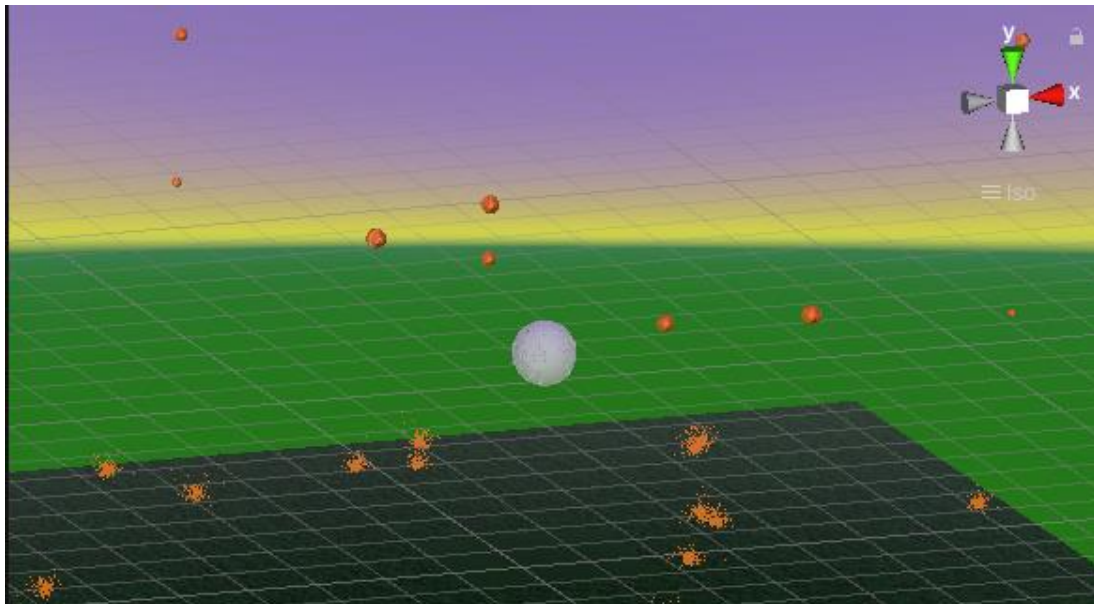


# Γραφικά Υπολογιστών και Συστήματα Αλληλεπίδρασης

Δεύτερη προγραμματιστική άσκηση με Unity 3D  
«Βρέχει Μπόλες!!»

Σολδάτου Χριστίνα Ολυμπία, Α.Μ. 4001  
23/12/2023

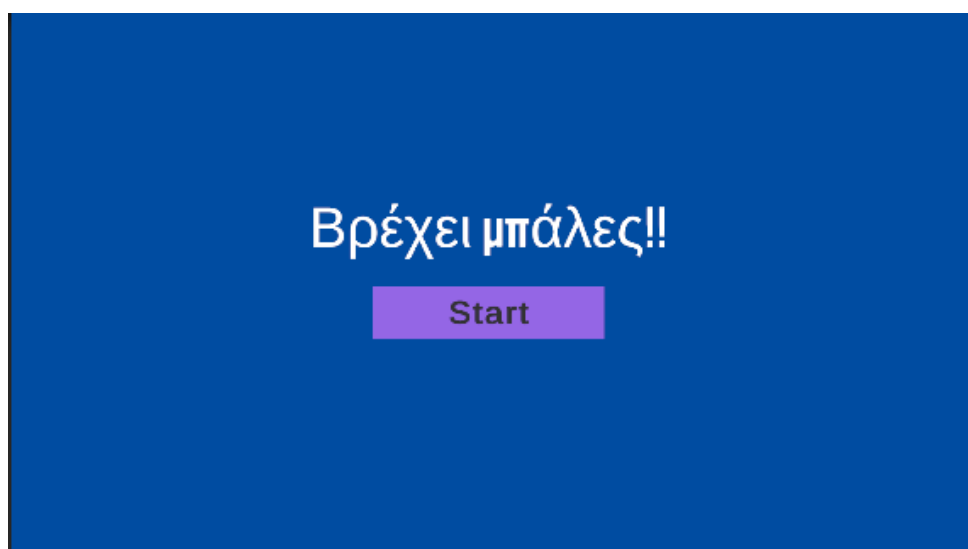
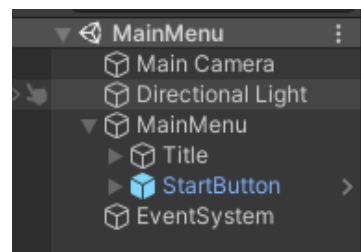


