Digital Media Final Project – CI7800

A Soul's Journey

Report

Portfolio Link:
https://www.anujpatwari.com

Game Link:
https://www.anujpatwari.com/asoulsjourney

Anuj Patwari

K1840581

MA in Game Development (Design)

TABLE OF CONTENTS

INTRODUCTION

HIGH CONCEPT

A Soul's Journey is a Puzzle-Platformer where the player has to place different kinds of platforms from his/her inventory at pre-defined locations to help the soul reach its goal.

ID CARD

Name – A Soul's Journey

Genre – 2D Puzzle Platformer

Platform – PC

Target Audience – Puzzle Game enthusiasts

Engine Used – Unity 2019.2.0f1

STORY

A child is crossing the road and is suddenly met with an accident. While he is lying on the ground, he enters this near death situation where his soul leaves his body. The boy is quickly taken to the hospital and this is where the first sequence of the game happens.

When the player completes 10 levels, the next cut-scene takes place where the doctor gives the first defibrillation shock. Since the soul is somehow still connected to the child, it regains one of its colours and the next sequence (Level 11 to 19) of puzzles start.

After this sequence, the second cut-scene takes place, where the doctor gives the second defibrillation shock and this time the soul regains another colour and the final sequence (Level 20 to 28) of puzzles start.

After all 28 levels have completed, the final cut-scene plays which shows the doctor giving the third defibrillation shock which this time lead to the lost soul finally being able to return to the child's body and the next morning, the child wakes up and is healthy once again.

GAMEPLAY IN A NUTSHELL

A Soul's Journey is a puzzle-platformer game where the player is given an inventory of platforms at the start of the level which he/she is allowed to place at certain pre-defined locations of the level. After placing these platforms, the player has to play through the level and try and get the soul to its goal (portal at the end of each level). If the player fails to do so, he/she can change the placements of the platforms according to his/her new strategy and try again.

TECHNOLOGIES & RESOURCES

GAME ENGINE

I have decided to go forward with using **Unity Engine 2019.2.0f1** for this project. I have decided to do so because of the following reasons:

1. **Familiarity –** Since I have used Unity Engine in the past, for Level Design and UI Programming, I am quite familiar with it in comparison to other Game Engines.
2. **2D Game –** Unity is a very good Game Engine to build any 2D game. Looking at examples like Ori and the Blind Forest, Cuphead, Inside and a lot more, it is safe to say that it has a very good track record in this aspect.
3. **Programming Support –** I have a lot of friends who are quite fluent in programming in Unity and I could ask them for help at any time if I got stuck anywhere. I also follow a lot of YouTubers who upload regular tutorials on Unity and I love learning new techniques from them. I used Brackeys' movement system and animation system for this project.
   Apart from the support from friends and YouTubers, the Unity Manual is very detailed and helpful and I have used it a lot during this project.

ART

I made all of the art in the game myself and have followed a simplistic art style, something very similar to what Thomas was Alone followed. I also got a lot of inputs and ideas from my friends which gave me a good insight into making the game more lively and engaging for the players.

VERSION CONTROL

To keep my game always backed up for any unforeseen problems was always on my list. So I began the project by making a **GitHub** repository for it before doing anything else. I feel this practice is a good one because version control is a huge part of many studios and this habit would be seen as a good habit by any potential employer in the future.

AIMS & OBJECTVIES

When showcasing an earlier version of this game at Develop: Brighton, I received a lot of positive feedback for the game and was asked by many people over there to further develop the game and expand it. These comments got me very motivated into going forward with the further development of this project.

As the project is the Digital Media Final Project, the main aim for me from this module was to have a fully functional prototype which is polished and gives a good experience to the player. I also wanted to make sure I implemented my Level Design philosophy as mentioned in the Game Design Document and a secondary objective was also set by me personally. This being that I want this game to be a completely original one and that nothing in this game is taken from anywhere, which included the art, music, design and basically anything at all.

To achieve the aims set above, I laid out certain objectives to be achieved from this project:

1. I needed to have at least 25 levels to showcase my Level Design process and philosophy through this game. This would also allow me showcase my understanding towards progression in terms of mechanic understanding and difficulty of the game.

2. Not knowing how to play any instruments and not having a good background in doing art, to achieve my secondary objective, I set out an aim of practicing some basic art and learn how to play the guitar for at least one hour every day. This would not only be of use for me for this project, but would eventually help me in the long run as well.

