

“SAMURUN” THE VIDEO GAME

Η ΑΠΟΛΥΤΗ ΔΟΚΙΜΑΣΙΑ ΣΤΗ ΓΕΦΥΡΑ ΤΟΥ ΠΑΡΑΔΕΙΣΟΥ



GAME DESIGN DOCUMENT

ΜΑΘΗΜΑ: ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΠΑΙΓΝΙΩΝ

Περιεχόμενα

Ομάδα σχεδιασμού	2
Gameplay	3
Characters/Story	3
Core Mechanics	3
Διεπαφή με τον χρήστη	4
Menus	4
Κύρια οπτική / Κάμερα	6
Τεχνικές Παράμετροι	7
Ήχος	11
Απαιτήσεις Hardware / Διαθέσιμες Πλατφόρμες / Προσβασιμότητα	12

Ομάδα σχεδιασμού

Το παιχνίδι δημιουργήθηκε στα πλαίσια του μαθήματος «Σχεδίαση και Ανάπτυξη Παιγνίων» του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

Οι σχεδιαστές του παιχνιδιού απαρτίζουν την ομάδα Γ του μαθήματος και είναι οι παρακάτω:

- ΛΑΛΙΔΗΣ ΓΕΩΡΓΙΟΣ
- ΚΑΠΑΡΟΥΝΑΚΗΣ ΓΕΩΡΓΙΟΣ
- ΑΝΑΣΤΑΣΙΑΔΗΣ ΠΑΝΑΓΙΩΤΗΣ
- ΚΑΤΣΑΡΟΣ ΧΡΗΣΤΟΣ
- ΖΕΑΚΗΣ ΜΙΧΑΗΛ
- ΣΟΥΜΠΑΣΗΣ ΙΩΑΝΝΗΣ



Gameplay

Το παιχνίδι διαδραματίζεται σε μία εναέρια γέφυρα στην οποία ο χαρακτήρας προσπαθεί να αποφύγει τα διάφορα αντικείμενα που βρίσκει καθώς τρέχει κατά μήκος αυτής. Σκοπός του παίχτη είναι να καταφέρει να μείνει στην γέφυρα αποφεύγοντας τα εκάστοτε εμπόδια και συνεπώς να μαζέψει τους περισσότερους δυνατούς πόντους. Όσο περνάει ο χρόνος η ταχύτητα -και κατ' επέκταση η δυσκολία- του παιχνιδιού αυξάνεται. Το score διαμορφώνεται ανάλογα με την ταχύτητα του παιχνιδιού.

Όσον αφορά τον χαρακτήρα, ο χρήστης μπορεί να ελέγξει μόνον την κατεύθυνσή του κατά πλάτος της γέφυρας (αριστερά – δεξιά). Στην αρχή του παιχνιδιού ο παίχτης δεν έχει καμία επίγνωση των εμποδίων που έρχεται αντιμέτωπος και η αλληλεπίδρασή του με αυτά είναι σταδιακή.

Το παιχνίδι τελειώνει όταν ο Samurai συγκρουστεί με κάποιο εμπόδιο ή εφόσον πέσει στο κενό έξω από τα όρια της γέφυρας. Στο τέλος του παιχνιδιού εμφανίζεται το μήνυμα “GAME OVER” και το score που κατάφερε να μαζέψει ο παίχτης. Σε περίπτωση που το score είναι το μεγαλύτερο που έχει επιτευχθεί στο παιχνίδι ανανεώνεται η ένδειξη “Highscore” στο Main Menu του παιχνιδιού.

Characters / Story

Το concept του παιχνιδιού είναι βασισμένου στην Ιαπωνική κουλτούρα και αφορά έναν και μόνο χαρακτήρα, έναν μαθητευόμενο πολεμιστή-samurai. Η ιστορία εκτυλίσσεται γύρω από την προσπάθεια του πρωταγωνιστή να φέρει εις πέρας την δοκιμασία του -να διασχίσει την γέφυρα του παραδείσου “Samurun”- έτσι ώστε να τελειώσει επιτυχώς την εκπαίδευσή του και να πραγματοποιήσει το όνειρό του, να γίνει Samurai.