## Κεντρική Ιδέα Παιχνιδιού

Στόχος της άσκησης είναι η υλοποίηση μίας εφαρμογής - παιχνιδιού με την χρήση του Unity η οποία θα φορτώνει ένα σκάφος που ελέγχεται από τον παίκτη. Το σκάφος θα πετάει πάνω από το έδαφος και θα κινδυνεύει να χτυπηθεί από μπάλες φωτιάς. Εάν κάποια μπάλα χτυπήσει το σκάφος τότε το παιχνίδι τελειώνει.

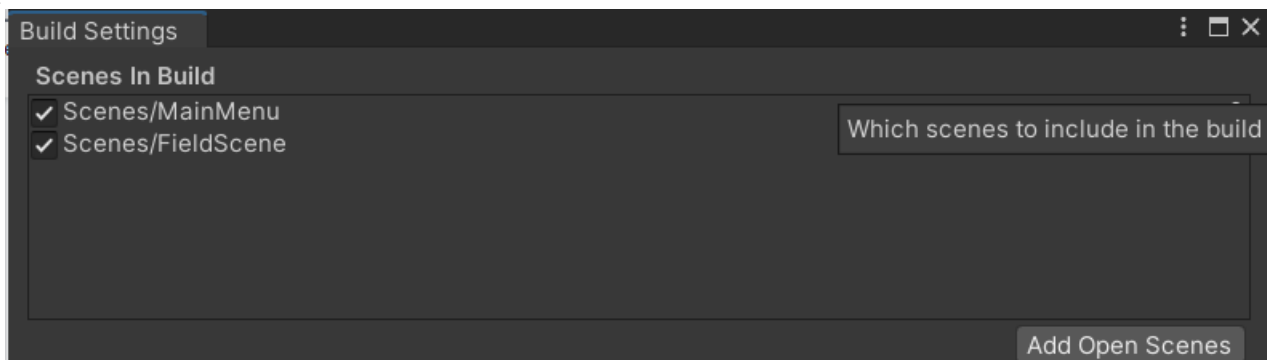
## Υλοποίηση Ερωτημάτων

(i) Αρχικά δημιουργήσα μια σκηνή και ρύθμισα το reference resolution 1024x768. Το background color άλλαξε σε μπλε και έβαλα τον τίτλο “Βρέχει μπάλες!!” με TextMeshPro. Αξίζει να σημειωθεί ότι για το Font Asset έκανα import ένα πακέτο με το όνομα \_GamesPlusJames Menu Assets. Τέλος, στη σκηνή τοποθέτησα ένα κουμπί “Start” και μετά από μερικές αλλαγές χρωμάτων στο κουμπί, έβαλα το script Main Menu στο οποίο αντιστοίχησα με On Click() το κουμπί με την μέθοδο.

