



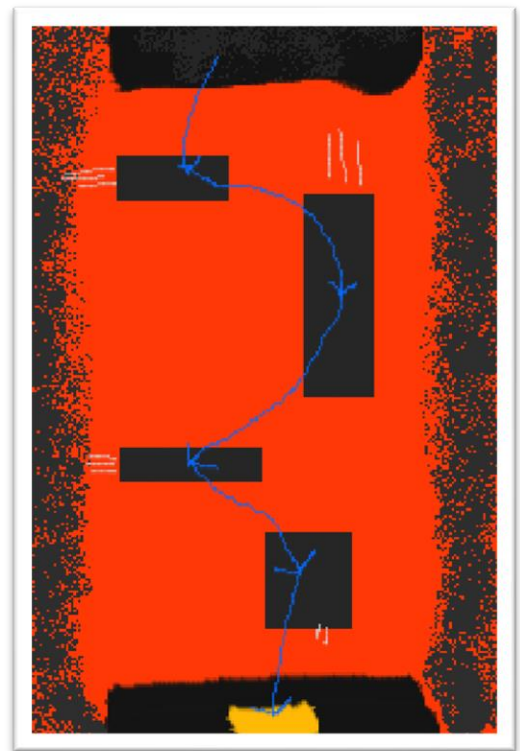
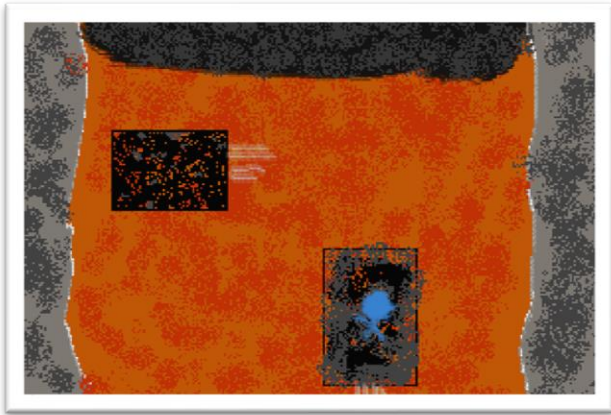
Game Design Document

Eric Spiteri 5.1 Interactive

Contents

Preliminary Sketches.....	2
Game Mechanics Conception.....	3
Formal Elements Conception/Inspiration	4
Development.....	5
Positive Loops	8
Negative Loops	9
Sound	10
Interface.....	11
Making it repetitive	12
Playtesting Plan	13
Playtesting Session1	14
Playtesting Session 2.....	15
Playtesting Session 3.....	16

Preliminary Sketches



Game Mechanics Conception

I wanted to make a simple platformer with several game mechanics present when the player jumps.

My plan was to give every single jump a unique challenge to keep the player engaged so that no two playthrough's feel the same.

I planned to make the player jump and land between platforms (primary mechanic). At

the same time, the player's jumpforce increases slightly (secondary mechanic) so that he is able to jump further as platforms are spaced higher after each step.

