“QWERTY Committee” - Test Plan and Results for *Project Arizona*

Unity Game Development Project by Jacob Morgan, Thomas Meyers, and Aidan Sorensen

I. Overall Test Plan

Our approach to testing our project game will consist of several different methodologies. First, we will test our game as we develop it. Due to the nature of game development in general, and especially in the Unity engine editor, games can be tested more readily and freely throughout the development process. After a function has been added or a variable has been tweaked, the game can quickly be executed in a given state, and variables or objects can be tweaked dynamically during this process to test various different outcomes. Since the whole point of games is to play them, it only makes sense to *play around* with it to make sure that it works correctly. Next, we will test the game more systematically with simulated data to ensure that all functionality handles exceptions accordingly under normal and abnormal conditions, and that abnormal conditions either cannot be reached by the player or do not break or crash the game. Lastly, we will playtest the finished, production game ourselves and have friends, family, or other students playtest it to give us feedback and potentially find additional bugs that we might not have foreseen.

II. Test Case Descriptions

DT1.1	**Development Testing 1**
DT1.2	Testing done throughout development of game
DT1.3	Run/Execute early game states as code/parameters are added or changed
DT1.4	Inputs: Test states, reasonably normal variables, user controls
DT1.5	Outputs: Functions may not work as intended or present conflicting outcomes
DT1.6	Normal
DT1.7	Blackbox
DT1.8	Functional
DT1.9	Integration (whole process)
DT2.1	**Development Testing 2**
DT2.2	Testing late in development to further stress-test code integrity
DT2.3	Run/Execute game in abnormal or specific conditions
DT2.4	Inputs: Abnormal states, variables, or parameters
DT2.5	Outputs: Game should not crash, show breaking errors, or lose performance
DT2.6	Abnormal
DT2.7	Whitebox
DT2.8	Performance
DT2.9	Unit testing
ST3.1	**Self Testing**
ST3.2	Developer self-playtesting
ST3.3	The dev team will playtest the game ourselves and try to break the game or find

any signs of abnormality in the finished states of the game as well as tweak the game to perform and feel how we want it to in terms of player control feedback, accessibility, and usability and performance of the game

ST3.4 Inputs: Normal controller inputs on various consumer hardware

ST3.5 Outputs: Game conditions and play feedback/responsiveness

ST3.6 Normal

ST3.7 Whitebox

ST3.8 Functional

ST3.9 Integration

PT4.1 **Player Testing**

PT4.2 Testing done by players outside of the dev team

PT4.3 We will allow friends, family, or other students to play our game with little to no outside influence or intervention to see how they react or respond to the game without any previous background information, help, or bias from the dev team.

PT4.4 Inputs: Normal controller inputs on various consumer hardware

PT4.5 Outputs: Game conditions and play feedback/responsiveness

PT4.6 Normal

PT4.7 Blackbox

PT4.8 Performance

PT4.9 Integration

III. Overall Test Case Matrix

Test Case ID	Normal/Abnormal	Black/Whitebox	Funct./Perform.	Unit/Integration
1	Normal	Blackbox	Functional	Integration
2	Abnormal	Whitebox	Performance	Unit
3	Normal	Whitebox	Functional	Integration
4	Normal	Blackbox	Performance	Integration