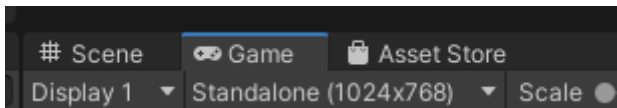


# 1. Περιγραφή της εργασίας

Θα κατασκευάσουμε μια εφαρμογή-παιχνίδι, στην οποία θα σχηματίζεται ένας χαρακτήρα, Ο Bob, από στερεά σχήματα, που θα ελέγχει ο παίκτης-χρήστης με το πληκτρολόγιο. Ο χαρακτήρας κινείται πάνω σε ένα επίπεδο πάνω στο οποίο είναι τοποθετημένα αντικείμενα (κύβοι) με διαφορετικές υφές. Όταν ο χαρακτήρας ακουμπάει ένα από αυτά τα αντικείμενα, «υιοθετεί» την υφή του. Επίσης κάποια από τα αντικείμενα τον «μεγαλώνουν» ενώ άλλα τον «μικραίνουν».

(i)

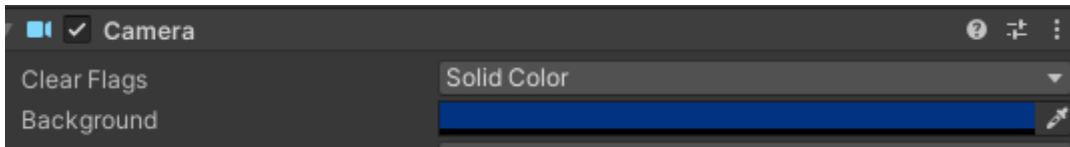
Φτιάξτε μια εφαρμογή Unity 3D που θα τρέχει με ανάλυση 1024x768.



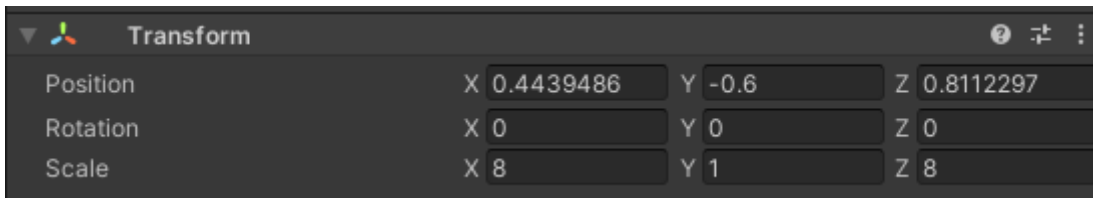
Η εφαρμογή θα έχει τίτλο «Bob in Wonderland».



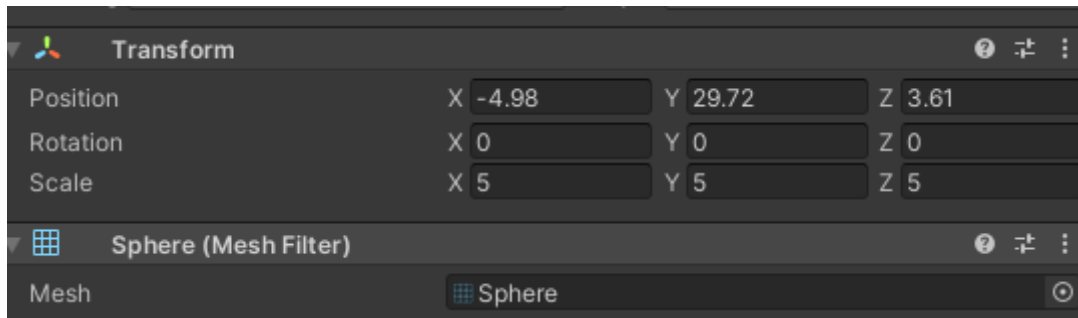
Στο υπόβαθρο (background) θα βάλετε μπλε σκούρο χρώμα.