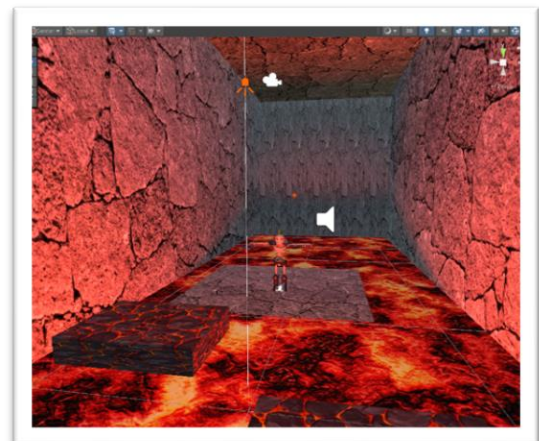


Formal Elements Conception/Inspiration

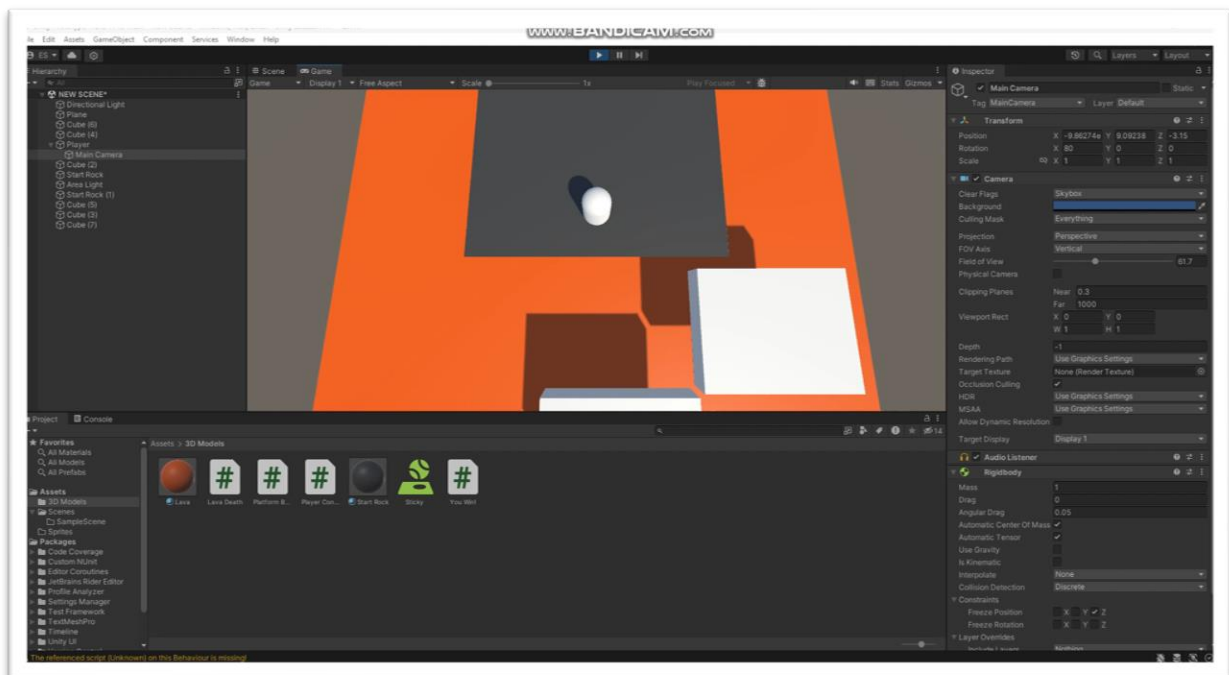
I had initially planned to give it a 2D pixelated feel, however I felt that the market was too oversaturated with these kinds of formal elements. Instead, I opted for a 3D low – poly style reminiscent of late 1990's shooters like Quake or ElderScrolls2.



I felt as though this would aide in the eerie feeling I intended to give off with this game. It also coincides with a short and simple self explanatory story of a skeleton escaping a deep abandoned mine.