Ωστόσο ο πρωταγωνιστής μας καλείται να διασχίσει μία ατέρμονη γέφυρα χωρίς να καταλάβει ότι η τελευταία του δοκιμασία είναι ακατόρθωτη. Όσο πιο πολύ διασχίζει τη γέφυρα, τόσο περισσότερο θα ωριμάσει και τελικά το νόημα της δοκιμασίας είναι να συνειδητοποιήσει ότι ακριβώς όπως η ζωή του θα είναι μια ατέλειωτη πρόκληση, έτσι και κάθε καλός Samurai δεν πρέπει να φοβάται να αντιμετωπίσει κανένα εμπόδιο που το επιφυλάσσει το μέλλον.



Core Mechanics

Η δομή του παιχνιδιού με εξαίρεση των δύο Menus (Main Menu και Game Over Menu), βασίζεται σε ένα και μοναδικό περιβάλλον, τη πίστα με τη γέφυρα SAMURUN. Η πίστα χωρίζεται σε tiles τα οποία δεν διακρίνονται από τον χρήστη, αλλά δίνουν την εντύπωση μιας ενιαίας δομής, την γέφυρα. Το σύνολο των tiles περιέχει ποικιλία εμποδίων, η οποία βασίζεται σε στοιβαγμένα αντικείμενα ιαπωνικής θεματολογίας. Τα εμπόδια με τον



τρόπο που είναι διατεταγμένα αναγκάζουν τον εκπαιδευόμενο Samurai να ελίσσεται δεξιά και αριστερά για να τα αποφύγει. Επιπλέον, το background του παιχνιδιού για να πλαισιώσει το θέμα της γέφυρας του παραδείσου αποτελείται από ένα ειδυλλιακό τοπίο με σύννεφα και γαλάζιο ουρανό.

Διεπαφή με τον χρήστη

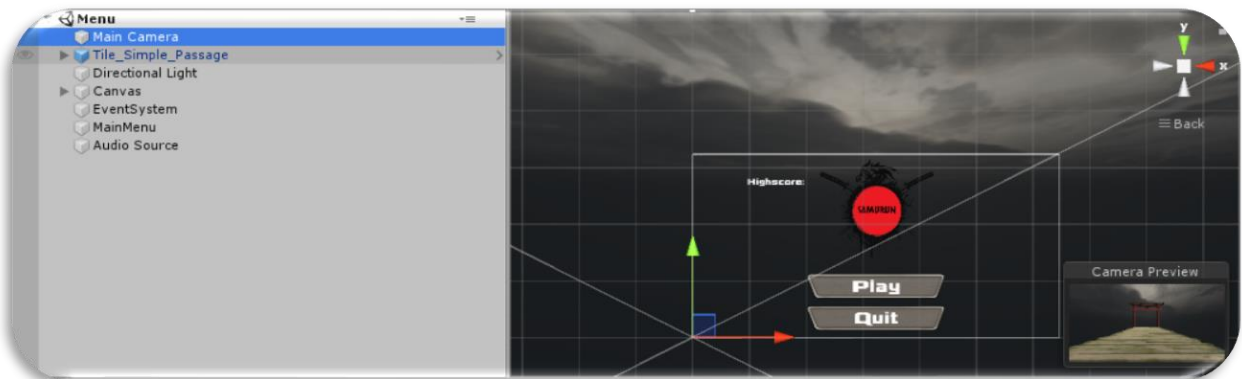
Η αλληλεπίδραση του χρήστη με το περιβάλλον του παιχνιδιού είναι άμεση και εύκολη, καθώς στο σύνολο του αποτελείται από τρία σκηνικά περιβάλλοντα, αυτά του του Main Menu, του Game Over Menu και της κύριας σκηνής-πίστας.

Η σκηνή της πίστας κατά την οποία ο παίκτης προσπαθεί να περάσει επιτυχώς τα εμπόδια μένοντας όσο περισσότερο χρόνο μπορεί χωρίς να χάσει, καταλαμβάνει και το περισσότερο χρόνο που αλληλοεπιδρά ο χρήστης με το παιχνίδι.

