ORACULAR

By Issy

Winner: Year 7-9 Open Platform

CONSTRUCTION/DESTRUCTION

Proudly supported by

CREATIVE VICTORIA

PAX AUS

IGEA
interactive games & entertainment association

ACER

2023 Australian STEM Video Game Challenge

Game Design Document

Game Name: *Oracular*

Team Member(s): *Issy Fox, Year 8 – Firbank Grammar School*

Open Platform

Game Overview

A summary of the game and its mechanics to provide context to the Game Design Document.

Game Title: ‘Oracular’

Platform Used: GDevelop

Summary: Oracular is a puzzle-platformer game where the player character, a pirate, is trapped inside a dungeon after stealing from it; awakening a spirit that locks the player character in the dungeon for their greed. To escape the dungeon, the player must solve puzzles and defeat enemies in each room of the dungeon, until they finally escape. The player character uses a mysterious staff they have stolen from the dungeon to destroy and create platforms and walls around the dungeon, which they use to avoid traps, bypass obstacles, and defend themselves from enemies.

Oracular has been created using the platform GDevelop, which is free to use, and utilises visual programming.

Oracular should take approximately 30 minutes to play and has eleven main levels.

Oracular can be run in a Microsoft Windows Operating System.

Planning

Organisation:

Responsibility

What are the major roles in the creation of my game?

As I am working on my game by myself, I will have to undertake every role necessary for the creation of my game. The major aspects of the game I will have to complete are as follows:

- Storyline design: Designing the storyline and any potential dialogue throughout the game in a way that can be easily incorporated with any actual gameplay. This may also tie into the creation of a title for my game, and any tutorials/instructions written throughout the game.
- Audio and Soundtrack: Creating a background track for the game, and for any menus or loading screens I create. I will also have to create different sound affects for the game, such as sound effects for the player jumping, using weapons, or otherwise interacting with the environment.
- Programming: The creation of the actual code used to run the game. Along with creating the code, I will also need to test and fix any glitches found during testing.
- Art/texture design: The design of the different characters, backgrounds, and elements of each part of the game. I will also have to design animations for the different actions/movements performed by each character/object. This will also involve the design of any menus, icons and labels/titles used.
- Recording/planning: Completing the Game Design Document and creating any timelines/plans so that I can keep all of the tasks I need to complete organised.
- Game mechanics: Designing the different puzzles and challenges throughout the game, and how the player will be able to interact with each element of the game.
- Testing: Testing the different elements of the game after I have designed them, and fixing any glitches/bugs that arise.

Submission Guidelines

How will you ensure your game meets submission requirements?

To make sure that my game meets submission requirements, I will carefully follow the points of both the rubric and official rules. I will go through each individual point and check them off once I feel that my game fits the requirements, and then review the rules and rubric once my game has been completed, to make sure that nothing has changed while I have edited/tested my game.

My mentor is also helping me to understand and interpret the rules of the game and giving me any additional information I need for following the rules and requirements of the competition.

My game must be suitable to be a ‘G’ rated game by the Australian Classification Board and the ‘E’ rating given to games by the Entertainment Software Rating Board. To make sure of this, I will make sure that any violence depicted in my game is unrealistic and very mild in impact, and I will not address any inappropriate themes or topics in my game. The graphic design of my game can also be used to make any violence, or more intense parts of the game mild in impact, which I will consider when creating the visual style of my game.

Workflow

In what order will you develop the components of your game?

I have created a timeline of the different tasks I will need to complete to develop my game. The timeline is sorted into sections. The tasks in each section may be completed in any order, depending on the results of previous tasks, and some tasks may have to be completed at the same time (e.g., Creating the soundtrack may rely on the type of setting I design). The dates next to each section is the amount of time I expect to spend finishing each section, although some sections may be finished earlier, giving me additional time for other sections. My schedule has me finishing my game before the due date, so that if I do need additional time I can readjust and review how I can make sure that I will finish the game on time.

Timeline:

Section 1: - 5/3 – 11/3

1. Plan brief outline of the game mechanics
2. Draw example of concept/what a part of the game could look like
3. Design the basic concept of the game
4. Start the planning section of the Game Design Document
5. Design basic storyline and setting

Section 2: 11/3 – 15/3

1. Justify connection to theme and look for inspiration within other games/media
2. Find relevant tutorials/examples of similar games
3. Design basic storyline
4. Complete the Planning section of the Game Design Document

Section 3: 15/3 – 18/3

1. Decide on basic controls, gimmicks/functions, and potential textures/art styles
2. Create draft storyline
3. Decide on basic setting
4. Create basic concept of main character
5. Begin the Designing section of the Game Design Document

Section 4: 18/3 – 25/3

1. Create a basic test of the concept of the game– created on GDevelop, as this is the platform that I will use to create the actual game
2. Edit plans and designs as necessary
3. Continue the Designing section of the Game Design Document

Section 5: 25/3 – 31/3

1. Design a draft/first version of the characters/setting(s)
2. Start to create the actual game (focusing on basic gameplay over textures, the soundtrack, and the other smaller details of the game)
3. Continue the Designing section of the Game Design Document

Section 6: 31/3 – 30/4

1. Start to create soundtrack
2. Continue to work on the main game
3. Change and create animations and characters
4. Start to incorporate storyline/any additional minor features of the game
5. Create menus/settings of the game
6. Continue the Designing section of the Game Design Document

Section 7: 30/4 – 22/5

1. Review/decide on game title
2. Finish storyline/any dialogue or cutscenes and additional characters that need to be created
3. Finish creating the main game
4. Test the game – if possible, get my friends to test it as well and give me feedback
5. Change/update any previous notes/parts of the Game Design Document that has been changed/does not end up working
6. Continue the Designing section of the Game Design Document

Section 8: 22/5 – 22/6

1. Change/fix any glitches/improve on any feedback received through testing
2. Finalise character designs and menus
3. Start to look at how the game will work when submitted/uploaded online
4. Create a summary of and describe the specifications of the game and any other necessary note taking/reviews for the Game Design Document, for when it is submitted
5. Finish creating sound effects, music, and final details
6. Continue the Designing section of the Game Design Document and begin the Reflecting section

Section 9: 22/6 – 30/7

1. Review the final game and test the final details of the game
2. Edit Game Design Document and add images/sources
3. Look through rubric, timeline, and official rules for a final time
4. Submit the game (or wait for when it can be submitted if finished early)

    - What is the reason for this order?

To effectively create the game, it is important that I have the chance to create drafts of different aspects of the game, so that I can make sure the game can realistically be made within the timeframe, and with my knowledge of coding. I will also need the chance to edit and remake any parts of the game that don't work or could be changed to improve upon the game. I will also need time to research any necessary coding/game design skills for the game, that I have not used before.

The timeline above gives me the opportunity for this, I have accounted for time that I need to spend reviewing the game, creating drafts and having the game tested by others.

