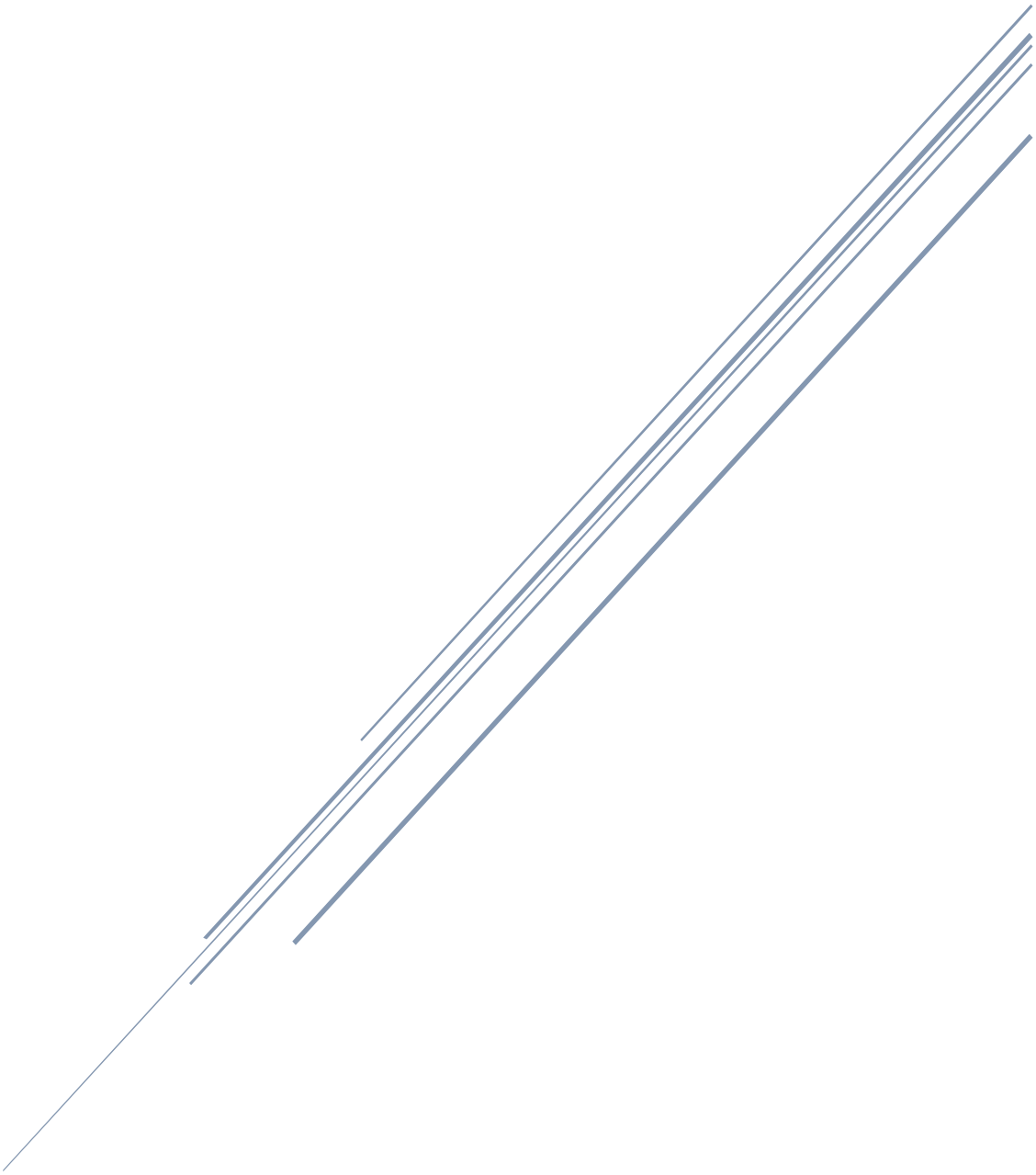
DES308 – REPORT

1904017



Abertay University

Analytics and Data-Driven Game Design

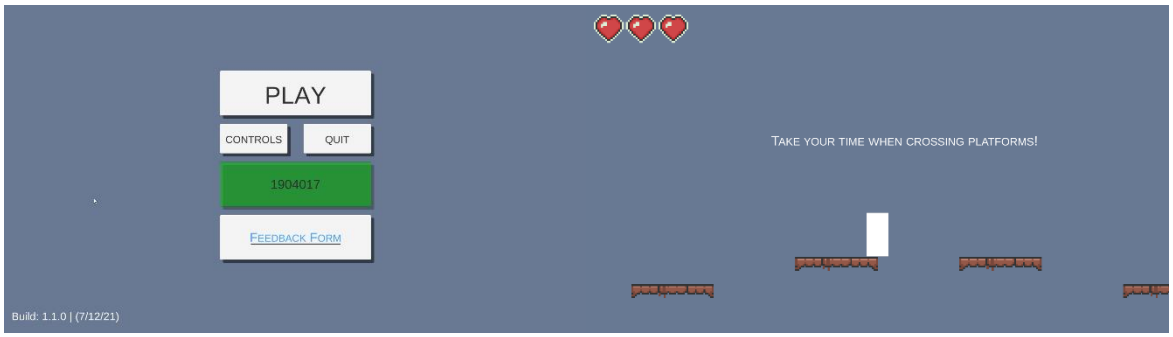
Table of Contents

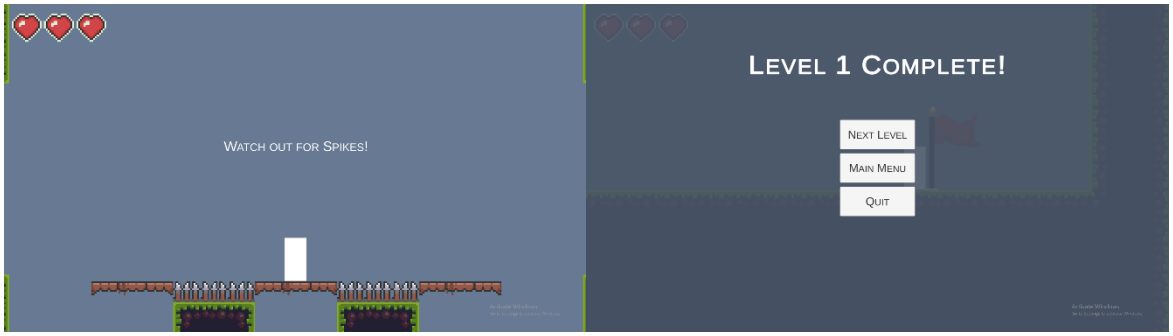
Prototype

For my prototype, I have chosen to do a 2D Platformer, created in unity 2020.3.0f1. It will consist of 3 levels plus a tutorial level. A main menu for settings, controls, audio etc. The tutorial level will be used as a basic introduction to movement, obstacles, collectibles. Each level will then introduce an obstacle to a deeper extent. The prototype is mainly inspired by the classic Super Mario, a simple platformer.

For my prototype, I needed movement, I followed this tutorial for my 2D movement. As for the art assets I used this pixel art set. I replaced the default heatmap icon with this.

(Below: Screenshots from multiple iterations bunched together)





Market Analysis

For my prototype, I also conducted market research for games of the same genre/style to see where my prototype would be situated within the current market. I went with the classic Super Mario Bro's to compare it to a more modern platformer such as celeste. This helps me to gain an understanding of current day game values and revenues.

Game Overview

2D-Platformer, created in Unity 2020.3.0f1, playable on PC

Inspired Games

- Super Mario Bro's
- Super Meat Boy
- Celeste
- Dadish

Market Insight

Game Name	Super Mario Bro's	Super Meat Boy	Celeste	Dadish
Downloads	58 Million	1-3 Million + (PC)	2-3 Million (PC)	20 Thousand (PC)
Developer	Nintendo	Team Meat	Extremely OK Games, Ltd.	Thomas K. Young
Publisher	Nintendo	Team Meat	Matt Makes Games Inc.	Thomas K. Young
Price	N/A	£10.99	£14.99	£2.09
Release	September 13, 1985 (JP)	30 November, 2010	25 January, 2018	22 February, 2021

Data Provided by: SteamDB

