Earth Shard
Test Strategy

MALEVOLENT
CROW

Version 3.0

1 CONTENTS

2 TESTS STRATEGY

I will be using a variety of testing methodologies, white box testing, grey box testing, black box testing/ play testing. For the white box tests, I will be using the following techniques: unit testing, integration testing, system testing along with static analysis and dynamic analysis, Ad-hoc techniques will be used too. Grey box tests will involve having other members of the class test parts of the earth shard to see if a fresh eye can identify issues I have not seen. The methods used by the grey box testers will most likely be ad-hoc tests, but if they wish to use a more formal testing strategy for deeper analysis that will be acceptable. For the black box tests, I will be using alpha and beta tests with third party participants by having them perform playtests as well as having internal playtests. For the internal playtests I will observe the third-party participants perform playtests without input unless they encounter a significant bug that requires developer input. For the alpha and beta tests I will distribute builds to participants and give them a survey to fill out after completing the build of the game.

3 JUSTIFICATION FOR CHOICES

The methodologies I have chosen to use are tried and true in the game development and programming field.

Whitebox testing allows me, the developer to have a clear understanding of what is going on within the program. The techniques I am using allow me to test from the smallest part of the program (unit testing) to testing the entire game (system testing) to find bugs or problems. Ad-hoc testing while not being a documented form of testing will allow me to quickly identify issues before they appear in any formal testing processes. Static and dynamic analysis also benefits ad-hoc testing.

Grey box testing is a more informal form of testing I'll employ when I am struggling with bugs that I can't quite crack. Having other members of the class make observations or test parts of my program can greatly benefit as other members of the class have different skills sets and experiences that could help identify issues I have not previously encountered. this has been incredibly beneficial in the past and should greatly benefit this project.

Black box testing/play testing is very valuable in catching stray bugs and improving gameplay that is not engaging the players. Using internal playtests and taking notes while observing without giving user input (unless required) will allow me to see where players maybe getting confused or stuck with features or the levels themselves. Also having the internal testers speak their mind while testing will help me understand what players are thinking. This is very beneficial as when earth shard is completed their won't always be a developer telling the player what to and what not to do so allowing testers to speak their mind while playing will allow me to create a more intuitive experience and address common problems before they occur. The alpha and beta tests will be done more remotely and will be on version of the game that are more complete. These tests will have users fill out surveys after playing and give feedback in the form of rating different parts of the game and giving any recommendations they or changes they feel could be added. Player feedback is incredibly valuable as this is a product for them.

4 When tests will take place

White box testing methodology and techniques are being used at the end of every feature that is considered complete to ensure it is functioning. Then at the end of a sprint a wider system test will be performed on the build of that sprint. After the second sprint is complete is when play tests will begin as the version of this game will be in a more complete and testable state compared to the previous build. The third sprint focuses on polish and testing earth shard and will involve the distribution of alpha then beta builds of earth shard to gather wider user feedback.

5 Documenting tests

The formal white box tests will be documented through use case tables and compiled into a large testing document. The tests will also be updated on the Gantt chat and if any further tests are needed the Gantt chart will be updated as necessary. As for the playtest results these will be automatically compiled by google forms which will be able to show a wide range of data with its visualisation tools.

Github is a tool I am using to track development and through the Github issue tracker I can document bugs I encounter and add them to the development boards for easy tracking.

5.1 Sample use case tables

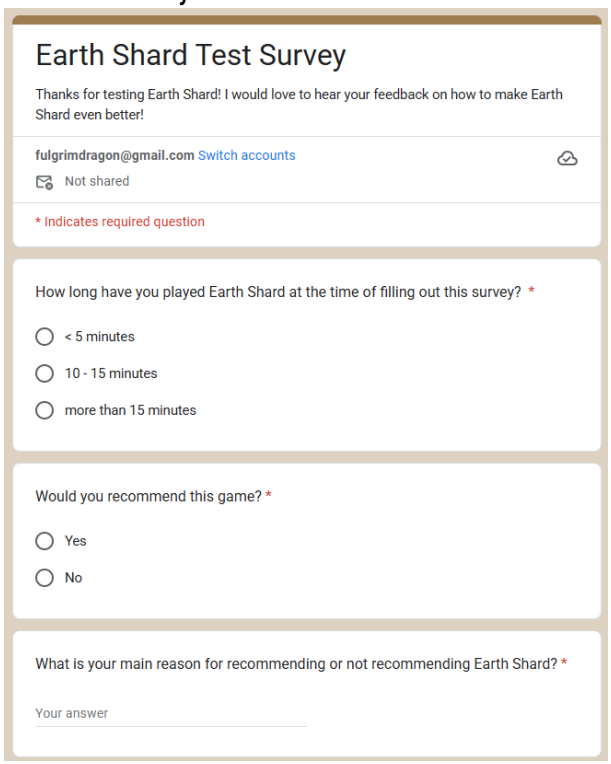
These tests are only hypothetical and do not represent any actual tests that have been conducted in earth shard.

