



**Πανεπιστήμιο Πειραιώς**

**ΗΜΕΡΟΜΗΝΙΑ 16/10/2022**

**Τμήμα Πληροφορικής  
ΟΝΟΜΑΤΕΠΩΝΥΜΟ :  
ΑΛΕΞΑΝΔΡΟΣ ΓΚΙΝΕΤΣΙ Π18028**

## Εισαγωγή

Στα πλαίσια της εργασίας, δημιουργήθηκε ένας τρισδιάστατος (3D) χώρος που αναπαριστά ένα μεσαιωνικό χωριό όπου περιλαμβάνει :Κάστρο,γεφυρά, ξενοδοχείο,εργαστήρι ξυλοκοπής,δημαρχείο,εκκλησία,αγαλμα,εργαστήριο αλχημίστη,παγκός μανάβη,πωλητής μικρών ειδών,μαυσιλείο,ανεμόμυλοι,έναν μολος βαρκών και μια φάρμα.Στο χωριό υπάρχουν επίσης πολεμιστές (NPCs).Το κάστρο αλλά και η γεφυρά είναι «οχυρωμένα» και απαγορεύεται η είσοδος.

## Σχεδίαση

Αρχικά σχεδιάστηκε το τερραίν . Τοποθετήθηκαν βουνά γύρω γύρω ως “φυσικό” εμπόδιο. Στη συνέχεια τοποθετήθηκε ένας ποταμός να διαπερνά το τερραίν και τοποθετήθηκε ένας καταρράκτης στην αρχή αυτού. Έπειτα δημιουργήθηκε μια γέφυρα όπου αρχικό σκοπό είχε να ενώσει 2 διαφορετικά χωριά, από ίσως διαφορετικούς “κόσμους”, κάτι το οποίο εγκαταλείφθηκε λόγω των λιγοστών ( όπως προέκυψαν ) δωρεάν assets που διαθέτει το asset store.

Κατόπιν τοποθετήθηκαν ανεμόμυλοι πάνω στο βουνό και στο κάστρο στην άλλη άκρη του τερραίν με την γέφυρα ως συνδετικό κρίκο. Οπότε ως τελευταίο έμεινε η δημιουργία δρόμου που συνδέει όλα τα παραπάνω και η σχεδίαση του χωριού.

Για την δημιουργία των καταστημάτων και του δημαρχείου έγινε χρήση πολλών μικρών prefabs, καθώς τα μοναδικά που βρέθηκαν έτοιμα είναι το άγαλμα στη μέση της πλατείας, την εκκλησία και το μαυσιλείο.Τέλος για να γεμίσει το τερραίν , τοποθετήθηκε παντού δάσικα αντικείμενα και τοποθετήθηκαν και 6 wind-zones για να κουνιούνται τα κλαδιά των δέντρων .

## Πηγαίος Κώδικας

Τα scripts που δημιουργήθηκαν και χρησιμοποιήθηκαν είναι τέσσερα. Οποιοδήποτε άλλο μπορεί να περιέχεται σε κάποιο αντικείμενο ( βλέπε χαρακτήρες χωριού), ήταν ήδη ενσωματωμένο με τα assets. Συνεπώς τα τέσσερα script είναι τα εξής:

1. AiMovement
2. rotateonly
3. PlayerLook
4. PlayerMove
5. MillsRotating
6. thoughts

AiMovement: Σε αυτό περιέχεται ο κώδικας που έχει χρησιμοποιηθεί για να μετακινούνται τα npc.

```
PlayerLook.cs | AIMovement.cs | rotateonly.cs | PlayerMove.cs
Assembly-CSharp | AIMovement

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class AIMovement : MonoBehaviour
6  {
7      public float movespeed = 3f;
8      public float rotspeed = 100f;
9
10     private bool iswondering = false;
11     private bool isrotatingleft = false;
12     private bool isrotatingright = false;
13     private bool iswalking = false;
14     void Start()
15     {
16     }
17
18
19
20     void Update()
21     {
22         if (iswondering == false)
23         {
24             StartCoroutine(Wander());
25         }
26         if (isrotatingleft == true)
27         {
28             transform.Rotate(transform.up * Time.deltaTime * -rotspeed);
29         }
30         if (isrotatingright == true)
31         {
32             transform.Rotate(transform.up * Time.deltaTime * rotspeed);
33         }
34         if (iswalking == true)
35         {
36             transform.position += transform.forward * movespeed * Time.deltaTime;
37         }
38     }
39 }
```

