**Testing strategy**

**Evaluation:**

There are several different methods that I ca use to collect data on the way that not only my game preforms, but also the way that the player reacts to it, both positive and negative. These different methods are as listed below:

* ALPHA testing
* BETA Testing
* Accessibility testing
* Functionality testing (Black Box Testing)
* Stress (Load) testing (White Box Testing)
* Compatibility testing

There are several reason for why I have chosen these methods of testing and why they are so important. When it comes to building and evaluating a game, extensive testing must be carried out to ensure that the game is ready to be released and played by its intended audience, it must also be able to accommodate for as many people as possible, so in by completing many different types of testing, a large ground can be covered in terms of identifying bugs and faults with the game.

It is also important to carry out, ALPHA, BETA and PLAY testing of the game (some of the most basic and compulsory testing). Alpha testing would be needed and carried out after creating the very basic level designs and functions. This would mainly to test to see if the coding so far has been created and also to find any issues with sprites/assets that have been implemented into the game.

I plan to complete the APLHA testing for my game, once I have created a very basic layout of both my levels and also have the playable character implemented into the game with very basic physics and a very basic and rough example of what the UI will look like for the game. I also plan to have only a couple screens working for this testing stage, i.e. The main menu screen, the actual game screen and the help screen.

The results that I will gather from the ALPHA testing stage will help indicate to me, from an early start, where I am going right and where I am going wrong, in terms of player interaction with everything in the game at a very basic level to make sure that my coding logic is correct. By doing this at an early stage and gaining these results, it should expose faults that I can rectify quickly and stop me from having so many in the BETA testing.

BETA testing would be completed further down the production stage, once the full game has reached a point where it would need polished. Beta testing is mainly used to find bugs in the game when the player interacts with the world around them or even when just running the game in general and the coding logic can’t be handled by the program and crashes.

BETA testing does not only include this basic testing, it also includes accessibility testing, functionality testing, stress testing and compatibility testing:

**Accessibility testing:**

Accessibility testing tests the ability of a game to be “accessible” for players. This mean that it tests the game to see if it can be played by people who have certain limitations that would otherwise restrict them from playing some games.

In my game, I would test to see if my game can be played by people who are colour-blind. This would mean that I would have to test for all the different types of colour-blindness that there is with the colour graphics on my game to make sure that it is actually playable for people who suffer from this condition.

This testing doesn’t only apply to this condition either, it can also apply to people who are either partially deaf or fully deaf. I will need to test my game with no audio to make sure that people who also suffer/struggle with this disability have a fair chance at playing the game, and not have the whole experience ruined by their disability.

The accessibility testing will also apply to players that may not speak or understand English, as we live in such a diverse country not, there will be a likely chance that there will be a child that will play the game that my struggle with understanding English (this can also apply to children who can speak English who aren’t foreign but just struggle to read), therefore I will need to make my game accessible to these players too and test this too see if my game does actually support their need when playing the game.

If when looking at these results, it comes back that my game was able to support the needs to the players that interact with my game, then it would be a success, however, if It came back that some players were struggling either a little or a lot with the game due to certain reasons, I would have to try and implement a way in which to support their needs to make it a positive gaming experience for everyone.

**Functionality testing:**

Functionality testing is a big part of any type of game testing as it is the type of testing that looks for bugs or errors in either code or map design. This can range from, again, simple logic errors that the program cannot complete or even asset clipping, in terms of clipping into walls and not being able to move or assets just not working at all. Without this type of testing being done, if the game was released, it would just be very buggy and give it a bad reputation; for example, the game PUBG is a very poorly optimised game for the pc, filled with bugs and errors in code, as the game became popular, it was rushed in being finished and also released unfinished to a degree. It started to appear that there wasn’t a lot of functionality testing taking place as players were running into huge issue and errors in the game that were not being addressed or just could have been fixed before launch, causing the reviews of the game to dip massively and end up having a poor representation. (1) (2)

To avoid this type of issue occurring with my game (even though it is in a much smaller scale) I will do extensive functionality testing to make sure that my game does not follow suit and will make sure that any issues that do arise, I will attempt to fix fully to make sure that it does not affect the players experience.

**Stress testing:**

Stress testing is also as important as functionality testing. Stress testing tests the games limits and attempts to crash the game. This not only tests the stability of the game, but it also tests the limits of the hardware available.

From the brief, I already know that I have a limited amount of storage and RAM to work with when it comes to the game. Therefore, I am going to have to make the game so that it does not exceed these set boundaries, Stress testing will allow me to see if I have manage to met the requirements by pushing it to the limit to see how much RAM it would actually end up using when being played at many different levels and also if it works well with other applications running in the background.