I also make sure to include time I will need to spend updating the Game Design Document, so that I can keep track of any information I need to add to it, as it will be easier for me to record information in the Game Design Document over time, rather than doing it all at once.

When does each component need to be complete?

I feel that completing the soundtrack and final animations of certain effects are some of the last components I need to complete, as these do not directly affect how other aspects of the game function, and instead enhance the experience of playing through the game and create an intriguing setting. While this is important, this is not necessary for testing or working on the rest of the game. Designing the storyline is a priority because every other aspect of the game (the setting, characters, goals of playing the game, etc.) is linked to the storyline, so I may have to remake parts of the game if I do not decide on my storyline quickly. This is why one of the first parts of my timeline is to create a basic outline of the storyline for my game.

The dates listed above each section of the timeline outline a rough idea of when I expect to be working on each section of the game, although I do expect the timeline of the game to change slightly once I actually begin to work on different parts of the game.

What things can you work on at the same time? Why?

I chose to put my timeline into sections of tasks that I may choose to finish at the same time, because I think that some tasks will be easier to complete if I do them at the same or similar times. For example, when I start to create the storyline, I will likely decide on who I want my main character to be, so it will be a good time to design the main character. I will also be able to test my game while I am still creating final details of the game such creating dialogue or any additional characters, as long as the game is still possible to complete without them.

Some of the tasks are also reliant on other parts of the game being completed. For example, I can only update the Game Design Document if I have actually changed enough of the game to have something to document, making this reliant on other tasks I would be working on at the same time.

Timeline

When does your game need to be ready for testing?

My timeline has me first testing my game from the 30/4-22/5. However, once I make any changes from the feedback I receive and finalise all of the details of my game, I will test it again. This ensures that any changes I make from the first round of testing will actually improve the game and helps me to find any glitches/bugs that may have been created by any changes in the code after the first round of testing.

How long will you allow for testing and receiving feedback?

In my timeline, I plan on testing my game at some between the 30/4 and the 22/5. I have allowed a lot of time for this first stage of testing as I want to have time to change any glitches I find in the game, and to then test the game again, until I have tested every aspect of the game and had it run successfully. I will also want to have time to discuss my game with my mentor, so that they can give me advice and make sure that my game complies with the official rules of the competition.

As I will likely end up changing different aspects of the game after testing it for the first time, I expect to have multiple tests of the game done at different times, so that I can test that any newer changes I have made are actually effective.

When does your game and Game Design Document need to be ready for submission?

While 11:59pm, 7th of August is the official due date, I would ideally like to have my game ready by the 1st of August or earlier, so that if I have any trouble submitting the game or have any final changes that I need to make, I will still have time to submit it.

Within my timeline, my work on the game finishes by the 30th of July, but realistically I know that I may need longer to work on the game, especially as different events, such as school camps may have me miss certain dates throughout the project. However, I may also finish certain sections faster than I expect to, so I do not think there will be a large variation between my set of dates and the actual date I am ready to submit my game.

Inspiration and Points of Originality:

What has inspired your game?

- Are there any cartoons, books, movies, or games that inspire you?

The game Celeste will be a good inspiration for the art style of my game, as the game has a relatively simple art style, which is used to add to the sense of adventure in the game, as the main character attempts to climb a mountain. The game Celeste is a platformer game that uses a limited number of controls, so that the player can focus on learning to adapt to different puzzles and challenges as the game progresses. This makes the game accessible to all skill levels, which I feel is an important part of designing my game.

The game Spelunky is also an inspiration for my game, as the main character in the game is a miner who explores a cave to find treasure. This is similar to the storyline of my game, where the main character will explore a dungeon to find treasure at the end. Both my game and Spelunky will be platformer games, and as the cave of Spelunky is similar to a dungeon I can use the Spelunky as inspiration for the design of the environments that my game is set within.

- How do you plan to use this inspiration in your game?

I will use this inspiration for designing the layout of each level of my game, as creating a game that is challenging for the player but not too difficult to complete is important, as puzzles I create will be easier for me to understand than someone who has never seen my game before playing it.

This also links to how different games approach the idea of progression over time, and how I can continue to add additional challenge to the game as it progresses. Looking at how the challenges of the games Spelunky and Celeste are incorporated into each game will help me to understand what a reasonable amount of challenge will be as I design my video game.

The graphics of each game will also inspire the graphics of my game. Both Celeste and Spelunky are 2D games, that utilise simple colour palates to maximise the impact of the art, which is something I would like to mirror within the art style of my game. Celeste uses pixel art, which is likely what I will use for the majority of my game, meaning that Celeste will be a good example of a style of graphics in a video game that I can attempt to emulate for my game.

Overall, having inspiration for the design of my game will be a great way for me to understand how I can design a game that is enjoyable to play, and something that can be easily played by a large audience.

Are there other games that have similar gameplay mechanics?

- Similar functionality?

The game *ElecHead* by Nama Takahashi has slightly similar gameplay mechanics to my game. The player character in *ElecHead* creates electricity, which powers different platforms and obstacles the character comes into contact with. This can cause platforms to move or obstacles to appear, meaning that the player has to be careful to only power on objects that will help them to progress through the game. This is similar to how platforms and obstacles can be caused to appear or disappear in my game - however this will be caused by magic weapons in my game, rather than electricity. Both games are platformer/puzzle games, where the player can control objects appearing/disappearing to help them progress through the game.

- Similar stories or characters?

The game *Uncharted: Drake's Fortune* has a similar story to my game as the player character, Nathan Drake, goes on a journey to find the treasure of El Dorado. This is similar to how in my story, the player character is a pirate who is searching for treasure. In Uncharted, the player searches a temple for the treasure, progressing through the different rooms, similar to how the player in my game searches through the rooms of a dungeon to find the treasure.

The character Linebeck, a recurring character in the *Legend of Zelda* series is also similar to the player character of my game. Linebeck is the captain of a steamboat which he uses to search for treasure. He helps the player character, Link in return for receiving treasure. This is similar to how the main character of my game is searching for treasure. Both characters travel by boat, and can be quite greedy, as they are both desperate to find treasure.

How will your game be different?

- Why would people prefer to play your game over these other games?

My game is different to other games because of the way I will combine solving puzzles and fighting enemies throughout the game. One of the main game mechanics of my game is how platforms and walls can be created and destroyed by the player character over long distances.

This is an important aspect of my game because it will be used to create different puzzles throughout the game, where the player has to find a way to move to the exit of the room by climbing platforms and moving around obstacles. However, this may also be used to make combat in the game more interesting, as enemies may be able to be blocked by walls that can be summoned or fall off platforms once they are destroyed. This makes the game unique, as to complete it, the player needs skill in both solving puzzles and defeating enemies to progress.

Is your game different enough to be worth making?