Αρχικά, η ταχύτητα ορίζεται με το movespeed και η ταχύτητα περιστροφής με το rotspeed. Έπειτα ορίστηκαν 4 booleans, για το αν κάνει κάποια ενέργεια, αν κινείται μπροστά, αριστερά ή και δεξιά. Στη συνάρτηση update παρατηρούνται οι ενέργειες που γίνονται αν κάποιο από τα booleans είναι αληθινό.

Τέλος, στη συνάρτηση wander ορίζεται για κάθε ενέργεια ένας τυχαίος αριθμός σχετικά με :

1. τον πόσο χρόνο κάνει περιστροφή,
2. τον πόσο χρόνο περιμένει για να κάνει περιστροφή,
3. το αν περιστραφεί δεξιά ( αν τύχει 1), αριστερά ( αν τύχει 2) ή καθόλου (αν τύχει 0),
4. το πόσο χρόνο αναμένει για να ξεκινήσει να περπατάει,
5. το πόσο χρόνο περπατάει εμπρός

```

IEnumerator Wander()
{
    int rotime = Random.Range(1,2);
    int rotwait = Random.Range(1, 2);
    int rotlorr = Random.Range(0,2);
    int walkwait = Random.Range(1, 2);
    int walktime = Random.Range(1,2);

    iswondering = true;
    yield return new WaitForSeconds(walkwait);
    iswalking = true;
    yield return new WaitForSeconds(walktime);
    iswalking = false;
    yield return new WaitForSeconds(rotwait);
    if (rotlorr == 1)
    {
        isrotatingright = true;
        yield return new WaitForSeconds(rotime);
        isrotatingright = false;
    }
    if (rotlorr == 2)
    {
        isrotatingleft = true;
        yield return new WaitForSeconds(rotime);
        isrotatingleft = false;
    }
    iswondering = false;
}

```

rotateonly : Εδώ έχει γίνει αντιγραφή τον κώδικα από το aimovement και έχει αφαιρεθεί η κίνηση προς τα μπροστά για να μπορεί το npc να περιστρέφεται μόνο.

```
7 public float movespeed = 3f;
8 public float rotspeed = 100f;
9
10 private bool iswondering = false;
11 private bool isrotatingleft = false;
12 private bool isrotatingright = false;
13 private bool iswalking = false;
14 void Start()
15 {
16 }
17
18
19
20 void Update()
21 {
22     if (iswondering == false)
23     {
24         StartCoroutine(Wander());
25     }
26     if (isrotatingleft == true)
27     {
28         transform.Rotate(transform.up * Time.deltaTime * -rotspeed);
29     }
30     if (isrotatingright == true)
31     {
32         transform.Rotate(transform.up * Time.deltaTime * rotspeed);
33     }
34 }
35
36 IEnumerator Wander()
37 {
38     int rotime = Random.Range(1, 2);
39     int rotwait = Random.Range(1, 2);
40     int rotlovr = Random.Range(0, 2);
41     int walkwait = Random.Range(1, 2);
42     int walktime = Random.Range(1, 2);
43
44     iswondering = true;
45     yield return new WaitForSeconds(walkwait);
46     iswalking = true;
47     yield return new WaitForSeconds(walktime);
48     iswalking = false;
49     yield return new WaitForSeconds(rotwait);
50     if (rotlovr == 1)
```