Η σχεδιαστική υλοποίηση πίσω από τα Main Menu και Game Over Menu είναι σκόπιμα λιτή και απλή ώστε να οδηγούν τον παίκτη χωρίς χρονικούς περιορισμούς, loading screens και περιττά animations πίσω στο βασικό παιχνίδι. Με τον τρόπο αυτό ενισχύουν το διασκεδαστικό και ταυτόχρονα εθιστικό στοιχείο του παιχνιδιού, τη συνεχή προσπάθεια του χρήστη να πετύχει μεγαλύτερο σκορ επαναλαμβάνοντας τη πίστα.

Menus

Main Menu:



Το Main Menu αποτελείται από το εξής στοιχεία: τα Play και Quit Buttons, το Πεδίο Highscore, το Logo του Παιχνιδιού και το background.

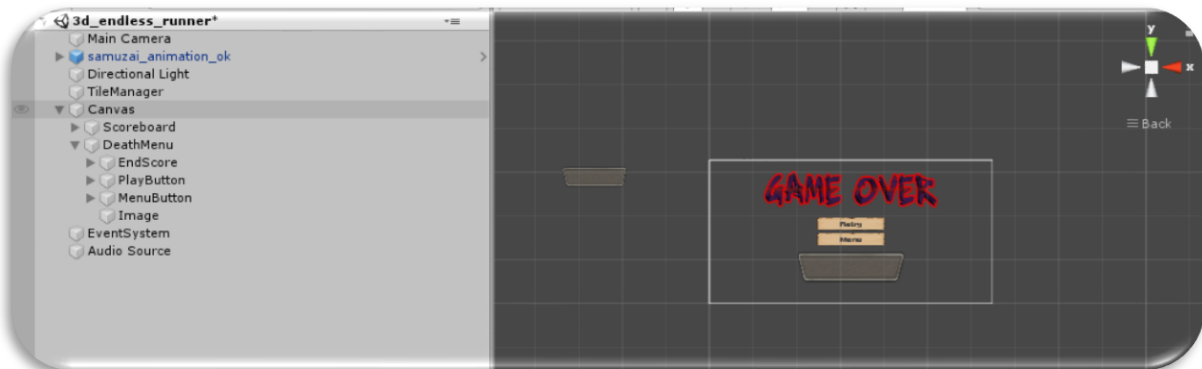
- Τα κουμπιά Play και Quit, όπως λέει και το όνομα τους, οδηγούν το πρώτο στην βασική πίστα του παιχνιδιού και το δεύτερο στον τερματισμό της εφαρμογής.

- Το πεδίο Highscore αποθηκεύει το τρέχον μεγαλύτερο Score που έχει επιτευχθεί και συνεπώς δείχνει στον χρήστη κατά το καλωσόρισμα της εφαρμογής, τον αριθμό που πρέπει να ξεπεράσει για να δει το δικό του Score τυπωμένο πάνω στο πεδίο.
- Το Logo SAMURUN βρίσκεται ψηλά στο κέντρο ως το αναγνωριστικό του παιχνιδιού και το background (το ιαπωνικό Tori και το κομμάτι της γέφυρας) εξυπηρετεί ως εισαγωγή στη μορφή της πίστας που θα ανακαλύψει ο παίκτης πατώντας το Play Button. Στο ίδιο λογικό πλαίσιο, η ομίχλη και τα μαύρα σύννεφα αποκρύπτουν για αισθητικό σκοπό το περιβάλλον της πίστας, κρατώντας έτσι μυστήριο τον πραγματικό κόσμο του παιχνιδιού.



Game Over Menu:

Το Game Over Menu του παιχνιδιού εξυπηρετεί ως η οθόνη που αντικρίζει ο παίκτης τη στιγμή που χάσει. Στο Menu αυτό, το κουμπί Retry οδηγεί τον παίκτη ξανά από την αρχή στη βασική πίστα, ενώ το κουμπί Menu επιστρέφει στο Main Menu. Το asset-εικόνα "GAME OVER" υποδηλώνει την ήττα του παίκτη και στο πλαίσιο κάτω από τα κουμπιά μπορεί κανείς να δει το σκορ που επιτεύχθηκε για το τρέχον playthrough.