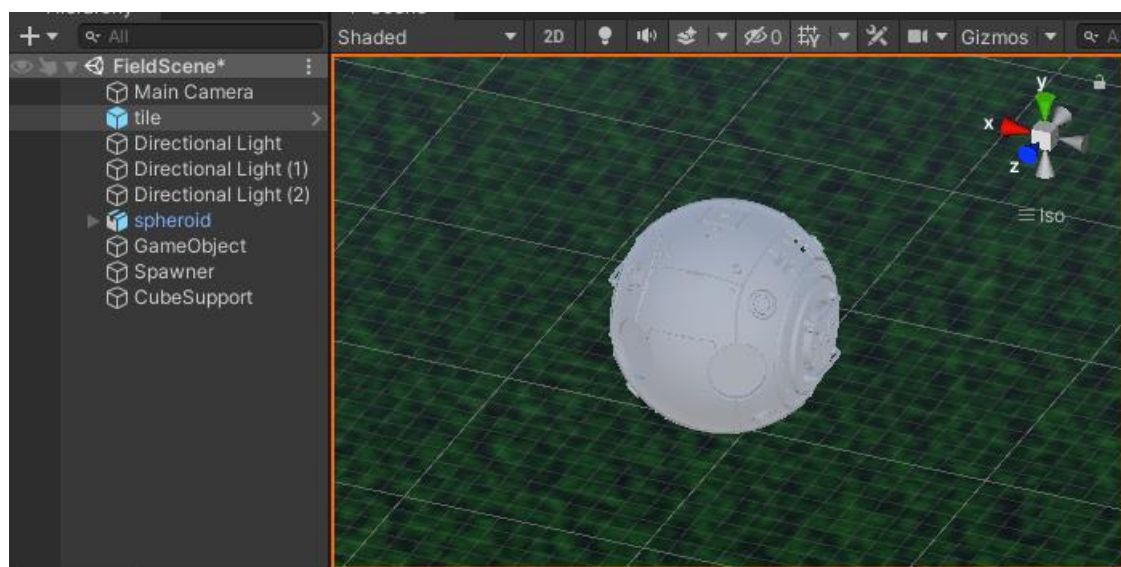
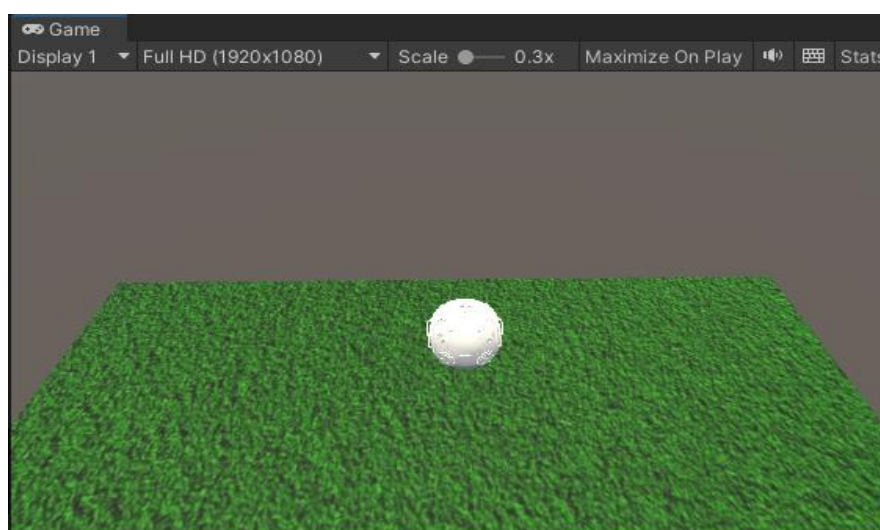


Η αρχική σκηνή.

Αφού πατηθεί το “Start” γίνεται μετάβαση στην επόμενη σκηνή η οποία είναι η FieldScene. Πηγαίνοντας στα Build Settings κάνω add και τις δύο σκηνές.

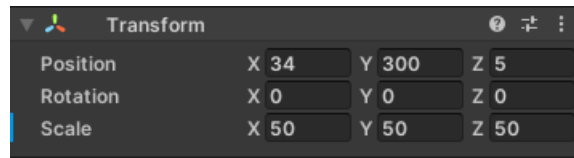


Ξεκινώντας την επόμενη σκηνή φορτώνεται το έδαφος και το σκάφος που κάναμε import.



Στην παραπάνω εικόνα φαίνονται όλα τα περιεχόμενα της σκηνής.

