*Software Testing*

# *Test Planning Project*

*Submitted*

*In partial fulfillment*

*For the award of the Degree of*

*Bachelor of Science*

*in Computing in Software Development (year 3)*

**

*Submitted By:*

*Muhammad Noman Junaid (G00351754)*

*GitHub Repository Link =* [*https://github.com/Nomijee/Software-Test-Planning-Project*](https://github.com/Nomijee/Software-Test-Planning-Project)

*Department of Computer Science*

*Table of Contents*

[*Test Planning Project 1*](#_Toc40379536)

[*Introduction 4*](#_Toc40379537)

[*OBJECTIVES AND TASKS 4*](#_Toc40379538)

[*SCOPE 4*](#_Toc40379539)

[*Testing Strategy 4*](#_Toc40379540)

[*Unit Testing 5*](#_Toc40379541)

[*Test 1: 5*](#_Toc40379542)

[*TEST 2 5*](#_Toc40379543)

[*TEST 3 5*](#_Toc40379544)

[*TEST 4 6*](#_Toc40379545)

[*Test 5 6*](#_Toc40379546)

[*TEST 6 6*](#_Toc40379547)

[*TEST 7 6*](#_Toc40379548)

[*Pause Menu Unit Tests: 7*](#_Toc40379549)

[*TEST 1 7*](#_Toc40379550)

[*TEST 2 7*](#_Toc40379551)

[*Test 3 7*](#_Toc40379552)

[*Gameplay Unit Tests: 8*](#_Toc40379553)

[*TEST 1 8*](#_Toc40379554)

[*TEST 2 8*](#_Toc40379555)

[*TEST 3 8*](#_Toc40379556)

[*System and Integration Testing 9*](#_Toc40379557)

[*Integration Testing Definition: 9*](#_Toc40379558)

[*Methodology: 9*](#_Toc40379559)

[*System Testing Definition: 9*](#_Toc40379560)

[*Methodology: 9*](#_Toc40379561)

[*Performance and Stress Testing 10*](#_Toc40379562)

[*Performance Testing Definition: 10*](#_Toc40379563)

[*Stress Testing Definition: 10*](#_Toc40379564)

[*Methodology: 10*](#_Toc40379565)

[*User Acceptance Testing 11*](#_Toc40379566)

[*Batch Testing 11*](#_Toc40379567)

[*Automated Regression Testing 11*](#_Toc40379568)

[*Beta Testing 11*](#_Toc40379569)

[*Test Schedule 11*](#_Toc40379570)

[*Control Procedures 11*](#_Toc40379571)

[*Features to Be Tested 11*](#_Toc40379572)

[*Features Not to Be Tested 11*](#_Toc40379573)

[*Resources/Roles & Responsibilities 11*](#_Toc40379574)

[*Schedules 11*](#_Toc40379575)

[*Risks/Assumptions 11*](#_Toc40379576)

[*Tools 11*](#_Toc40379577)

# *Introduction*

*This game will be a 2D side-scrolling platformer, inspired by the likes of ‘Salt and Sanctuary’, ‘Shovel Knight’, and ‘Fancy Pants’, with elements from ‘Skyrim’ (mainly in the way the player character and enemy characters attack). The artwork will be inspired mainly by Shovel Knight, which uses mainly pixel art to create its characters and world. The gameplay will be inspired by ‘Salt and Sanctuary’ and ‘Dark Souls’ and ‘Skyrim’, which will see the player navigate progressively difficult levels with a wizard type character that uses magic a lá ‘Skyrim’. Each level will have several enemies that the player must defeat to progress. Each level will also have a boss that the player must defeat to progress to the next level. Each level will contain pickups for the player, such as health pickups to replenish the player’s health.*

# *OBJECTIVES AND TASKS*

*The main aim of game testing is to find bugs in the game software. When a game is developed, the development lead and the QA lead develop test cases. The game testers then execute test cases to find bugs. They report these bugs to the developers, who then fix the bugs.*

# *SCOPE*

*Scope of game testing is quite narrow, but they are extensive individually. The game testing involves testing whether the game has the desired gameplay and functions (health pickups, control mechanism, levels, load and save game etc.), whether it works properly on all the devices it is intended to without crashing.*

# *Testing Strategy*

*In a simplistic view, testing is to identify bugs found in the software, so the problem can be removed. There are different forms of tests and testing that can be categorized as “Black-Box” testing and “Clear-Box” testing (“Clear Box” testing is also known as “White-Box” testing in the software industry).*

## *Unit Testing*

*Test 1:*

*The "Play Game" button will be clicked by the tester and they will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the Game has loaded appropriately.*

*Expected Results: This test will show the Game’s first level on the screen, text will appear on-screen informing the player of the control screen. The proper number of players lives, and boss lives displayed to the player at the start and the enemies will begin to assault the player as normally.*

## *TEST 2*

*Test Description: The "settings" button will be clicked by the tester and they will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the settings screen appears and contains the proper buttons, allowing the player to edit game settings, such as sound level and music level and ensure they function as desired.*

*Expected Results: This test will show the contents of the settings screen, the button for the option for controlling levels of the music and sound effects.*