Κύρια οπτική / Κάμερα

Η οπτική του παιχνιδιού βασίζεται στη σταθερή γωνία, από την οποία κοιτά η κάμερα τον χαρακτήρα-Samurai, την γέφυρα και τα εμπόδια που εκτυλίσσονται μπροστά του. Η συγκεκριμένη οπτική μένει απaráλλαχτη κατά τη διάρκεια του παιχνιδιού και εξυπηρετεί δυο κύριους στόχους. Συγκεκριμένα, η κάμερα βρίσκεται πίσω από τον χαρακτήρα και ελάχιστα ψηλότερα από αυτόν, με τρόπο ώστε ο παίκτης να παρατηρεί από τη μία ξεκάθαρα το περιβάλλον των εμποδίων που αναπτύσσεται μπροστά του και από την άλλη να μη νιώθει ξένος από αυτό, αλλά κομμάτι της πρόκλησης που αντιμετωπίζει ο χαρακτήρας. Επιπλέον, η κάμερα τοποθετήθηκε κοντά στον Samurai-χαρακτήρα, καθώς έτσι γίνεται δυσδιάκριτο από μακρινή απόσταση το μοτίβο κινήσεων καθώς και το χρονικό σημείο που πρέπει να το πραγματοποιήσει ο παίκτης, αυξάνοντας συνεπώς τη δυσκολία του παιχνιδιού και εντείνοντας την ανάγκη να προγραμματίσει τις ενέργειες που πρόκειται να κάνει.



Μικρή εξαίρεση όσον αφορά την οπτική, αποτελεί το animation στην έναρξη του παιχνιδιού, κατά το οποίο η κάμερα βρίσκεται σε top down οπτική γωνία προς το παίκτη και σταδιακά αλλά σε σύντομο χρόνο, κατέρχεται για να οριστικοποιηθεί στη σταθερή γωνία. Η κινησιολογία αυτή της κάμερας αλλά και όλη η συμπεριφορά της, υλοποιήθηκαν σε σχετικό script σε C#.

```
public class CameraMotor : MonoBehaviour
{
    private Transform lookAt;
    private Vector3 startOffset;
    private Vector3 moveVector;

    private float transition = 0.0f;
    private float animationDuration = 2.0f;
    private Vector3 animationOffset = new Vector3(0, 5, 5);
    // Start is called before the first frame update
    void Start()
    {
        lookAt = GameObject.FindGameObjectWithTag("Player").transform;
        startOffset = transform.position - lookAt.position;
    }

    // Update is called once per frame
    void Update()
    {
        moveVector = lookAt.position + startOffset;

        //X
        moveVector.x = 0;

        //Y
        moveVector.y = Mathf.Clamp(moveVector.y * 2, -3, 5);

        if (transition > 1.0f)
        {
            transform.position = moveVector;
        }
        else
        {
            //Animation at the start of the game
            transform.position = Vector3.Lerp(moveVector + animationOffset, moveVector, transition);
            transition += Time.deltaTime * 1 / animationDuration;
            transform.LookAt(lookAt.position + Vector3.up);
        }
    }
}
```



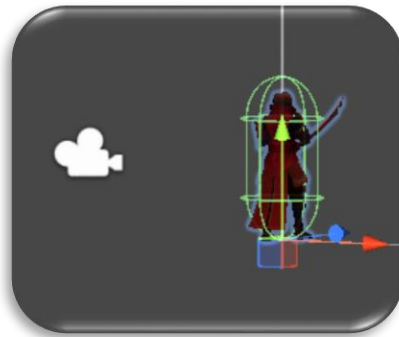
Τεχνικές Παράμετροι

Assets:

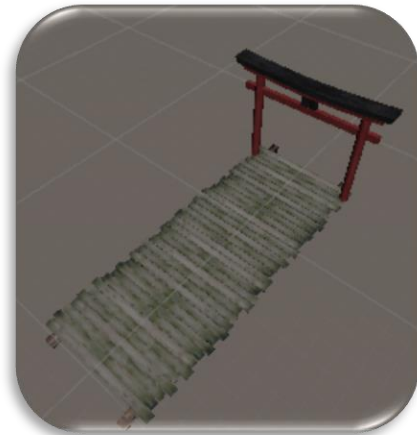
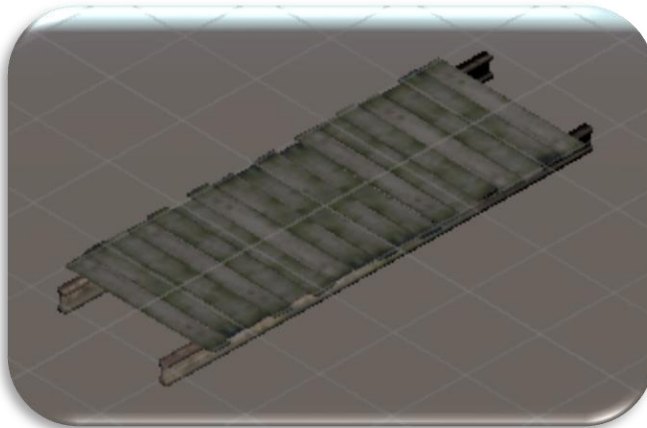
Η υλοποίηση του παιχνιδιού έγινε ολοκληρωτικά στη μηχανή του Unity -3D 2019 και τα assets που ενσωματώθηκαν, πάρθηκαν εξ ολοκλήρου από δωρεάν πακέτα του Unity Asset Store. Συνεπώς η ιδέα και το concept του παιχνιδιού καλύφθηκε από έτοιμα Assets του καταστήματος που πληρούσαν τις απαιτήσεις του σχεδιασμού.

Η ιαπωνική κουλτούρα, όντας το βασικό στοιχείο γύρω από την ιδέα του SAMURUN, αποτέλεσε το κυρίαρχο κριτήριο επιλογής των Assets που χρησιμοποιήθηκαν. Αναλυτικά:

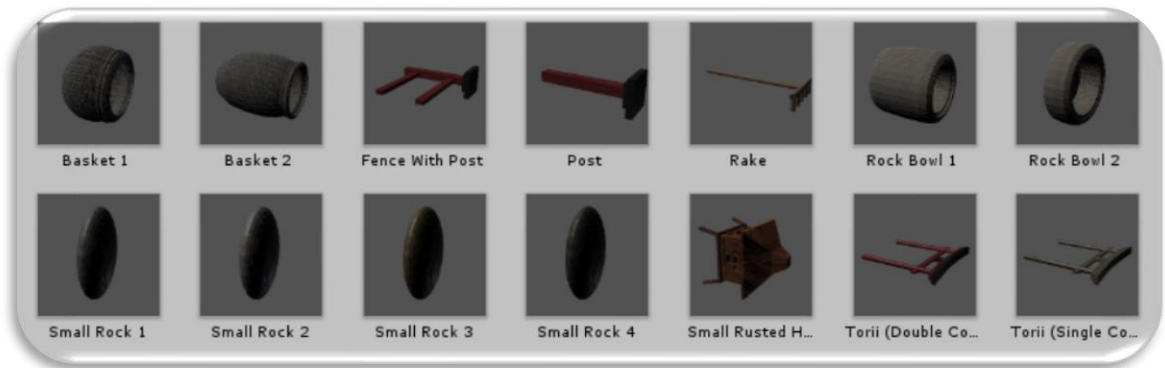
- **Red Samurai Character Pack:** Το Asset του Samurai χρησιμοποιήθηκε ως ο πρωταγωνιστής του παιχνιδιού με τον οποίο αλληλεπιδρά ο παίκτης.



- **Modular Wooden Bridge Tiles Pack:** Το συγκεκριμένο πακέτο περιέχει τα κομμάτια που συντέλεσαν τα tiles της ξύλινης γέφυρας SAMURUN, δηλαδή της βασικής πίστας του παιχνιδιού.