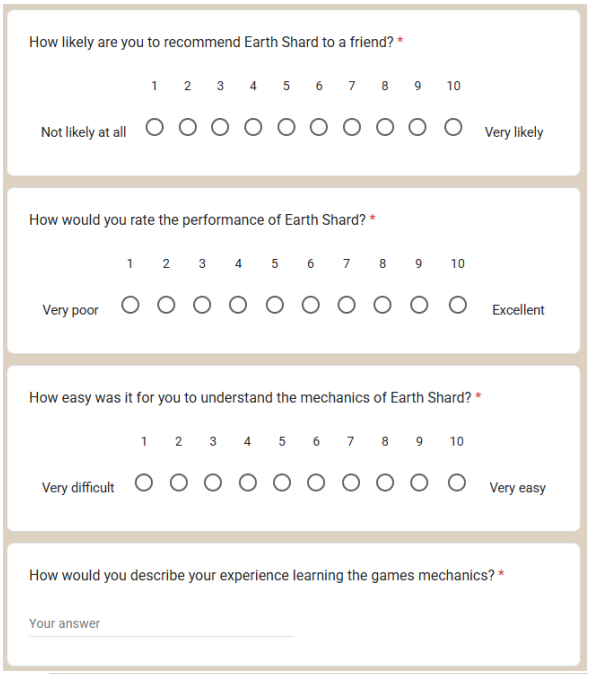
Integration Tests				
Test No.	Description of Test	Expected Result	Actual Result	Revisions
1	collision detection between player projectile and switch	Console should output “collision detected” when a collision occurs.	Console only output collision half of the time.	Changed collider to dynamic collisions.
2	Enemy death counter opens door at 10 dead enemies	Door should open once all enemies in zone are killed.	Door opened when all enemies died.	None required.

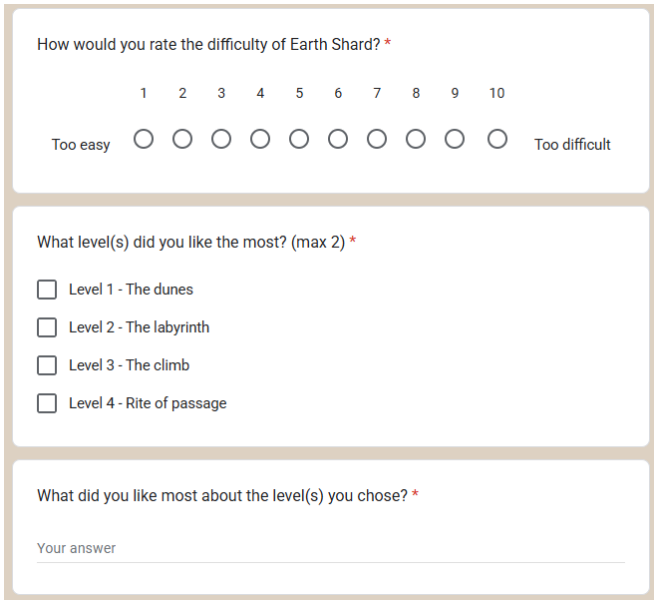
5.2 Sample survey

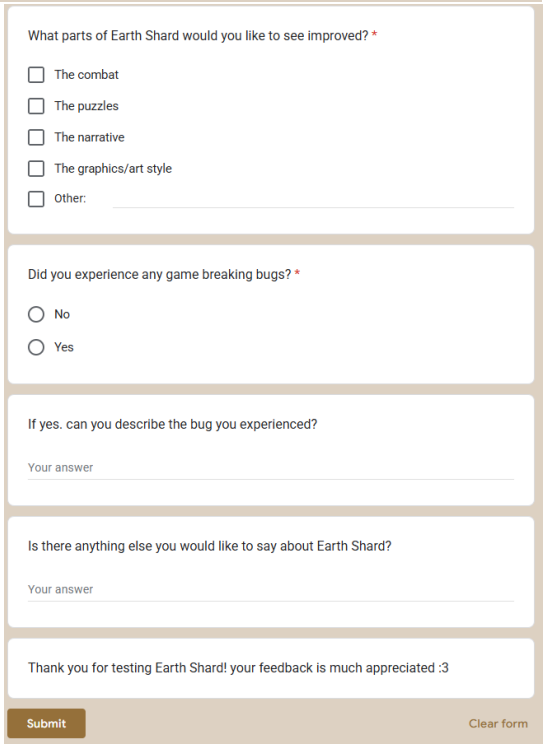
Survey link Here!

5.2.1 Survey Screenshots









6 RESOURCES

- PC with keyboard and mouse
- In house testers
- Google forms
- Third party testers
- Microsoft word

- Internet access
- Build of earth shard
- Unity
- github