- Why/why not?

Yes, as both platformer and puzzle games are popular genres, and by combining the two in a simple way, many different people, of many different skill levels will be able to play my game and have an interest in the different gameplay mechanics.

The visual style of the game is also unique, as it combines more retro pixel art with the modern effects and styles that can be created with GDevelop, creating a unique combination between the two styles.

One of the main ways that my game is unique is also through the way that the theme of ‘Construction and Destruction’ is used. While the player is able to destroy and create platforms, it is different to how this concept is often used in games to allow the player to build and destroy their own creations. My game has the player create and destroy objects that are in fixed positions in each level of the game. This unique approach causes the player to have to think about the potential effects of the placement of each object within the level, and causes them to be more thoughtful about how they influence the environment around them, creating a unique experience for the player.

References:

IGN. (n.d.). Spelunky. Retrieved from https://www.ign.com/games/spelunky.

Linebeck. (2023). Retrieved from https://zelda.fandom.com/wiki/Linebeck

McWhertor, M. (2022). Clever puzzle-platformer ElecHead is coming to Switch. Retrieved from https://www.polygon.com/23065717/elechead-consoles-nintendo-switch-release-date.

Petite, S. (2018). 'Celeste' Review. Retrieved from https://www.digitaltrends.com/gaming/celeste-review/

Stevenson, R. (2022). The Uncharted Games Explained: What You Need To Know Before You See The Movie. Retrieved from https://www.looper.com/742509/the-uncharted-games-explained-what-you-need-to-know-before-you-see-the-movie/

Tahir, S. (2023). 6 Games With The Best Art Style. Retrieved from https://gamerant.com/games-with-the-best-art-styles/#:~:text=6%20Games%20With%20The%20Best%20Art%20Styles%201,Cuphead%20...%206%201%20Persona%205%20Royal%20

Tyler, D. (2023). What Makes a Good Game so Much Fun? Retrieved from https://www.gamedesigning.org/gaming/great-games/

Technical Requirements:

Development Environment

In which operating system will the finished product run? (Windows/Mac/both)

My game will run on Windows as GDevelop, the platform I am using to develop my game, will allow me to export my game to a Windows operating system.

What platform will you use to build your game?

- What are the advantages and disadvantages of this development environment?

I will use the platform GDevelop to build my game. GDevelop is a visual programming platform, that is specialised in 2D pixel art games. As my game will be designed using pixel art, and is a 2D game, this is perfect for what I need to design.

One of the main advantages of GDevelop is the amount of different resources it has for people who are new to game design. There are many different tutorials built into the platform, as well as many example games that demonstrate different techniques or skills that can be used.

GDevelop also has an integrated program Piskel, for pixel art, and JFXR, for sound design. I will use these two platforms for the sound and art of my game, allowing me to keep most of the elements of my game on the one platform, rather than having to use multiple different programs to design different elements of the game, which will allow me to be more efficient.

While visual programming is an advantage for me, as it allows me to properly use the platform without having to learn a programming language, this may be considered a disadvantage to someone who knows a programming language, as they will not be able to use it within GDevelop.

System Requirements

What sort of system, specifications or peripherals will the player require to play your game?

A mouse or trackpad will be required to navigate through the different menus of my game. My game will run on a Windows operating system, meaning that it will be required to play my game. A keyboard will be needed to play my game. However, my game does not have many peripherals required as neither the camera or microphone will be used.

While not required, headphones can improve the experience of playing my game, so that the sound effects and music can be more immersive.

Resourcing/Capability

What tools will you need access to in order to fulfil your technical requirements?

My game will need to be able to be played without the player having to download additional software. To do this, I will need to use GDevelop's publishing feature to turn my game into a file that can be distributed/exported to other people for them to easily play my game. For this, I will use the GDevelop 'Wiki' which has information on how to publish a game using the platform.

GDevelop is able to export games to be played on Windows OS, which is also a technical requirement of my game.

I will also need to make sure that the file size of my game is as small as it can possibly be to make it easier to download and export. This means that I will need to learn about optimising the code of my game, and I will also need to delete any unnecessary files or resources that I download/design, but end up not requiring (e.g. unused fonts, unused characters, redundant code, etc.).

Where will you learn new skills to help with game design and development?

GDevelop, the platform I am using to develop my game, has many different resources for people new to the platform.

One of the main ways I will learn any necessary skills I need is through GDevelop’s tutorials and example games that are designed to teach people common elements of game design (e.g. adding health bars to the game, or calculating the score of a player). This will help me with many of the more common skills I will need to learn. I can also look at the example games for inspiration for my art style and audio, as I will be able to see varying art styles that can be achieved using GDevelop.

Another way I will learn skills needed for the development of my game is through different tutorials online. GDevelop has both a YouTube channel and online forum, which explains many of the different capabilities of the program. As GDevelop uses its own visual programming format, this will help me to learn the exact meanings of different aspects of the platform.

I will also look at different tutorials from game developers online, as getting information from multiple sources (rather than just GDevelop), may help with different problems I encounter.

Designing

Game title

What is your game called?

My game is called ‘Oracular’.

Why did you choose this name?

The word Oracular means advice that is hard to understand, or unclear/confusing. I think this fits the theme of the game, as, being a puzzle game, the player will have to solve puzzles and navigate levels with little to no instructions – making the game ‘Oracular’. The word oracular can also be used to describe something connected to or from an oracle, and while this is not directly relevant to the game, the fantasy elements of the setting – a magical dungeon with monsters and other creatures in it, relates to the idea of a mysterious oracle.

Does this name help players to know what the game might be about?

The name does give the player an idea of what the game might be about, as while they may not know the definition of the word oracular, the mysterious sound of the word alludes to the idea that the game may set in a fantasy setting. The meaning of the word also relates to the game, as the spirit of the dungeon that speaks to the player in the start of the game gives the player vague advice,

to build suspense during the first scenes, so the meaning will become clear to the player once they begin the game.

Game description

What is your game about?

Oracular is about a pirate (the player character) who was exploring a dungeon in an attempt to steal treasure, and in doing so, awakens a mysterious spirit that traps them inside it. To escape this dungeon, the player must navigate the different rooms by defeating enemies in melee combat, and by solving puzzles. To solve these puzzles, the player must destroy and create platforms and other objects in each of the rooms of the dungeon, using a magical staff, which will allow them to overcome obstacles and avoid traps. This will allow them to progress and eventually escape the dungeon, after finally proving themselves by defeating the challenges of the eleven main rooms of the dungeon.

What will players do in the game?