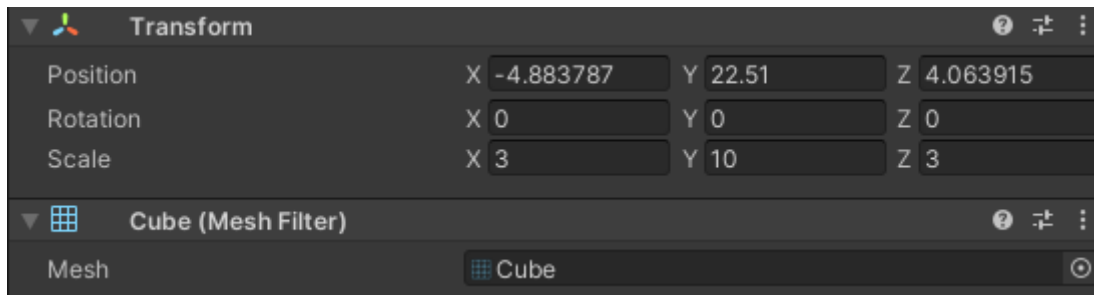
Όταν θα ξεκινάει η εφαρμογή θα φορτώνονται ο Bob (στο κέντρο του εδάφους) και το «έδαφος» (επίπεδο) στο οποίο θα κινείται ο Bob. Το έδαφος είναι ένα επίπεδο από 80 x 80 τετράγωνα, σχεδιασμένο πάνω στο xz επίπεδο, στο οποίο θα εφαρμόσετε μια υφή εδάφους.



Ο Bob θα σχηματιστεί από στερεά αντικείμενα: μία σφαίρα για κεφάλι με κλιμάκωση 5,



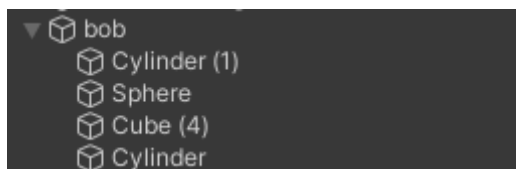
ένα παραλληλεπίπεδο για σώμα με διαστάσεις 3\*10\*3 (x,y,z)



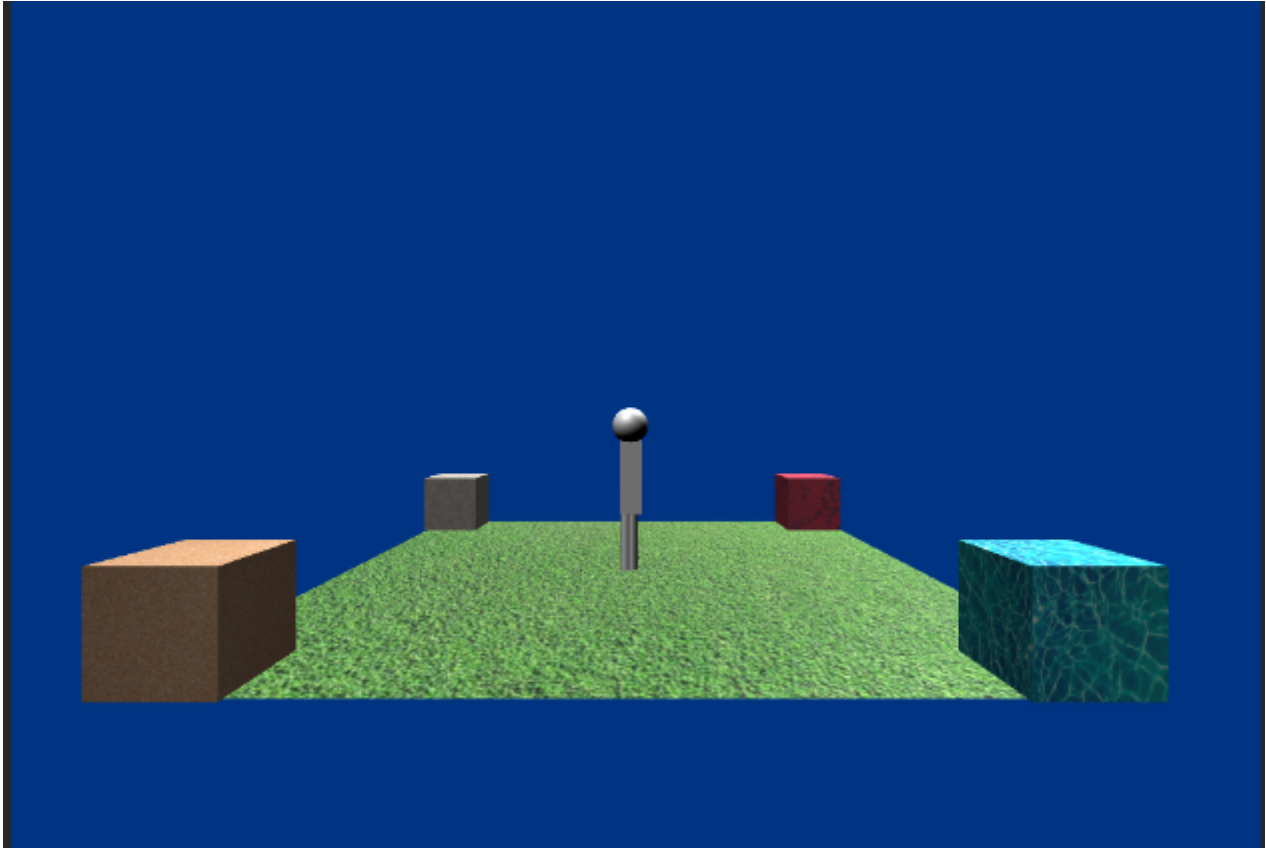
και δύο κυλίνδρους για πόδια με διαστάσεις 1.5\*4\*1.5 (x,y,z).