This is crucial as if I had created the game and not tested this, I could end up launching the a game and it would just crash or run really slow just because of the fact that there is not enough storage or RAM available to actually run the game, causing the player to have a negative experience and make the game unappealing not only to the player, but also to the client as well.

**Compatibility testing:**

Compatibility testing tests the game to see if it works well with the controls that are going to be used to actually play the game. Meaning it will also test to see if it is optimised well for the controls to work.

For my game, the testing will look at the way that the player character is controlled (Joystick) and test the coding for the controls to see if it responds well when being used. This means that the player moving in the direction that the joystick is being used and also quick movement changes as well.

This can also be said for the buttons as well, the buttons will also be tested to see if the command carried out the desired action that they are made for, for example, the button for jump will be tested to male sure that the player does indeed jump, it will also test for lag between the button press this will be the same for the crouch button as well.

These controls will not only be tested for in game too, but also for out of the game as in, menu navigation as well.

I plan to compete the BETA testing for my game once I have implemented all the different assets with their animations into the game, once have also added the whole level design with graphics an added the UI for the game and all the different screens working and all being linked together.

The results that I will gain from the beta will be crucial in telling me what I have done correct so far and what will need fixed, this testing will be very similar to the APLHA testing but the results from this test may be a little more “nit-picky” rather than being a major issue. I will have to look at the way that the game played, both with game physics and also asset interaction. I will also have to look at different issues that may have occurred, as in coding logic not being correct and also the way that the player manoeuvres around the map, changing heights or lengths of platforms or the distance of the enemies’ patrol, or the sizes of the UI on the screen or even placement. As stated above in the different methods of testing. (3)

**Evaluation Documents:**

Below I have created a survey that can be filled out once the player has played the game. This is to gain valuable feedback so that I can improve upon what I have already built: (4)

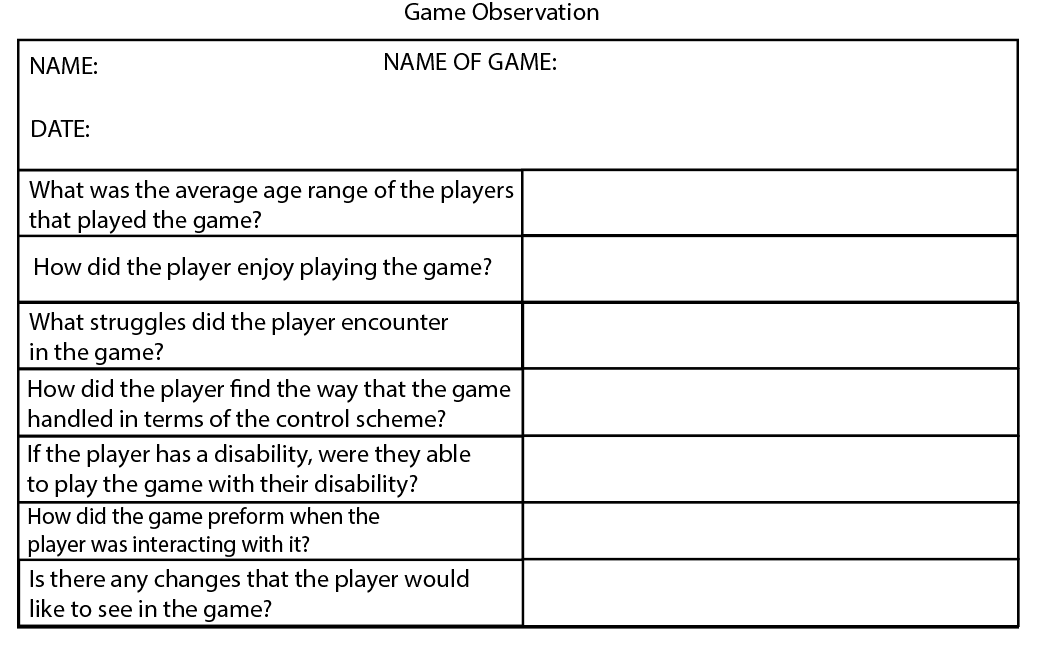
Surveys are a very common form of gaining back feedback from users or specific target demographic for products and services. These are a good way of collecting all different kinds of information from your target demographic (in my case, kids and young teens, aged 5-15).

The different information that can be collected from the survey can range from the way that the person interacts with the game, to their first impressions of the game as well. This helps to give me a large amount of different feedback that I can then use to improve my game to not only meet the brief, but also make it fun to play.

Surveys are also built to be very short, only having about 6-10 questions that only take maximum of 5 minutes to complete. The main reason for this is that when having people complete the survey, you want to have them stay interested and motivated to finish it so that they don’t get bored and start giving short, blunt or even half answers, which can affect the accuracy of the actual feedback that you could get in a negative way.