Throughout the game, the player will need to be proficient in both combat, and puzzle-solving, as enemies will be spread throughout the different rooms of the dungeon the player navigates. The player will aim to escape the dungeon they are trapped in, which they will do by reaching the exit door of one room of the dungeon, bringing them to the next room, until they reach the final exit. To reach each door, the player will have to solve puzzles by evading traps, defeating enemies using melee combat and overcoming obstacles. The player can do this by using the magical staff the player character is able to use to destroy and create different glowing objects in each room. The player will have to think about how destroying and creating different objects will affect their environment, as something that may seem helpful may later hinder their progress. The melee combat in the game will be simple, a sword the player character has from the start of the game will be the player’s main way of defeating enemies. However the player may be able to use different traps and obstacles to their advantage to defeat enemies (e.g. destroying platforms that the enemies stand on, or creating walls between the player character and any enemies).

What is the objective of the game?

The objective of the game is to escape the dungeon by progressing through each of the eleven rooms of the dungeon (each room of the dungeon is equal to a single level). To do this, the player will have to defeat enemies using melee combat and solve puzzles by destroying objects that prevent them from reaching the exit, and creating objects that will helps them to bypass traps. When a player dies, their progress in the room of the dungeon they are in is reset, meaning they need to reach the exit of each room of the dungeon without dying to progress.

The player character wants to escape the dungeon because they have been trapped inside it after stealing treasure, disrupting a spirit that lives in the dungeon in the process. This means that the player’s objective is tied to the goals of the player character, making the storyline more immersive for the player.

Who is your intended audience?

My intended audience is people who enjoy puzzle games and platformer games – as my game incorporates the two of these genres, this allows it to reach a larger target audience. I have attempted to make my game enjoyable for a wide audience, as, although some of the puzzles and

challenges of the game can be difficult, I have added a ‘help’ option to the game that will give the player hints and additional information, which will allow less experienced or younger players to enjoy the game as much as an older player will.

What makes your game fun or interesting?

My game is made fun by the way that combat and puzzles are intertwined within the game. What could be an otherwise simple puzzle is made more complicated when the player has to consider how they will avoid or defeat enemies in the room of the dungeon. The player also has the opportunity to use the elements of the puzzle to their advantage by creating walls to stop enemies from progressing, destroying platforms to cause them to fall, or even catching enemies in the traps of the dungeon. This allows the player to be creative and try many different strategies for overcoming even the simpler levels of the game. This also makes the game more enjoyable to play multiple times.

Does your game have characters or objects?

- What role do the characters or objects play in the story?
- What is the motivation for these characters or objects within the game?

There are many different objects and characters used in Oracular. The player character is the main way the player interacts with the game, through using the controls of the player character to create and destroy objects, and to damage and evade enemies. The other main character if the game is the spirit that traps the player character in the dungeon at the start of the game, who the player character fights in the final room of the game. This creates the ending for the game, and the most difficult level of the game – the final boss battle.

One of the most important objects in the game are the platforms and walls that can be destroyed or created by the player. These platforms are used to add an extra element of the challenge to the game. For example, a wall blocking the player, once destroyed, may make it easier for an enemy to damage the player, or a platform later needed for the player character to stand on, may first need to be destroyed to allow the player to progress through the room.

Other objects throughout the game are primarily used to create puzzles for the player to solve, such as levers and buttons that move platforms. These objects are important to the game, as the puzzle elements of the game make it so that the player has to properly explore each of the levels, making the platforms and objects that can be created and destroyed more important than they would be otherwise.

There are also different enemies and traps throughout the game, which can deal damage to the player. This means that the player has to be careful of how they move throughout the level, as often the platforms the player can create and destroy can be used to their advantage. For example, enemies placed on platforms the player character needs to stand on can be avoided if the player destroyed the platform, causing the enemy to fall to the floor, and then the player can create the platform again for the player character to stand on.

Environment

Where does the game take place?

The game takes place in a magical dungeon that the player character found and stole from on their journeys to find treasure. The player is then trapped in the dungeon by a mysterious spirit, with a strange staff they stole from the dungeon, that they later discover can destroy and create certain objects. Each level of the game takes place in a different room of the dungeon. The games ends once

the player character is able to escape the dungeon, after they have progressed through 11 rooms of the ancient dungeon.

The dungeon the character is trapped in is shown to be on an island surrounded by jungle through the different cutscenes within the game.

Under what conditions does your game take place?

-Do these conditions have any effect on the gameplay that you might need to consider?

One of the main conditions that affects the gameplay of Oracular is the setting. Because the game is set in a dungeon that the player is trapped in, most of the levels are set in relatively small, enclosed spaces. This creates an emphasis on puzzle solving that is unique to Oracular, because relatively small changes to the level (creating a platform, destroying an object, or moving an element of the level), can massively affect whether the player will be able to complete the level. The smaller areas of the game also encourages the player to defeat enemies, rather then attempting to run past them, as this is difficult to do within the narrow tunnels of each level.

Another condition of the game is the fact that it has a fantasy theme. To represent this within the game, many of the objects and creatures I use within the game will not based off of any real creatures. This will allow the art style to properly reflect the conditions of the game.

Theme

This year’s theme

What is this year’s theme?

This year’s theme is ‘Construction and Destruction’.

Link to your game

How is the theme incorporated in your game?

The theme appears in my game through one of the major game mechanics available to the player. In my game, the player character receives a staff that, when used, can destroy, or create different platforms and objects throughout the game. When the ‘Q’ key of the keyboard is pressed an orange projectile is shot from the staff that will destroy any glowing objects it collides with in the level, and when the ‘W’ key is pressed, a blue projectile appears instead, which rebuilds any of objects that have been destroyed. This relates to the theme as the player is able to ‘destroy and ‘create many of the different elements of the game, which they will need to do to progress through the levels.

The theme is also very important to the player’s strategy throughout the game, as the platforms that can be created and destroyed can be used to the player’s advantage to navigate traps, defend themselves for enemies and solve different challenges within the level.

As the game is a puzzle-platformer game, the player will need to jump along platforms, and climb parts of the room to reach the next level. Having the player be required to create their own path to progress through the game creates a unique spin on the genre, that makes the theme ‘Construction and Destruction’, central to the game.

What parts of your game relate to the theme and why?

One of the main game mechanics of my game – the player’s ability to create and destroy objects throughout the game – incorporates the theme. Throughout the game, in the different rooms of the

dungeon, there will be glowing objects, such as specific platforms, staircases and walls in each room. These glowing objects can be destroyed and created when the player uses the magical staff they have been given at the start of the game. The player will have to think about when to destroy and create different objects in the game, as destroying a wall to move closer to the exit of one room of the dungeon may bring them closer to enemies, into traps, or bring them to dead ends. This means that they will have to create creative paths through each room of the dungeon by creating and destroying objects to defend themselves from enemies, and to bypass any traps.