Το έδαφος έχει δημιουργηθεί με το Object tile και λόγω προβλημάτων, με το obj CubeSupport βάζω ένα ακόμα επίπεδο κάτω από το έδαφος το οποίο είναι διάφανο. Δημιούργησα το script GenerateTiles το οποίο φτιάχνει το object board κάθε φορά που πατιέται το play. Αρχικός σκοπός ήταν να υλοποιηθεί το έδαφος όπως απαιτεί η άσκηση δηλαδή με 80x80 τετράγωνα. Δυστυχώς αυτό δεν ήταν δυνατόν, καθώς δεν μπορούσε να το υποστηρίξει ο υπολογιστής μου. Έτσι το έδαφος δημιουργήθηκε με ένα μεγάλο τετράγωνο 1500x1500 και με υφή που πήρα από την προηγούμενη άσκηση. Το ύψος είναι  $y=0.001$  καθώς η υφή δεν εμφανίζονταν σωστά στο  $y=0$ . Στο τέλος στο σκάφος δόθηκαν τα ακόλουθα χαρακτηριστικά.



(ii) Για το 2ο ερώτημα δημιούργησα ένα script με όνομα sphere\_move στο οποίο ορίζω:

1. Με ποια πλήκτρα πραγματοποιούνται οι κινήσεις του σκάφους:

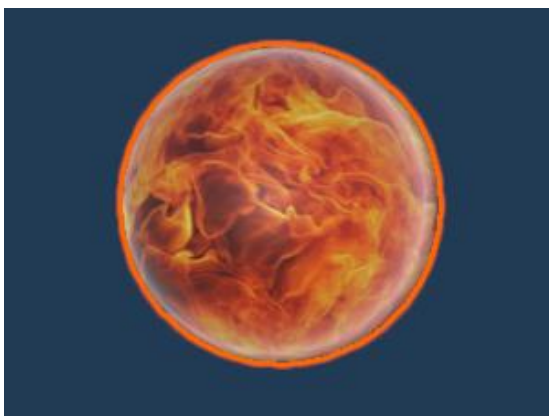
```
// Sphere Controls
if (Input.GetKey(KeyCode.A))
{
    Vector3 newPosition = new Vector3 (1f, 0f, 0f);
    transform.Translate(newPosition * Time.deltaTime * speed) ;
}
```

Ενδεικτικό παράδειγμα.

2. Την σταθερή ταχύτητα του σκάφους.
3. Τα όρια πτήσης του σκάφους με τον παρακάτω τρόπο:

```
Vector3 newPos = new Vector3(Mathf.Clamp(transform.position.x, -750, 750),
Mathf.Clamp(transform.position.y, 300, 700),
Mathf.Clamp(transform.position.z, -750, 750)) ;
transform.position = newPos;
```

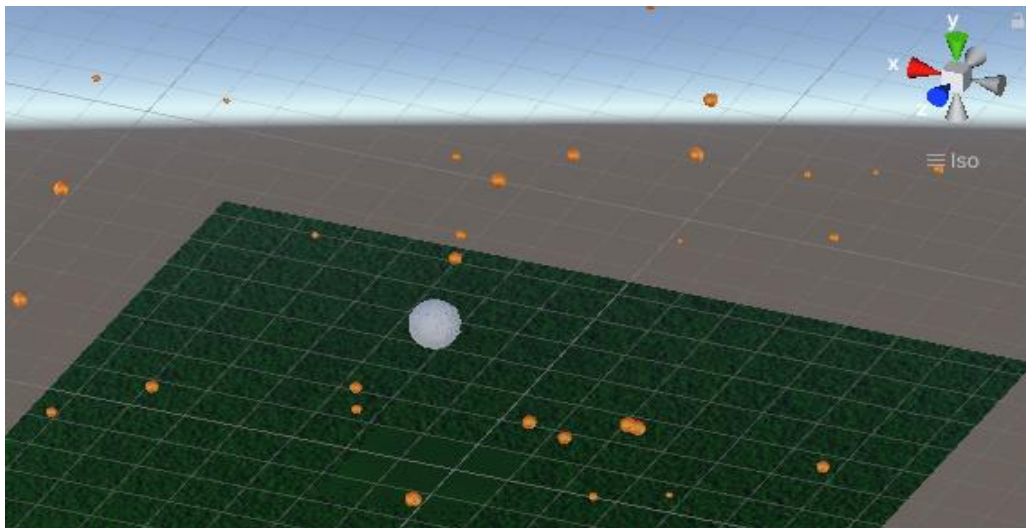
(iii) Για το ερώτημα αυτό δημιούργησα το Object fireball χρησιμοποιώντας sphere και βάζοντας texture φωτιάς από τα αρχεία στις 1Γ άσκησής μας. Τοποθέτησα το fireball στο φάκελο Prefabs και το διέγραψα από το Hierarchy. Στη συνέχεια, με το script RandomFireballs δηλώνω:





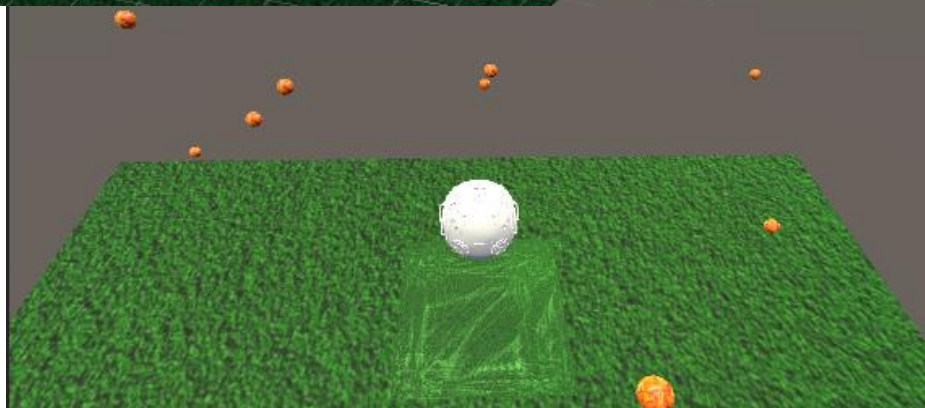
1. Ότι οι fireballs θα πρέπει να εμφανίζονται σε τυχαία σημεία εντός των ορίων του εδάφους.
2. Τα όρια ύψους που θα εμφανίζεται η fireball.
3. Η ακτίνα της σφαίρας.

Τέλος, δημιούργησα ένα object Spawner στο οποίο κάνω Add Component το script RandomFireballs και στο πεδίο Fireball Prefab έβαλα το fireball. Κάνω Add απαραίτητα και το Rigidbody γιατί χρειάζονται τα physics και πιο συγκεκριμένα η βαρύτητα.



Scene view

Game view



(iv) Για την υλοποίηση μίας απλής κάμερας έφτιαξα το script CameraControls στο οποίο δίνεται σε κάθε key μία συγκεκριμένη κίνηση στους άξονες x,y και z με σταθερή ταχύτητα.