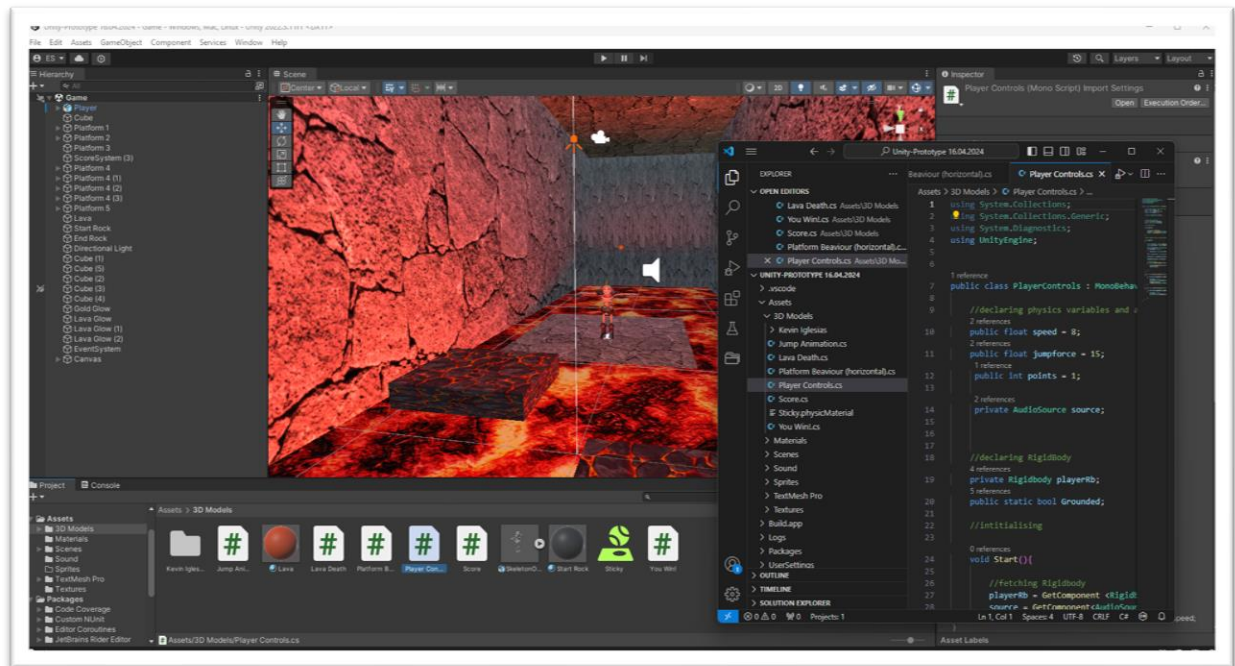


Development



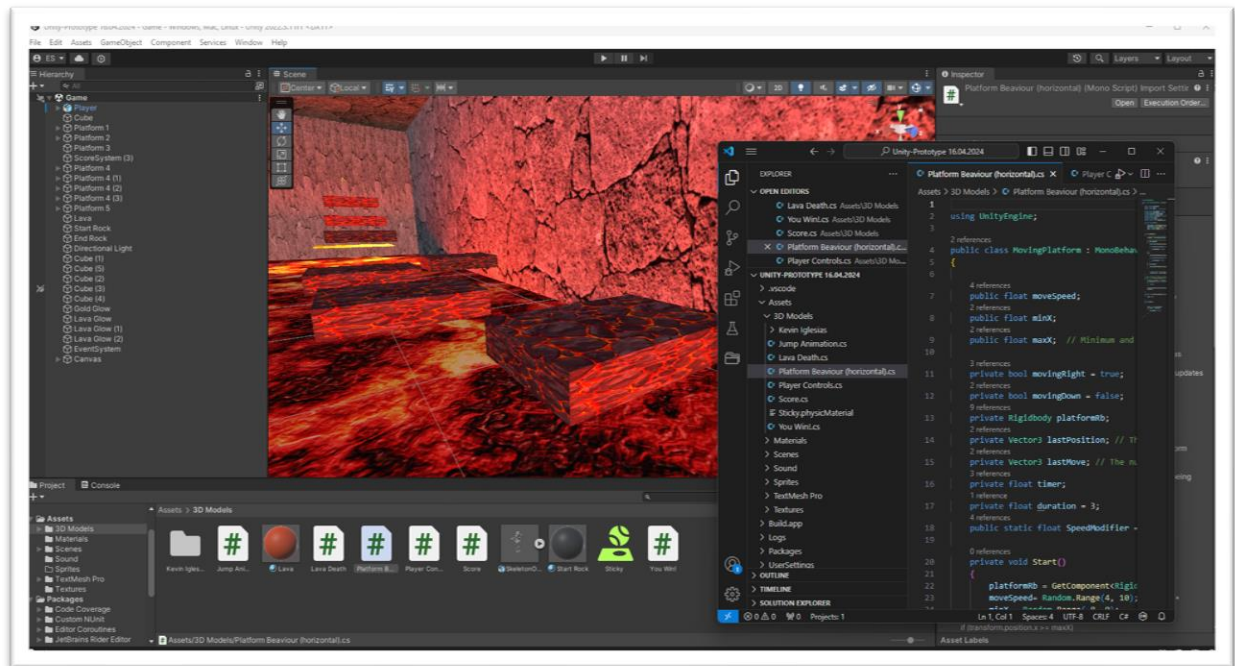
I had began creating the game world with simple geometric shapes to code the core mechanics such as jumping and movement as well as the collision of the platforms. I used CSharp Unity to code everything.