In the current market across all different device platforms and price points there is a multitude of platformers, many have stood the test of time and are still known to this day. Looking at the price point for my prototype, I would start on the low end of price points and potentially free depending on final release platform availability. Microtransactions for future features. If the prototype were to head in the direction of speed-running, there is a community for such titles.

Module Suitability

Title	Simple Platformer
Genre	Platformer
Platform	PC
Control	Keyboard
Core Mechanics	Horizontal Movement
	Jumping
	Obstacles
Skill Level	Medium
Game Length	3 levels
Session Length	10 Minutes

Module Suitability
Short Game Length
Short Session Length
Obstacles: Speed/ Damage/ Placement
Fails due to Obstacle type
Fails due to falling off level
Per-player fails due to obstacle type
Tutorial: Start/ Complete/ Skip
Per-Level timer

Hooks

For my quantitative data within my prototype, I implemented the below hooks using Unity Analytics / Data Recorder Script and Discord Webhook.

My reasoning for adding these hooks is as follows:

- **Level Progression:** Allows me to visualise how each player is progressing and if any trends appear across players such as quitting after level 2.
- **Level Fails/Completions:** How many players are failing levels and comparing it to those who complete the levels.
- **Player Death Positions:** To see where each and all players collectively "die" and see if any common places of death areas appear.
- **Time to Complete Level:** This data can be used to see if any players are taking unusually long to progress through a level and if so, I can deduct what may be causing the problem through other hooks.
- **Damage Taken:** I can use the hooks to see when/where and by what the player is taking damage from whether that be a saw/spike/suspended trap.
- **Health Restored:** Using this hook I can see if the hearts within the level are sufficient and not too over-powering.
- **Restarts/Quits:** This hook allows me to see the total amount of restarts per level as well as where players are quitting the game from.
- **Menu Viewed:** This hook allows me to see if players are making use of the Audio and Controls menu.

