ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΓΡΑΦΙΚΑ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ 2

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2022-2023

ΟΜΑΔΑ 4651-4786

ΓΙΑΚΟΥΜΑΚΗ ΕΛΕΝΗ, ΑΜ:4651

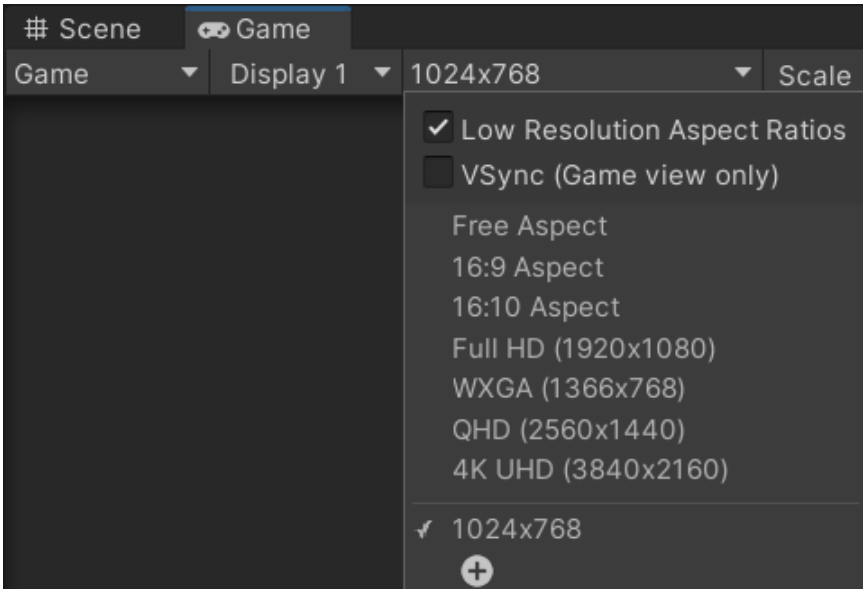
ΣΙΑΚΑΒΑΡΑ ΘΕΜΙΣΤΟΚΛΕΙΑ, ΑΜ:4786

23 Δεκεμβρίου 2022

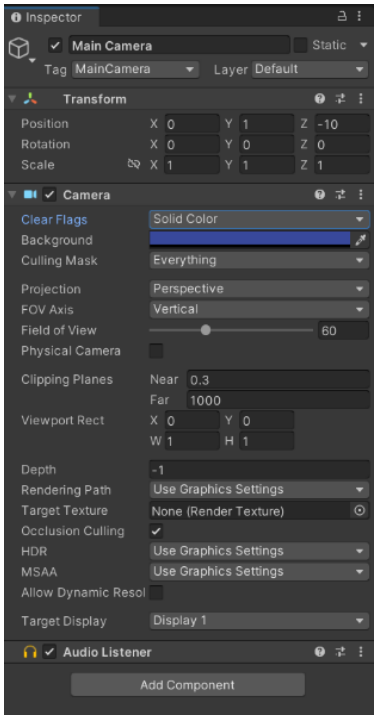
Περιγραφή της εργασίας

Ερώτημα i)

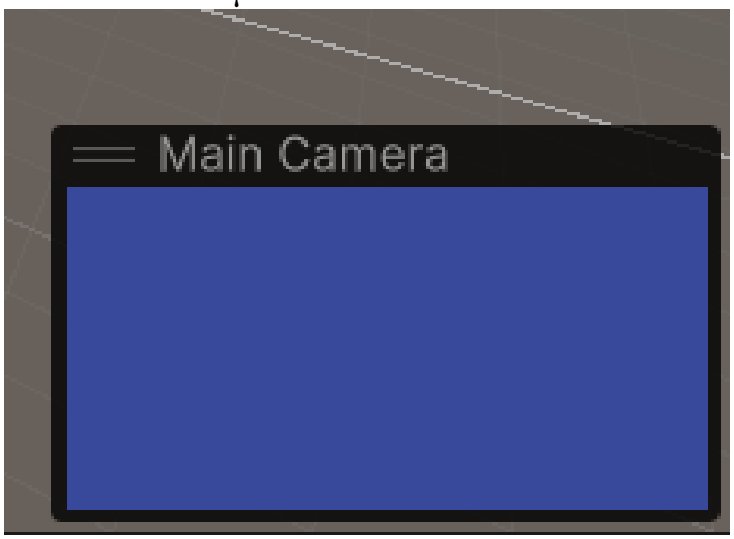
Αρχικά, για την ανάλυση της εφαρμογής Unity, στο section “Game”, ορίσαμε εμείς την ζητούμενη ανάλυση (1024x768) όπως φαίνεται παρακάτω:



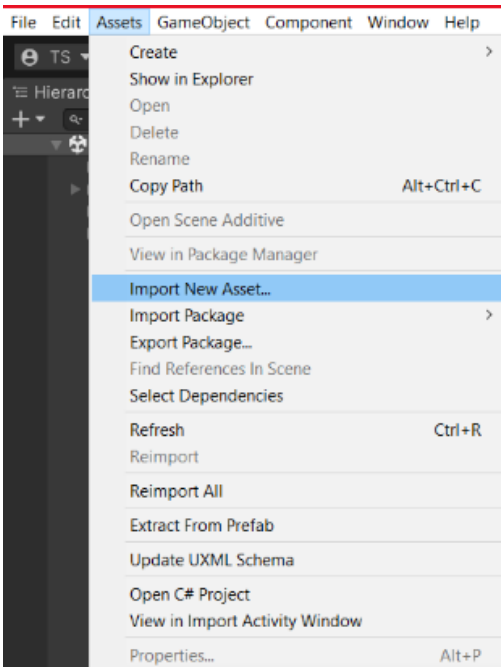
Το φόντο έγινε μπλε χρώμα ως εξής:



Με αποτέλεσμα:

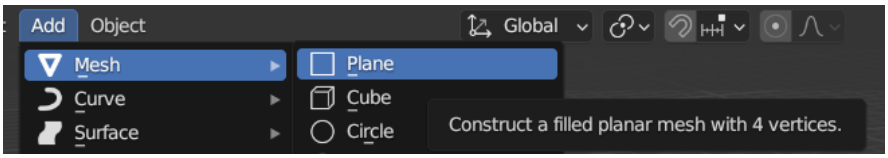


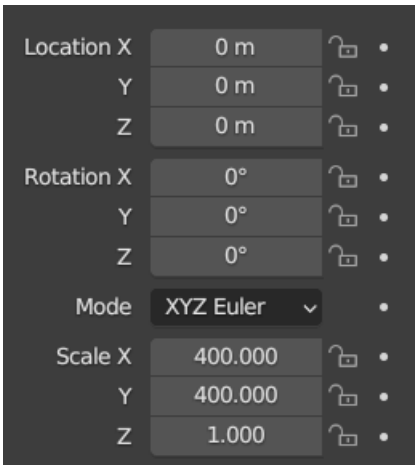
Τέλος, προσθέσαμε τα αντικείμενα spheroid.obj και floor2.obj. Και με drop and drag τα τοποθετήσαμε στη σκηνή μας. Επίσης με την ίδια μέθοδο εφαρμόσαμε και το texture "ground1" στο έδαφός μας.



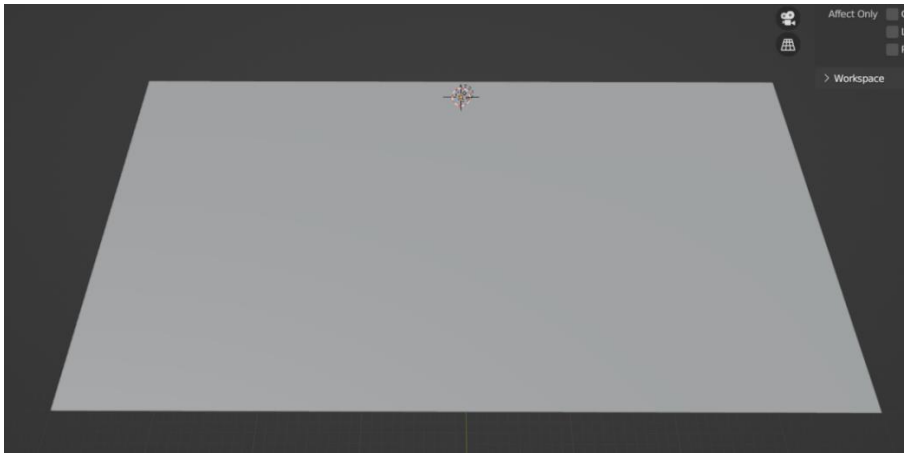
Το αντικείμενο floor2.obj το δημιουργήσαμε μέσω blender:

Προσθέσαμε ένα πλέγμα με κέντρο το (0,0,0) και υπολογίσαμε τις διαστάσεις έτσι ώστε να αποτελείται από 80x80 τετράγωνα (κάθε τετράγωνο ήταν 5m, άρα οι διαστάσεις που προκύπτουν είναι 400x400m).