I have personally used surveys before in the past for multiple different projects and they have worked really well in gathering feedback, therefore I fell that it would be very suitable for this project as well. I have also created the survey through the website “Survey Monkey” (5), this website is really reliable and great for creating simple surveys. This website also allows for different responses for the questions on the survey to help give a more accurate response too.

I have also created an observation document that can be seen below:



The reason for why I have also created an observation sheet, as well as a survey to complete, is to try and get different variations of feedback from different people so that I have as much feedback as possible.

Observation forms are a great way to have a 3rd party’s opinion on how they think that the game went as well as including the players opinion as well. For example, when the player is playing the game, they will be engrossed and not playing attention to what is going on around them, they will be mainly just focusing on their character. I also know this from past experience when using these as I have used them before in college for anther game project where I personally observed and game being played. I found this method very helpful and a really good way to get feedback and info about the game that I was testing/observing.

However, if another person is watching the game that is being played, they may be a little less focused on what the character is doing and will not only be able to see what the player is doing on screen, but also how they are using the controls so that they can see if the player is truly struggling or not. This is important as some players may feel that if they say that they have struggled to play it or find the controls difficult, they will be seen as bad at playing the game and therefore may not actually tell the truth on whether they found it easy or not and having that third-party input can help show the possible correct feedback that I will need.

I have based some of the questions in the observation sheet off of the survey, again, to help gather the two different views that the people completing these forms may have, and also as I have said I have also added some other questions in the observation from to help give me the most broad and accurate results that I can get.

**Testing Features:**

Functionality:

* Game counts the score of the player when they collect items in the game
* Player interaction works – player collecting treasure, player dying to enemies or taking damage from enemies when touched.
* Timer counts down when the game starts – finishes the game when it reaches 0
* The navigation of the main menu screen
* The text box to input the name of the player on the game over screen
* The leader board screen showing the different names of the players and their highest scores in the game.

Usability:

* The way that the player controls the character i.e. with joystick and buttons
* Using the controls to input your name into the leader board textbox
* Using the controls to navigate through the menus
* Seeing if people with certain disabilities can play the game i.e. people who are colour-blind or even people who are deaf

Performance:

* Smooth transitions between the different menu screens
* Small loading times between loading menu screens and even loading the game
* No lag, i.e. high frame rate
* Good response time between activating certain buttons or movements of the joystick to it appearing on screen
* All assets loading correctly and all assets heights and sizes correct to the game ratio.
* How well the game is received – Replay ability

**Functionality Feature Testing:**

When it comes to testing the functionality features of a game, specifically my game, I will need to have certain actions take place so that the criteria are met so that I can test for the certain features that are listed above.

When it comes to counting the score for the player there are multiple ways on which I can do this. I can simply pick up one piece of treasure and it if adds to the overall score then that means that it is working. However, I can also collect more pieces of treasure to see if it will correctly calculate the score depending on how much treasure I have collected. I can also complete the game to see if the score will be correctly calculated when the time and score are added together to give the total final score that will be used on the leader board.

Testing for the player interaction can also be partly done at the same time as the score testing. This will mean that I will test to see if the game will register that the item or treasure has been picked up. I will also test to see if the player will take damage and can also die to an enemy, best way to do this is to walk right into an enemy and see if they do indeed damage me and also end the game if my health is too low.

The test for the timer is a very simple one. I will start the game and if the timer starts counting down from the second that I start to play, then it will mean that the timer is working.

Testing the text box for functionality would include, typing a name into the box and submitting the name, then heading to the leader board and checking to see if not only the program has stored the name but it also displays it as well.

Whilst testing this, I can also test to see if navigating between the different menus is easy and works. I would do this by simply attempting to go to different pages or menus within the game and being able to go back to the one that I was previously on.

**Usability Feature Testing:**

When testing the usability of my game, I will be looking at the way in which the player interacts with the game and if the player actually is able to interact with the game in the intended way.

One of the main tests that will need to be carried out for this is the controls to not only move the player, but also to navigate the screens. I will be testing to see if the user is able to control the character and navigate the menus with ease. This will mean that they can move the joystick and press the buttons necessary to be able to complete certain actions/movements to get to certain platforms within the game.

As with the others, I can check too see if the player can easily type into the textbox at the end of the game that will allow them to input their name into the textbox. I want to make it as easy and as quick as possible so that it allows for more people to play the game in a shorter period of time.

I will also again have to test to see if the player can navigate the menus easily, this time round its less of if it will actually work and more about if the player, using the controls, can easily move between each screen and also understands how to do that.