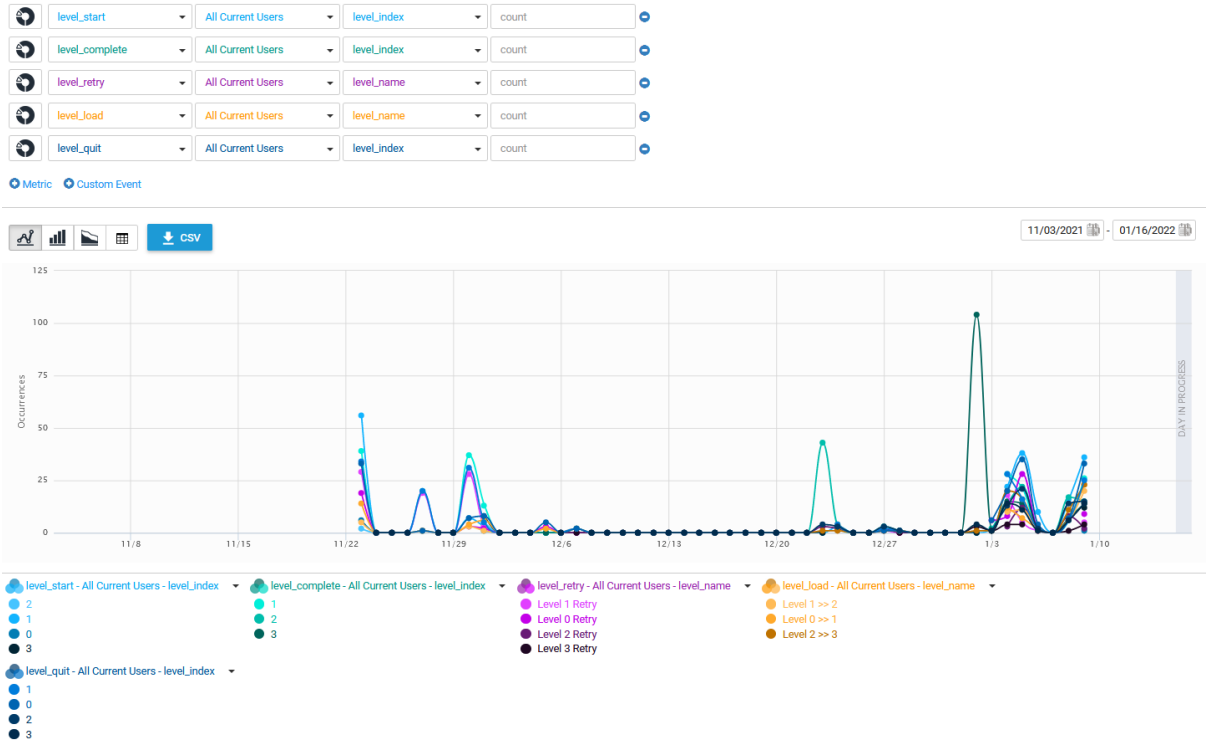
For my prototype the above-mentioned hooks are what I thought would be most important to keep track off as they are the main components within the game. These hooks can be used for both new and returning players as they can be compared against each other.

Due to these hooks, my game improved tremendously over the course of development. I took a different approach to my testing, however. I did "focus-testing" on each level, and each iteration represents this, by this I mean that in version 1.0.0 it contained the tutorial and level 1, with each major version (1.2.0 / 1.3.0) I then opened the next level for testing. I believe this was the best way to go about testing my prototype as it allowed me to work on each level independently and get the most testing and improvements made within each new iteration.

Data Collection Integration

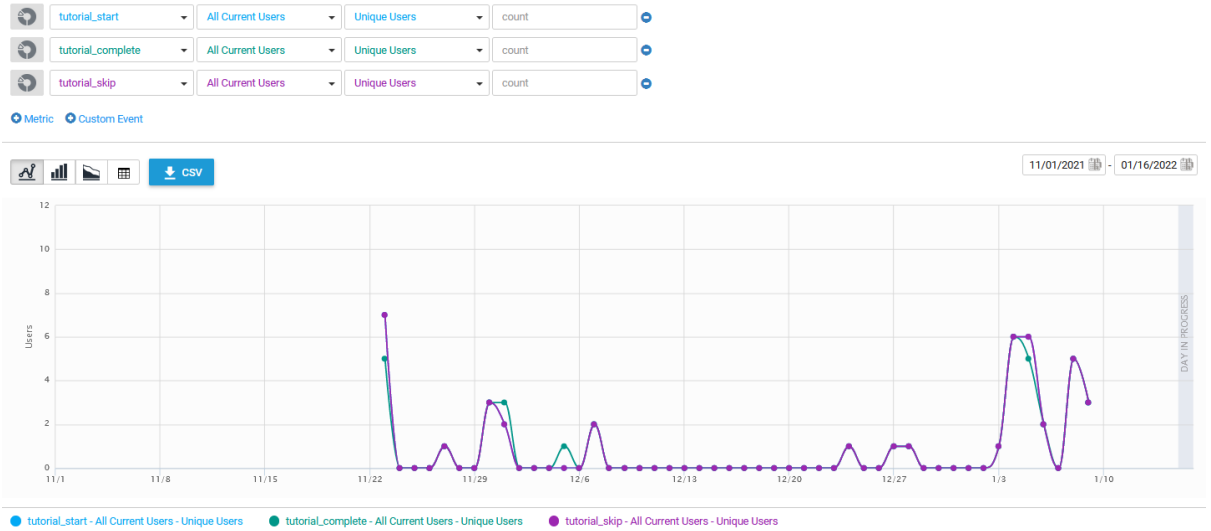
A lot of unity analytics hooks were used and tracked through each new iteration, it took some time to get to grips with unity analytics, as it was a whole new thing to work with, their dashboard is however, very useful and easy to navigate. I did set-up a funnel, but it only sent data for the first level, so I made a work-around funnel seen in the below Data Sheet called “level_load” which tracked level progression.