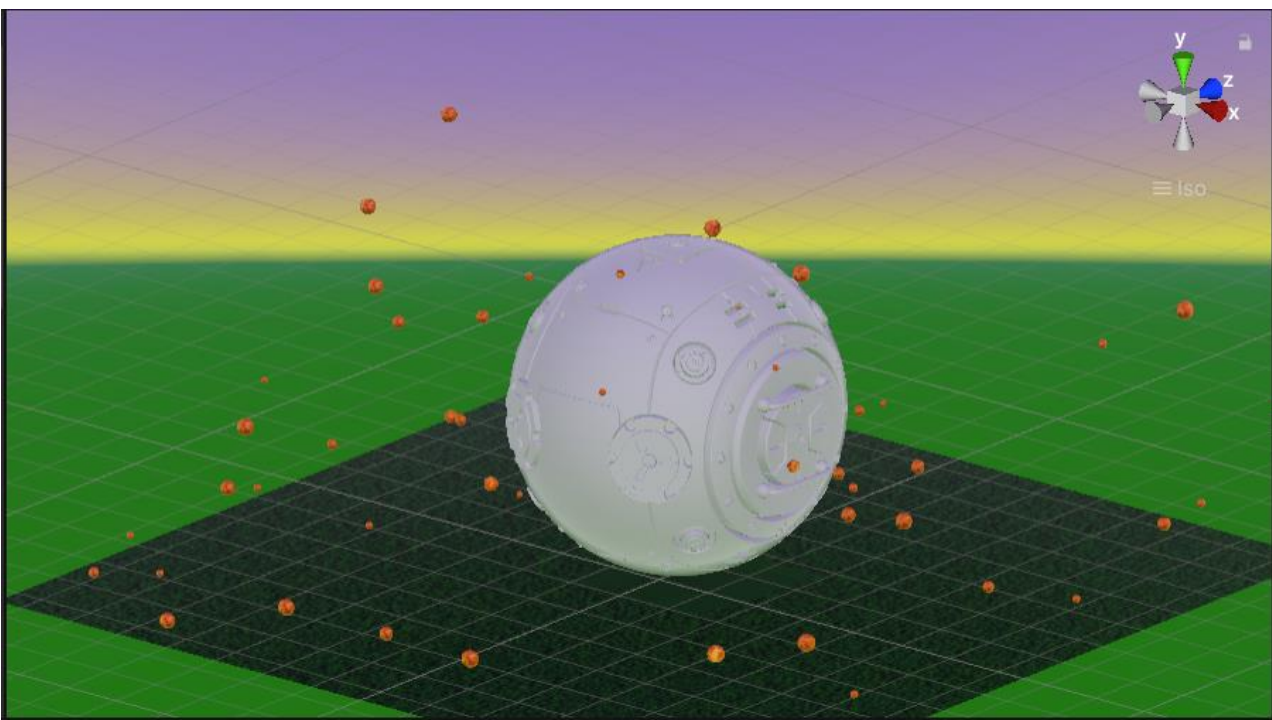
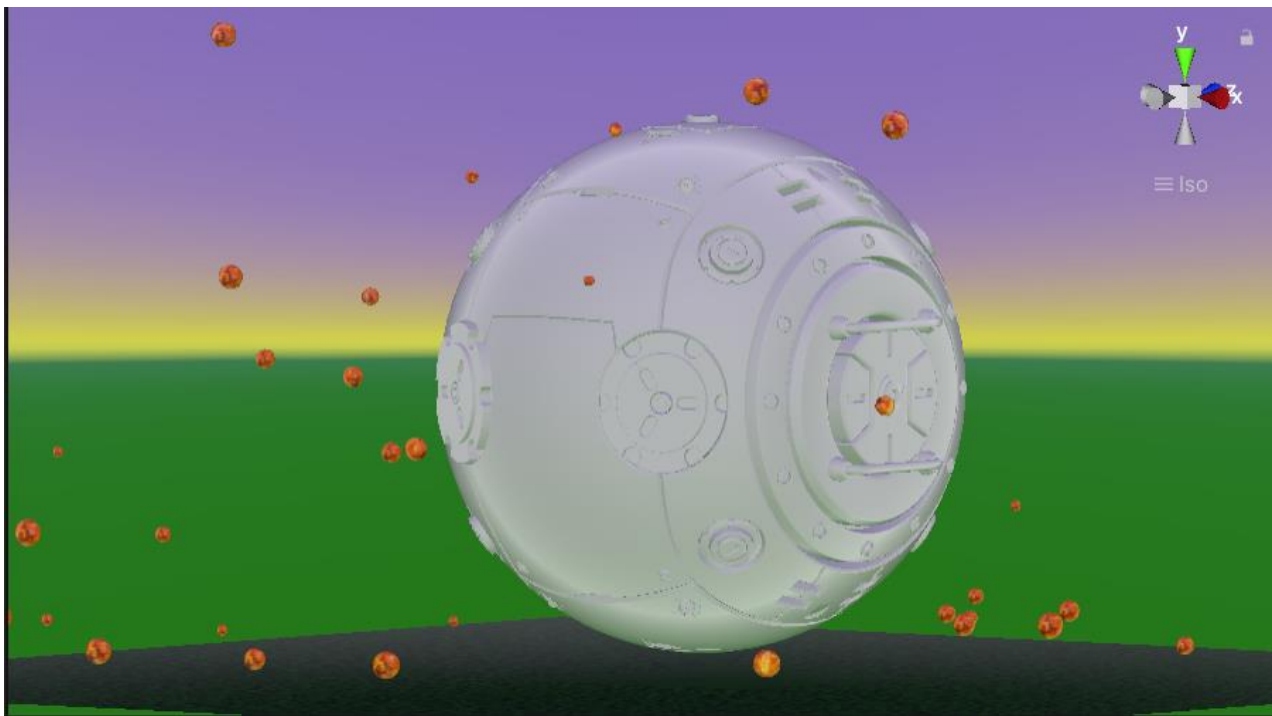
```
//Camera translation

//Go forward
if(Input.GetKey(KeyCode.UpArrow))
    transform.Translate(Vector3.forward * moveSpeed * Time.deltaTime);
```

