Software Implementation and Testing Document

For

Group 16

Version 1.0

Authors:

Kevin Sturge

Robert Romero

Valeria Torres

Asher Adams

1. Programming Languages (5 points)

We used GDScript, this is the language used in the Godot engine.

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

Platforms: Godot, Github, Trello

Technologies: Git

We program and test everything in Godot and do version control and collaboration in Github and trello. Trello has our kanban board and information on each card.

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).*

Functional testing was performed to the best of our ability by running the program in isolated scenes depending on which element was being testedand playtesting to ensure implementations are working correctly with no chance for softlocks, bugs, etc. The game loop has not been implemented yet as we have not finished sufficient game systems, but on an individual system level, each system is tested.

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).*

Performance and user experience is a critical part of a fun and enjoyable game, which is why it must be a priority. Performance is kept in mind during development implicitly, but when systems merge, it can hard to tell how performance is impacted. After every change and newly implemented feature, the performance and user experience is gauged at an individual level.

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

Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).

At various points during development, we have done code reviews where we go over all the recent changes to the project and make sure that the new systems work with the existing codebase. This is important when merging pushes to make sure that the code is compatible even without running the code. And it avoids code silos.