[Level 0 (Tutorial) to Level 3 Data]



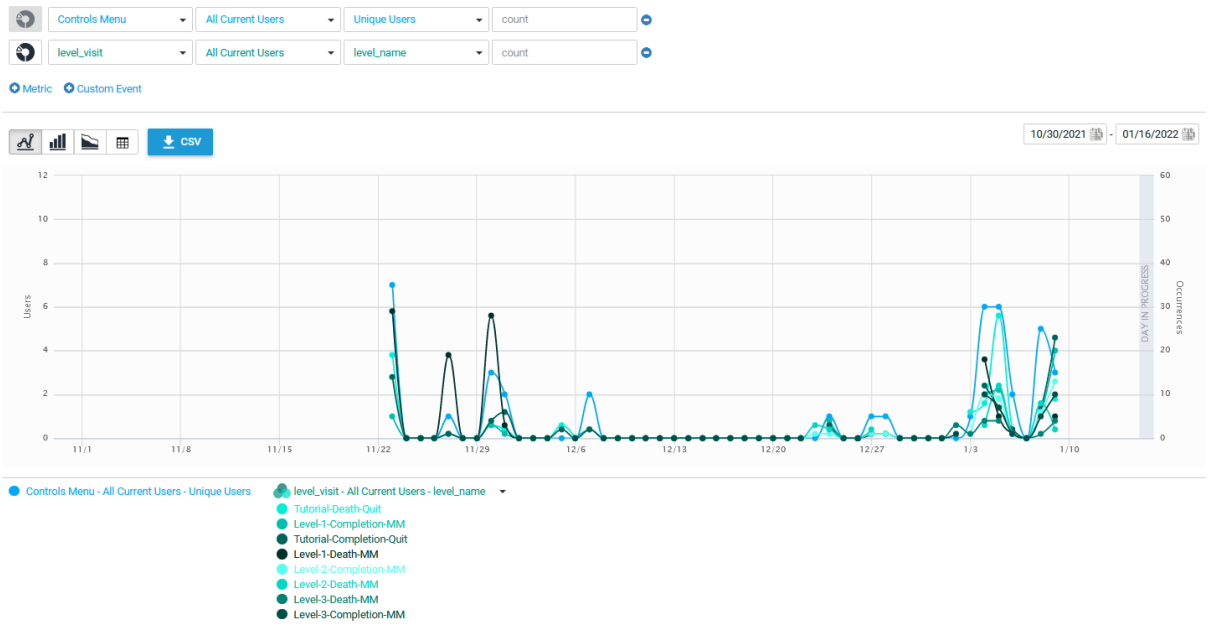
[Tutorial Data]

I separated my Tutorial Data into its own sheet, for clarity. This allowed me to see how many people were starting, completing, or completely skipping the tutorial and I could then relate this to the Discord Webhook, which also kept track of these things to see if they were a returning player or not.



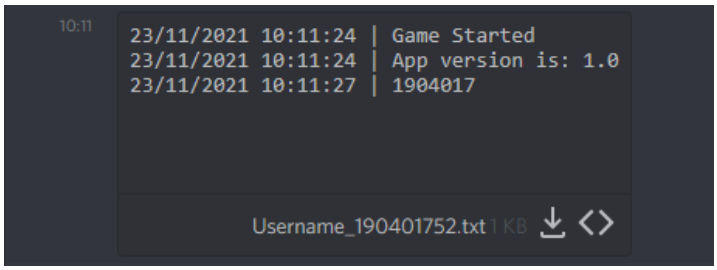
[Menu Visits Data]

Another small data collection I did was seeing if players were viewing the controls menu or not as well as when they quit from the game and if it was a quit-on level fail/completion and if depending on what it was, I could once again relate it to the Discord Webhook, see what user it was and then look at their feedback form to see their reasoning for quitting.

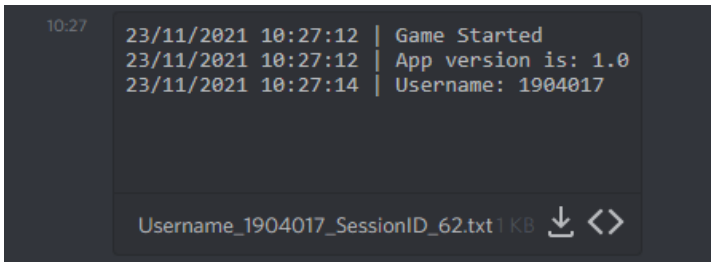


Another way I implemented data collection was using Discord Webhook which a class colleague provided, this allowed me to instantly receive player data. Compared to Unity's multi-hour wait time. I was able to implement the same hooks as unity analytics. However, it did not provide the charts and graph view like unity, so unity was still used to visually represent all the data, where-as Discord was used for each player run of the game.