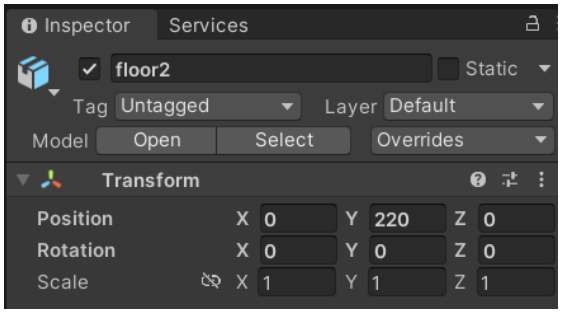
Έτσι, δημιουργείται το εξής έδαφος:



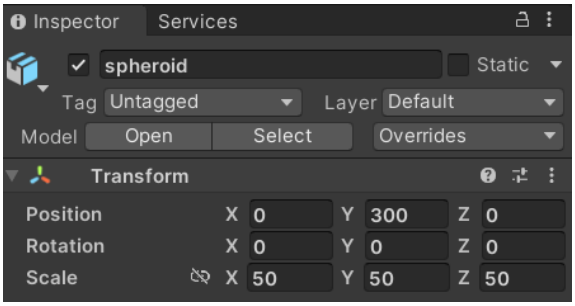
Κάνοντας Export → Wavefront(.obj), μπορέσαμε και το χρησιμοποιήσαμε για την δημιουργία του παιχνιδιού μας.

Στο Unity, αλλάξαμε τη τοποθεσία των αντικειμένων (και το scale του σκάφους):

Έδαφος

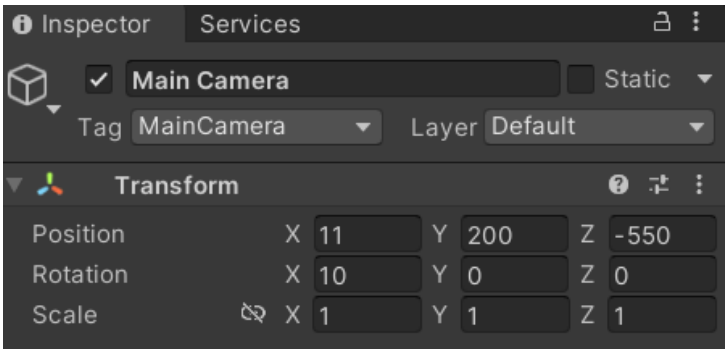


Σκάφος

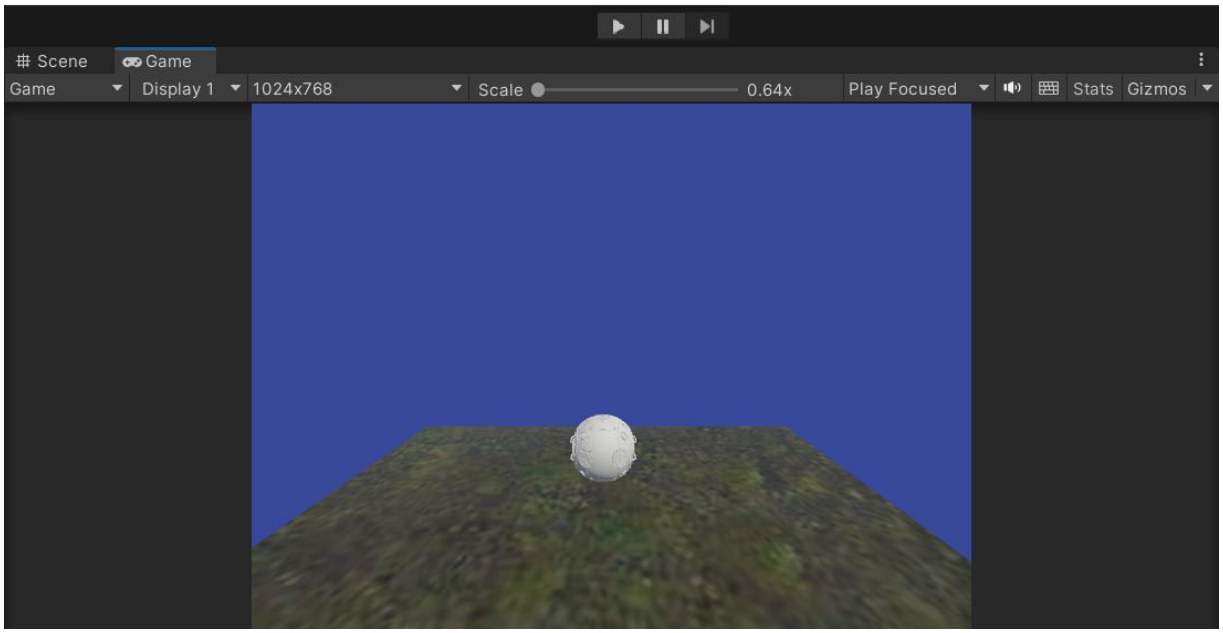


Το κέντρο του εδάφους είναι το (0,0) οπότε x=z=0.