PROCESS

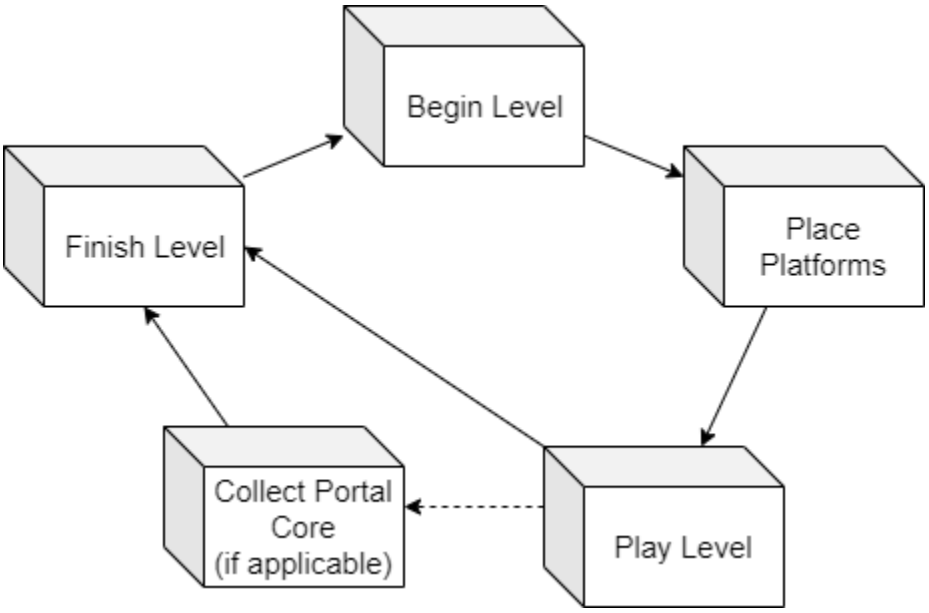
PHASE 1 – BRAINSTORMING

At the very beginning of the project, I wasn't entirely sure of the concept. So after about a week into submitting my proposal, I sat down and started brainstorming about how the mechanics I mentioned in my proposal would work. It took me a few days to understand the scope of my project, the kinds of platforms I want and the kinds of levels I want to build for my players.

I also had to think of a story that would somehow attach the player to the character. I came up with a number of different stories, but none of them felt like they would make the player connect with the character but eventually I came up with this story and after pitching it to a few people, I decided to go ahead with it.

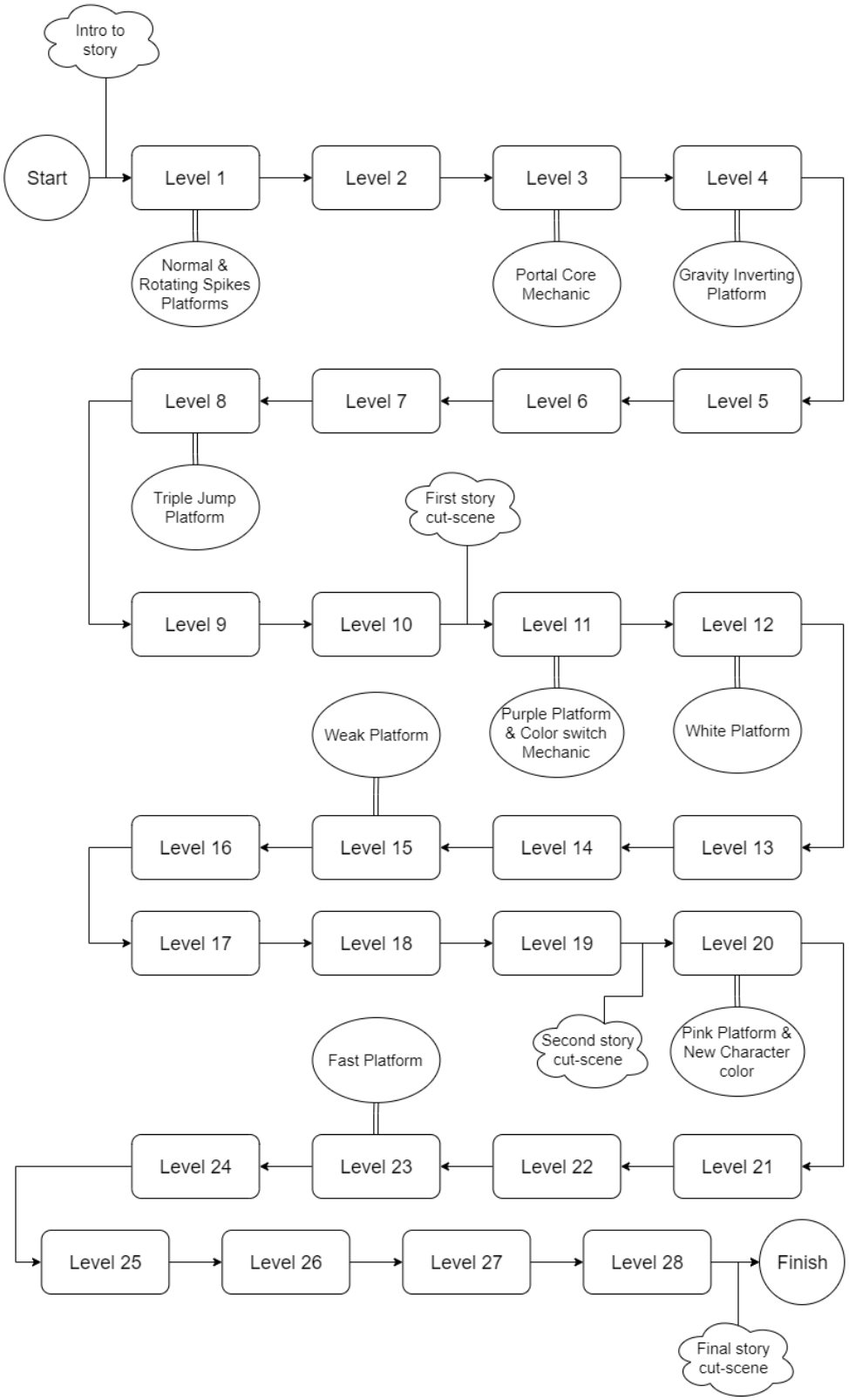
I made a few flowcharts on various aspects just to have a clear goal throughout the course of the project. Here are the designs I came up with:

GAME LOOP



This diagram represents the core Game Loop followed through each level of the game. The explanation being that at the beginning of the level the player has to place the platforms according to his strategy to take the soul to its goal and then play the level based on the platforms he/she has placed. If the level's portal's core is not in the portal, the player has to collect the core before he reaches the goal, otherwise he will not be able to proceed onto the next level. After collecting the core, the player just needs to reach the goal without dying.

GAME FLOW

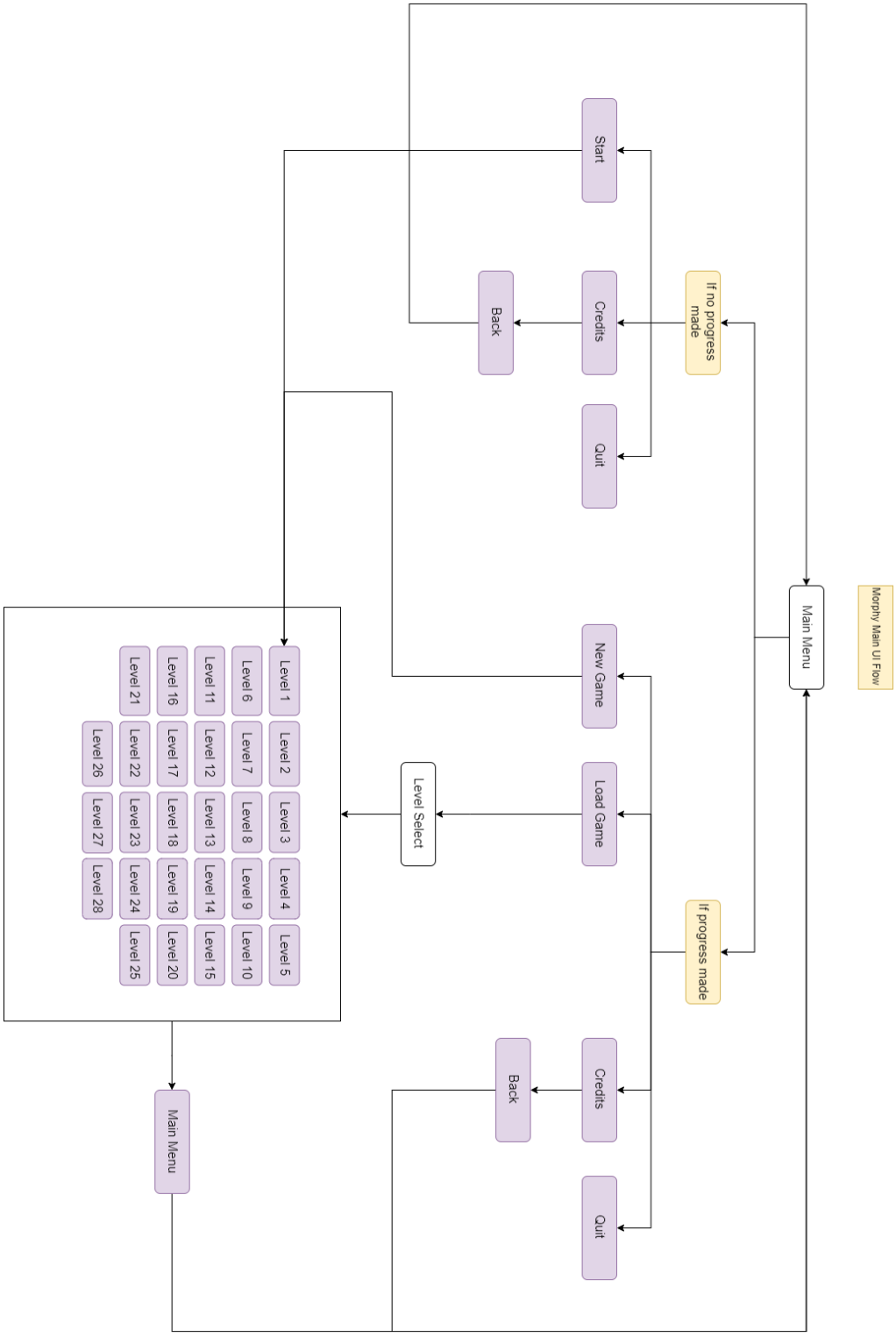


The above flowchart represents the **Game Flow** indicating how many levels and where each mechanic is to be introduced to the player.