This relates to theme of ‘Construction and Destruction’ as the player will have to create and destroy objects, with this being one of the major gimmicks of the game. The way the theme is incorporated into my game is creative, as the character does not have to be standing next to an object to destroy it physically, using tools in a way commonly found in many other games. Instead, the player is able to destroy and create objects over a distance using the magical staff they have been given. The destruction and creation of objects is a focus of the game, as the player does not receive any resources, points, or any other incentive for creating/destroying objects, other than the fact that they will need to do so to complete the game, and to solve the puzzles of each room of the dungeon.

Gameplay/mechanics

Objectives/Goals

What is the aim of the game?

-Can a player win the game? How?

The aim of the game is to make it through each of the eleven levels of the dungeon, to then defeat the final boss of the dungeon, allowing them to leave. This is how a player will win the game, as the story of the game will conclude with the player escaping the dungeon and preparing to return to their pirate ship, to sail back home. Whenever the player dies, the room of the dungeon they are in is reset, meaning that they will have to complete every challenge in any individual room of the dungeon without dying to reach the next room. This means that to win the game, while the player can die an infinite number of times and still win the game, they have to properly complete each level eventually to progress.

What is the player trying to achieve?

- Can a player finish the game? How?

The player is trying to escape the dungeon, by solving puzzles, avoiding obstacles and traps, and defeating enemies to reach the next room of the dungeon, until they finally reach the exit of the last room and escape. If the player dies, the room the player is currently in is reset, meaning that the player has to be careful to not be damaged by enemies or traps to win the game. The puzzles in each room of the dungeon become more difficult closer to the end of the game, preparing the player character for the final boss battle against the spirit that has trapped them in the dungeon, allowing them to finally escape the dungeon.

The game has a storyline that is completed when the player character escapes the dungeon.

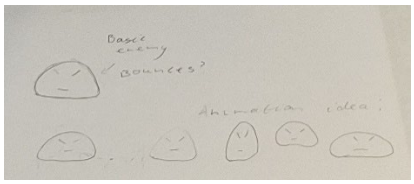
How does the player progress through the game?

- Are there multiple levels or does the game get more difficult or introduce new goals over time?

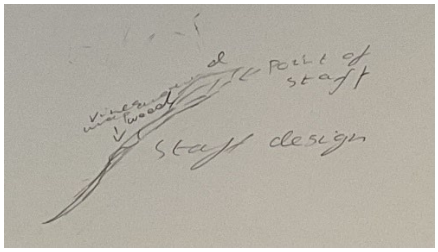
The different rooms of the dungeon the player travel through throughout the game act as levels. Each room will progress in difficulty, presenting the player with different challenges and skill to learn, along with enemies that are harder to defeat. The enemies in the later game may move quicker, have higher health, deal more damage, or have more erratic movements than previous enemies, making it a more difficult challenge for the player to defeat them. The levels become longer as the game progresses with more complicated obstacles to overcome. The puzzles also become more difficult over time, with more elements introduced to the puzzles in later levels.

To keep the game interesting, levels introduce new elements and objects over time, such as keys that need to be used to unlock the exit door of the level, moving platforms and levers. This prevents Oracular from becoming too repetitive or boring over time.

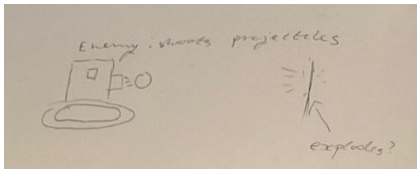
Concept drawings and storyboards:



Design of an enemy (unused).

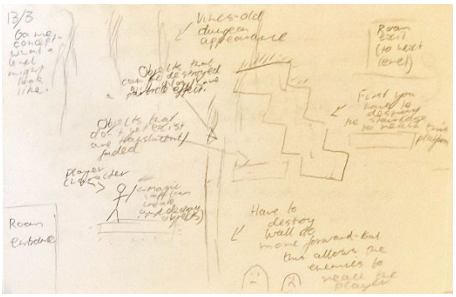


Design of the staff the player uses.



Secondary enemy design.

Start of game cutscene storyboard.



Sketch of example level design.

Perspective

What is the player’s perspective when playing the game?

- Do they experience the game from a first-person point of view?
- From the side (like a platformer)?
- From a top down perspective?
- Is it two dimensional(2D) or three-dimensional (3D) game?

The player’s perspective when playing the game is from the side. This is typical for a platformer game, as it allows the player to easily see the different obstacles the player character will encounter before they would be able to from a different perspective.

As the game is two-dimensional, it will appear as if the player can see through one of the exterior walls of the dungeon the player character is trapped inside of.

Controls

How do players actually play or interacts with the game?

The player controls the player character, who is the vessel for all of the different ways the player can interact with the actual game. The player uses numerous different controls on their keyboard that are introduced to do them within the different levels of the game. The player character often needs to be physically near objects to interact with them (e.g. to switch a lever on, which is important in many levels of the game, the player character needs to be standing next to the lever), making it important for the player to be skilled at manoeuvring the player character.

The player can also use the cursor to open the settings menu, or to access the ‘help’ menu, which will give them different hints and information about each level.

What are the controls?

- How do they work?

The controls used throughout the game are as follows:

‘W’ Key – Shoots a ‘Construction’ projectile from the player character’s staff, creating any previously destroyed objects the projectile hits. The projectile is shot directly forwards from the direction the player character is facing and will move at a consistent speed until it collides with an object and explodes.

‘Q’ Key – Shoots a ‘Destruction’ projectile from the player character’s staff, destroying any glowing objects found throughout the game. The projectile is shot directly forwards from the direction the

player character is facing and will move at a consistent speed until it collides with an object and explodes.

Spacebar – Allows the player character to jump.

‘Right’ Arrow Key – Allows the player character to walk to the right.

‘Left’ Arrow Key – Allows the player character to walk to the left.

‘A’ Key – Allows the player character to perform a melee attack, swinging the player character’s sword in front of them. The player can also trigger levers by pressing the ‘A’ key while the player character is standing next to a lever. In the settings and start menu, the ‘A’ key is used to select options on the menu.

‘W’ Key and Down Arrow simultaneously – A ‘Construction’ projectile appears as previously mentioned; however, the projectile is shot downwards.

‘W’ Key and Up Arrow simultaneously – A ‘Construction’ projectile appears as previously mentioned; however, the projectile is shot upwards.

‘Q’ Key and Up Arrow simultaneously – A ‘Destruction’ projectile appears as previously mentioned; however, the projectile is shot upwards.

‘Q’ Key and Down Arrow simultaneously – A ‘Destruction’ projectile appears as previously mentioned; however, the projectile is shot downwards.

‘S’ Key – Allows the player to see the positions of any destroyed objects in the level.

‘E’ Key – Resets the level, putting the player back at the start of the level and resetting the states and positions of any enemies and other objects.

Either ‘Shift’ Key (while the player character is jumping) – Makes the player dive to the ground, this damages any enemies standing on the ground near where the player lands.

’Up’ Arrow Key - Allows the player character to grab and climb a ladder if they are in front of it.

