



Week 12 Presentation

Group 1:

Lyall Campbell

Laura Milkevičiūtė

Connor Grundy

Declan Wylie

Dale Mehmet

The Team

- Lyall Campbell: Game Design, UI Design
- Laura Milkevičiūtė: Art, Playtesting
- Connor Grundy: Game Design, Playtesting
- Declan Wylie: Programming
- Dale Mehmet: Art

Presentation Outline

- Game Concept
- Towers: Design, Behaviour & Art
- Enemies: Design, Behaviour & Art
- Levels: Design, Art & Implementation
- UI Design
- Playtesting
- Game Trailer
- Q&A

Game Concept

CINEMA + TOWER DEFENCE = ?

Game Concept CINEMA THEME

- Avoid limiting our design possibilities too early on.
- Allow as much freedom as possible despite the limits.
- Film series: different levels, different themes.
- Time travel: four different settings, each level is an instalment.
- Wrapped in with movie posters, cinema and transition to “movie”

Game Concept

TOWER DEFENCE

- Time traveller defending against attackers of each time period.
- 8 placeable towers, same across each level but different appearance.
- Similar for enemies: same stats, different appearance.
- 4 placeable traps with different effects.

Towers

BASIC	Standard tower.
FAST	Less damage but quicker rate of fire compared to basic.
AREA OF EFFECT	Shoots a projectile which deals damage in a radius.
PIERCING	Projectiles can pierce through multiple enemies.
LONGSHOT	More damage and larger range but lower rate of fire.
SLOW	Shoots a projectile which slows enemies.
DAMAGE OVER TIME	Applies a dot to enemies, dealing damage over time to enemies.

Tower Behaviour

- Towers use `Physics.OverlapSphere` to store all enemies within range into an array.
- The tower then goes through the array and targets each enemy going through the array. Enemies are taken out if they are out of range and new enemies are added when they enter the tower's range.
- Each tower type shoots a different projectile. These projectiles all have the same properties apart from a few because the bullet effect is determined by the enemy with a `.tag` check.