UI FLOW

This flowchart shows the exact flow of all the UI Screens and Levels in the game. This helped me later on in the polishing stage to see exactly what I needed to make for the game without forgetting anything important.

After everything was decided, I moved onto the next phase.



PHASE 2 – PROTOTYPING

After the completion of the brainstorming phase, I went on to Unity and immediately started trying to build the mechanics and since I was still fairly new to Game Programming itself, it was a very rough start.

I used Brackeys' tutorial to build my movement system which was pretty straightforward and then I wanted to start with the Rotating Spikes Platform and the initial way I was planning to make that mechanic was apparently very complicated and wasn't necessary. I obviously came to know about this once I spoke to a friend who is good at programming in Unity and with his help I was able to build my first mechanic. At that moment, I realized that there was a very simple way for me to build the mechanic, by just using Booleans for every rotation. I would've saved a lot of time doing this, but I think I can cut myself some slack because this was my first time trying to code entirely on my own.

After this I built the Gravity Inverting Platform, which was made very easy because I was using the movement system by Brackeys and this had a lot of variables and functions that I could use to make this platform. Then I went on to build the Weak Platform, which with the help of Coroutines wasn't hard to do. Although I am missing feedback, it doesn't feel like it after I had observed some playtesting sessions held with my friends.

Implementing the Triple Jump Platform wasn't hard either, but in the beginning I was missing feedback making the game very hard to predict. I left this as it is and decided to work on it in my polishing phase. So I added a feedback of showing up a faded out platform just before the jump when the platform actually shows up. This reduced the difficulty to an acceptable level.

Implementing the weak platform I think was the easiest of all of the platforms, because it required just one Coroutine and that was it. But implementing the coloured platforms was a bit tricky since I needed them to enable and disable collisions based on the colour what the player was at that moment. But with the help of setting integers at each colour, it became significantly easy to manage the code.

The Fast platform took some time to get the values right because if the player became exceptionally fast, it would make it very hard to control and if the player became fast, but not fast enough, it would make it very simple for the player to solve the puzzle. So getting the right values and the right acceleration was key to this mechanic.

After all the mechanics were ready, it was finally time to implement them into levels. I made lots and lots of paper designs for levels and implemented them into Unity. This was the most fun part of the whole prototyping session since this is where I was actually able to see my Game Design coming together and it was also playable, which gave me a vision into what the game is going to look like eventually.

PHASE 3 – POLISHING

After building all the mechanics, it was time to polish out the bugs and other flaws in the game. I started off by building a reset function where the level resets itself upon the player dying, i.e. the rotations of all the platforms, destroyed platforms coming back, the player colour resetting, etc.

I then went on to import the Art Assets into the game. I made basic outlined platforms in photoshop and then to achieve the look as mentioned in the Game Design Document for each of the platforms, I decided to use Shader Graphs to animate them and give them more life. It would also bring about variations in the different platforms and make each platform stand out from the other as well.

I also wrote and played the music in the game myself. All of it was played on the guitar which I had learnt just for this project. I selected the guitar specifically for this project because the story touched upon a very sensitive topic and the guitar can do wonders with the kind of music this game needed. So I borrowed

recording equipment from the University IT department and with the help of that I was able to record the music for the game and it sounded great and perfectly fit the game.

After the art and music were implemented, I had to check all the levels once again, play through all of them, fix any small bugs I could find and simplify any over complicated levels. Did this multiple times through all 28 levels that were implemented and I felt like the sequence was right and it felt like the game I wanted it to be.

I then had a playtesting session by sending the game out to a small group of friends and found a few more bugs with the collisions and character animations which I duly fixed after the playtesting session. I also found a few issues with a few levels being too difficult suddenly and found solutions to each of them and fixed them immediately.

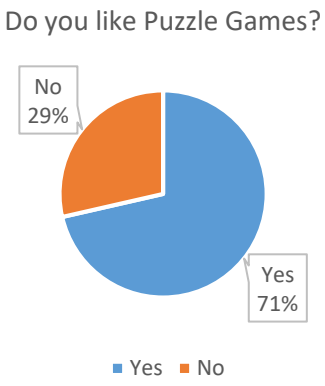
This wasn't the end of it since I still needed another playtest session with everything in the game, but I wasn't yet ready to send out the game for another playtest session. I wanted everything to be in the game before I had a final playtest session. So I started working towards all the other things like the Save function, death counter, UI Flow, Main Menu, Credits Scene, Tutorial in the first few levels, Level Select and Quit functions. All these took me about another week and now I felt like I'm ready for another playtest session.