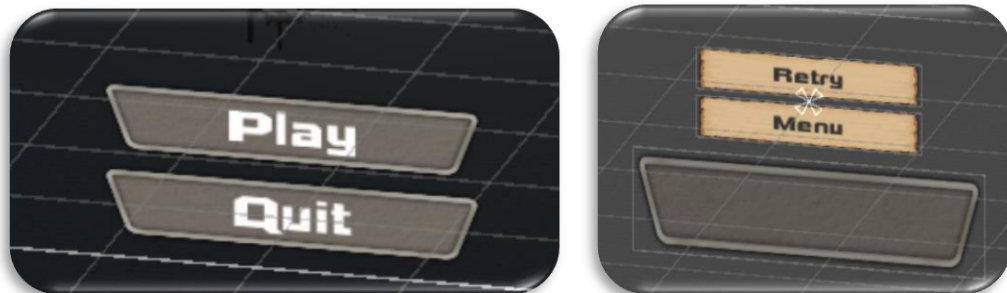
- **Japanese Garden Pack:** Εκτός από το asset του samurai, το συγκεκριμένο πακέτο από assets υποδεικνύει και αυτό τον ιαπωνικό χαρακτήρα του παιχνιδιού, διότι περιέχει το ιαπωνικό Torii, διάφορα καλάθια, βαρέλια και αντικείμενα σχετικά με το θεματικό του παιχνιδιού. Τα Assets αυτά τοποθετήθηκαν ως εμπόδια στη γέφυρα SAMURUN.



- **Medieval Barrels and Boxes Pack:** Πακέτο με βαρέλια και κουτιά μεσαιωνικού χαρακτήρα που συμπληρώνουν τα εμπόδια της πίστας μαζί με τα ιαπωνικά αντικείμενα.

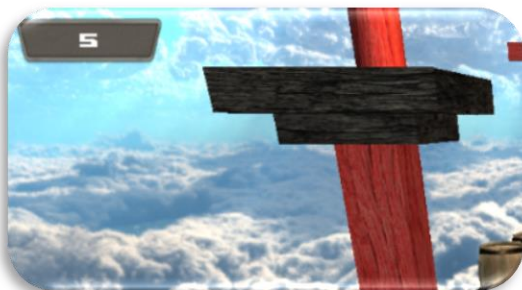


- **Fantasy Wooden GUI Pack:** Το Graphical User Interface του παιχνιδιού (Buttons, πλαίσια Score και Highscore) βασίστηκε σε αυτό το πακέτο.



- **Skyseries Freebie Skybox Pack:** Ο σκοτεινός ουρανός του Main Menu και τα σύννεφα στον γαλάζιο ουρανό που αποτελούν το background της βασικής πίστας.





- **Εικόνες:** Για τις ανάγκες του παιχνιδιού και ιδιαίτερα για το Game Over Menu και το Logo της εφαρμογής, δημιουργήθηκαν και ενσωματώθηκαν ως assets στο γενικό project οι ακόλουθες εικόνες. Τα συγκεκριμένα κομμάτια δεν αποτελούν προϊόν του Asset Store.



GAME OVER

Συναρμολόγηση της βασικής πίστας:

Η γέφυρα SAMURUN που αποτελεί το επίπεδο στο οποίο ο χαρακτήρας τρέχει προσπερνώντας τα εμπόδια, απαρτίζεται από την συνένωση των παραπάνω assets. Στο πυρήνα της η πίστα είναι ένα σύνολο από διαφορετικά Tiles, τα οποία παράγονται τυχαία το ένα μετά το άλλο κατά τη διάρκεια του παιχνιδιού και προσφέρει την εντύπωση μιας γέφυρας απείρου μεγέθους. Η ιδιότητα αυτή επιτυγχάνεται μέσω κώδικα σε C#, ο οποίος πάντα παράγει διαφορετικό Tile από το αμέσως προηγούμενο, ενώ κατά το σύντομο εναρκτήριο animation της κάμερας, εμφανίζει μόνο Tiles χωρίς εμπόδια.

Τα tiles που κατασκευάστηκαν, είναι σε αριθμό οχτώ και περιέχουν ποικιλία εμποδίων. Τα εμπόδια απαρτίζονται από τα assets που προαναφέρθηκαν και χαρακτηρίζονται από διαβαθμιζόμενη δυσκολία, αφού άλλα καταλαμβάνουν πολύ χώρο στη πίστα και άλλα λιγότερο.