PlayerLook: Εδώ κινείται η κάμερα με το ποντίκι μας. Ορίζουμε το ποντίκι με τον άξονα x για οριζόντια και άξονα y για κάθετα πολλαπλασιασμένο με την επιτάχυνση ποντικιού και τον χρόνο που κάνουμε την ίδια κίνηση. Στη συνέχεια, ορίζονται όρια ώστε το ποντίκι να μη μπορεί να πάει παραπάνω από 90 μοίρες πάνω και 90 μοίρες κάτω.

```

public class PlayerLook : MonoBehaviour
{
    [SerializeField] private string mouseXInputName, mouseYInputName;
    [SerializeField] private float mouseSensitivity;

    [SerializeField] private Transform playerBody;

    private float xAxisClamp;

    private void Awake()
    {
        LockCursor();
        xAxisClamp = 0.0f;
    }

    private void LockCursor()
    {
        Cursor.lockState = CursorLockMode.Locked;
    }

    private void Update()
    {
        CameraRotation();
    }

    private void CameraRotation()
    {
        float mouseX = Input.GetAxis(mouseXInputName) * mouseSensitivity * Time.deltaTime;
        float mouseY = Input.GetAxis(mouseYInputName) * mouseSensitivity * Time.deltaTime;

        xAxisClamp += mouseY;

        if (xAxisClamp > 90.0f)
        {
            xAxisClamp = 90.0f;
            mouseY = 0.0f;
            ClampXAxisRotationToValue(270.0f);
        }
        else if (xAxisClamp < -90.0f)
        {

```

```

        }
        else if (xAxisClamp < -90.0f)
        {
            xAxisClamp = -90.0f;
            mouseY = 0.0f;
            ClampXAxisRotationToValue(90.0f);
        }

        transform.Rotate(Vector3.left * mouseY);
        playerBody.Rotate(Vector3.up * mouseX);
    }

    private void ClampXAxisRotationToValue(float value)
    {
        Vector3 eulerRotation = transform.eulerAngles;
        eulerRotation.x = value;
        transform.eulerAngles = eulerRotation;
    }
}

```

PlayerMove : Σε αυτόν τον κώδικα ορίζεται η κίνηση του παίκτη. Στη συνάρτηση playermovement ορίζεται για το αν μετακινηθούμε οριζόντια ή κάθετα. Αν πατηθεί w θα πάρει την τιμή 1 και συνεπώς θα κινηθούμε μπροστά ανάλογα με την ορισμένη ταχύτητα ενώ αντίστοιχα αν πατηθεί s θα πάρει την τιμή -1 και θα κινηθεί προς τα πίσω. Κάπως έτσι λειτουργεί και οριζόντια με το d να παίρνει την θετική τιμή και να κινείται δεξιά ενώ το a να δέχεται την αρνητική τιμή για να κινηθεί αριστερά.

Τέλος, τοποθετήθηκε η συνάρτηση jumpInput το οποίο δέχεται το πλήκτρο που έχουμε ορίσει σαν input ( space στη συγκεκριμένη περίπτωση ), μετατρέπει το boolean isjumping ως αληθές και ξεκινάει την jumpEvent το οποίο κάνει τις απαραίτητες ενέργειες για να πραγματοποιηθεί το άλμα.

```
public class PlayerMove : MonoBehaviour
{
    [SerializeField] private string horizontalInputName;
    [SerializeField] private string verticalInputName;
    [SerializeField] private float movementSpeed;

    private CharacterController charController;

    [SerializeField] private AnimationCurve jumpFalloff;
    [SerializeField] private float jumpMultiplier;
    [SerializeField] private KeyCode jumpKey;

    private bool isJumping;

    private void Awake()
    {
        charController = GetComponent<CharacterController>();
    }

    private void Update()
    {
        PlayerMovement();
    }

    private void PlayerMovement()
    {
        float horizInput = Input.GetAxis(horizontalInputName) * movementSpeed;
        float vertInput = Input.GetAxis(verticalInputName) * movementSpeed;

        Vector3 forwardMovement = transform.forward * vertInput;
        Vector3 rightMovement = transform.right * horizInput;

        charController.SimpleMove(forwardMovement + rightMovement);

        JumpInput();
    }
}
```



```

private void JumpInput()
{
    if (Input.GetKeyDown(jumpKey) && !isJumping)
    {
        isJumping = true;
        StartCoroutine(JumpEvent());
    }
}

private IEnumerator JumpEvent()
{
    charController.slopeLimit = 90.0f;
    float timeInAir = 0.0f;

    do
    {
        float jumpForce = jumpFalloff.Evaluate(timeInAir);
        charController.Move(Vector3.up * jumpForce * jumpMultiplier * Time.deltaTime);
        timeInAir += Time.deltaTime;
        yield return null;
    } while (!charController.isGrounded && charController.collisionFlags != CollisionFlags.Above);

    charController.slopeLimit = 45.0f;
    isJumping = false;
}