Tower Behaviour

- Towers can be upgraded, before the tower is upgraded, a check is performed.
- Some towers have different behaviours, for instance, there is a bomb tower that will shoot a standard projectile. However, on collision, the projectile be destroyed and spawn a growing explosion.
- Another example of this is the Slow Tower. This tower shoots normal projectiles at enemies but on hit, will slow the enemies down to a certain percentage of their original speed temporarily.
- These additional projectiles have separate scripts for each behaviour.

```
IEnumerator ScaleExplosion()
{
    float timer = 0;

    while (timer <= time) {

        timer += Time.deltaTime;
        float size = Mathf.Lerp(initialSize, finalSize, timer/time);

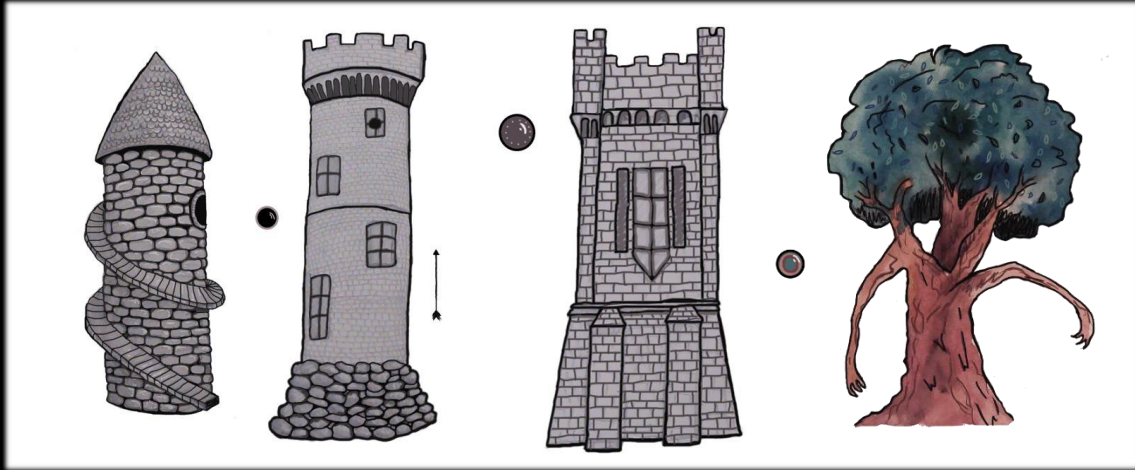
        gameObject.transform.localScale = new Vector3(size, size, size);

        yield return null;

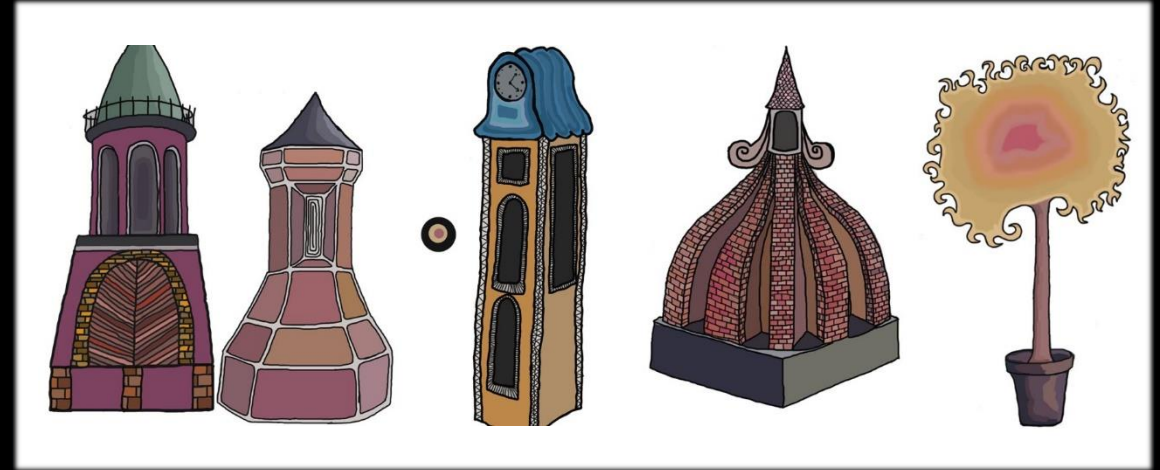
    }

    Destroy (gameObject);
}
```

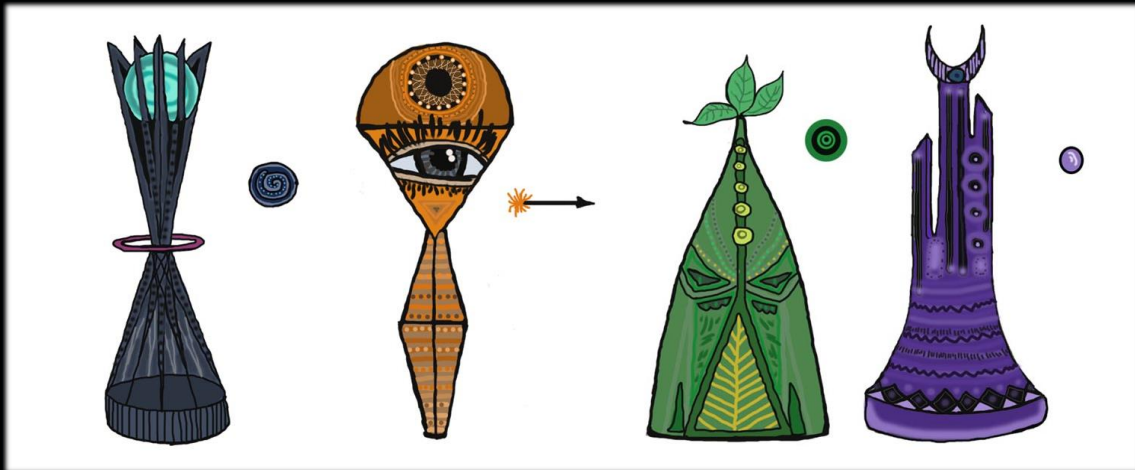
Tower Art



MEDIEVAL



VICTORIAN



SCI-FI

- 2D
- Sketches hand-drawn using watercolour.
- Assets made using Photoshop.
- Colour palette used to represent each era.
- Black outlining to make it look unrealistic.

