag καθώς η Unity φτιάχνει ένα Tag για τον player αυτόματα. Χρειάζεται απλά να το αναθέσουμε στο object. Εκτός αυτού θα χρειαστεί και ένα Rigidbody2D ο παίκτης έτσι ώστε να γίνεται προσομοίωση της φυσικής και άρα να έχει βαρύτητα.

OddBall

Εφόσον τελειώσαμε με τα διαδικαστικά ξεκινάμε με τον κώδικα του παίκτη:

```
Player.cs x
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  using UnityEngine.SceneManagement;
6  using UnityEngine.UI;
7
8  public class Player : MonoBehaviour
9  {
10
11     private Rigidbody2D rb;
12     private int score;
13
14     public GameObject scorePoint;
15     public GameObject scoreText;
16
17     public float jumpSpeed; // used
18
19     // Start is called before the first frame update
20     void Start()
21     {
22         score = 0;
23         rb = GetComponent<Rigidbody2D>();
24     }
25
26     // Update is called once per frame
27     void Update()
28     {
29         if (Input.GetMouseButtonDown(0)) {
30             rb.velocity = Vector2.up * jumpSpeed;
31         }
32     }
33
34     // Called when colliding with "isTrigger" collider objects
35     void OnTriggerEnter2D(Collider2D other) {
36         if (other.tag == "ScorePoint") {
37             score += other.gameObject.GetComponent<ScorePoint>().GetScore();
38             scoreText.GetComponent<Text>().text = " Score: " + score.ToString();
39             Destroy(other.gameObject);
40
41             int yRand = Random.Range(-4, 5);
42             Instantiate(scorePoint, new Vector2(0, yRand), transform.rotation);
43
44             Debug.Log("Score: " + score.ToString());
45         }
46
47         if (other.tag == "Death") {
48             Debug.Log("Game Over!");
49             Destroy(gameObject);
50             SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
51         }
52     }
53 }
54
```

OddBall

DeathRectangle (Τα εμπόδια μας):

```
DeathRectangle.cs x
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class DeathRectangle : MonoBehaviour
6  {
7
8      public float speed;
9      public bool goingRight;
10
11     private GameObject DM;
12
13     // Start is called before the first frame update
14     void Start() {
15         DM = GameObject.Find("DeathManager");
16     }
17
18     // Update is called once per frame
19     void Update()
20     {
21         if (goingRight) {
22             transform.position += new Vector3(speed * Time.deltaTime, 0, 0);
23         } else transform.position -= new Vector3(speed * Time.deltaTime, 0, 0);
24
25         if (transform.position.x > 15 || transform.position.x < -15) {
26             Destroy(gameObject);
27             DM.GetComponent<DeathManager>().setSpawn(true);
28         }
29     }
30
31 }
32
33
```

OddBall

DeathManager(Για να δημιουργούνται τα εμπόδια δυναμικά):

```
DeathManager.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class DeathManager : MonoBehaviour
6  {
7      public GameObject DeathRectangleLeft;
8      public GameObject DeathRectangleRight;
9
10     private bool spawn;
11     // Start is called before the first frame update
12     void Start() {
13         setSpawn(true);
14         CreateDeath();
15     }
16
17     void Update() {
18         if (spawn) CreateDeath();
19     }
20
21     public void CreateDeath() {
22
23         if (spawn) {
24             int left = Random.Range(1, 3);
25             int right = Random.Range(1, 3);
26
27             for (int i=0; i<left; i++) {
28                 int yRand = Random.Range(-4,5);
29                 Instantiate (DeathRectangleLeft, new Vector2(15, yRand), transform.rotation);
30             }
31
32             for (int i=0; i<right; i++) {
33                 int yRand = Random.Range(-4,5);
34                 Instantiate (DeathRectangleRight, new Vector2(-15, yRand), transform.rotation);
35             }
36
37             setSpawn(false);
38         }
39     }
40
41     public void setSpawn(bool spawn) {
42         this.spawn = spawn;
43     }
44 }
45
46
```