Τοποθετούμε την κάμερα ενδεικτικά σε μία τοποθεσία για να μπορέσουμε να δούμε τις επόμενες αλλαγές:



Επομένως, έχουμε αυτό το αποτέλεσμα:



Ερώτημα ii)

Για την κίνηση του σκάφους, δημιουργήσαμε ένα καινούργιο script "PlayerMovement" στο spheroid.obj με το *Add Component*. Γράψαμε τον παρακάτω κώδικα:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class playerMovement : MonoBehaviour
{
    public float speed = 10;

    void Update()
    {
        //x axis
        if (Input.GetKey(KeyCode.A) && transform.position.x > -320)
            transform.Translate(Vector3.left*speed*Time.deltaTime);
        if (Input.GetKey(KeyCode.D) && transform.position.x < 320)
            transform.Translate(Vector3.left*(-speed)*Time.deltaTime);
        //z axis
        if (Input.GetKey(KeyCode.S) && transform.position.z < 320)
            transform.Translate(Vector3.forward * speed * Time.deltaTime);
        if (Input.GetKey(KeyCode.X) && transform.position.z > -320)
            transform.Translate(Vector3.forward * (-speed) * Time.deltaTime);
        //y axis
        if (Input.GetKey(KeyCode.W))
            transform.Translate(Vector3.up*speed*Time.deltaTime);
        if (Input.GetKey(KeyCode.E) && transform.position.y > 220)
            transform.Translate(Vector3.up*(-speed)*Time.deltaTime);
    }
}
```

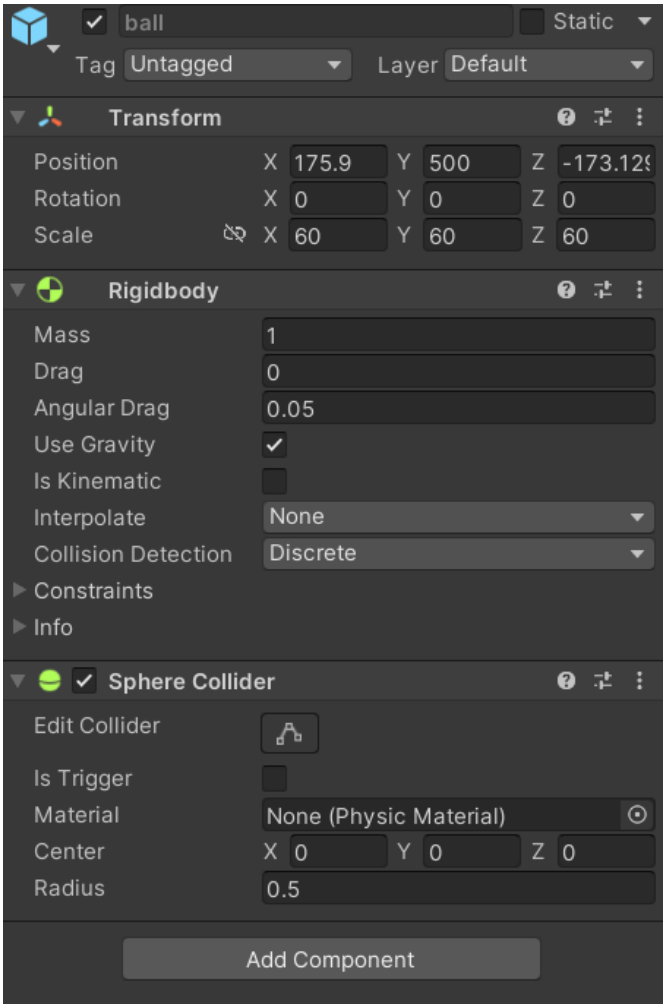
Όπως αναφέρεται και στην εκφώνηση η κίνηση σε σχέση με το πλήκτρα είναι η εξής:

- A : αριστερά
- D : δεξιά
- S : πίσω
- X : μπροστά
- W : πάνω
- E : κάτω

Και για να είναι στα όρια του εδάφους, στην συνθήκη if προσθέσαμε μια συνθήκη σχετική με την τοποθεσία του σκάφους. Εφόσον το έδαφος έχει διαστάσεις 400 x 400 (όπως αναφέραμε και στο ερώτημα ι), βάλαμε να μην ξεπερνά το -320 ή το +320, αναλόγως την κατεύθυνση.

Ερώτημα iii)

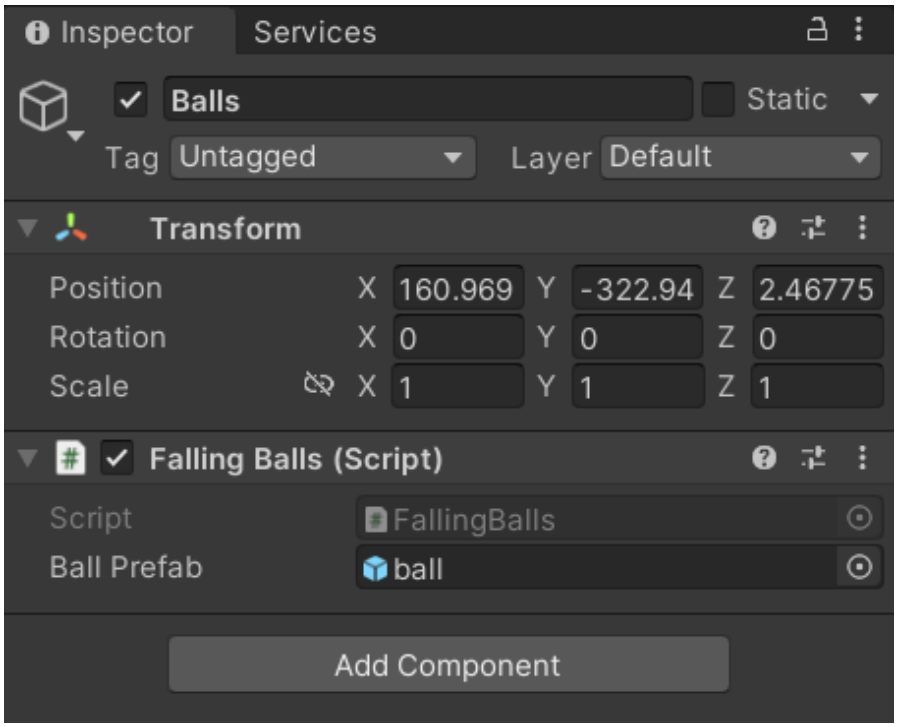
Αρχικά, εισάγουμε το ball.obj (πήραμε την μπάλα της προηγούμενης εργασίας), και το τοποθετούμε στη σκηνή μας. Με drag – drop εφαρμόζουμε την υφή “fire” στην μπάλα. Για να πέφτει προς το έδαφος, πρέπει να εφαρμοστεί βαρύτητα, επομένως προσθέτουμε τα εξής components (Rigidbody, Sphere Collider):



Στη συνέχεια, για να δημιουργούνται πολλές μπάλες σε τυχαία σημεία, δημιουργούμε ένα empty object με το όνομα Balls, στο οποίο δημιουργούμε ένα νέο script “FallingBalls”. Γράφουμε τον εξής κώδικα:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FallingBalls : MonoBehaviour
{
    public GameObject ballPrefab;

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            Vector3 randomPos = new Vector3(Random.Range(-400, 400), Random.Range(600, 1001), Random.Range(-400, 400));
            Instantiate(ballPrefab, randomPos, Quaternion.identity);
        }
    }
}
```