Τα στερεά αυτά να ομαδοποιηθούν ώστε να κινούνται και να μετασχηματίζονται στη συνέχεια ως ένα αντικείμενο



## Το αποτέλεσμα:



(ii)

Ο παίκτης μπορεί να πλοηγήσει τον Bob πάνω στο έδαφος. Δεν μπορεί να βγει έξω από τα όρια του εδάφους. Ο Bob κινείται από τον χρήστη, με σταθερό βήμα, με τα εξής πλήκτρα (όταν τα πλήκτρα δεν πατιούνται ο Bob θα παραμένει ακίνητος).

Για να γίνει αυτό θα δημιουργήσουμε ένα script στην ομαδοποιημένα αντικείμενα που παρουσιάζουν τον Bob(Move.cs):

Αρχικοποιούμε τα όρια :

```
public class Move : MonoBehaviour
{
    public float xRange = 45.0f;
    public float MxRange = -35.0f;
    public float zRange = 37.0f;
    public float MzRange = -43.0f;
    public float yRange = 10.0f;
    public float MyRange = -10.0f;
```

## Μεσα στην συναρτηση Update()

- <a>, <d> κίνηση παράλληλα στον άξονα x του συστήματος παγκόσμιων συντεταγμένων.

```
//x a3onas
if (Input.GetKey(KeyCode.D)) {
    if (transform.position.x > xRange)
    {
        transform.position = new Vector3(xRange, transform.position.y, transform.position.z);
    }
    else
    {
        transform.Translate((float)0.1, (float)0.0, (float)0.0);
    }
}
else if (Input.GetKey(KeyCode.A))
{
    if (transform.position.x < MxRange)
    {
        transform.position = new Vector3(MxRange, transform.position.y, transform.position.z);
    }
    else
    {
        transform.Translate((float)-0.1, (float)0.0, (float)0.0);
    }
}
```

- <s>, <x> κίνηση παράλληλα στον άξονα z του συστήματος παγκόσμιων συντεταγμένων

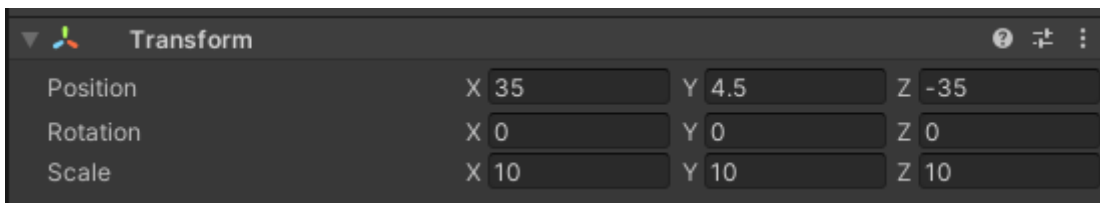
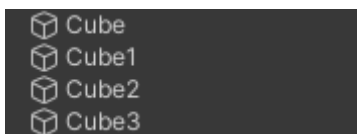
```
//z a3onas
if (Input.GetKey(KeyCode.S))
{
    if (transform.position.z > zRange)
    {
        transform.position = new Vector3(transform.position.x, transform.position.y, zRange);
    }
    else {
        transform.Translate((float)0.0, (float)0.0, (float)0.1);
    }
}
else if (Input.GetKey(KeyCode.X))
{
    if (transform.position.z < MzRange)
    {
        transform.position = new Vector3(transform.position.x, transform.position.y, MzRange);
    }
    else
    {
        transform.Translate((float)0.0, (float)0.0, (float)-0.1);
    }
}
```

