



Software Testing

Test Planning Project

Value: 30%

Due Date: Thursday 14th May 2020 11.59pm

Game Development International Ltd have requested you to develop a test plan for the new game that they have just designed. The game's requirements and design document are outlined in Appendices 1 and 2 respectively.

In appendix 3 there is sample template for the test plan which is an adapted version of a document which can be found on: <https://www.softwaretestinghelp.com/test-plan-sample-softwaretesting-and-quality-assurance-templates/> You are requested to utilise to this template as the basis of your test plan but you may add additional sections if you deem it appropriate.

Deliverables

- Each student must submit their Test Plan using GitHub. (**Note To keep your GitHub repository private you can add me to your repository using my GitHub ID: martinshynes001**)
- In addition to the Test Plan document each student **must** submit a short screencast presenting the contents of their Test Plan.

Submission Deadline

Project Submissions to be completed by 11.59pm on the 14th May 2020.

IMPORTANT NOTES

- You must use GitHub to track your continual commits to the Test Plan document.
- Plagiarism is **NOT** acceptable.
- You may be asked to present and explain your document.
- Late Submissions will be deducted by 15% per day. This will be accumulative – e.g. A student who submits their work three days late will be given a deduction of 45%.

Appendix 1

Game Requirements Document

Create a design and storyboard for a game. This is a 2D game to be developed as part of the module requirements. The types of game that are acceptable to design are:

- Shooters – Classic, horizontal or vertical scrolling
- Platform – Classic or 2D
- Traditional Game – Board Games (requires a simple AI)

You need to research these game types before deciding which you will design. Look at examples of the game types and understand what makes them popular to different players. Bear in mind that “clone and tweak” is a valid design method in which you pick an existing game and then modify the game world to suit your design. You may design from first principles.

If you are taking the “clone & tweak” approach, you need to identify clearly the game or games you are using as basis and present the evidence of the type of game it is within the options provided.

For the design, you need to create the following components

1. **Front End:** A term applied to all menus and screens that occur outside of the gameplay. This takes the player from the title screen to the point that gameplay begins.
2. **In-Game Menus:** A set of menus and screens accessed in-game, often from a pause menu. These form part of the game mechanisms rather than being distinctly separate.
3. **Control Mechanisms:** The way in which the player controls the game entities. Many games have just one control mechanism.
4. **The Game:** The gameplay screens showing the initial setup, how the action starts, a midpoint in play and the winning/progression conditions depending on the game you are designing. If the game is episodic in nature, then explain how episodes are defined and how the player moves between them.

Appendix 2

Game Design Document

Purpose of this document

This document is intended to provide a background of the application and the purpose of it. It provides a walkthrough of the design with notes about the user experience included. This document serves as the basis of implementation and records services and technologies that are used to achieve the development of the application.

Research on game types

Shooters

Shooter games are a subgenre of action video games, which often test the player's spatial awareness, reflexes, and speed in both single player or multiplayer environments. Shooter games encompass many subgenres that have the commonality of focusing on the actions of the avatar engaging in combat with a weapon against both code-driven NPC enemies or other avatars controlled by other players. Most commonly, the purpose of a shooter game is to shoot opponents and proceed through missions without the player character being killed or dying as a result of the player's actions. A shooting game is a genre of video game where the focus is almost entirely on the defeat of the character's enemies using the weapons given to the player.

Examples of shooter games would be Call of Duty, Destiny, Borderlands, Overwatch, and Counter Strike.

Platformer

Platform games, or **platformers**, are a video game genre and subgenre of action games. In a platformer the player-controlled character must jump and climb between suspended platforms while avoiding obstacles. Environments often feature uneven terrain of varying height that must be traversed. The player often has some control over the height and distance of jumps to avoid letting their character fall to their death or miss necessary jumps. The most common unifying element of games of this genre is the jump button, but now there are other alternatives like swiping a touchscreen.

Examples of platform games would be Shovel Knight, Mario, Sonic the Hedgehog, Donkey Kong, and Castlevania.

Board Game

A **board game** is a tabletop game that involves counters or pieces moved or placed on a pre-marked surface or "board", according to a set of rules. Some games are based on pure strategy, but many contain an element of chance; and some are purely chance, with no element of skill. Games usually have a goal that a player aims to achieve. Early board games represented a battle between two armies, and most modern board games are still based on defeating opponents in terms of counters, winning position, or accrual of points.