‘Up’ and ‘Down’ Arrow Keys – Used in the settings and start menus of the game to highlight different menu options.

Cursor – used to select the menu, and to turn the audio of the game on and off in the settings menu.

‘R’ Key – Skips text in the tutorial.

‘P’ Key – Continues to the next line of text in the tutorial.

Many of the more complicated controls are introduced later during the game. For example, climbing ladders is only necessary for the player once they reach the third room of the dungeon, so the controls are only introduced to the player once they reach that level.

Instructions/Tutorials

What features did you include to help the player learn to play the game?

The first few scenes of the game explain the story of the player character. This informs the player of the context of the game and gives them an idea of the challenges they will face while playing the

game, the player then goes through a tutorial which explains the basic controls of the game. Player then has an opportunity to test against different objects in the room (e.g., destroying a platform that blocks the exit to the room.

The first few levels of the game are much easier than the rest of the game, so that the player has the chance to test the different controls of the game, and to learn how different game mechanics work and interact with each other. Making the tutorials of the game interactive will make it more interesting and memorable for the player.

One of the most important features I have added to the game to help the player is the ‘help’ button. This is a button that can be pressed in any level of the game, that gives the player hints and advice about how to solve different puzzles, or where to go next in the level. This helps players learn to play the game if they have forgotten any information, or don’t understand a certain aspect of the game.

Visual and Audio Design

Style

What is the visual style of your game?

- Is it inspired by an artist, art movement, time, or place?

All the art in my game has been created using pixel art. This creates a blocky, cartoonish look throughout the game, as most of the sprites and textures are 16x16 pixels – making different colours and patterns within them stand out. The art style also uses mostly brown, red and grey – these colours make the environment feel more like the dark, underground dungeon that the game is set in, without actually making the game too dark for the player to see.

While the visual style of my game is not inspired by a specific artist, art movement, time or place, there are many other famous video games that use similar pixel art, such as Celeste, which, as previously mentioned, is a good example of a platformer game that is well known for its art style.

To find inspiration for my art style, I looked at many similar artworks or art styles of different games that utilise pixel art, and used this to better understand how I could create a cohesive art style for my game.

The pixel art in my game is used to make it more appealing to younger players, and it makes different elements of each puzzle easy to spot and recognise.

Have you linked the visual design to this year’s theme? Explain.

While the art style is mostly not directly linked to this year’s theme, the visual design of the game is very important for how I have chosen to incorporate the theme. The platforms in the game that can be created and destroyed all have a glowing ‘godray’ effect used, to make them stand out among the regular platforms of the game. Most of the levels in the game also use primarily grey, brown, and red colours, making the rooms of the dungeon feel as though they are old and decaying over time, similar to the theme of destruction.

The basic art style of the game makes the different elements of the game easy to distinguish and remember, allowing the player to focus on the puzzles of creating and destroying the objects in each level.

How do your audio choices relate to your visual style?

The sound effects used in my game generally are not intended to be realistic. Instead, they sound cartoonish and easy to hear, this is similar to how the art style of the game is colourful and designed in a way to make it easy to distinguish objects from one another. This works well with the dungeon setting of the game, as the typically dark atmosphere would normally make it difficult and frustrating for the player to see the finer details of the level.

I also occasionally chose to use more realistic sound effects, especially for the different monsters and ambient noise within the cave. This ties to the fact that, although the visual style of the game is relatively bright, the storyline and environment of the game are more serious, as the player character attempts to overcome many dangerous challenges throughout their journey.

Did you use any platform assets in your game?

- Which assets and why?

Other than the font used for certain parts of the game, I have not used any assets for the art or audio of the game. I have created all of the art using Piskel, a pixel art program built into GDevelop. Piskel has not provided me any assets or resources beyond basic brushes and line/shape tools built into the program. I wanted to create all of the art for my game myself so that I could make sure it all had a consistent style, and so that I could easily adjust the art in my game whenever I wanted/needed to make changes.

All of the sound and music in the game were created by me using JFXR and BandLab Assistant, two free tools for sound design, with JFXR being used primarily for sound effects, and BandLab Assistant for music.

Provide example sketches, storyboards, and images.

Process

How did you go about achieving your desired visual style?

I created all of the art for my game using the program Piskel within GDevelop. I did not have any previous experience in making pixel art for my game, so to start off with, I spent a lot of time watching and reading tutorials, and looking at examples of other games that use pixel art. Over time, I found that making most of the sprites in my game 16 x 16 pixels created a style that I liked for the game, and while I often scaled the sprites to larger sizes to add details or to make them fit into the level better, most objects in the game were designed as 16 x 16 sprites/textures.

As my game is in a dungeon, the colours most often used are grey and brown (for the stone/bricks and different platforms), black (for finer details and outlines), and red (for the lights in rooms, enemies, and the colour of the player characters coat. Using these colours consistently throughout the game makes the game feel more cohesive, and the simple limited detail of the objects combines well with the limited colours of the game to create a distinctive visual style.

I also used various special effects, such as warm light emanating from torch objects within each room of the dungeon, glowing objects to emphasise certain elements of the level and shadows cast from different objects to further enhance the environment of each level. The dynamic lighting of the level allows the objects of the room to feel responsive to the player's movement, creating a more interactive environment.

How did you create your music and sound effects?

To create the music and sound effects for my game, I used multiple different programs. One of the main programs I used to create sound effects is called JFXR, which is a free sound effect generation program that has been integrated into GDevelop.

I used JFXR to create many of the simpler sound effects in the game, such as the jump sound effect, damage sound effects and the sounds of elements used in the menus. JFXR was a good way for me to quickly generate sound effects, as it would automatically create a default sound effect, which I could then adjust aspects of (e.g., frequency, sustain and harmonics), to create an appropriate sound effect.

For the more complicated sound effects in the game, I used a program called BandLab Assistant. BandLab Assistant is a Digital Audio Workstation (DAW) similar to programs like garage band, which has different tools for creating and editing audio and music.

I used BandLab Assistant's virtual instruments to create the songs used throughout the game, as well as some of the different sound effects in the background of each level. I also used BandLab Assistant for the sound effects in the background of some of the dialogue throughout the game.

Most of the songs I created used mostly percussion instruments, because I felt like this blended into the background of the game best, while still adding atmosphere to the game. BandLab Assistant has a selection of different drum machines which can be used to emulate the sound of a drumkit. This was one of the major tools I used for many of the songs, such as the one played in the final level, during the boss fight.

I also used a selection of different instruments to create sound effects of monsters played at random intervals in the background of each level. BandLab Assistant has a variety of different filters and effects that can be used to alter the sound of an instrument, meaning that I was able to distort the sound of different basses and guitars to create sounds that sounded like the growling of different monsters.

I think that the using multiple programs for the development of the sound and music in my game helped me to effectively use the strengths of each program, to create audio that matched the art style and genre of Oracular.