- <w>, <e> κίνηση παράλληλα στον άξονα y του συστήματος παγκόσμιων συντεταγμένων(αλλαγή ύψους).

```
//y a3onas
if (Input.GetKey(KeyCode.E))
{
    if (transform.position.y > yRange)
    {
        transform.position = new Vector3(transform.position.x, yRange, transform.position.z);
    }
    else {
        transform.Translate((float)0.0, (float)0.1, (float)0.0);
    }
}
else if (Input.GetKey(KeyCode.W))
{
    if (transform.position.y < MyRange)
    {
        transform.position = new Vector3(transform.position.x, MyRange, transform.position.z);
    }
    else
    {
        transform.Translate((float)0.0, (float)-0.1, (float)0.0);
    }
}
```

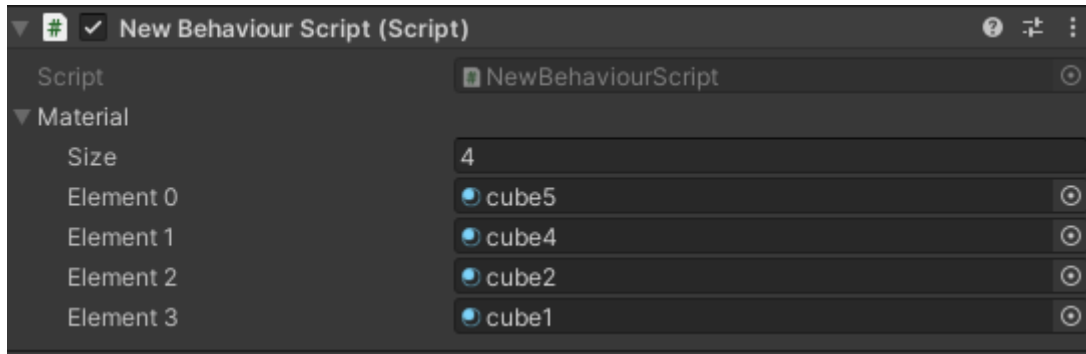
Η λογική είναι να πατησει κάποιο από τα παραπάνω πλήκτρα με την εντολή `transform.Translate` να μετακινηθεί κατά 0.1 ή -0.1 ο καταλληλός άξονας σε περίπτωση που ο χαρακτήρας βγει έξω από τα όρια με την εντολή `transform.position` να τον επαναφέρει πίσω με αποτέλεσμα να μην μπορεί να βγει έξω από τα όρια

- (iii) Στις 4 γωνίες του εδάφους υπάρχουν 4 κύβοι (με διάσταση 10) και διαφορετικές υφές.



**Οι κύβοι θα είναι ακίνητοι και όταν ο Bob ακουμπάει έναν κύβο, θα εφαρμόζεται σε αυτόν η υφή του κύβου που άγγιξε.**

Θα δημιουργήσουμε ένα καινούριο script και το προσθετούμε σε κάθε κομμάτι του μπομπ ξεχωριστά