Παρακάτω παρατίθενται τα Tiles, καθώς και το Script TileManager που είναι υπεύθυνο για την τυχειότητα στην αναπαραγωγή τους.



```

public class TileManager : MonoBehaviour
{
    public GameObject[] tilePrefabs;
    private Transform playerTransform;
    private float spawnZ = -12.0f;
    private float tileLength = 9.5f;
    private int amnTilesOnScreen = 7;
    private float safeZone = 25.0f;
    private int lastPrefabIndex = 0;

    private List<GameObject> activeTiles;

    // Start is called before the first frame update
    void Start()
    {
        activeTiles = new List<GameObject>();
        playerTransform = GameObject.FindWithTag("Player").transform;
        for (int i = 0; i < amnTilesOnScreen; i++)
        {
            if (i < 3)
                SpawnTile(1);
            else
                SpawnTile();
        }
    }

    // Update is called once per frame
    void Update()
    {
        if (playerTransform.position.z - safeZone > (spawnZ - amnTilesOnScreen * tileLength))
        {
            SpawnTile();
            DeleteTile();
        }
    }
}

```

```

private void SpawnTile(int prefabIndex = -1)
{
    GameObject go;
    if (prefabIndex == -1)
        go = Instantiate(tilePrefabs[RandomPrefabIndex()]) as GameObject;
    else
        go = Instantiate(tilePrefabs[prefabIndex]) as GameObject;
    go.transform.SetParent(transform);
    go.transform.position = Vector3.forward * spawnZ;
    spawnZ += tileLength;
    activeTiles.Add(go);
}

private void DeleteTile()
{
    Destroy(activeTiles[0]);
    activeTiles.RemoveAt(0);
}

private int RandomPrefabIndex()
{
    if (tilePrefabs.Length <= 1)
        return 0;

    int randomIndex = lastPrefabIndex;
    while (randomIndex == lastPrefabIndex)
    {
        randomIndex = Random.Range(0, tilePrefabs.Length);
    }
    lastPrefabIndex = randomIndex;
    return randomIndex;
}

```

Animations:

Το παιχνίδι συμπεριλαμβάνει έναν αριθμό από διαφορετικά animations. Στα κυριότερα από αυτά συγκαταλέγονται το εναρκτήριο animation της κάμερας που προαναφέρθηκε (βλέπε **Κύρια Οπτική / Κάμερα**) αλλά και το animation της σωματικής κίνησης του χαρακτήρα (η ιδιότητα του να φαίνεται πως τρέχει), το οποίο ήταν έτοιμο και ενσωματωμένο στο asset του samurai. Τα υπόλοιπα animations λειτουργούν ως συνδεδεμένος κρίκος στην σχεδίαση και αφορούν την εμφάνιση των Menus και τη προβολή της αύξησης του score σε πραγματικό χρόνο, για τα οποία έχει γραφτεί το κατάλληλο C# Script.

Υλοποίηση της αλληλεπίδρασης του χαρακτήρα με τη πίστα:

Ιδιαίτερη σημασία πρέπει να δοθεί στη κίνηση του χαρακτήρα, η οποία αποτελεί την άμεση διεπαφή του παιχνιδιού με τον παίκτη. Τις ανάγκες της διάδρασης αυτής, καλύπτει ο βασικός κώδικας της υλοποίησης και συγκεκριμένα το script `PlayerMotor`. Συνοπτικά, ο κώδικας υλοποιεί τις εξής λειτουργίες:

