

Software Implementation and Testing Document

For

Group <3>

Version 1.0

Authors:

Grace B

Elias D

Jason G

Miguel M

Santiago R

1. Programming Languages (5 points)

The programming language we are using in the project is C#. The reason we are using C# is because the game engine utilizes C# scripts in order to produce different actions and events that the user may want. C# is used in all occurrences for scripts.

We also use the HLSL and Shaderlab code generated by TextMeshPro.

2. Platforms, APIs, Databases, and other technologies used (5 points)

TextMeshPro on menus to have the resolution of text scale to different screens and look cleaner than basic Unity text.

Other Unity Store packs (primarily for visual assets and FX). These will be used throughout the entirety of the project.

3. Execution-based Functional Testing (10 points)

*Describe how/if you performed functional testing for your project (i.e., tested for the **functional requirements** listed in your RD).*

1. Post Processing Effects
 - Tested on 3 machines of varying power to make sure they were not bottlenecking the performance
2. Surfaces Affecting Player Motion
 - This was tested by experimenting with the different parameters (ex: cooldown, speed, etc) for each surface that affects player movement.
 - Each surface was tested and modified to make the “feel” seem more natural. These features include: the unidirectional and multidirectional slowdown, unidirectional and multidirectional speedboost, and the two different types of jump boosts.
 - For each surface, I would run the player through and test WASD and jump motion.
 - I also had other people who were not working on this project experiment with the surfaces that affect player movement.
3. Dialogue System
 - This was tested by creating sample dialogue objects then playing the game in the dialogue scene to test the dialogue.
 - I made sure to implement various forms of punctuation and dialogue of varying lengths to ensure that the different features that were implemented during this increment were all properly and sufficiently tested.
4. Teleportation
 - This feature was tested by having the player run through the teleport from different angles and speeds.
 - The feature was also tested to make sure that if the player jumped above the collision range that they would not receive the teleportation.
 - The teleportation feature was also tested by moving the target around the world to ensure that the player was teleporting to the correct location each time.
5. Damage

- This feature was tested by having multiple enemies on the testing scene and debugging the amount of damage they would produce, until it was fine tuned to a comfortable value
- I added and removed enemies to make the feel correct
- 6. Enemy Movement
 - This feature was tested by placing obstacles in front of the enemy's patrol points and seeing how it would react.
 - I would add tall objects and wide objects and make sure the enemy could go from patrol point to patrol point without getting stuck.
 - I also tested this when the enemy is chasing the player. I went around walls and over walls and the enemy was able to stay on track and continue chasing.
- 7. Spotlight Search
 - This feature was tested by alerting the enemy and making sure that their spotlight would turn red and I would go a certain distance away from the enemy and make sure that when it went back to patrolling it would turn white again.
 - This was a simple test that made sure the spotlight of the enemy would change colors correctly and that it had a good range / aesthetic appeal.
- 8. Waypoint Marker
 - Tested by moving the player camera through all possible angles, ensuring that the marker can be seen at the position of its attached object and that the marker does not move off of the screen.
 - Distance tracker was tested by moving towards and away from multiple goals.
- 9. Objective List
 - Tested by clicking on the object with the attached script
 - Tested distance at which object can be interacted with to ensure that the player cannot click on the object from across the map
 - Ensured that each object can only be interacted with once, so that the max number of objectives reached cannot exceed the number of objectives that exist
- 10. Level Selector
 - Tested by loading each level on the menu.
 - Tested loading a level then quitting to main menu then loading again.
- 11. Save Game/Load Game
 - Tested by loading and saving, saves level and position.
 - Tested distance quitting the game completely and the load still works.
- 12. Pause menu
 - Tested pause menu in all scenes.

4. Execution-based Non-Functional Testing (10 points)

*Describe how/if you performed non-functional testing for your project (i.e., tested for the **non-functional requirements** listed in your RD).*

Tested prototyped level by having a few people run through it and tuned to keep an even difficulty throughout.

To test the “feel” of the surfaces that affect player movement, I reached out to some friends and acquaintances to test out the features for me. I had 5 of them use my laptop and see how they felt about the different surfaces and player movement. Overall, they were pleased with the different features and

appreciated the variety in these surfaces. Particularly, they were most interested in the teleportation feature, which is something that I enjoyed implementing in this project. In this increment, I made sure to implement feedback that I received. For instance, in the last last increment, I wrote about how the jump boost was not intuitive, so I added in a more intuitive jump boost that applies to all players, regardless of if they are jumping or not.

5. Non-Execution-based Testing (10 points)

We reviewed each other's work visually through the Unity game engine and suggested ways to improve the project. Just anything to help each other was our form of a demo/inspection. Any impediments or blockages were handled through the discord channel to give everyone on the team an understanding of what the other person was working on and how they were able to fix it / what they needed to fix it.