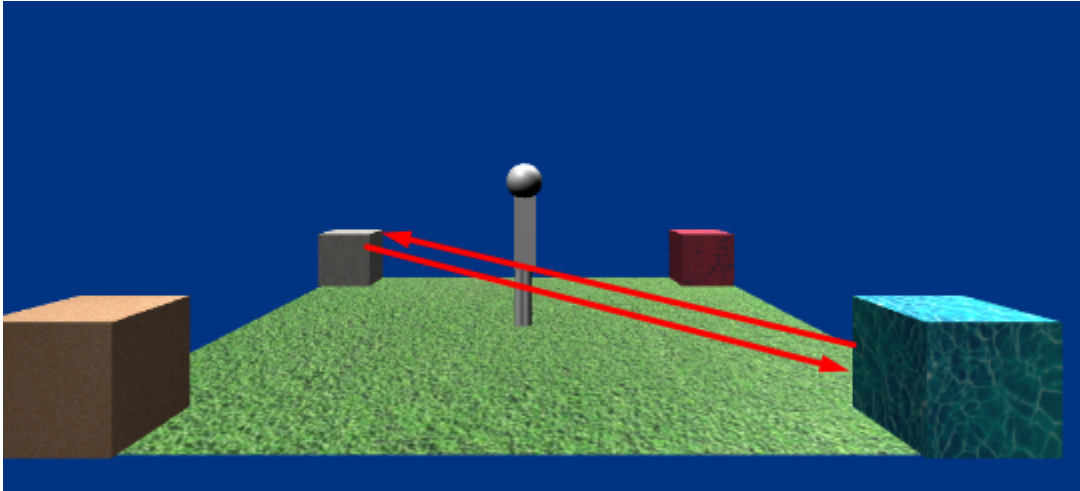
Αυτό το ερώτημα θα μπορούμε να το κάνουμε και με το collision όταν ακουμπήσει τον κύβο να αλλάζει αλλά επιλέξαμε να το κάνουμε με συντεταγμένες με την ίδια λογική που χρησιμοποιούμε στα όρια με την εντολή.

```
if (transform.position.x > 27f && transform.position.z < -28f)
{
    rend.sharedMaterial = material[0];
}
else if (transform.position.x < -28f && transform.position.z < -28f)
{
    rend.sharedMaterial = material[1];
}
else if (transform.position.x > 28f && transform.position.z > 28f)
{
    rend.sharedMaterial = material[3];
}
else if (transform.position.x < -28f && transform.position.z > 28f)
{
    rend.sharedMaterial = material[2];
}
```

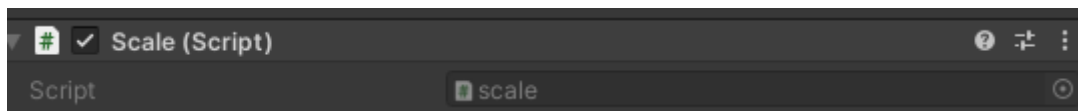
Με την εντολή sharedMaterial επιλέγουμε το texture που θέλουμε.

**Δύο από τους 4 κύβους (σε μία διαγώνια του εδάφους) θα αλλάζουν και τον Bob: ο ένας κύβος θα τον «μεγαλώνει» (κλιμάκωση), ενώ ο άλλος θα τον «μικραίνει» (σμίκρυνση)**

Επιλέξαμε αυτούς τους κυβους



Ο μπροστα κυβος θα τον μεγαλωνει κλιμακα ενω ο πισω θα τον μικραινει οποτε θα δημιουργησουμε script και θα το προσθεσουμε στο ομαδοποιημενο object που αναπαραστα τον χαρακτηρα.



Στην αρχη οριζουμε ποσο να μεγαλωνει και να μικραινει το scale και το προσθετουμε σε μια μεταβλητη γιατι αμα το βάλουμε μέσα στην update() θα μεγαλωνει η μικραινει αντιστοιχα συνεχεια με αποτελεσμα να κρασάρει το προγραμμα οποτε επιλεξαμε να εχει προκαθορισμενο μεγαθος.

```
void Start()
{
    scaleUp = new Vector3(0.5f, 0.5f, 0.5f);
    scaleDown = new Vector3(-0.5f, -0.5f, -0.5f);
    x = transform.localScale;
    up = x + scaleUp;
    down = x + scaleDown;
    check = new Vector3(0f, 0f, 0f);
}
```

Ξανα με την λογικη με τις συντεταγμενες και χρησιμοποιωντας την εντολη `transform.localScale` να οριζει το καινουργιο scale ενω στους αλλους κυβους επιστρεφει στο κανονικο του μεγεθος.