## *TEST 3*

*Test Description: When the "Load Game" button is clicked on the Main Menu, the tester will ensure that " Load Game" does nothing when there IS NO prior saved data and the button will be grayed out.*

*Expected Results: If no prior saved game data exists, then the "Load Game" button will be grayed-out and the button should serve no real purpose other than to inform the player that no prior saved game data exists.*

## *TEST 4*

*Test Description: The "Load game" button will be clicked by the tester and they will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the level that is presented will be the appropriate level as per the saved game data; note: this will vary depending on when the user's game is saved, so no game data is currently available. for this test, the user will first play the game, then have the game save her progress to the data file, and then engage in this test, test 4*

*Expected Results: This test will show the appropriate level on the screen with the proper number of lives available to the player at the start and the enemies will begin to assault the kingdom as normally. the level is dependent upon the saved game data that is available at the time of this test.*

## *Test 5*

*Test Description: The tester will click the “Exit Game” button on the main menu and the game will exit without any errors*

*Expected Results: Command will return once more to the operating system and the game will have exited with no errors.*

## *TEST 6*

*Test Description: When the " Delete Game" button is clicked on the Main Menu, the tester will ensure that " Delete Game" does nothing when there IS NO prior saved data and the button will be grayed out.*

*Expected Results: If no prior saved game data exists, then the " Delete Game" button will be grayed-out and the button should serve no real purpose other than to inform the player that no prior saved game data exists.*

## *TEST 7*

*Test Description: The " Delete Game " button will be clicked by the tester and they will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the level that is selected is deleted, so no game data is currently available. for this test, the user will first play the game, then have the game save her progress to the data file, and then engage in this test.*

*Expected Results: This test will Delete saved game data that is available at the time of this test.*

## *Pause Menu Unit Tests:*

## *TEST 1*

*Test Description: The tester will ensure the game saves the player's progress correctly by playing the game's level, going back to the main menu, and selecting "Load game" button*

*Expected Results: The game data loaded when the player selects "Load Game" from the main menu will be the same game data (and the game will have been saved in the correct spot) that the player saved several moments earlier after selecting "Exit Game" from within the game's pause screen. this test will also be performed after the game has been shut off and then turned back on.*

## *TEST 2*

*Test Description: The "settings" button will be clicked by the tester and they will assess whether it is behaving as well (also as fast) as it should behave, ensuring that the settings screen appears and contains the proper buttons, allowing the player to edit game settings, such as sound level and music level and ensure they function as desired.*

*Expected Results: This test will show the contents of the settings screen, the button for the option for controlling levels of the music and sound effects.*

## *Test 3*

*Test Description: The tester will click the “Exit Game” button on the main menu and the game will exit without any errors*

*Expected Results: Command will return once more to the operating system and the game will have exited with no errors.*

## *Gameplay Unit Tests:*

## *TEST 1*

*Test Description: The tester will play through the game and ensure that the player/enemy animations occur at appropriate times and character animations don't appear jerky; instead they are smooth animations*

*Expected Results: The enemies and player have animations that are smooth while they walk along the path, additionally, all sound effects occur at logical times, i.e., an attack sound is only heard when a player-character's attack animation is seen and it is only seen when an enemy is attacked.*

## *TEST 2*

*Test Description: The tester will play through the game and ensure that the player moves according to the controls move forward, backward, jump, crouch, attack and pause/resume game.*

*Expected Results: The player moves forward with Right arrow key/D, backward with Left arrow key/A, jumps with Up arrow key/W, crouch with c and attack with Left mouse click/R and the game is paused or resume with spacebar.*

## *TEST 3*

*Test Description: The tester will play through the game and ensure that the player loses lives upon collision with enemy projectile. and gains lives when collides with health pickup.*

*Expected Results: The players lives increase by 1 diamond upon health pickup and decreases by 1 diamond upon collision with enemy projectile.*

## *System and Integration Testing*

*System Integration Testing is defined as a type of software testing carried out in an integrated hardware and software environment to verify the behavior of the complete system. It is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirement.*

### *Methodology:*

*System integration testing is a testing process that exercises a software system's coexistence with others.*

*Test Description: The tester will play through the game and ensure that the player loses lives upon collision with enemy projectile. and gains lives when collides with health pickup. Then tester will test the pause menu and then will test the main menu of the game.*

## *Performance and Stress Testing*

*Performance testing is carried out to check the system's performance under varying loads. Stress testing is carried out to check the behavior of the system under the sudden increased load. It contains load and stress testing as components.*

*Test Description:*

*The Unity Performance Test Extension will be used to test the performance that can be run using the Unity Test Runner. And the Unity Performance Benchmark Reporter will be used to compare the performances.*

### *Methodology:*

*The Unity Profiler is a tool you can use to get performance information about your application. Tester will connect it to devices on the network or devices connected to the machine to test how game runs on the intended release platform. Tester will also run it in the Editor to get an overview of resource allocation of the game.*

## *User Acceptance Testing*

## *Batch Testing*

## *Automated Regression Testing*

## *Beta Testing*

# *Test Schedule*

# *Control Procedures*

# *Features to Be Tested*

# *Features Not to Be Tested*

# *Resources/Roles & Responsibilities*

# *Schedules*

# *Risks/Assumptions*

# *Tools*