Early testing with Discord Webhook, allowed for the username to be stored as the file name as well as their session ID in the below image the "52" at the end of the string corresponds to the session ID. It also time stamped each new line that was stored.



Improved Version for better readability:

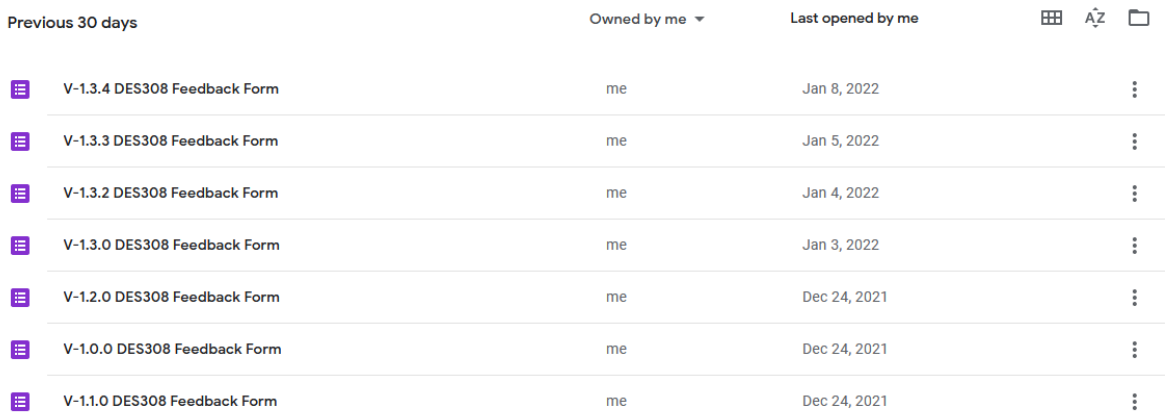


A full data-webhook example:

```
09/01/2022 16:36:41 | Game Started 94
09/01/2022 16:36:41 | App version is: 1.3.4
09/01/2022 16:36:42 | Username: 1904017
09/01/2022 16:36:45 | Player Viewed: Controls Menu
09/01/2022 16:38:06 | Player Viewed: Tutorial Menu
09/01/2022 16:38:06 | Player Viewed: Tutorial Menu
09/01/2022 16:38:08 | Player Clicked: Tutorial - Yes
09/01/2022 16:38:08 | Player started Tutorial
09/01/2022 16:38:27 | Player restored 1HP in level: TutorialScene
09/01/2022 16:38:33 | Player took 1HP, from Suspended Trap, in level: TutorialScene
09/01/2022 16:38:33 | Player took 1HP, from Suspended Trap, in level: TutorialScene
09/01/2022 16:38:34 | Player restored 2HP in level: TutorialScene
09/01/2022 16:38:38 | Player Reached the finish at level: TutorialScene with a time of: 22.28846 with 3HP Left
09/01/2022 16:38:40 | Player started Level 1
09/01/2022 16:38:50 | Player restored 1HP in level: Level_1
09/01/2022 16:38:56 | Player Reached the finish at level: Level_1 with a time of: 16.59129 with 3HP Left
09/01/2022 16:38:57 | Player started Level 2
09/01/2022 16:39:06 | Player restored 2HP in level: Level_2
09/01/2022 16:39:09 | Player took 0.5HP, from SawBlade, in level: Level_2
09/01/2022 16:39:09 | Player took 0.5HP, from SawBlade, in level: Level_2
09/01/2022 16:39:11 | Player restored 1HP in level: Level_2
09/01/2022 16:39:15 | Player Reached the finish at level: Level_2 with a time of: 17.79221 with 3HP Left
09/01/2022 16:39:17 | Player started Level 3
09/01/2022 16:39:19 | Player took 0.5HP, from SawBlade, in level: Level_3
09/01/2022 16:39:19 | Player took 0.5HP, from SawBlade, in level: Level_3
09/01/2022 16:39:22 | Player restored 1HP in level: Level_3
09/01/2022 16:39:26 | Player restored 2HP in level: Level_3
09/01/2022 16:39:31 | Player took 0.2HP, from Spike Trap, in level: Level_3
09/01/2022 16:39:31 | Player took 0.2HP, from Spike Trap, in level: Level_3
09/01/2022 16:39:31 | Player took 0.2HP, from Spike Trap, in level: Level_3
09/01/2022 16:39:31 | Player took 0.5HP, from SawBlade, in level: Level_3
09/01/2022 16:39:31 | Player took 0.5HP, from SawBlade, in level: Level_3
09/01/2022 16:39:36 | Player took 1HP, from Suspended Trap, in level: Level_3
09/01/2022 16:39:37 | Player restored 2HP in level: Level_3
09/01/2022 16:39:39 | Player Reached the finish at level: Level_3 with a time of: 21.59055 with 2.4HP Left
09/01/2022 16:39:41 | Player completed level: Level_3 and returned to main menu
09/01/2022 16:39:41 | Username: 1904017
09/01/2022 16:39:43 | Player Viewed: Quit Options
09/01/2022 16:39:43 | Player Viewed: Quit Options
09/01/2022 16:39:43 | Player Clicked: Quit Options - Yes
09/01/2022 16:39:43 | Player Quit game from: MainMenu
```

Username_1904017_SessionID_27.txt 3 KB

Finally, I had my google-forms feedback form which I implemented into the games main-menu:



Version 1.0.0

Version 1.3.4

(You can make up a name for viewing the above file, just clear the data so you don't submit a fake report.)