Και τέλος, ορίζουμε τη μπάλα ως το αντικείμενο που θα χρησιμοποιείται, δηλαδή το ballPrefab που έχουμε στον κώδικα:



Ερώτημα iv)

Για την κίνηση της κάμερας, δημιουργήσαμε ένα empty object “CameraObject” στο οποίο θα “κοιτάει” η κάμερά μας. Σε αυτό το object τοποθετήσαμε την main camera και δημιουργήσαμε ένα νέο script με το όνομα “ControlCamera”. Γράψαμε τον εξής κώδικα:

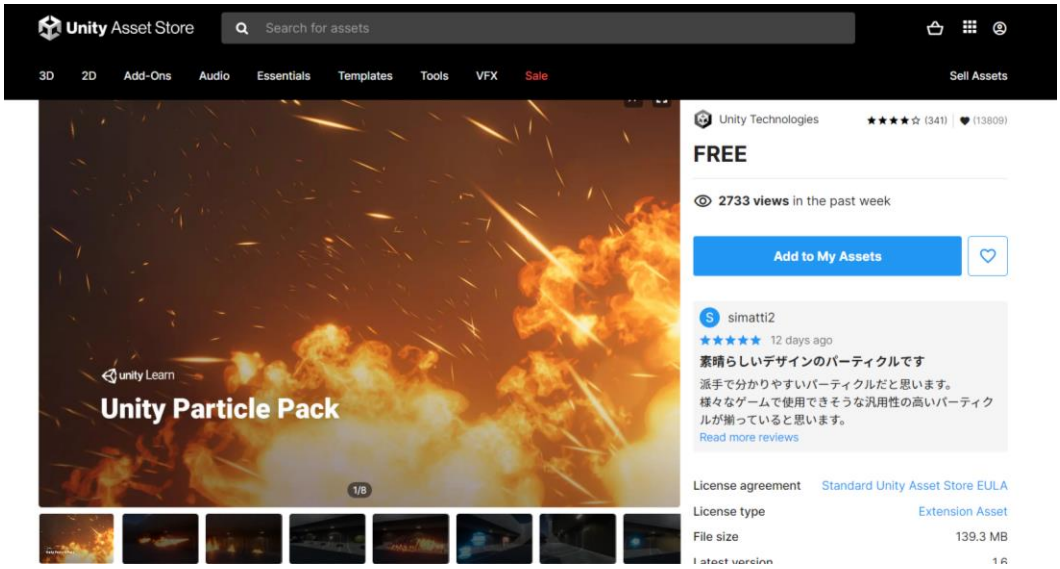
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ControlCamera : MonoBehaviour
{
    public float cameraSpeed = 50;
    public float cameraTime = 100;
    public Vector3 newPos;

    public float rot = 10;
    public Quaternion newRot;