Examples of board games would be Chess, Monopoly, Scrabble, Backgammon, and 2048.

Game Design Overview

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 la '**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.

Game Characteristics

As this will be a platformer, this game will:

- Allow the player to control a specific character, that has an important fictional/narrative role.
- Have game statistics and/or relational attributes with other game objects, enemies, and/or the player character.
- Allow the player to take on and navigate the levels using an easy-to-use user interface.
- Have obstacles that the player must overcome, such as enemies and bosses.

Research on similar games

- **Dark Souls**

Dark Souls is an action role-playing game developed by FromSoftware and published by Namco Bandai Games. A spiritual successor to FromSoftware's *Demon's Souls*, the game is the second instalment in the *Souls* series. The game is played in third person. The game is 3D, however the game has many game mechanics that will be incorporated into this game. The player HUD is minimalistic; on the top left of the HUD, the player's health and stamina bar is displayed. Underneath these bars, any status effects (positive or negative), are displayed. On the bottom left of the HUD,

the player's currently equipped weapon, shield, consumable, and magic are displayed. On the bottom right, the amount of 'souls' the player has accumulated are displayed. 'Souls' in this series of games act as XP that the player can use to level up and buy items.



- **Salt and Sanctuary**

Salt and Sanctuary is a 2D action role-playing video game developed and published by Ska Studios. It is heavily inspired by the *Souls* series by Japanese developer FromSoftware. The game features 2D hand drawn visuals. The HUD used in Salt and Sanctuary is similar to the one used in *Dark Souls*: the player's health/stamina is displayed on the top left.



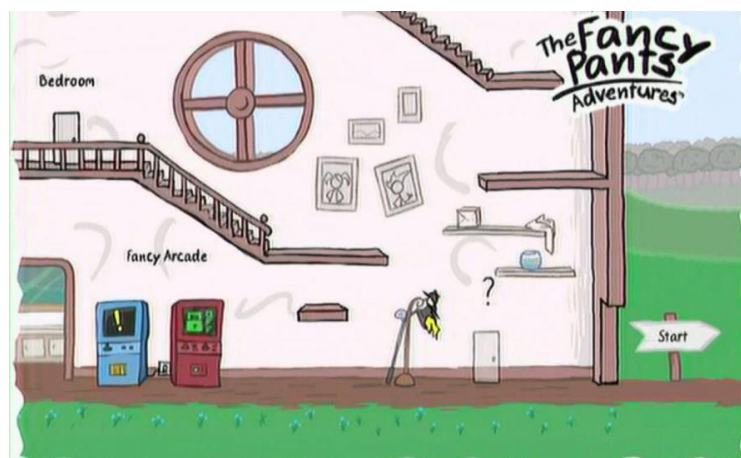
- **Shovel Knight**

Shovel Knight is a 2D side-scrolling platform game developed and published by Yacht Club Games. *Shovel Knight* is inspired by gameplay and graphics of platformer games developed for the Nintendo Entertainment System. Like the above games, the HUD in Shovel Knight is simple and minimalistic; the amount of Gold the player has is located on the top left corner. The amount of life points the player has is located on the top centre. The boss' life points (if one is present) is displayed on the top right corner.



- **Fancy Pants Adventures**

Fancy Pants Adventures is a series of free side-scrolling Flash games created by American developer Brad Borne. Three worlds have been released so far. *Fancy Pants Adventures* features Fancy Pants Man as the playable character of the series. He is a two-dimensional teenage stick figure with spiky, choppy hair, wearing only a pair of coloured triangular-shaped pants, hence his name. Again, the HUD is simple and displays only the most vital information; the player's health bar is displayed on the top left of the screen. Collectibles and the amount the player has collected is also displayed on the top left of the screen.