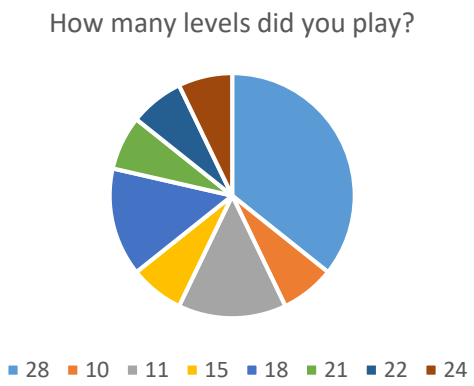
PHASE 4 – PLAYTESTING

Although I have already had multiple playtests before, I call this the playtesting session because this was the final playtest before I could say the game is completed. I sent out the game to around 20 people and the feedback was overwhelmingly positive and I couldn't be happier with the feedback. Everyone liked the game and I also got some critical feedback on the level design and the feedback which I rectified immediately. They also reported a few very small bugs which were fixed immediately as well.

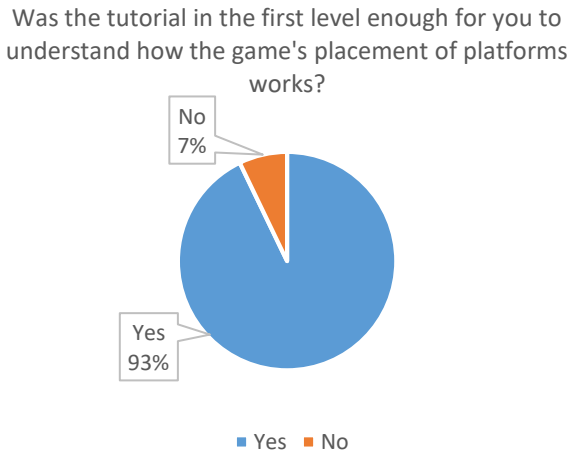
After all this was done, I felt the need to give my special thanks to everyone who playtested the game throughout the Playtest sessions since without them, I wouldn't be able to produce what I have produced.

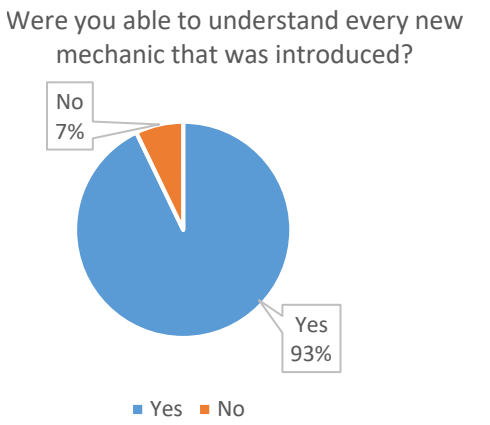
I asked all my playtesters to fill out a form which would give me some statistics and insights into how they all felt about the game. I got a total of 14 responses to my playtesting survey and here are some of the statistics:

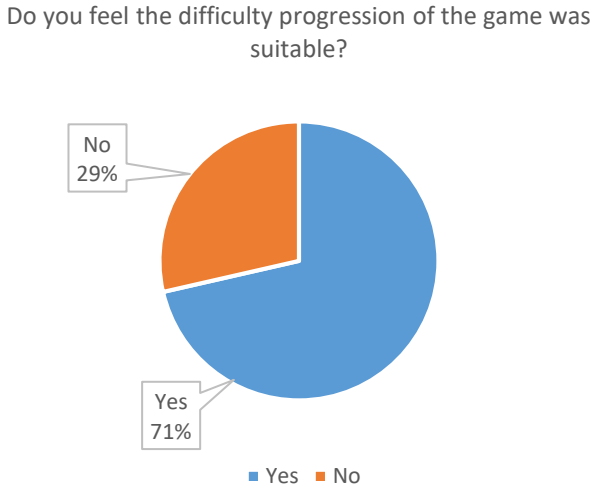




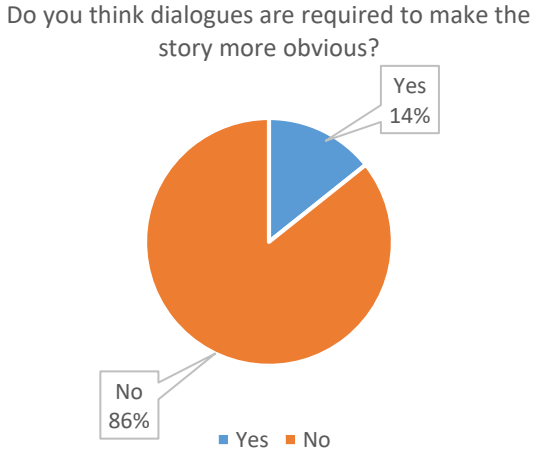
I had to choose a demographic of both kinds of people even though my game is concentrated mainly towards Puzzle game enthusiasts. Out of my playtesters of 14 people, 10 were people who like Puzzle games while the remaining 4 are the ones who don't particularly enjoy them.