- Κίνηση του χαρακτήρα δεξιά-αριστερά με τα πλήκτρα βέλη, το ποντίκι ή την αφή στα κινητά.
- Αυξανόμενη ταχύτητα του χαρακτήρα με το πέρασ του χρόνου μέχρι μια μέγιστη τιμή.
- Ήττα του παίκτη όταν συγκρουστεί με εμπόδιο (Εμφάνιση Game Over Menu).
- Ήττα του παίκτη όταν πέσει στο κενό έξω από τα όρια της γέφυρας.

```
// Start is called before the first frame update
void Start()
{
    controller = GetComponent<CharacterController>();
    startTime = Time.time;
}

// Update is called once per frame
void Update()
{
    if (isDead)
        return;

    if( Time.time - startTime < animationDuration )
    {
        controller.Move(Vector3.forward * speed * Time.deltaTime);
        return;
    }

    moveVector = Vector3.zero;

    if (controller.isGrounded)
    {
        verticalVelocity = -0.5f;
    }
    else
    {
        verticalVelocity -= gravity * Time.deltaTime;
    }

    //X => Left and Right
    moveVector.x = Input.GetAxisRaw("Horizontal") * speed;
    if ( Input.GetMouseButton(0) )
    {
        //We are holding touch on the right
        if (Input.mousePosition.x > Screen.width / 2)
            moveVector.x = speed;
        else
            moveVector.x = -speed;
    }
}
```

```
//Y => Up and Down
moveVector.y = verticalVelocity;
if (moveVector.y < -10)
    Death();

//Z => Forward and Backward
moveVector.z = speed;

controller.Move(moveVector * Time.deltaTime);

public void SetSpeed (float modifier)
{
    speed = 5.0f + modifier;
}

private void OnControllerColliderHit(ControllerColliderHit hit)
{
    if (hit.point.z > transform.position.z + (controller.radius))
        Death();
}

private void Death()
{
    isDead = true;
    GetComponent<Score>().OnDeath();
}
```

Ήχος

Το SAMURUN έχει δυο πηγές ηχητικής υπόκρουσης, οι οποίες είναι σχετικές με την ιαπωνική κουλτούρα που υιοθετεί. Το πρώτο ηχητικό απόσπασμα εμφανίζεται στο Main Menu και αποτελεί ένα instrumental κομμάτι με στοιχεία heavy metal είδους μουσικής και είναι επιπλέον βασισμένο στην ιαπωνική μουσική και τα τοπικά μουσικά όργανα. Είναι δομημένο με τρόπο ώστε να γίνεται έντονα αντιληπτό στον παίκτη που τρέχει την εφαρμογή και να τον προετοιμάζει για τη διασκέδαση που ακολουθεί πατώντας το Play Button. Ομοίως με το πρώτο, το δεύτερο ηχητικό κομμάτι που ακούγεται κατά τη διάρκεια του playthrough στη πίστα, συνδέεται με την Ιαπωνία και τα μουσικά όργανα-ταμπούρια. Ηχητικά, συμβαδίζει απόλυτα με την κλιμάκωση της δυσκολίας του παιχνιδιού, καθώς σταδιακά γίνεται εντονότερο.

Απαιτήσεις Hardware / Διαθέσιμες Πλατφόρμες / Προσβασιμότητα

Το παιχνίδι SAMURUN σχεδιάστηκε ώστε να μην απαιτεί κάποιο εξειδικευμένο hardware (πχ κάμερα) για να μπορεί να τρέξει. Έχει ελάχιστες απαιτήσεις σε υπολογιστική δύναμη που τις καλύπτει για παράδειγμα ένας υπολογιστής χωρίς εξωτερική κάρτα γραφικών. Παράλληλα είναι ιδιαίτερα οικονομικό σε θέματα αποθηκευτικού χώρου και μνήμης.

Μέσω της μηχανής Unity το παιχνίδι εξάχθηκε για λογισμικά Windows και Android και επομένως είναι συμβατό με ηλεκτρονικούς υπολογιστές και smartphones.

Ως προς την προσβασιμότητα του παιχνιδιού, η απλή σχεδίαση και το περιεχόμενο του το κάνει προσβάσιμο σε όλες τις ηλικίες και κοινωνικές ομάδες, ενώ δεν απαιτεί κάποια γνώση ή συγκεκριμένες ικανότητες για να χρησιμοποιηθεί. Μελλοντικά, κύριος σχεδιαστικός στόχος αποτελεί η βελτίωση της εφαρμογής για την καλύτερη εμπειρία των ατόμων με ειδικές ανάγκες, βασισμένη στο δικό τους feedback πάνω στην υπάρχουσα έκδοση του παιχνιδιού.