- **The Elder Scrolls V: Skyrim**




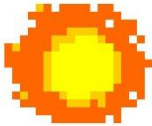

The Elder Scrolls V: Skyrim is an action role-playing video game developed by Bethesda Game Studios and published by Bethesda Softworks. It is the fifth main instalment in *The Elder Scrolls* series, following *The Elder Scrolls IV: Oblivion*. The game's main story revolves around the player character's quest to defeat Alduin the World-Eater, a dragon who is prophesied to destroy the world. The player may freely roam over the land of Skyrim which is an open world environment consisting of wilderness expanses, dungeons, cities, towns, fortresses, and villages. The main feature from Skyrim that will be incorporated into this game will be how the player






uses magic, which will be the primary method of attack in the developed game; the player can shoot balls of flame from their hands, with a delay in between ‘shots’.



Game Assets

The game uses several assets to create its levels and characters. These assets were made primarily in Aseprite, which lets you create 2D sprites and animations. The background was ‘made’ using Pixelator, which converts images into pixel art sprites. In the project folder provided, the following assets are included:

Asset Name	Asset
Background	
Player Character	
Enemy Character	
Player/Enemy Projectile	
Ground Asset	

Platform One Asset	
Platform Two Asset	
Health Pickup	
Rock Asset	
Menu Logo	

Sample Level



Sample level as shown. Background can be used repeatedly to make the level 'longer'. Shown above (from top left to top right, bottom left to bottom right): player's health, represented by red diamonds, platform with health pickup, enemy on platform, boss' health, player character attacking, enemy character attacking, boss character attacking).

Game Components

- Front End

The game will have three options on start-up: 'Play', 'Settings', and 'Exit Game'.

Selecting 'Play' will take the player into the game and the player will begin at Level 1.

If a save system is able to be implemented, the player will begin at their last saved point. 'Settings' will allow the player to edit game settings, such as sound level and music level. 'Exit Game' will quit the application'. OPTIONAL: Include a 'Load Save', 'Save Game', and 'Delete Save' option. This will allow the player to create multiple save files and be able to choose which one to load.

Sample Main Menu



A sample main menu. The player will be presented with this when they first start up the game. The player has the option of playing the game, accessing settings, or exiting the game. Optionally, if it can be implemented, a load game and delete game will also be options.

- In-Game Menus

The game will include a number of options when the game is paused, similar to those available on start up. The player will be able to resume the game, access settings, restart the level, and exit the game. The player can resume the game by selecting the appropriate option or simply pressing the assigned button for pausing/resuming the game. The player can access the settings and can adjust the sound level and music level, for example. Choosing to restart the level will reset the entire level. This entails

resetting the player's position to where they were when they first started, resetting the enemies' position, and resetting any and all pickups.

Sample In-Game Menu



A sample in-game menu. Once the game has been paused, the player can access settings or exit the game. Again, if a save system can be implemented, a save game option will also be presented, allowing the player to save the game state.

- Control Mechanisms

Basic controls:

<i>Action</i>	<i>PC</i>	<i>Mobile</i>
Move Forward	Right arrow key/D	Arrow on screen
Move Backwards	Left arrow key/A	Arrow on screen
Jump	Up arrow key/W	Arrow on screen
Crouch	C	Arrow on screen (hold)
Attack	Left mouse click/R	Dedicated button
Pause/Resume	Spacebar	Button in top right of screen

- The Game

Once the player has opened the game, they will be presented with three options: 'Play', 'Settings', and 'Exit Game'. Selecting 'Play' will take the player into the game and load the first level. The game will start immediately. From here, the player can progress through the level and once completed and once completed, the next level will load. At the start of the first level, text will appear on-screen informing the player of the control screen. The game will feature at least three levels, with each increasing in difficulty. This could range from having more and more enemies in the progressive levels, enemies having more health, the player character having less enemies, etc. Once the player has completed all the levels, the player will be presented with the option to either start again from the first level or quit the game. If the player chooses

'Settings' instead, they can instead adjust the sound level or music level. 'Exit Game' will close the game.

Once inside the game, the player can pause the game at any time. From here, they can access a number of options: resume the game, access settings, restart the level, or exit the game.