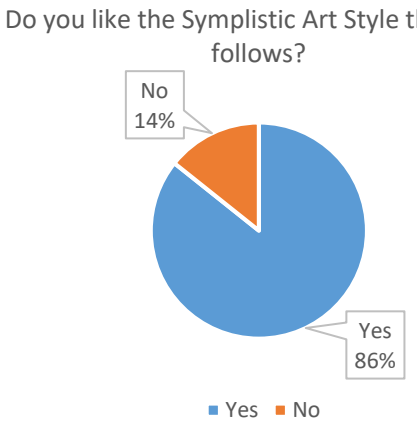


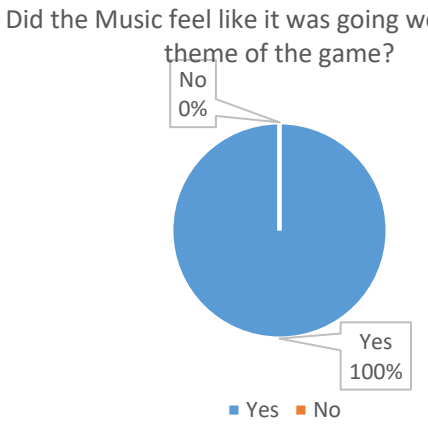


After talking to all the playtesters, I asked them where they felt the progression felt off and upon receiving feedback about level 18 as the main source of criticism, I fixed the difficulty of that level and made it easier to fit the progression of the game.

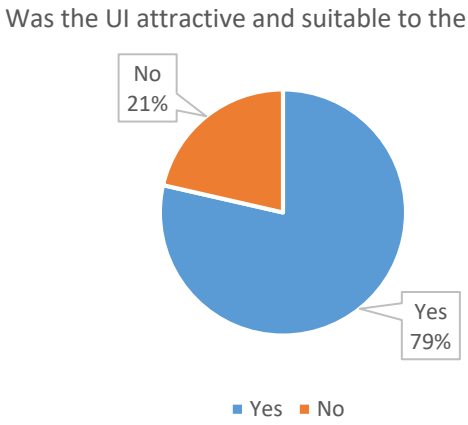


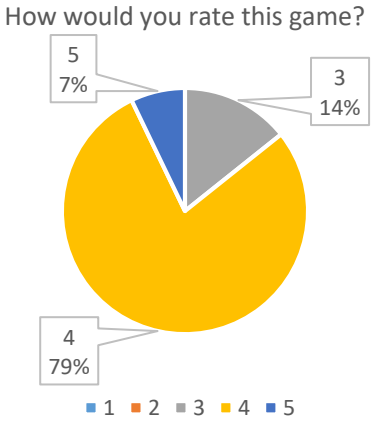
One of the questions in the survey was asking the playtesters what they thought the story of the game was and most of them replied with the correct answer. Which meant that dialogues were not required.





Since I have never really ever touched upon the Art and Music of a game myself before, it was really important for me to understand if what I have done here actually works well with my audience and after looking at the results of the playtesting session, I could not be any happier.