Enemies

BASIC	Standard enemy, nothing special.
REGULAR	Similar to basic: more health, more damage.
FAST	Fast enemy, low health and deals small damage.
SPEEDY	Even faster, very low health but deals the same damage as fast.
POWERFUL	Improved version of regular: even more health, even more damage.
LEADER	Reasonably strong, somewhat fast and deals considerable damage.
GROUP	Similar stats to leader but deals more damage.
TANK	Lots of health and deals great amount of damage but moves slowly.
TANK II	Slower but has increased health and deals a very high amount of damage.

Enemy Behaviour

- Enemies follow a set of points or “nodes” which are stored in an array.
- The pathfinding script guides the enemy from node-to-node by working out the next direction from the current index using some simple vector math. (The movementDirection is normalised because we don't care about the distance!)
- The enemies will deal damage to the player if they hit the last node. This damage is decided by the enemy type.
- Money and score is awarded when enemies have been defeated by a tower.

```
void OnTriggerEnter(Collider other)
{
    if(other.gameObject.name == nodePoints[currentNodeIndex].name)
    {
        currentNodeIndex++;
        if(currentNodeIndex >= nodePoints.Length)
        {
            Destroy(gameObject);
        }

        else
        {
            movementDirection = (nodePoints[currentNodeIndex].transform.position - transform.position).normalized;
        }
    }
}
```

Enemy Behaviour

- The damage done to the enemies is dependant on the projectile's tag or name that collided with them.
- Status effects originating from utilities are handled on the enemy whereas effects originating from projectiles are handled on the projectile themselves.
- Enemies of different types have different speeds, damage to the player and health along with different sprites to accompany them.

```
if (other.GetComponent<Collider>().name == "FreezePrefab(Clone)")
{
    speed = 0.0f;
    utilitySlowed = true;
    enemyHealth -= 5;
    FreezeUtilityScript freezeUtility = other.GetComponent<Collider>().GetComponent<FreezeUtilityScript>();
    freezeUtility.utilityDurability--;
    remainingUtilitySlow = Time.time + 3.0f;|
}
```

```
void OnTriggerEnter(Collision collision)
{
    if (collision.collider.tag == "Enemy")
    {
        if(!collision.collider.GetComponentInChildren<SpeedHandlerScript>())
        {
            EnemyScript enemyScript = collision.collider.GetComponent<EnemyScript>();

            GameObject startSpeed = Instantiate(startSpeedRestoreObject) as GameObject;
            startSpeed.transform.parent = collision.collider.transform;

            SpeedHandlerScript startSpeedRestore = startSpeed.GetComponent<SpeedHandlerScript>();
            startSpeedRestore.revertTime = slowTime;
            startSpeedRestore.startSpeed = enemyScript.speed;

            enemyScript.speed *= slowAmount;
        }
    }
}
```