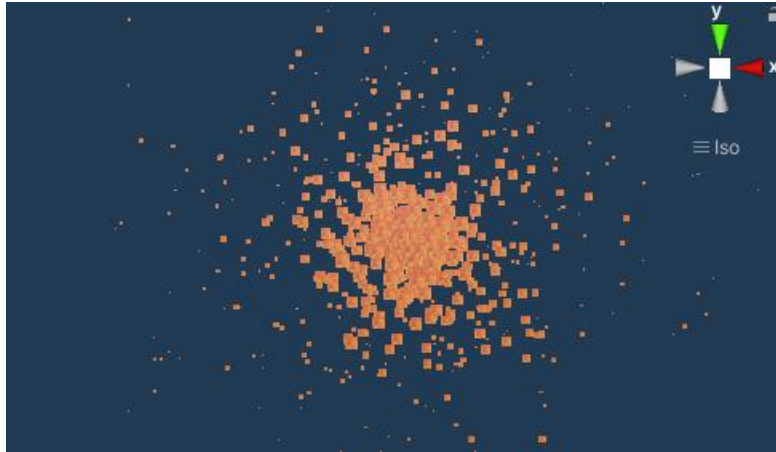
Ενδεικτικό Παράδειγμα

(v) Για το τελευταίο ερώτημα δημιούργησα το script Destroy το οποίο το προσθέτω στο Prefab fireball. Με το script αυτό πραγματοποιείται σύγκρουση και καταστροφή αντικειμένων. Η χρήση των tags δίνει τα εξής αποτελέσματα:

1. Tag “sphere”: Μόλις δει αυτό το tag θα πρέπει το fireball να καταστρέψει το spheroid και τον εαυτό του.
2. Tag “ground”: Μόλις δει αυτό το tag θα πρέπει το fireball να καταστραφεί μόνο τον εαυτό του. Εικόνες Fireballs – Spheroid – Ground πριν το collision



Εικόνες Fireballs – Spheroid – Ground πριν το collision



Αξίζει να σημειωθεί ότι έγινε χρήση particle systems τα οποία χρησιμοποιήθηκαν για τις εκρήξεις. Τέλος, με Build and Run έφτιαξα το εκτελέσιμο αρχείο (.exe) με όνομα Unity Project Ex.2.

## Γενικές Πληροφορίες

Πληροφορίες σχετικά με την υλοποίηση:

- Λειτουργικό Σύστημα: Windows 10
- Έκδοση Unity: 2020.3.42f1
- md5 checksum: b6195d1840499006a25d04b3fdf433ee
- Project link- Drive:

[https://drive.google.com/drive/folders/1u0bGdw3PP9uj7CSsaVrjdZwrHpc\\_4vxd?usp=sharing](https://drive.google.com/drive/folders/1u0bGdw3PP9uj7CSsaVrjdZwrHpc_4vxd?usp=sharing)

Links:

<https://www.youtube.com/watch?v=IbiwNnOv5So>

<https://www.youtube.com/watch?v=I4RFdxDIl2A>

[https://www.youtube.com/watch?v=76WOa6IU\\_s8](https://www.youtube.com/watch?v=76WOa6IU_s8)

<https://www.youtube.com/watch?v=ZAnub1kUMrM&t=237s>

<https://www.youtube.com/watch?v=Q6ffRoyqtq4&t=732s>

[https://www.youtube.com/watch?v=6eUt1yHvH\\_U](https://www.youtube.com/watch?v=6eUt1yHvH_U)

<https://www.youtube.com/watch?v=9n6pL6KFtes>

Ευχαριστώ πολύ για τον χρόνο σας.