How did you get from the concept stage to the finished product?

The original concept of my game is very different to how the final version of my game ended up. To start off with, I decided on how I wanted to interpret the theme for my game, and what genre I wanted the game to be. Once I decided on the idea of having platforms/objects in the game that can be created or destroyed, making a puzzle game quickly became the best option for my game. Once I had decided on this, I looked for a platform to develop the game on, and to test the game mechanics. I decided that GDevelop would be my best option because of how it specialised in creating two-dimensional games (which I thought would work best with my interpretation of the theme), and the number of different resources available to learn the program. Once I had decided on using GDevelop, I created a few shorts tests of different functions of the game, such as creating a moving player that could move along platforms, to test that I would realistically be able to use GDevelop for my game.

After this, I then spent time looking at example games on GDevelop, so that I could create a realistic idea of how my game could look/act. I also found tutorial videos made by GDevelop and watched them until I felt confident that I would be able to use the platform.

I then decided on a more specific concept of my game; the characters, setting and overall storyline, as well as the types of combat and puzzles the player may have to solve.

To start off with, the first part of the game I designed was what ended up being the first level of the game, rather than the opening scenes, so that I could make sure that the concept of my game would work. I designed the textures of the player character, platforms, and some of the enemies and objects used throughout the game. I also worked on basic animations of the player character and enemy and implemented both the combat health systems used into the game. In addition to this, I created the system for having platforms be destroyed/created by the player and tested having multiple scenes in the game by creating a settings menu.

Once all of the basic game mechanics worked, I was able to work on creating other more complicated levels and elements of the game, as well as creating the opening scenes, and menus throughout the game.

As I worked on the later levels of the game, I would often decide to add more controls or elements to the game. For example, when I was creating the second level of the game, I realised that the player may not be able to remember the locations of platforms they have destroyed, which led to me making it so that pressing the ‘s’ key shows the player the locations of any destroyed objects.

Creating the art for the game was one of the things I found most challenging, and as I learned of different techniques and effects, I could add to different objects I would often go back and redesign them. I spent a lot of time designing the player, as the number of different animations the player character had meant that every time I changed the appearance of the player character, I would have to change it across every frame of every animation, which I wanted to avoid doing unnecessarily.

While I created the game, along with working on the basic levels of the game, I would also work on improving the ‘feel’ of the game – the ease with which the player could use different controls, and the use of visual effects to make the game feel more responsive.

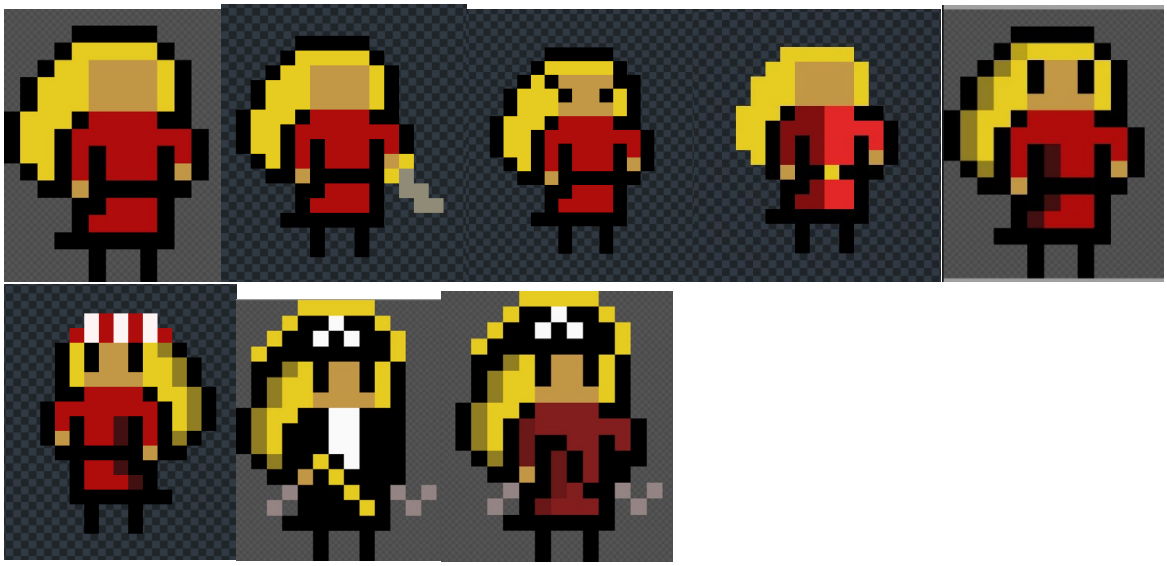
I would test each of the levels of the game many times as I worked on it, testing each of the player’s controls and testing for simple bugs; making sure that the player couldn’t get stuck or somehow escape the boundaries of the level. I also had many of my friends, and my sister play the game, and they would give me feedback on the general difficulty of the game, and any things they felt could be improved. I made sure that the game was tested thoroughly during many points during my development of the game, so that I could work on any feedback I was given as soon as possible. It was often easier to work on feedback for the game as early as possible before anything to be changed had affected multiple levels of the game.

I worked on the sound effects and music of the game last. Although the sound effects of the game do not directly affect the gameplay, they do need to be connected to the animations and visual style of the game, and as I worked to improve upon these frequently, if I had created the sound effects earlier, they would also need to be changed. The sound effects of the game help to alert the player of different things, such as the player character being damaged, or the start of the player character jumping. This means that the animations of these actions need to be tied into the sound effects played at the same time. I wanted the music of the game to link to the atmosphere of each level, meaning that I needed to wait until I had finished creating each level to do this.

Once the game had been finished, I did some final tests of the game, and worked on optimising the code; deleting unnecessary files and extensions to make the game take less time to load. I also

adjusted some final details of the game, such as the speed certain objects move at, and the timing of final effects.

Finally, I tested the different potential ways of exporting the game, so that I could find the way that worked best for the competition. To do this, I sent the game files to myself and to some of my friends and had them check that they could play the game properly. This was the final step of creating my game, and once this was done, and I had completed the Game Design Document, my mentor and I prepared to submit my entry to the Australian STEM Video Game Challenge.



Images of some of the different designs of the player character over time.

This design processed is mirrored in the designs of many smaller elements within the game. All of the photos above are different versions of the player character I created before creating the player character used in game (the bottom right photo). I wanted my character to stand out easily from the background of the game, which is why most versions of the character have a red coat. However, I later tried experimenting with different colours, such as the black, yellow, and white player character (second to last, bottom row).

As I experimented with the different styles of art I could use in the game, I started to become better at using techniques such as shading, and so many of the later versions of the player character are much more detailed, and use more shading, and finer details in the image.