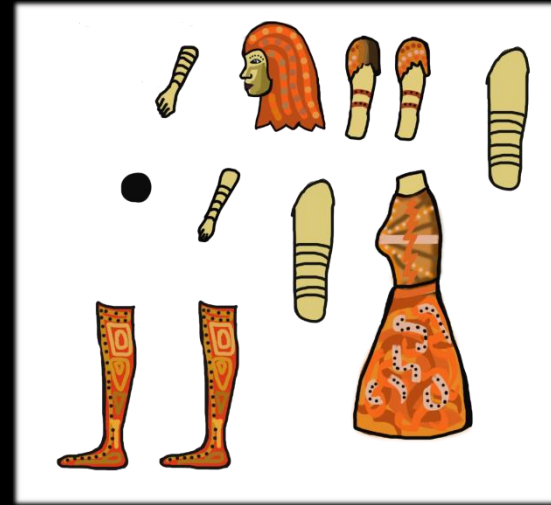
Enemy Art



MEDIEVAL

SCI-FI

VICTORIAN



BODY PARTS

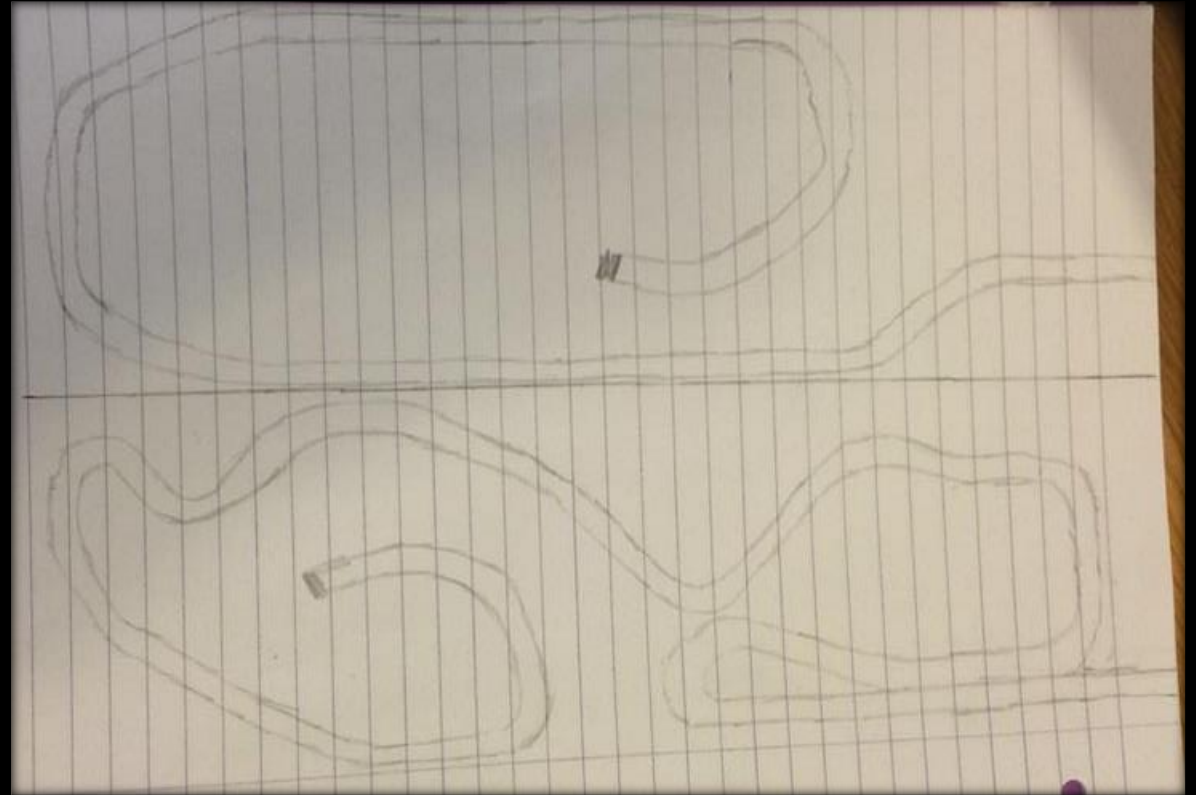
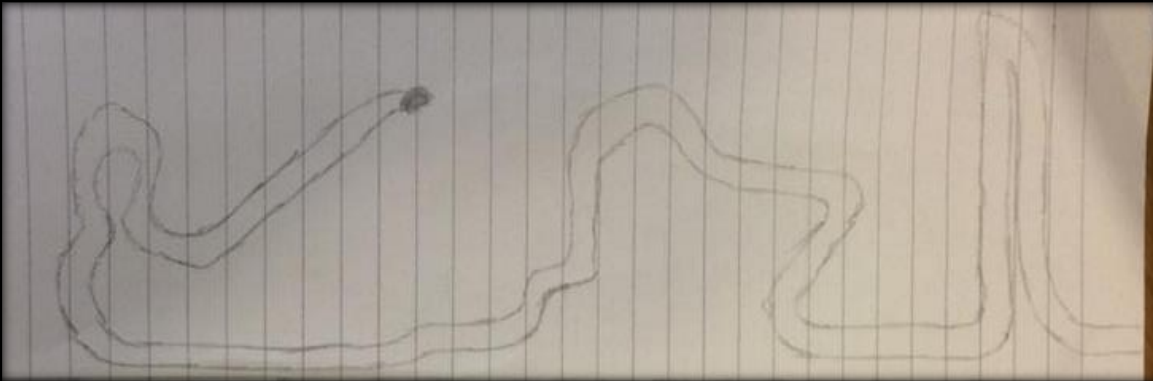