I had problems with the Platforms “min” and “max” positions being too far apart or too close however I had later resolved this with the implementation of a specified range of random values.



I had later managed to implement the secondary mechanic of the player increasing its jumpforce every time the player lands on a platform. This allowed me to make the platforms wider apart (vertically) for a feel of progression which I will later get into when discussing loops.

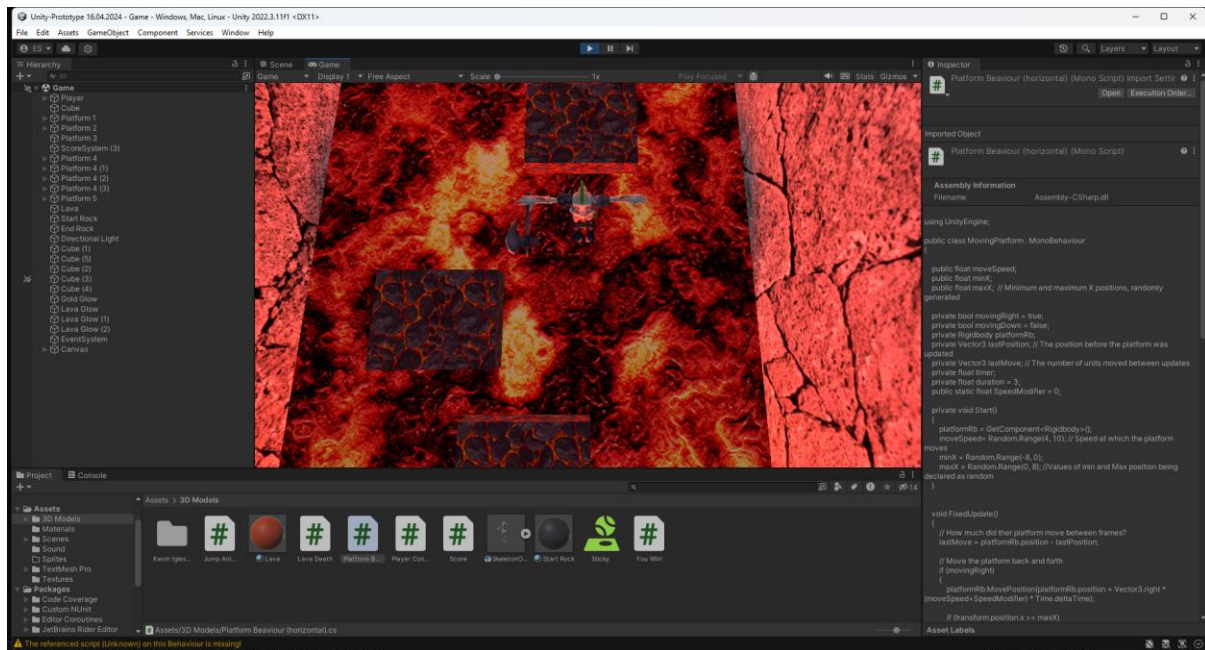
Moreover, I implemented the death function upon touching the lava to create the obvious challenge in the game.



I had later managed to make the platforms crumble after 5 (later 3 seconds for greater difficulty) to create a greater challenge and sense of urgency to the player.

Moreover I modified the speed of everything when the player jumps which I will later discuss in loops.