```

MillsRotating: ο απλούστερος κώδικας της εργασίας, απλά περιστρέφεται άξονας ζ κατά 1.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class millsrotating : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      void Start()
9      {
10
11      }
12
13     // Update is called once per frame
14     void Update()
15     {
16         transform.Rotate(0,0,1);
17     }
18 }
19

```

Thoughts: Σε αυτόν τον κώδικα τοποθετήθηκαν οι σκέψεις του χρήστη ( εμφανίζονται στη κονσόλα), όταν φτάνει σε έναν προορισμό. Οι σκέψεις πρέπει να εμφανιστούν με την σειρά ώστε να δώσουμε κάποιο νόημα στο παιχνίδι.

```
// Start is called before the first frame update
void Start()
{
    Debug.Log("Finally i arrived");
}

// Update is called once per frame
void Update()
{
    thought();
}

public void thought()
{
    if (transform.position.z > 227 & transform.position.z < 228 & transform.position.x > 414 & transform.position.x < 418 & a == 0)
    {
        Debug.Log("Why are there so many fighters ?");
        a += 1;
    }
    if (transform.position.z > 193 & transform.position.z < 200 & transform.position.x > 425 & transform.position.x < 428 & a == 1)
    {
        Debug.Log("This town is so beautiful !");
        a += 1;
    }
    if (transform.position.z > 218.5 & transform.position.z < 224.5 & transform.position.x > 362.5 & transform.position.x < 363.5 & a == 2)
    {
        Debug.Log("This forest feels so...magical");
        a += 1;
    }
    if (transform.position.z > 283 & transform.position.z < 286 & transform.position.x > 286 & transform.position.x < 291 & a == 3)
    {
        Debug.Log("Why is the bridge protected?");
        a += 1;
    }
    if (transform.position.z > 205.5 & transform.position.z < 211.5 & transform.position.x > 136 & transform.position.x < 137 & a == 4)
    {
        Debug.Log("That castle has too many soldiers...Are they preparing for an invasion?!");
        a += 1;
    }
    if (transform.position.z > 350 & transform.position.z < 353 & transform.position.x > 260.5 & transform.position.x < 288 & a == 5)
    {
        Debug.Log("This forest feels deserted and haunted!");
    }
}
```

## Κριτήρια αξιολογησης

**Αληθοφάνεια** : Ικανοποιείται, λόγω κινήσεων των δέντρων με τον άνεμο, με την ρίψη νερού από τον καταρράκτη στο ποτάμι , καθώς και η ροή του νερού στο ποτάμι.

**Περιεχόμενο** : Το περιεχόμενο θυμίζει τον πραγματικό κόσμο, λόγω των κτιρίων, την παρουσία ηρς, την αληθοφανή κίνηση αυτών και του φυσικού περιβάλλοντος.

**Πληρότητα** : Ο κόσμος είναι ένα ολοκληρωμένο συστατικό τρισδιάστατου κόσμου, αφού εκτός από τον ίδιο τον χαρακτήρα που βλέπει και αλληλεπιδρά σαν να είναι μέσα σε αυτόν τον κόσμο, υπάρχει πιστή αναπαράσταση των κινούμενων στοιχείων και δεν υπάρχουν ασυνέχειες και ασυνήθιστα πράγματα ( π χ. παραβίαση φυσικών νόμων ).

**Σχεδιασμός :** Ο σχεδιασμός του κόσμου έγινε με λεπτομέρεια, αφού δόθηκε μεγάλη σημασία στα χρώματα και στη μορφολογία στο κάθε σημείο του τερραίν από τα βουνά μέχρι και το ποτάμι.