ANIMATION SCREENSHOTS

- 2D
- Sketches drawn by hand with watercolour.
- Made in Photoshop by drawing over sketches.
- Characters' body parts were drawn separately.
- Using Unity, body parts were put together.
- Bone based animation.

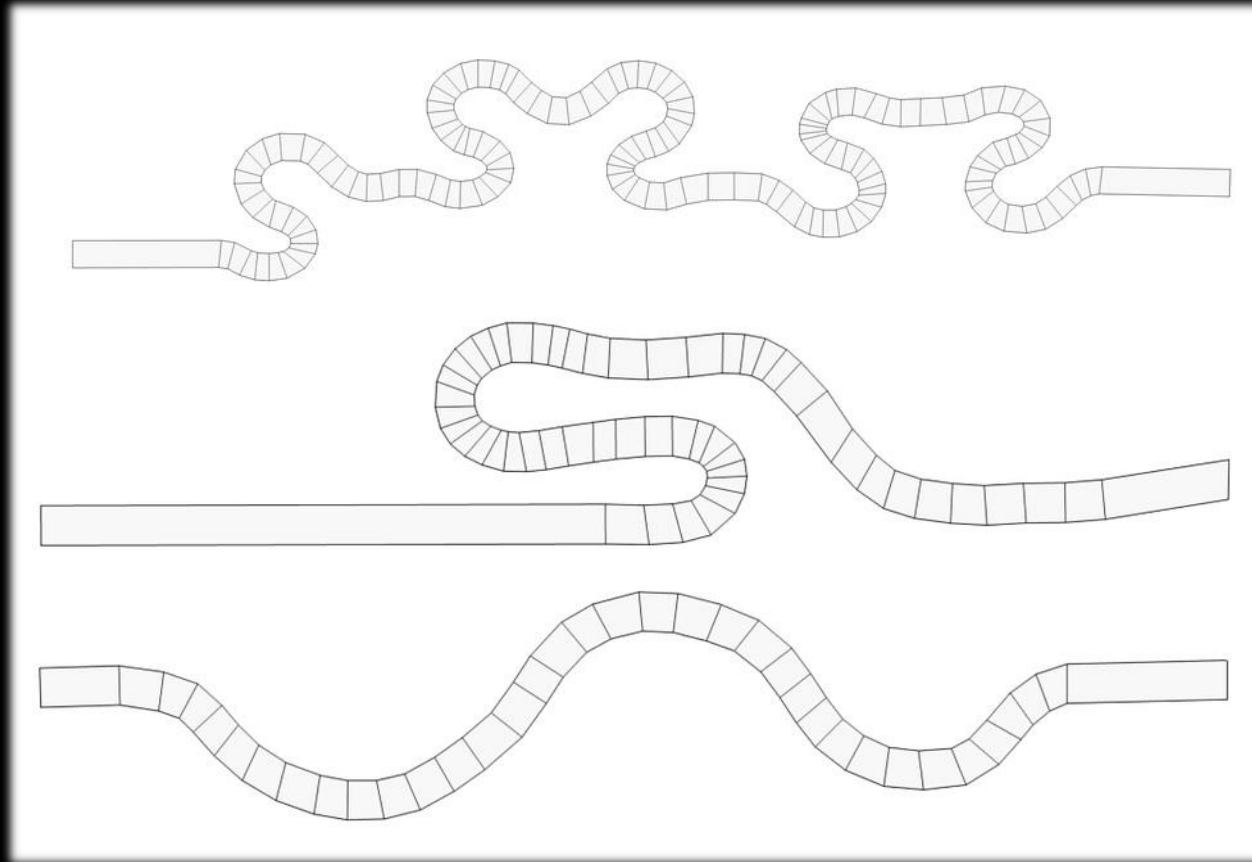
Level Design

- Various sketches were done to begin with until the ones to be used were agreed upon.