Positive Loops



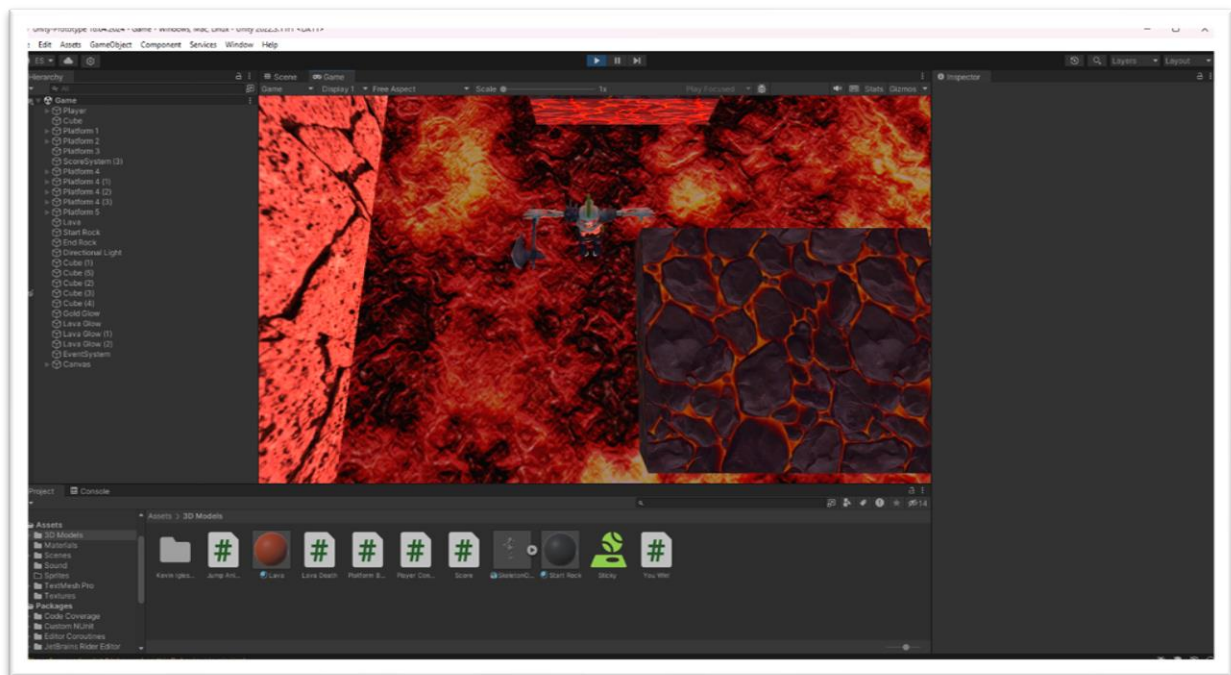
Early on in development, I realised that the game was quite stale and unoriginal by just being able to jump from one platform to the other.

As a result, I created some feedback loops to help encourage some behaviours.

Every time the player lands on a platform, the jumpforce increase slightly, encouraging the player to make progress. However this is counteracted by another negative loop which the player has to account for.

Negative Loops

While the jumpforce increases every time the player lands on a platform, every time he jumps his movement speed is also reduced, meaning that the increase in jumpforce is only vertical and not horizontal, this makes the player think carefully and plan his jump accordingly.

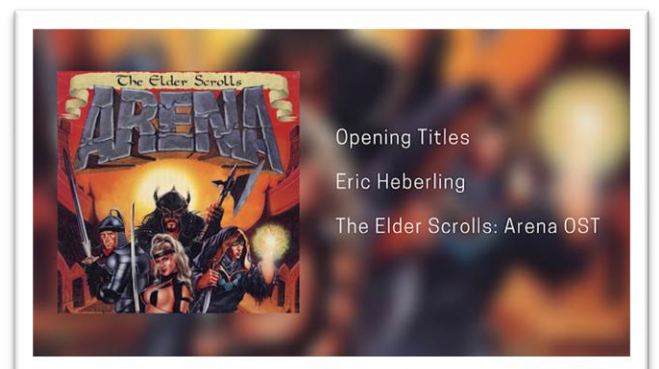


Sound

I used retro 1990's dungeon crawler music (credits to Bethesda Softworks) to add to the eerie and retro formal elements previously mentioned.