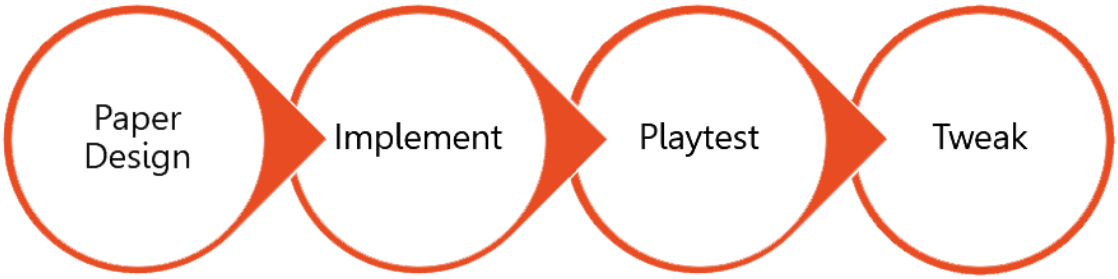
IMPLEMENTATION

GAME DESIGN

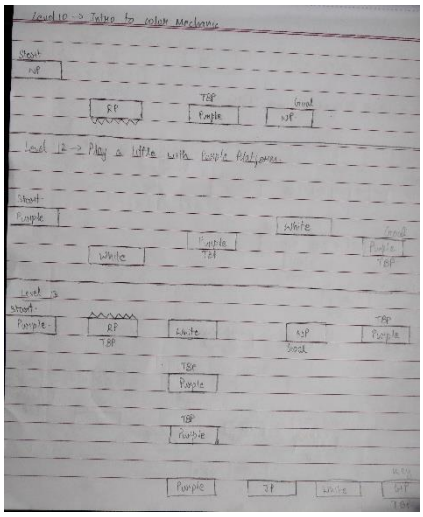
From a Game Design perspective, everything I had originally designed for the game have been included in the final prototype including all the UI screens, animations, the mechanics and level design. More about each of them have been mentioned below.

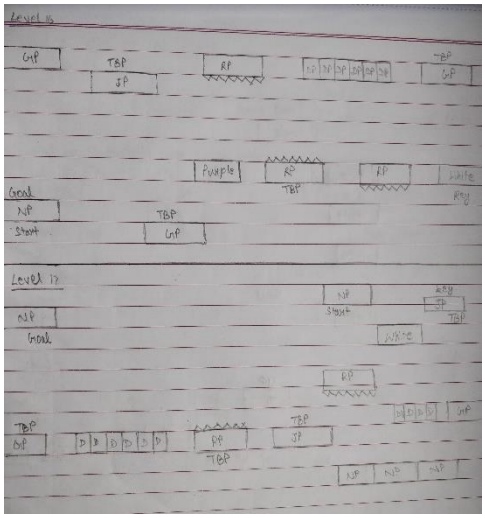
LEVEL DESIGN

I wanted to follow the Kishōtenketsu philosophy in the Level Design process. A full understanding of this has been detailed out in the Game Design Document. This philosophy was brought to life in A Soul's Journey by using the following process:



Below are some of the paper Level Designs from the game:





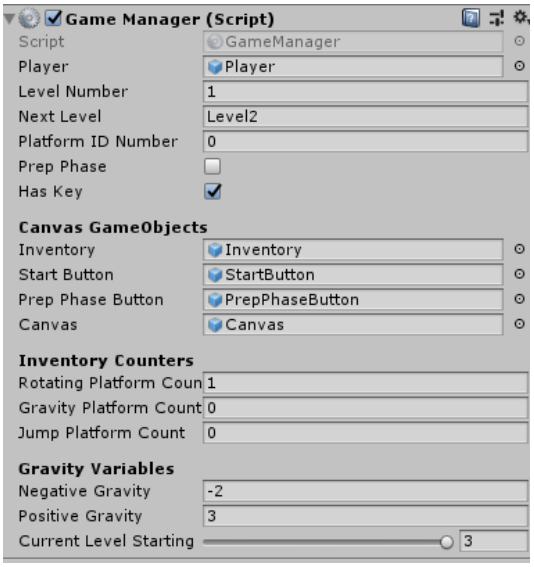
PROGRAMMING

Programming was a huge challenge for me since I had briefly done Game Programming in Unity before. It was as exciting as it was challenging for me to program this game, but in the end, all my efforts of building the game paid off and I am very proud of the final result of the game.

I think keeping the code understandable and clean was one of the key reasons for me not to get too confused while I was programming this game. I always knew where I could find what I was looking for and this saved me a lot of time. Commenting everything, naming functions appropriately and just separating code from other

code based on their functions are all small steps to having very clean and understandable code. Below is a screenshot of my GameManager script which was used in every single level:

```
[HideInInspector] public Vector3 startingCoordinates;
[SerializeField] GameObject player;

public int levelNumber;

[Tooltip("Type the name of the intended next level in this field. Do not forget to include that level into the Build Settings.")]
public string nextLevel;

public float platformIDNumber;

//Preparation phase Variables
public static UnityEvent PrepPhaseEnded = new UnityEvent();
public static UnityEvent PrepPhaseStarted = new UnityEvent();
public bool prepPhase;
public bool hasKey;

Goal goal;

[Header("Canvas GameObjects")]
[SerializeField] GameObject inventory;
[SerializeField] GameObject startButton;
[SerializeField] GameObject prepPhaseButton;
[SerializeField] GameObject canvas;


InventoryCountDefiner invCount;
[Header("Inventory Counters")]
public float rotatingPlatformCount;
public float gravityPlatformCount;
public float jumpPlatformCount;
[HideInInspector]public GameObject rotatingPlatformCountText;
[HideInInspector]public GameObject gravityPlatformCountText;
[HideInInspector]public GameObject jumpPlatformCountText;


[Header("Gravity Variables")]
//Gravity Platform Variables
public float negativeGravity;
public float positiveGravity;
[Range(-2,3)]
public float currentLevelStartingGravity = 3f;
```