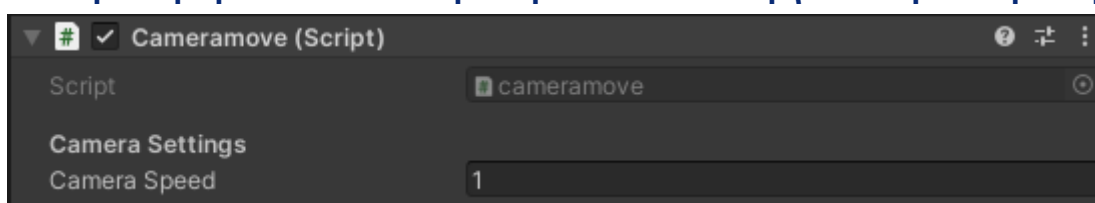
```
void Update()
{
    if (transform.position.x > 34f && transform.position.z < -32f)
    {
        transform.Translate((float)0.0, (float)-0.01, (float)0.0);
        transform.localScale = up;
    }
    else if (transform.position.x < -21f && transform.position.z < -32f)
    {
        transform.Translate((float)0.0, (float)0.01, (float)0.0);
        transform.localScale = x;
    }
    else if (transform.position.x > 33f && transform.position.z > 24f)
    {
        transform.Translate((float)0.0, (float)0.01, (float)0.0);
        transform.localScale = x;
    }
    else if (transform.position.x < -24f && transform.position.z > 24f)
    {
        transform.Translate((float)0.0, (float)0.01, (float)0.0);
        transform.localScale = down;
    }
}
```

Παρατηρηθηκε οτι σε καποια σημεια ο χαρακτηρας μετα το scale επεφτε κατω απο το επιπεδο για τον λογο αυτο προστεθηκε η εντολη που χρησιμοποιησαμε ωστε ο χαρακτηρας να ανεβαίνει ψηλά στον άξονα y (`transform.Translate`) με αποτελεσμα να διορθωθεί.

(iv)

Υλοποιήστε μια απλή κάμερα ώστε ο χρήστης να βλέπει τη σκηνή από οποιαδήποτε γωνία και θέση και από οποιοδήποτε ύψος. Η κάμερα ελέγχεται από το χρήστη με τα βελάκια του πληκτρολογίου για κίνηση στους άξονες x και z του συστήματος παγκόσμιων συντεταγμένων και τα πλήκτρα `<+>/<->` για κίνηση κατά μήκος του άξονα y του συστήματος παγκόσμιων συντεταγμένων (αλλαγή ύψους)

Για την καμερα θα υλοποιήσουμε ενα καινουργιο script στην καμερα





Με την ίδια λογική αμα πατηθεί κάποια απο τα πληκτρα να κανει το αναλογο αλλαζοντας το position της καμερας με την εντολη Camera.Position στην εκφωνησει λειει να η κίνηση πρεπει να γινει στους άξονες x,y και z του συστήματος παγκόσμιων συντεταγμένων αλλα ετσι δεν μπορούμε να δουμε απο καθε γωνια για τον λογο αυτο γιατι δεν κατανοηθηκε απο την εκφωνησει προσθεσαμε να μπορεί να κανει και rotate με το πλήκτρο R

```
void Update()
{
    if (Input.GetKey(KeyCode.RightArrow))
    {
        CameraPosition.x += CameraSpeed/10;
    }
    if (Input.GetKey(KeyCode.LeftArrow))
    {
        CameraPosition.x -= CameraSpeed / 10;
    }
    if (Input.GetKey(KeyCode.UpArrow))
    {
        CameraPosition.z += CameraSpeed / 10;
    }
    if (Input.GetKey(KeyCode.DownArrow))
    {
        CameraPosition.z -= CameraSpeed / 10;
    }
    if (Input.GetKey(KeyCode.KeypadMinus))
    {
        CameraPosition.y -= CameraSpeed / 10;
    }
    if (Input.GetKey(KeyCode.KeypadPlus))
    {
        CameraPosition.y += CameraSpeed / 10;
    }
    if (Input.GetKey(KeyCode.R))
    {
        transform.RotateAround(transform.position, Vector3.up, 360 * Time.deltaTime);
    }
    this.transform.position = CameraPosition;
}
```