For the last test I will need to test if my game is suitable for people with disabilities. Therefore, there are two ways that I can test this. Have someone who has one or both of these disabilities and have them play the game, or I can have a person play the game with the sound off. I will need to have someone play the game who is colour-blind as I cannot test to see if it is playable as I am not.

**Performance Feature Testing:**

Linking to the other tests that would have been already carried out, I will have to look at how the game actually runs, in terms of not only speed but also quality. By doing this, I can make sure that my game looks finished and looks like a professional product, this also stops the possibility of players being put off playing it if it looks bad.

I will need to test both the transition speed of the menus in the game and also the smoothness of those transitions as well. This means that I will have to constantly switch between different menus to gauge whether there is any “Screen Tare” or “Lag” when swapping between them.

Going on from “lag” I will also have to test the frame rate of the game when it is being played, so it too will not encounter “lag” and make it harder and a less pleasant experience for the player. The main way I’m going to have to test this is juts by playing the game and loading a lot of assets into one area to see how it will affect the overall frame rate.

This again will also apply to testing the input lag of the controls for the game. I will have to test all the different functions that require a button press or a joystick movement, this will both include the main game itself, the menus and also the input for the leader board name too.

When building the sprites and creating the map, I will have to use measurements so that I can make the assets all the correct size in relation to the map size. However, if I get these measurements wrong, I can see this in this method of testing and rectify it before release. The main way that I will test for this is by trying to move the sprite around the map and if it gets stuck on certain objects, I can edit it to fit.

This one, is more observation, however, this will also count towards the overall performance of the game as in the fact that I will be looking at how the game, as a whole, preformed when being played and also the impression that it also left. This will allow me to see what I can change to make the game more appealing for others to play.

Replay ability is a factor to consider when testing for the performance of my game. My game should be able to be played multiple times without the player becoming bored. The way in which I will have to test for this will be that I will have to have a player play the game and attempt to complete it, I will then have that player take a break, either playing a different game or reading etc, I will the have them have another shot at my game and record their review of the game.

**Testing Plan:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of testing** | **What’s being tested** | **Input Data & Test Step** | **Expected Results** |
| Functionality | Game adds score | Add 1 to score when “treasure” is picked up. | Add 1 to the score |
| Functionality | Game adds score | Adding the time left to the score on top of original score. Complete game | Adds the time left to the original score and calculates total score |
| Functionality | Navigation of the menu screens | Selecting the Nav buttons on the screen and move between different screens | When clicking the nav button, move to the correct screen |
| Functionality | Player interaction | Player will connect with a skeleton and lose some health | Player will lose health when they come into contact |
| Functionality | Player interaction | Player will connect with a skeleton and die when on last life | Player will die on contact with skeleton |
| Usability | Player moves | Joystick moves left and character on screen moves left | Character moves left on screen without input lag |
| Usability | Player Jump | Button pressed and player jumps | Character jumps on screen without any input lag |
| Usability | Game accessibility | Change the colour scheme to suit the individual that is colour-blind | Player is able to play the game and enjoy it, seeing the colours that I’ve intended for the player to see |
| Usability | Game accessibility | Turn the game audio off when playing the game to see if game can be played by someone who’s deaf | Player will be able to play the game without the audio and not have a different experience from others. |
| Performance | Loading assets | Allow the program to access the file that contains the assets for the game and load one into the game scene. | The sprite is loaded into the game scene correctly, meaning that the size and height of the sprite is correct according to the size of the scene. |
| Performance | Loading assets | Allow the program to access the file that contains the assets for the game and load all assets into the game scene. | All sprites are loaded into the game correctly meaning that they are in the correct place and the correct size. |
| Performance | Loading times | Test the overall gameplay of the game and time the loading times for the game to see how long they take. | There will be a very small loading time between the game launching and loading different menu screens. |
| Corner Case | The hardware for the system that the games running on may be upgraded or downgraded. | Upgrading the hardware of the system | Being able to add more levels and assets into the game to flesh it out a bit more |
| Corner Case | The hardware for the system that the games running on may be upgraded or downgraded. | Downgrading the hardware of the system | Having to remove certain assets for abilities within the game so that it will meet the hardware requirements. |

References:

1. <https://store.steampowered.com/app/578080/PLAYERUNKNOWNS_BATTLEGROUNDS/>
2. <https://store.steampowered.com/app/578080/PLAYERUNKNOWNS_BATTLEGROUNDS/#app_reviews_hash>
3. <https://www.testbytes.net/blog/game-testing-methodology/>
4. <https://www.surveymonkey.co.uk/r/KRP3N3Q>
5. <https://www.surveymonkey.com>