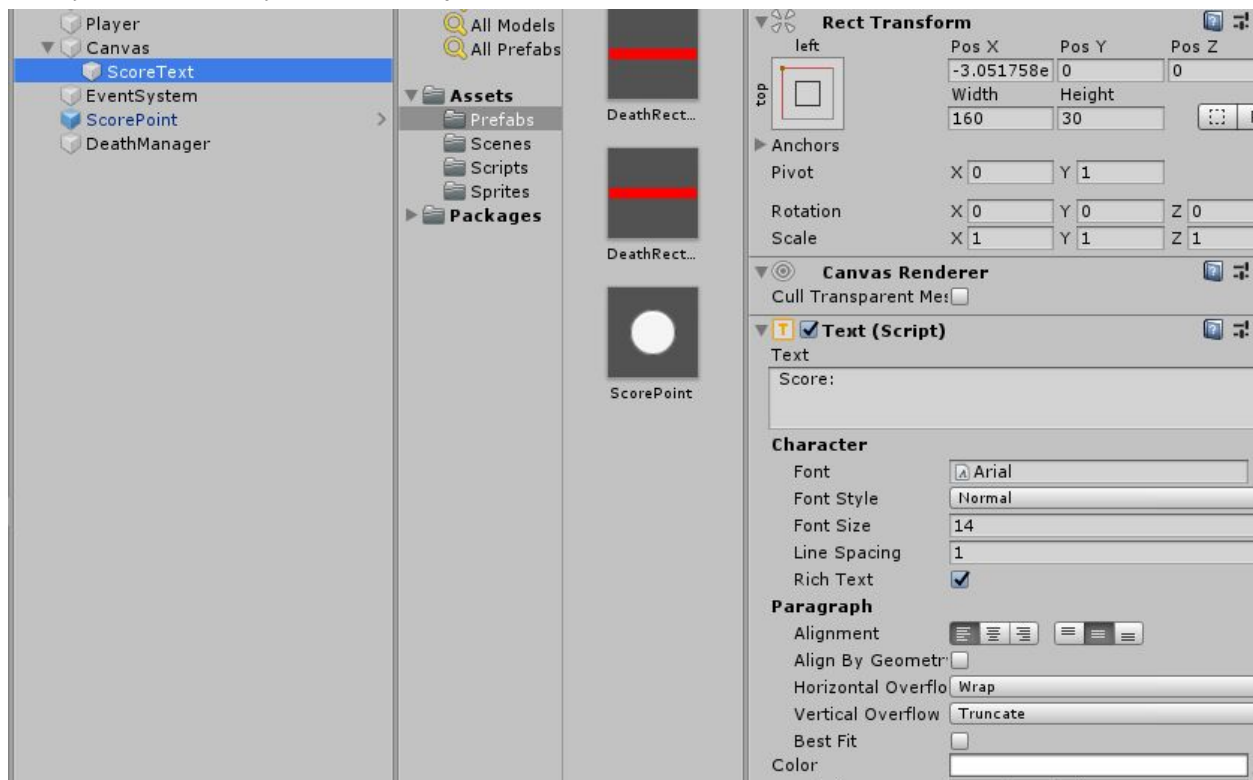
OddBall

ScorePoint:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class ScorePoint : MonoBehaviour
6  {
7      private SpriteRenderer sr;
8      private int score;
9
10     public Color colorBlue;
11     public Color colorOrange;
12     public Color colorGreen;
13
14     // Start is called before the first frame update
15     void Start()
16     {
17         sr = GetComponent<SpriteRenderer>();
18         SetRandomColor();
19     }
20
21     // Update is called once per frame
22     void Update()
23     {
24     }
25
26     void SetRandomColor() {
27         int index = Random.Range(0, 3);
28
29         switch(index) {
30             case 0:
31                 sr.color = colorBlue;
32                 break;
33             case 1:
34                 sr.color = colorOrange;
35                 break;
36             case 2:
37                 sr.color = colorGreen;
38                 break;
39         }
40
41         if (sr.color == colorGreen) {
42             score = 5;
43         } else if (sr.color == colorOrange) {
44             score = 10;
45         } else score = 15;
46     }
47
48     public int GetScore() {
49         return score;
50     }
51 }
52
53
```


OddBall

Τέλος για το Score θα χρειαστεί να φτιάξουμε στο hierarchy ένα UI object text το οποίο θα φτιάξει αυτόματα και έναν canvas. Ανάλογα με το που θέλουμε να βάλουμε την ένδειξη για το score υπάρχουν αρκετές ρυθμίσεις που μπορούν να μπουν στο συγκεκριμένο object. Στην δική μας περίπτωση θέλουμε να είναι πάνω αριστερά. Θέτουμε “Screen Space - Camera” στο render mode του Canvas και την main camera μας στο Render Camera setting. Επιλέγουμε το ScoreText και θέτουμε το transform πάνω αριστερά. Επίσης βάζουμε το κατάλληλο Text και αλλάζουμε αν χρειάζεται το font style/size κλπ.



Αφού γεμίσουμε όλες τις public μεταβλητές από τα scripts στους inspectors του κάθε object το παιχνίδι θα μπορεί να παίξει. Εννοείται ότι αυτό το εξετάζουμε και κατά την διάρκεια της δημιουργίας του project κάνοντας τα κατάλληλα debugs και τρέχοντας το παιχνίδι με την κάθε αλλαγή που πιστεύουμε ότι μπορεί να υπάρξει πρόβλημα.

Κάτι σημαντικό να θυμόμαστε είναι επίσης ότι στα prefabs μπορούμε να προσθέσουμε και άλλα πράγματα όπως είναι ο DeathManager και ο Player μας. Αυτό θα βοήθαγε ιδιαίτερα στην περίπτωση που είχαμε παραπάνω από μία σκηνές. Στην δική μας περίπτωση ωστόσο καθώς είναι ένα απλό project παραλήφθηκε.