Outcomes of Prototype Testing

I conducted several testing sessions, multiple in-person and multiple online where I sent my game to willing testers, I received both quantitative and qualitative data doing this, as mentioned with hooks but also, I had a feedback form prepared for each major iteration of the game. In total I had 10 unique testers, within these 10, I had consistent iteration 1-x testers as well as new people testing in later iterations, this allowed for more diverse feedback.

Through my testing I gathered a lot of insight in both in-person and online as with in-person I could watch them as they played and see how they felt when certain occurrences such as fails/wins occurred.

One thing I did not expect however, was for the testers to play "together", a couple of the testers played the game concurrently and challenged each other with completing levels the fastest, through their qualitative data I was able to conclude that adding a in-game timer would be beneficial for such players, known as "speed-runners" in the gaming industry. This provided them a much easier time comparing their times.

Lots of arguably smaller changes were made to the game, such as UI adjustments, some buttons being re-located/adjusted, font changes and improvement to readability, which all game together to make a better experience.

I was able to use Unity Analytics to view the data in more readable formats such as pie-charts (figure 1), this allowed me to compare the player data against each other and see the percentage of players completing/failing levels, and if the majority was succeeding, I made small adjustments to make certain aspects easier for the minority of players.

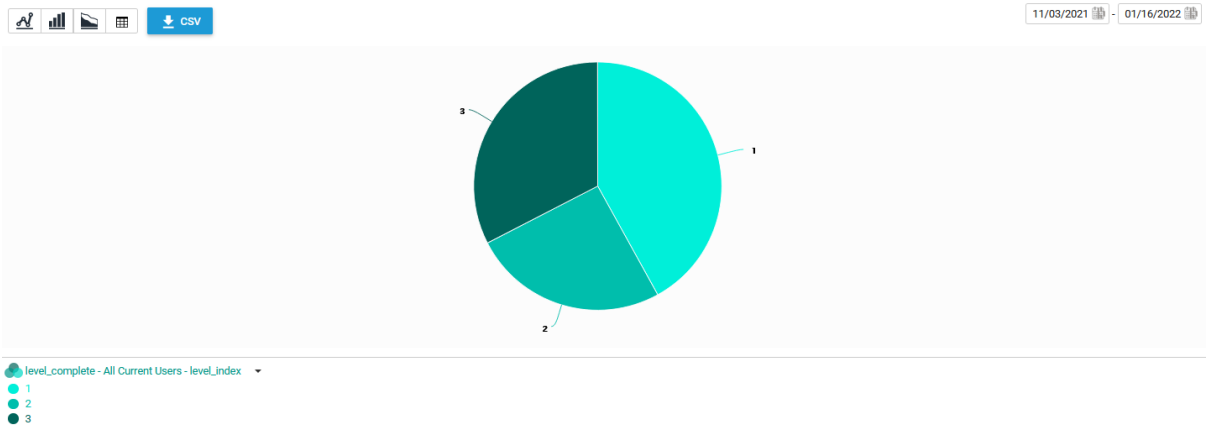


Figure 1 Total Level Complete

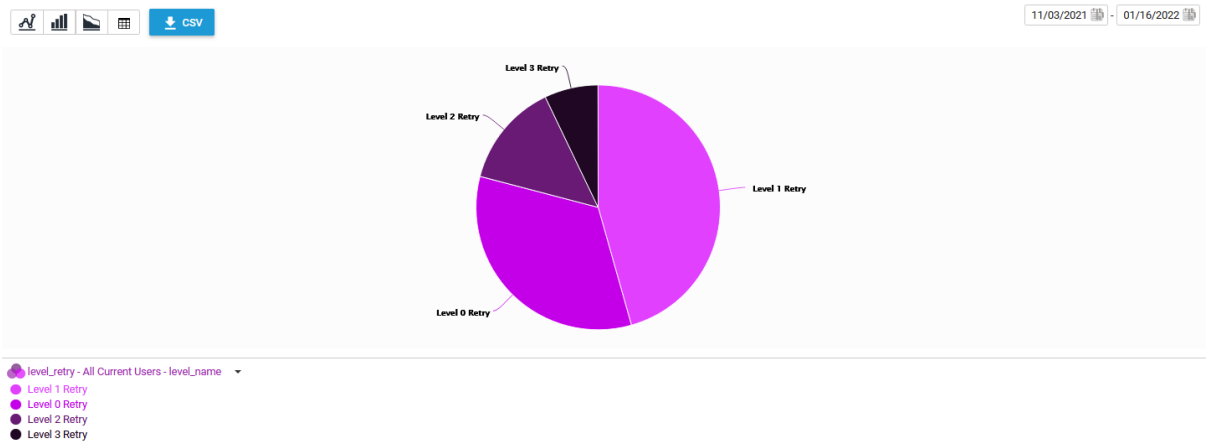


Figure 2 Total Level Retry

A flaw that I did however pick-up on whilst viewing them, is since I did my focus-testing, comparing these statistics against each other doesn't work, as sometimes more people would test certain builds, so the data wasn't comparable. I had to use the data as per-level statistics rather than overall statistics. As seen in figure 2, more people retried level 0 and 1, which comes down to those levels having more playtime as they were available for longer.