**Αισθητική :** Η αισθητική παρουσιάζεται απλή και ευπροσάρμοστη για τον χρήστη, καθώς μπορεί να αντιληφθεί πολύ γρήγορα τον περιβάλλον χώρο και δεν υπάρχουν πολλές αν όχι καθόλου απορίες για κάποιο αντικείμενο.

**Πρωτοτυπία :** Διερευνήθηκε με λεπτομέρεια ο σχεδιασμός και η υλοποίηση κάποιων συγκεκριμένων κτιρίων που δε μπορούσαν να βρεθούν στο asset store. Παράδειγμα αυτών είναι η δημιουργία υπαίθριων καταστημάτων στην αγορά καθώς και η υλοποίηση της γέφυρας.

**Χρηστικότητα :** Η παρουσία δρόμων, γέφυρας που ενώνει δύο σημεία και η ύπαρξη βαρκών στον μόλο καθώς και η ύπαρξη κάρων και φορτίων δίνουν την εντύπωση πως ο κόσμος αυτός χρειάζεται και παρέχει την μετακίνηση σε άλλα χωριά και την αλληλεπίδραση με αυτά.

**Κίνηση :** Υπάρχει πολλαπλή κίνηση στον κόσμο. Η κίνηση του ίδιου του χρήστη, των npc και του φυσικού περιβάλλοντος.

**Λειτουργικότητα :** Ο χρήστης μπορεί να περιηγηθεί μέσα στον 3D κόσμο μέσω ενός first person controller.

**Ανάπτυξη :** Η εφαρμογή υλοποιήθηκε σε unity version: 2019.3.0f6 με την βοήθεια του Microsoft Visual Studio 2019.

## Πηγές-assets

Δημιουργία δρόμων : <https://www.youtube.com/watch?v=1IWuvocBSuA>

Asset: <https://assetstore.unity.com/packages/3d/characters/easyroads3d-free-v3-987>

Δημιουργία Καταρράκτη : <https://www.youtube.com/watch?v=XhSp8nFLUi4>

Κώδικας για κίνηση npc: <https://www.youtube.com/watch?v=aEPSuGlcTUQ>

Κατανόηση και εφαρμογή wind zones : <https://www.youtube.com/watch?v=Lz8wEsvCWMs>

Κώδικας για fps : <https://www.youtube.com/watch?v=n-KX8AeGK7E>

### **Λοιπά Assets:**

1. για ζώα: <https://assetstore.unity.com/packages/3d/farm-animals-set-97945>
2. για knight: <https://assetstore.unity.com/packages/3d/characters/humanoids/strong-knight-83586>
3. για warriors: <https://assetstore.unity.com/packages/3d/characters/medieval-cartoon-warriors-90079>
4. για ξίφη και ασπίδες: <https://assetstore.unity.com/packages/3d/props/weapons/free-shield-and-sword-asset-pack-128672>
5. για εκκλησία :  
<https://assetstore.unity.com/packages/3d/environments/historic/church-model-110307>
6. για μαυσωλείο:  
<https://assetstore.unity.com/packages/3d/environments/fantasy/mausoleum-128753>
7. για άγαλμα: <https://assetstore.unity.com/packages/3d/environments/fantasy/angel-statue-27594>
8. για βοηθητικά prehab:  
<https://assetstore.unity.com/packages/3d/props/exterior/medieval-modular-city-bridge-and-ground-118277>  
<https://assetstore.unity.com/packages/3d/environments/fantasy/medieval-town-exterior-27026>  
<https://assetstore.unity.com/packages/3d/environments/historic/medieval-buildings-34770>  
<https://assetstore.unity.com/packages/3d/environments/fantasy/mega-fantasy-props-pack-87811>  
<https://assetstore.unity.com/packages/3d/environments/fantasy/glowing-forest-79686>  
<https://assetstore.unity.com/packages/3d/props/weapons/medieval-cannon-61834>  
<https://assetstore.unity.com/packages/3d/props/interior/treasure-set-free-chest-72345>  
<https://assetstore.unity.com/packages/2d/textures-materials/sky/skybox-4183>  
<https://assetstore.unity.com/packages/audio/music/piano-music-160494>