Another one of the main focuses of my designs was to make it clear that the player character was a pirate. After showing one of my friends the version of the player character (furthest to the right, top row) that I had originally planned to use as the player character, they mentioned that this wasn’t very clear. This led to many of the other versions of the player character holding swords, wearing bandanas or with different colour schemes until I settled on the final version of the player character.

This is an example of the design process I went through for each of the different enemies, objects, textures, menus, and sound effects throughout the game. The player character is one of the most important objects in the game, so having a detailed design process to create it was key to having well-designed animations and actions for the player as developing the game progressed.

Reflecting

Testing, fixing and project execution

Testing

How many people tested your game?

Seven different people tested my game.

What did they need to focus on when testing?

When people tested my game, I tried to avoid giving them any help or advice, to see if they could understand the different controls and objective of the game. Because I obviously will not be able to speak to every judge/person who plays my game, this allowed me to see any aspects of the game that I needed to explain better, and to better gauge how challenging each of the levels of the game are.

Another thing they also needed to focus on was testing for any glitches. This meant that I would ask them to look at the different menus, and less important parts of the game, as well as being thorough in testing each control and every possible way of solving each level. This helped me to see any bugs within the game that I may have overlooked when I tested it myself.

Another focus of the people testing my game was the pace of my game. Sometimes, they would feel that different information was redundant, or that parts of the game dragged on. Conversely, sometimes they felt that different parts of the game were too fast-paced or confusing. This allowed me to adjust the amount of information presented to the player at different points, to make the game feel easier to follow.

Fixing

What problems were found during testing?

Many of the problems found during testing were to do with different instructions either not being clear enough, or different parts of the game being too slow/fast to follow easily. For example, one of the people testing my game felt that the text in the beginning of the game moved very slowly, making it harder to remember what had been written previously, and less engaging for the player.

There were also various glitches found during testing, including the game freezing completely during the tutorial if the ‘a’ button was pressed, or the player getting stuck in different objects. These types of problems were generally entirely reliant on mistakes in the code of the game, rather than level design.

One of the major issues found by the majority of the people testing, was the use of the ‘s’ key, which is supposed to reveal the positions of any missing objects, not being clear. This then caused them to misuse it, which mad it much harder for them to complete the game. This was a problem with level design, and while I did not change the actual use of this control, I did change how it was explained to the player throughout the game, to make it easier to understand.

How did you fix these problems?

I fixed many of the different glitches in the game through minor changes in the code, or by slightly changing the positions of different objects in each level to prevent and of the bugs previously found from occurring.

For problems related to the experience of the player, e.g., when the text was moving to slowly in the tutorial, I would add different features to mitigate them. For example, I added a way to skip text, and reduced the time text would take to appear on the screen in the tutorial level. Often, the person testing the game would have suggestions for how the problems could be improved (such as being able to skip dialogue), which I would follow when I could.

For things that the people testing didn’t understand, I would often be able to fix this by editing the wording of the instructions or adding more mentions of a certain control/function. Originally, all of the instructions given to the player were given as scrolling text that would then disappear, but I later chose to change some of this text to instructions that would stay in place on the background of the level, pointing to relevant areas. This meant that the player would not have to memorise every instruction given to them, which majorly improved the difficulty level of the game for many people.

People struggling to understand different aspects of the game is also what lead to me adding the ‘help’ button to every level of the game; an option that gives the player additional hints and information when activated. This makes the game more accessible, and encourages players to continue to play the game, as they know that they can bypass any overly challenging sections with this option.

Project execution

Did you finish your game? Were you able to include everything you planned into your submitted game?

I completely finished Oracular a few weeks before the submission date. While I had included everything I had expected to include in the game, there were certain ideas I had that I would have included if I had had more time.

For example, I originally wanted to include a store in the game, which could be visited to buy various items and power-ups with coins that I would place throughout each level of the game or be dropped from enemies. However, I felt that the game was too short for this to be a necessary feature once I had finished the main levels. If there was more time to work on the game, this would have been one of the main aspects of the game I would have added.

When I first started to develop the game, I also had no idea of how quickly I would be able to make unique levels of the game. Originally, I expected to have at least twenty different levels of the game, but I later decided that having fewer, longer levels made the levels more interesting. This lead to me only creating eleven levels of the game.

What skills or abilities do you need to learn for next time?

While I think GDevelop was a good program to use for my first attempt at programming a game, I do think that being able to use a more complicated game engine could allow me to have access to more resources and possibilities within my game. However, many of these different game engines have more complicated interfaces, require knowledge of a programming language, or even have their own programming languages that need to be used. This means that I would like to begin to learn one

of the more common programming languages (e.g., Java, C#, or C++) as well as experimenting with using different game platforms.

I could also do the same as this with different Digital Audio Workstations to design the audio of my game, and different graphic design tools for the art within my game. While I think all of the programs I used in the design of my game worked well, as I learned more about pixel art, developing the audio of my game and game design, I think I would be able to use more specialised programs to expand my capabilities and challenge myself.

As this was my first attempt at pixel art, audio and sound design, coding and gameplay design continuing to practice all of these skills will allow me to continue to improve for the future, meaning that I will likely continue to practice making games.

What worked well?

I think that I was able to learn all of the skills I needed for creating Oracular quickly through the use of different resources offered by GDevelop. GDevelop has a YouTube channel, where there are many different video tutorials explaining the functions of the platform, and many of the skills and functions commonly used in developing a game. This helped me when I was first beginning to program the game, when I struggled with understanding how some different functions of the platform work (e.g., timers and variables, or the scene editor used to edit the appearance of each scene).

I also was able to mostly use the programs integrated within GDevelop in the creation of my game, which allowed me to save time on searching for the programs necessary. Piskel, which I used for the design of every object in the game, and JFXR, which I used for many of the sound effects in Oracular. This allowed me to move from one facet of the development of Oracular to another much faster than I would have otherwise.

Overall, my original expectations of how developing the game would work, and what I had hoped the end result of my game would be ended up being quite realistic. This is due to the detailed planning and testing I did to properly understand the full potential of the programs I was using, and the extent of the skills I had.

What would you do differently next time?

Next time, I think I would slightly rearrange the order that I developed different parts of the game. When I created Oracular, I left all of the sound effects and music to be designed after I had finished programming and creating all of the art of the game. This left me with less time to create the sound effects of the game than I would have had otherwise, and also meant that I had to look through the whole game again to remember what objects and sections of the game needed sound effects or music, when it would have been faster to do while I was still designing the game.

I also think that I could have left more time for testing the game, as although I was able to test the game many times, other people testing the game took a lot longer than I had originally expected to complete the game. This meant that they often weren’t able to test the game as thoroughly as I had hoped, and that I didn’t have as much time to change the game to follow their feedback.

The skills I have learnt from developing Oracular means that I think I would be able to create a game at a much faster pace than I did with Oracular, meaning that I would have more time to experiment and explore different opportunities in any future competitions.