    // Start is called before the first frame update
    void Start()
    {
        newPos = transform.position;
        newRot = transform.rotation;
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.RightArrow))
            newPos += (transform.right * cameraSpeed);
        if (Input.GetKeyDown(KeyCode.LeftArrow))
            newPos += (transform.right * -cameraSpeed);
        if (Input.GetKeyDown(KeyCode.UpArrow))
            newPos += (transform.forward * cameraSpeed);
        if (Input.GetKeyDown(KeyCode.DownArrow))
            newPos += (transform.forward * -cameraSpeed);
        if (Input.GetKeyDown(KeyCode.KeypadPlus))
            newPos += (transform.up * cameraSpeed);
        if (Input.GetKeyDown(KeyCode.KeypadMinus))
            newPos += (transform.up * -cameraSpeed);
        if (Input.GetKeyDown(KeyCode.R))
            newRot *= Quaternion.Euler(Vector3.up * rot);
        if (Input.GetKeyDown(KeyCode.P))
            newRot *= Quaternion.Euler(Vector3.up * -rot);
        if (Input.GetKeyDown(KeyCode.F))
            newRot *= Quaternion.Euler(Vector3.right * rot);
        if (Input.GetKeyDown(KeyCode.L))
            newRot *= Quaternion.Euler(Vector3.right * -rot);

        transform.position = Vector3.Lerp(transform.position, newPos, Time.deltaTime * cameraTime);
        transform.rotation = Quaternion.Lerp(transform.rotation, newRot, Time.deltaTime * cameraTime);