I also added a lava sound effect that plays continuously to coincide with the main threat of the game, the lava beneath.

Moreover, I included a “grunt” sound effect from quake when the player jumps as it gives the player feedback on what they're doing as well as it being a throwback to 1990's style videogames.

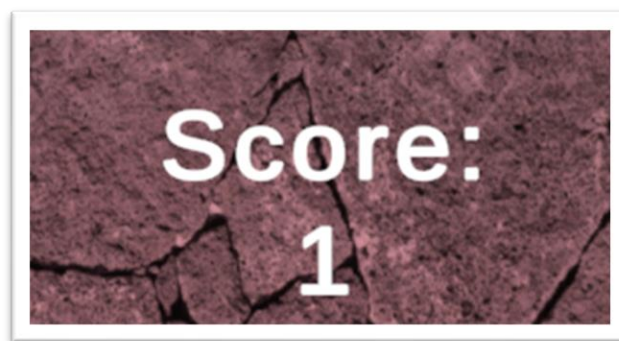


Interface

A score counter at the top of the screen was added that increments everytime the player lands on a platform.

The secondary objective of this game is to get as much of a high score as possible which is why this was crucial to add.

A counter which tells you how much time is left until the platform collapses was also planned however this was scrapped due to coding difficulties.



Making it repetitive

```
Debug.Log("You Win!");  
int currentSceneIndex = SceneManager.GetActiveScene().buildIndex;  
SceneManager.LoadScene(currentSceneIndex);
```

```
platformRb = GetComponent<Rigidbody>();  
moveSpeed= Random.Range(4, 10); // Speed at which the platform moves  
minX = Random.Range(-8, 0);  
maxX = Random.Range(0, 8); //Values of min and Max position being declared as random
```

Towards the end of development, code was included so as to make the game virtually infinite.

In the top picture, one can see that every time the player lands on lava, or reaches the goal, the game resets (along with the counter).

In the bottom picture, one can see that the moving speed of the platform, and their minimum and maximum X positions are generated completely randomly each time the game is launched and reset. This makes for a unique experience each time.

Playtesting Plan

I plan to conduct three playtesting sessions each with different people from various backgrounds. The types of questions will be different depending on the question and will be executed through a live interview on teams. Their gameplay and live reaction will also be recorded (with consent). This data will all be qualitative.

I plan to playtest with three different friends of mine. One of them has a background (and degree) in game design so this allows me to ask more technical questions regarding feedback loops and core mechanics.

The other friend is a hypercasual gamer and has a background in art which allows me to gauge the more dramatic elements of the game.

The third friend also has some experience in playing games but doesn't have a background in either art or game design so I will ask simple questions such as "is it fun" "is it frustrating" etc.

Playtesting Session1

Name: Anonymous

Background: Game Design

Intentions of Playtest: gather feedback on game mechanics/bugs

Data collection: Live play session and interview

Session Outline: First allow him to play the game (while recording) to see how he learns the mechanics, then allow him to try and find bugs and comment on them. After the play session stops ask him about the core mechanics and feedback loops.

Deliverables: consent paper

Results: an understanding on how the player learned to play the game, core mechanic outlines and potential bug fixes.

Evidence of session: Live and trimmed recording.

Playtesting Session 2

Name: Anonymous

Background: Art

Intentions of Playtest: gather feedback on Dramatic Elements

Data collection: Live play session and interview

Session Outline: First show her the game, then allow her to play for a bit (not expecting her to get the hang of it). Ask her opinion on the artstyle

Deliverables: consent paper

Results: an opinion of the overall aesthetics of the game from an art student.

Evidence of session: Live and trimmed recording.

Playtesting Session 3

Name: Anonymous

Background: Casual Gamer

Intentions of Playtest: gather feedback on the overall feel of the game

Data collection: Live play session and interview

Session Outline: Allow her to play the game, see whether she enjoys it and asks what frustrates her.

Deliverables: consent paper

Results: an opinion of the overall feel of the game

Evidence of session: Live and trimmed recording.