Data Collection Interpretation

The main goal was to analysis and interpret the data to improve my prototype. For my quantitative data collection, I used this mainly for death positions, as I could use the data presented to see what was causing the deaths and where, for examples was it an obstacle or a platform? I was able to deductively decide based upon the data if changes were needed in certain areas, or if it was player-error and they just needed more time to learn the controls/movement.

A major change that was discovered through both qualitative and quantitative data was within Level 1. I quickly began to see across the testers there was a common area where they were failing the level. I was able to look at the data and visualize this data using the Data Recorder script and decide that indeed it was too difficult to manoeuvre across this section and therefore made a change that would ultimately decide how platforms would be throughout the rest of the levels within the prototype.

In figure 3, there is a total of 62 deaths across 3 players, which was a huge amount to start out with.

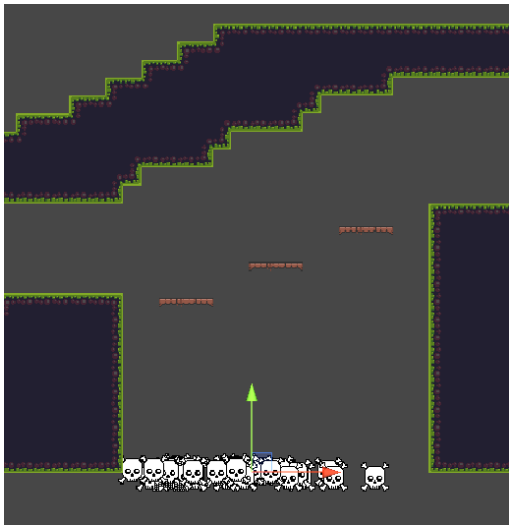


Figure 3 Level 1 Before Changes

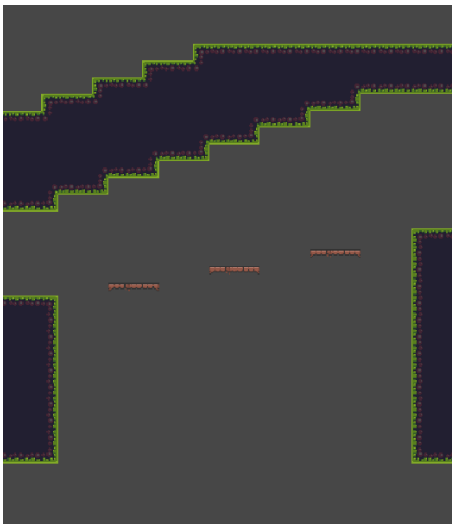


Figure 4 Level 1 After Changes

The change made was the jump height between each platform before it was 2-tiles and after it is reduced to 1-tile. This proved to be the correct decision as rate of player failure declined after this.

On-top of this, with each new iteration I provided a new feedback form to the players, with each new batch came new feedback which I used to improve the prototype. Things such as UI were brought up, and I improved the readability and positioning in some instances as seen in figures 5,6,7.

Level Complete Screen:

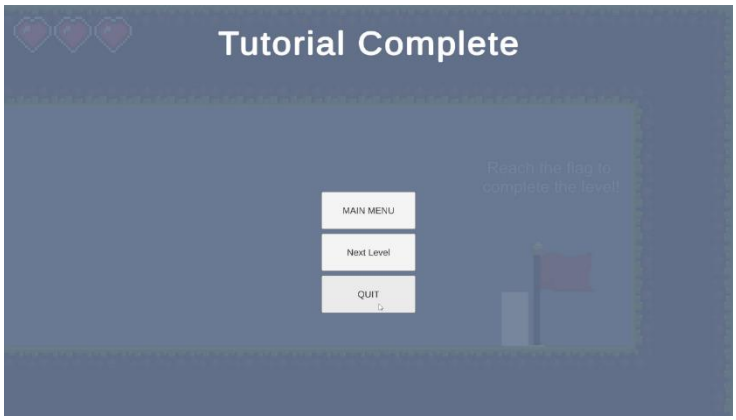


Figure 5 V-1.0.1 Completion Screen

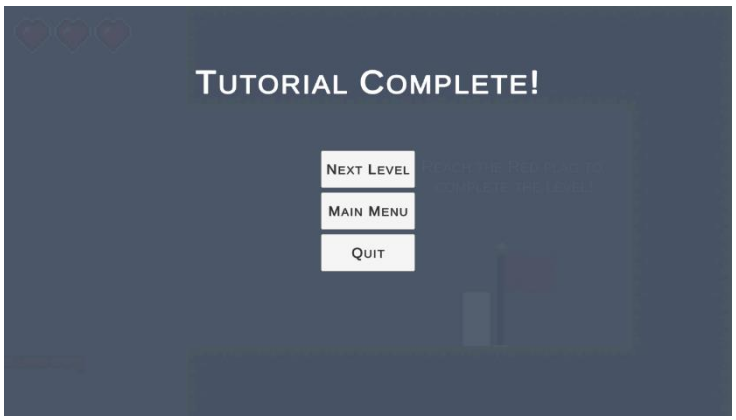


Figure 6 V-1.1.0 Completion Screen

Figure 7 V-1.3.4 Completion Screen

The above figures showcase one of these small examples, where over-time and through iteration the menu changed slightly but was directed by feedback to the result in figure 7.