    }
```

Η κάμερα λειτουργεί ως εξής:

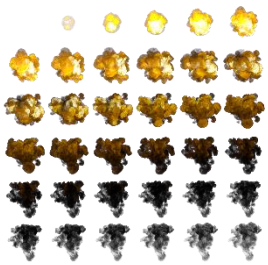
- Βελάκι ← : προς τα αριστερά (άξονας x)
- Βελάκι → : προς τα δεξιά (άξονας x)
- Βελάκι ↑ : προς τα πίσω (άξονας z)
- Βελάκι ↓ : μπροστά (άξονας z)
- + από το keypad: αύξηση ύψους (y)
- - από το keypad: μείωση ύψους (y)
- R : περιστροφή αριστερά στον άξονα x
- P: περιστροφή δεξιά στον άξονα x
- F: περιστροφή πίσω στον άξονα z
- L: περιστροφή μπροστά στον άξονα z

Ερώτημα ν)

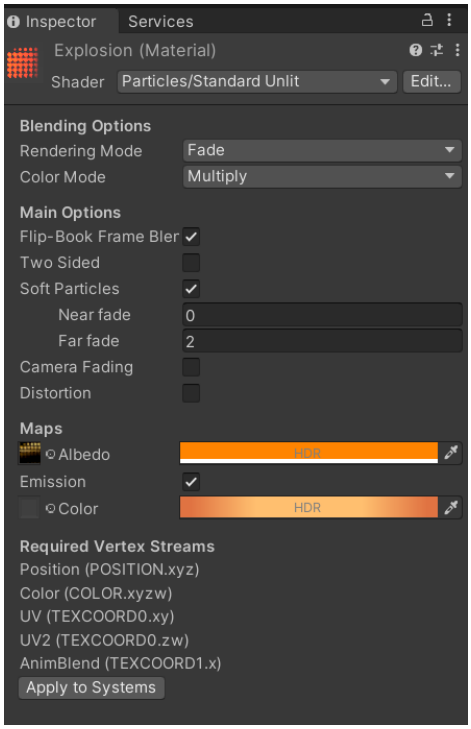
Για να έχουμε εφέ έκρηξης, κατεβάσαμε το παρακάτω πακέτο



Και πιο συγκεκριμένα, χρησιμοποιήσαμε:

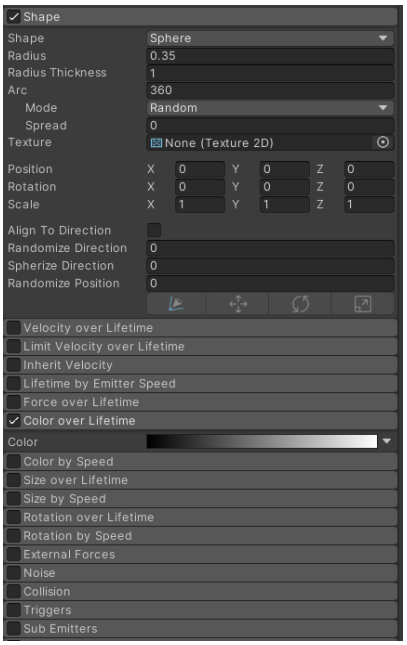


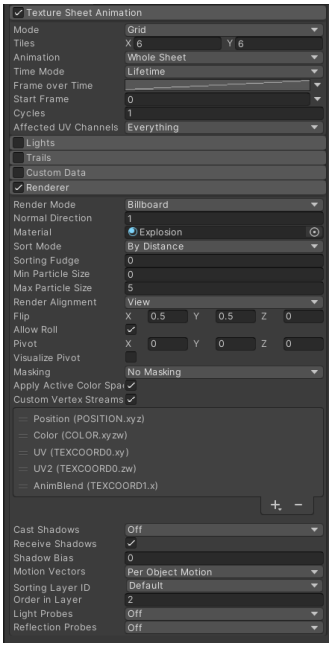
Δημιουργήσαμε ένα Material, στο Shader επιλέξαμε Material → Particles → Standard Unit και κάναμε τις εξής ρυθμίσεις:



(Στο πλαίσιο Albedo έχουμε κάνει drag and drop την έτοιμη εικόνα που δείξαμε παραπάνω)

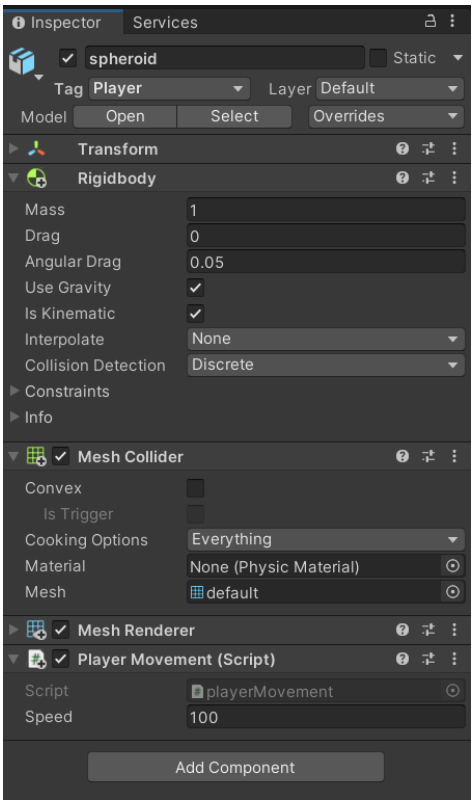
Στη συνέχεια, δημιουργήσαμε ένα Particle System στο οποίο κάναμε τις παρακάτω αλλαγές:



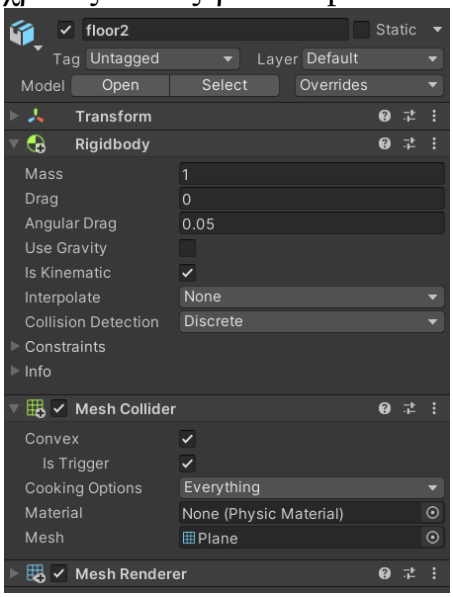


Στην τελευταία εικόνα, στο Material, επιλέξαμε το material “Explosion” που δημιουργήσαμε προηγουμένως.

Για να καταφέρουν να αλληλεπιδράσουν τα δύο σώματα, πρέπει να προσθέσουμε και στο σκάφος μας κάποια components.



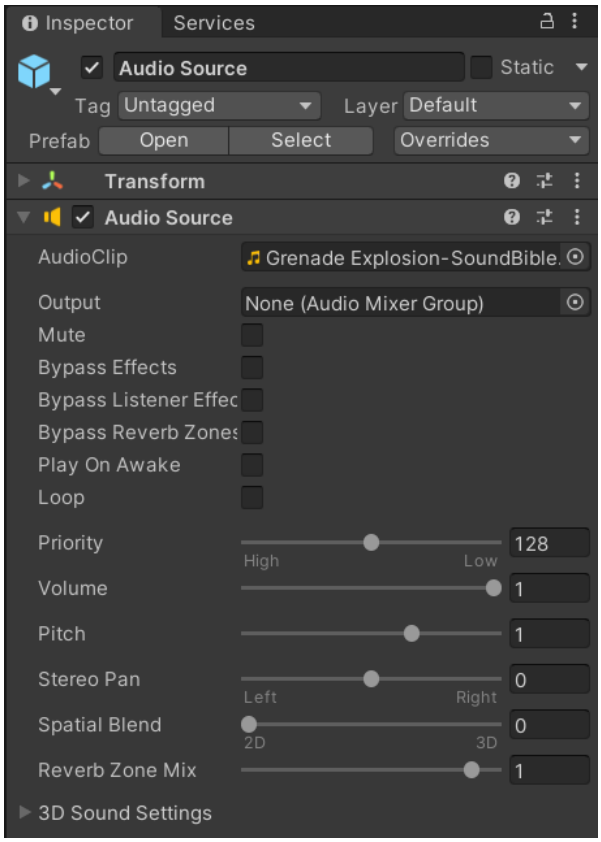
Και αλλάζουμε το Tag σε “Player”.
Ομοίως, και το έδαφος χρειάζεται έξτρα components.



Πέρα από την έκρηξη, θέλουμε να έχει και ήχο όταν συγκρούονται το σκάφος με τη μπάλα.

Αρχικά, κατεβάσαμε έναν έτοιμο ήχο έκρηξης από το “soundbible.com” και τον συμπεριλάβαμε στο project.

Δημιουργήσαμε ένα Audio Source με τις εξής επιλογές:

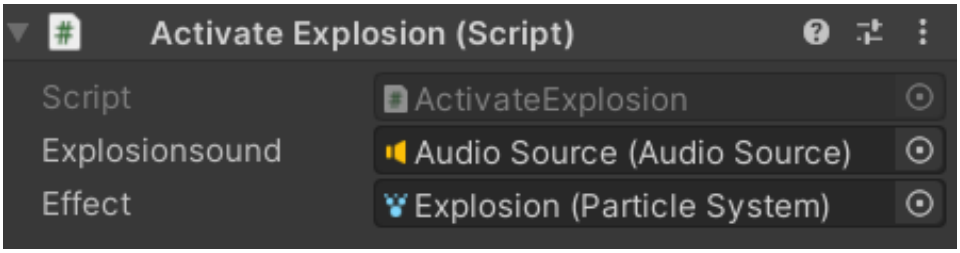


Αφού δημιουργήσαμε όλα τα απαραίτητα, στο Prefab “ball”, προσθέτουμε άλλο ένα script, με το όνομα “ActivateExplosion” το οποίο θα διαχειρίζεται την έκρηξη.

Ο κώδικας:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ActivateExplosion : MonoBehaviour
{
    public AudioSource explosionsound;
    public ParticleSystem effect;

    private void OnTriggerEnter(Collider collision)
    {
        if (collision.tag == "Player")
        {
            explosionsound.Play();
            effect.Play();
            Destroy(collision.gameObject);
            Destroy(gameObject);

        }
        else
        {
            explosionsound.Play();
            effect.Play();
            Destroy(gameObject);
        }
    }
}
```