Level Design

- The 3 paths that were chosen contribute to the increase in difficulty as the player progresses through levels.
- Less vantage points requires more strategy and tactical tower placement to succeed.



Level Art



MEDIEVAL

Level Art



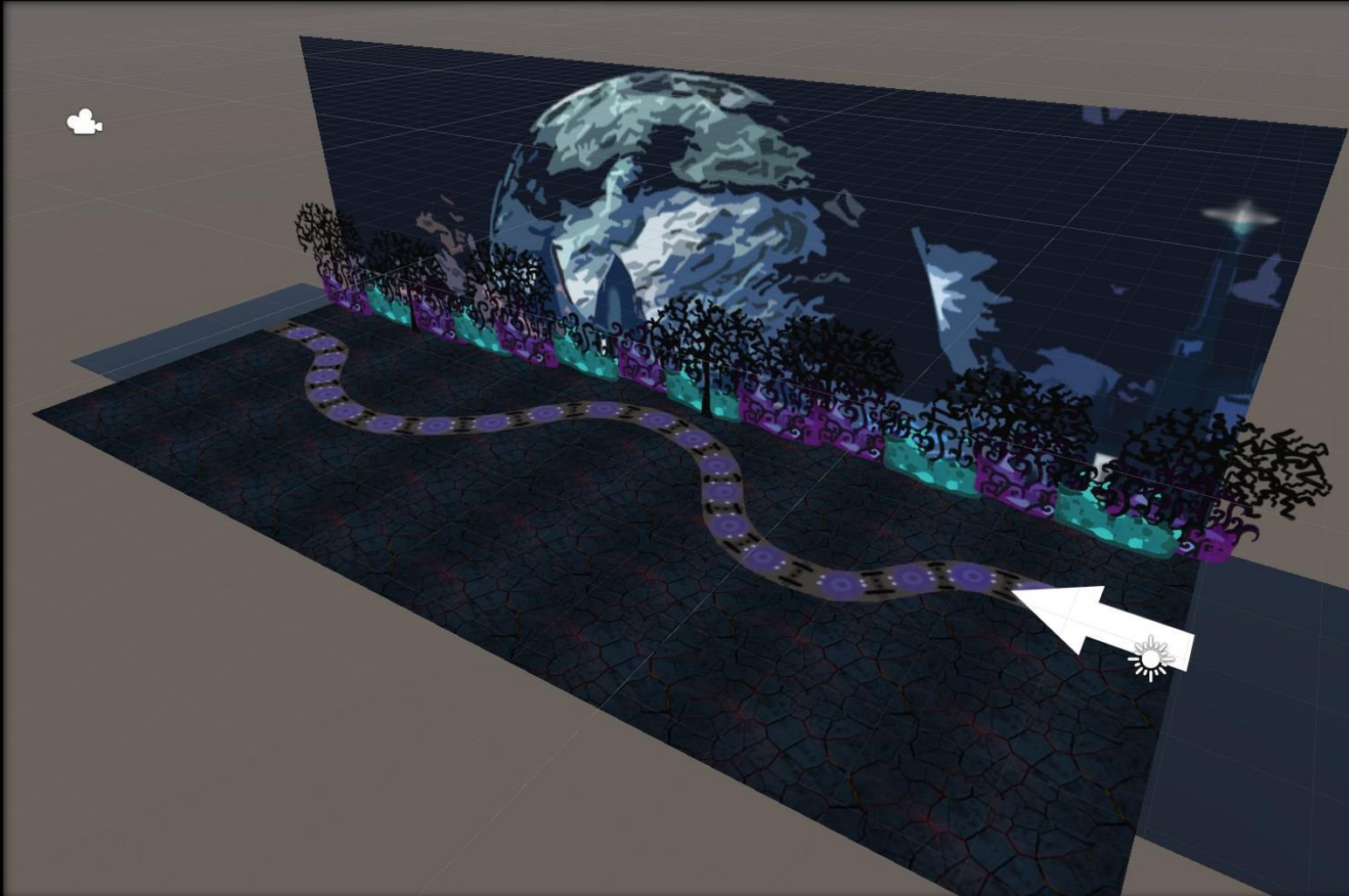
VICTORIAN

Level Art



SCI-FI

Game Implementation



- Sprites used for almost the entirety of the game.
- 3D model hidden below the path sprite for collision detection.
- Each level has same setup but uses different assets.
- Repositioned nodes and new path.
- Audio components for each prefab played by using the appropriate method.

UI Design



*A game by:
Lyall Campbell, Laura Milkevičiūtė, Connor Grundy, Declan Wylie
and Dale Mehmet. Additional music by Roman Polevecko.*

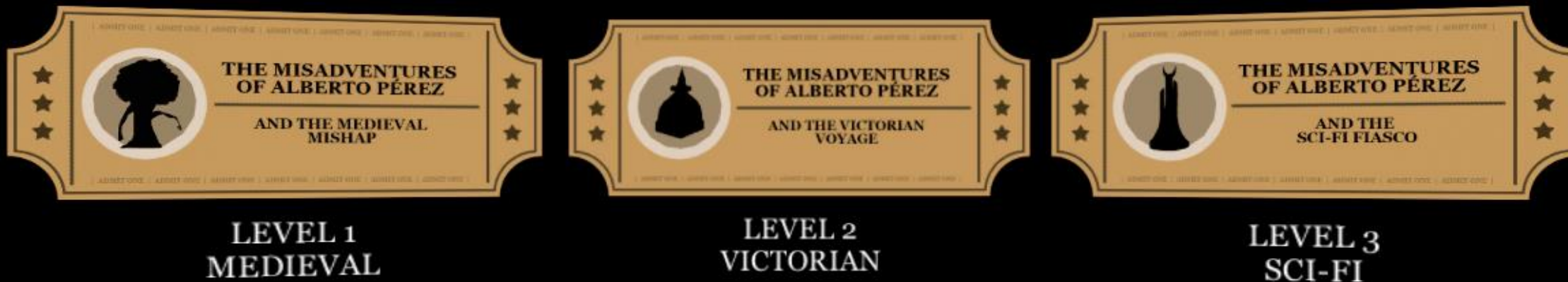
START GAME

OPTIONS

HOW TO PLAY

Version 1.0

UI Design



RETURN TO MENU

UI Design

+ 100%

\$ 750



PRESS PLAY TO START

WAVE: 1/30
SCORE: 0



LEVEL 3 - SCI-FI
SELECT A TOWER OR TRAP FOR INFORMATION



500



250



1000



1500



750



750



1000



60

80



40



100

Playtesting

- The methods used for playtesting were think aloud and players filling out a survey.
- The survey included 10 questions covering gameplay and art.
- The feedback received was mostly positive but uncovered a few bugs that were fixed promptly.
- A major issue uncovered was no indication of where enemies spawned from. This was fixed with an arrow informing players of the enemy spawn direction.



"Time Towers" playtest

You are about to start playing our game for as long as you choose until we otherwise instruct you to stop. If you lose, you start playing again. We encourage you to give your thoughts during the playtest, however, we cannot help you with the game. The purpose of the playtest is not winning the game.

The game will be provided via a web link. After you finish, please answer the questions below honestly and give your feedback where possible.

1. How many minutes did you feel you were playing for?

5 10 15 20

2. What was your strategy?

.....
.....
.....
.....

3. To what extent did you feel like you were in control of the outcome of the game?

1 2 3 4 5 6 (1 is not being in control at all, 6 is being a god)

Comments

.....
.....

4. Name the game you have played that is most similar to the game you playtested

.....
.....

5. What was most exciting about this game?

.....
.....
.....

Q&A