Another major addition to the prototype was level checkpoints, this was added based upon quantitative data, upon analysing the data, I decided to experiment by adding checkpoints throughout the levels. Doing this I also had to add it to the tutorial level to inform the player what their use was.

Overall, they were received well, and players mentioned how they were a nice addition to help those who struggled more than others. Ultimately reducing the total level-reset punishment of falling off the world.

Figure 8 Checkpoint system introduction in tutorial.

Feedback Version 1.3.3:

- A restart button or a pause menu to do so, should be available.
- The knockback is far too big on all the threats.
- A timer would be nice for speedrunners

Timer for people who speedrun (maybe leaderboards?)

A feature I held off, was a pause menu. I did not add this until Version 1.3.4, in the feedback it was never brought up that one was missing until version 1.3.x was put into testing, thus I added it in.

A feature I did not plan however, was a level timer but as discovered when players started to compete against each other for fastest level completion time, I decided to add one in, which was well received among both those who requested it and those who did not.

Through quantitative data, I was assured that the “Saw” obstacle was the one causing the player the most damage, which made sense as these obstacles were the majority in each level. I did small adjustments through levels, only by changing the speed and or distance of which they travelled, as in some cases these obstacles were a bit too fast for most players. Overtime the damage being taken dropped off as returning players began to learn the level layouts and timings better, whilst new players still took a reasonable amount of damage from these obstacles but nothing too over the top.

Knock-back first mentioned V-1.0.0

> Saws could have more feedback when you are hit, e.g bouncing off them or being hit in a direction. Possibly flash the character when they take damage.

A user asking for Knock-back, V-1.3.2:

> The game is slightly too easy in my opinion, it feels like you can just run through without a huge challenge. Could possibly make the traps and other objects reset you to a checkpoint or have a hitbox of sorts that stops a player just walking through them.

Knock-Back Feedback, V-1.3.3:

> The added bounce to the saws added more challenge since you could no longer just damage boost through them

> Spikes have a shitload of knockback to the point where you can fly across parts of the level.
>
> The floor spikes can sometimes not damage directly from above.
>
> I used the spike knockback to escape the map and fall off to where there was no death barrier.

Knock-back Feedback, V-1.3.4:

> Nicely balanced now. possibly a tad easy but I have played it a fair amount. The new knockback feature definitely made me think more about how I play.

Through feedback I was able to gradually get this feature to a balanced state.

Prototype Changes

An overview of changes can be viewed in this video

V-1.0.0

- Base Prototype for testing
- Level 0 (tutorial) and 1 available to test

V-1.0.1 (not tested)

- (Bug-Fix) – Tile map collider issues fixed

V-1.0.2

- This version was used for testing session 1
- Above versions fall into this one

V-1.1.0

- (New) – Added colour to input field, for visual feedback
- (New) – Build version added to main menu
- (Adjusted) – Controls Menu UI improved
- (New) – Checkpoint System
- (Adjusted) – UI Font, improved readability, Adjusted button positions in end screen
- (New) – New obstacle introduced to prototype “Suspended-Trap”
- (New) – Tutorial section added for platform manoeuvrability
- (Adjusted) – Platforming section in level 1 adjusted

V-1.2.0

- (New) – Level 2 Added for testing
- (New) – Gold Hearts added to level 2

V-1.3.1

- (Adjusted) – Re-programmed checkpoint system

V-1.3.2

- (New) – Level 3 added for testing
- (Bug-Fix) – Fixed checkpoint system not resting on level progression
- (New) – New obstacle introduced to prototype “Spikes”
- (Adjusted) – UI updates, Controls added to in-tutorial UI, Gold Hearts added to tutorial, Level number added to end screen
- (New) – “W” Added as a jump key
- (New) – Jump-pad added to level 3

V-1.3.3

- (New) – Settings Menu
- (New) – Audio
- (New) – Added knock-back effect to Saw/Spike
- (New) – Added new UI to each level informing the user which level they are on.

Audio Feedback:

What are your thoughts on the Audio currently in the game?

6 responses

Music is not distracting, and the sound effects are representative of the action being taken, hence are effective and useful.

Nice sound effects, could be a little more precise with the audio slider

Adds to the gameplay, feels more accomplishing a and enjoyable finishing a level/picking up hearts etc.

Useful.

Little loud to start

Good they fit the theme and the actions they were assigned to

V-1.3.4

- (New) – Per-level Timer
- (Adjusted) – Reduced default volume of Music and SFX (1.3.3 Feedback)
- (Adjusted) – Saw/Spike knock-back effect:
  - The added bounce to the saws added more challenge since you could no longer just damage boost through them
- (New) – Pause Menu Added
- (New) – Directional signs added for alternate routes
- (Adjusted) – replaced vertical platforms in level 2 in favour of “jump-pad”:
  - Some of the parts where you just jump up brown platforms like in the middle of level 2 and the end of level 3 can feel a lil wack and take a few attempts

V-1.4.0

- (Adjusted) Reduced Saw/Spike Knock-back effect // Adjusted due to feedback from previous version mentioning it being too strong.