Και επειδή το εφέ και ο ήχος μας είναι public, κάνουμε drag and drop αυτά που δημιουργήσαμε παραπάνω:



Περιγραφή δυσκολιών υλοποίησης -προβλήματα που συναντήθηκαν – ερωτήματα που δεν υλοποιήθηκαν:

Στο ερώτημα iii, η μπάλα που δημιουργείται δεν είναι τυχαίας ακτίνας. Έχει σταθερή ακτίνα.

Στο ερώτημα v, ενώ έχουν δημιουργηθεί όλα, και τα αντικείμενα εξαφανίζονται όταν συγκρουστούν, η έκρηξη δεν εμφανίζεται.

Στο α μπόνους ερώτημα δημιουργήσαμε τον ήχο αλλά δεν ενεργοποιείται σωστά.

Δεν υλοποιήσαμε άλλα μπόνους ερωτήματα.

Πληροφορίες σχετικά με την υλοποίηση:

Η εργασία υλοποιήθηκε σε λειτουργικά συστήματα Windows και χρησιμοποιήθηκαν οι πιο πρόσφατες εκδόσεις της Unity, καθώς και του Visual Studio.

Λειτουργία ομάδας και συνεργασία:

Η ομάδα λειτούργησε συνεργατικά. Αρχικά είδαμε μαζί την εκφώνηση της άσκησης, και συζητήσαμε ιδέες για τη λύση της. Έπειτα από κάποιες συναντήσεις που έγιναν μέσα στην εβδομάδα, βγήκε αυτό το αποτέλεσμα.

Πηγές

- https://docs.unity3d.com/ScriptReference/KeyCode.html
- https://docs.unity3d.com/ScriptReference/Object.Instantiate.html
- https://docs.unity3d.com/ScriptReference/Transform.html
- https://docs.unity3d.com/ScriptReference/Transform-position.html
- https://assetstore.unity.com/packages/essentials/tutorial-projects/unity-particle-pack-127325
- https://www.youtube.com/watch?v=Umcg1cBHcxM&list=WL&index=47
- https://www.youtube.com/watch?v=m8evpNWSzIA&list=WL&index=31
- https://www.youtube.com/watch?v=MEqgo5sP4qY&list=WL&index=32
- https://www.youtube.com/watch?v=0NCTAuP3BgU
- https://soundbible.com/1467-Grenade-Explosion.html
- https://www.youtube.com/watch?v=cvQiQglPI18&list=WL&index=44&t=6s