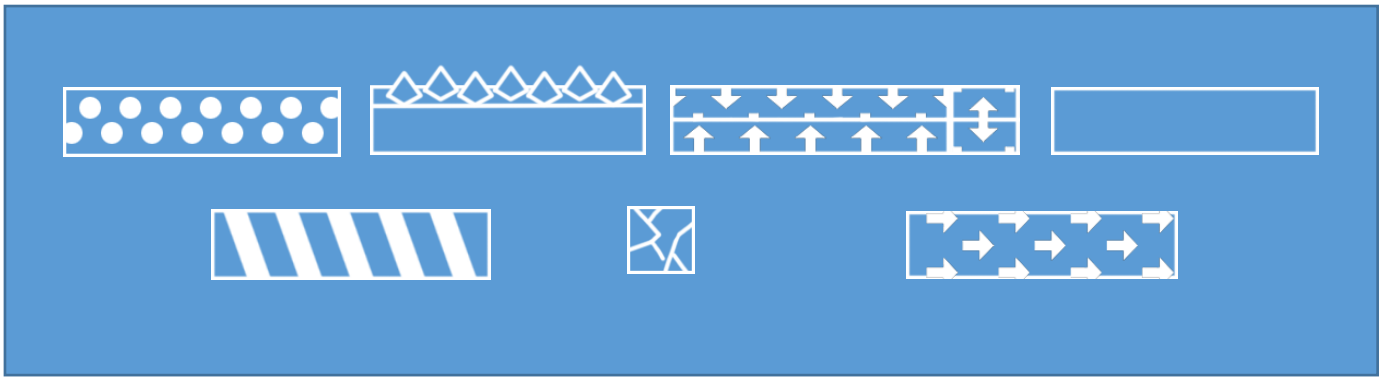
Here in the code you can see me commenting, making tooltips, headers, etc. to organize my inspector view in the Game Engine. I believe these kinds of practices can help a person work efficiently in a team as these practices enable one to explain what he/she has done to others with ease.

ART

At the start of the project I decided to make all of the art in the game by myself and this was a huge challenge for me because I am not particularly good at art. So I took inspiration from Thomas was Alone which followed a very simplistic art style. I made basic shapes and fill them in with different animated shapes with the help of Shader Graphs. In the end, it turned out to look pretty good while the animated platforms gave life to the backgrounds.

I also wanted to give the backgrounds some elements which are “kid friendly” because the game is based on a child’s soul and having elements of things that children like in the game, would suit the theme of the story.

Here are the platforms finally used in the game:



PORTFOLIO

A Soul's Journey is currently featured on the home page of my Portfolio website at https://www.anujpatwari.com and on the same website, A Soul's Journey also has its own dedicated page at https://www.anujpatwari.com/asoulsjourney where the gameplay video can be watched and a few screenshots from the game are also put up. Along with all these, the page also describes the game in short for anyone to read.

The Game Design Document of the game can also be downloaded from the bottom of this page.

The dedicated page also mentions the platform it was made for, genre, target audience & that this project is solely developed by me.

DEVLIERABLES

In this table you can find all the deliverables from my side towards this project:

Deliverable	Name
Game Design Document	AnujPatwari_FinalProject_ASoul'sJourney_GDD
Prototype – Windows Export	AnujPatwari_FinalProject_ASoul'sJourney_WindowsPrototype
Prototype – Mac Export	AnujPatwari_FinalProject_ASoul'sJourney_MacPrototype
Unity Project	AnujPatwari_FinalProject_ASoul'sJourney_UnityProject
Gameplay Video	AnujPatwari_FinalProject_ASoul'sJourney_GameplayVideo
Final Report	AnujPatwari_FinalProject_ASoul'sJourney_Report

LEARNING OUTCOMES

I am very proud and happy with how the final game has turned out and absolutely overwhelmed from the response I have received from a lot of my friends for this game. I could not be happier with the way it has turned out.

What I take away from this project is truly unbelievable. At the start of this project I never expected to have learnt so much in terms of art, music, programming and design. I was able to implement the desired philosophy and I have learnt a lot about level design progression through constant playtesting and feedback.

Feedback from anyone cannot be ignored. Feedback is what the games industry thrives on. Feedback from players helps any game to improve and also brings out the best in both the game and the designer behind the game.

A FINAL THANK YOU

Finally I would like to thank Mr. Jarek Francik and Mrs. Hope Caton, without whom this project would not have been possible. I would also like to thank the following people for the various ways they have all helped me through this project:

Abhimanyu Chattopadhyay, Abhinav Singh, Advait Dhumne, Arunachalam Ayyappan, Bhuvanesh Tekavade, Hussain Dabhiya, Iman Syed, Kishan Kumar, Pulkit Singh, Raul Ravi, Shreegovind Patwari, Shubham Bengale, Sreenath PS, Thaman Cariappa, Theodore Menezes, Uday Kampani and Vladimir Ravichandran.