Appendix 3

Test Plan Template

Test Plan Template:

(Name of the Product)

Prepared by:

(Names of Preparers)

(Date)

TABLE OF CONTENTS

1.0 INTRODUCTION

2.0 OBJECTIVES AND TASKS

2.1 Objectives

2.2 Tasks

3.0 SCOPE

4.0 Testing Strategy

4.1 Unit Testing

4.2 System and Integration Testing

4.3 Performance and Stress Testing

4.4 User Acceptance Testing

4.5 Batch Testing

4.6 Automated Regression Testing

4.7 Beta Testing

5.0 Test Schedule

6.0 Control Procedures

7.0 Features to Be Tested

8.0 Features Not to Be Tested

9.0 Resources/Roles & Responsibilities

10.0 Schedules

11.0 Risks/Assumptions

12.0 Tools

1.0 INTRODUCTION

A brief summary of the product being tested. Outline all the functions at a high level.

2.0 OBJECTIVES AND TASKS

2.1 Objectives

Describe the objectives supported by the Master Test Plan, eg., defining tasks and responsibilities, vehicle for communication, document to be used as a service level agreement, etc.

2.2 Tasks

List all tasks identified by this Test Plan, i.e., testing, post-testing, problem reporting, etc.

3.0 SCOPE

General

This section describes what is being tested, such as all the functions of a specific product, its existing interfaces, integration of all functions.

Tactics

List here how you will accomplish the items that you have listed in the "Scope" section. For example, if you have mentioned that you will be testing the existing interfaces, what would be the procedures you would follow to notify the key people to represent their respective areas, as well as allotting time in their schedule for assisting you in accomplishing your activity?

4.0 TESTING STRATEGY

Describe the overall approach to testing. For each major group of features or feature combinations, specify the approach which will ensure that these feature groups are adequately tested. Specify the major activities, techniques, and tools which are used to test the designated groups of features.

The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation of the time required to do each one.

4.1 Unit Testing

Definition:

Specify the minimum degree of comprehensiveness desired. Identify the techniques which will be used to judge the comprehensiveness of the testing effort (for example, determining which statements have been executed at least once). Specify any additional completion criteria (for example, error frequency). The techniques to be used to trace requirements should be specified.

Participants:

List the names of individuals/departments who would be responsible for Unit Testing.

Methodology:

Describe how unit testing will be conducted, including a description of tests to be carried out. Who will write the test scripts for the unit testing, what would be the sequence of events of Unit Testing and how will the testing activity take place?

4.2 System and Integration Testing

Definition:

List what is your understanding of System and Integration Testing for your project.

Participants:

Who will be conducting System and Integration Testing on your project? List the individuals that will be responsible for this activity.

Methodology:

Describe how System & Integration testing will be conducted, including a description of tests to be carried out. Who will write the test scripts for the unit testing, what would be sequence of events of System & Integration Testing, and how will the testing activity take place?

4.3 Performance and Stress Testing

Definition:

List what is your understanding of Stress Testing for your project.

Participants:

Who will be conducting Stress Testing on your project? List the individuals that will be responsible for this activity.

Methodology:

Describe how Performance & Stress testing will be conducted, including a description of tests to be carried out. Who will write the test scripts for the testing, what would be sequence of events of Performance & Stress Testing, and how will the testing activity take place?

4.4 User Acceptance Testing

Definition:

The purpose of acceptance test is to confirm that the system is ready for operational use. During acceptance test, end-users (customers) of the system compare the system to its initial requirements.

Participants:

Who will be responsible for User Acceptance Testing? List the individuals' names and responsibility.

Methodology:

Describe how the User Acceptance testing will be conducted, including a description of tests to be carried out Who will write the test scripts for the testing, what would be sequence of events of User Acceptance Testing, and how will the testing activity take place?

4.5 Batch Testing

4.6 Automated Regression Testing

Definition:

Regression testing is the selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still works as specified in the requirements.

Participants:

Methodology:

4.7 Beta Testing Participants:

Methodology:

5.0 TEST SCHEDULE

Include test milestones identified in the Software Project Schedule as well as all item transmittal events.

Define any additional test milestones needed. Estimate the time required to do each testing task. Specify the schedule for each testing task and test milestone. For each testing resource (that is, facilities, tools, and staff), specify its periods of use.

6.0 CONTROL PROCEDURES

Problem Reporting

Document the procedures to follow when an incident is encountered during the testing process. If a standard form is going to be used, attach a blank copy as an "Appendix" to the Test Plan. In the event you are using an automated incident logging system, write those procedures in this section.

Change Requests

Document the process of modifications to the software. Identify who will sign off on the changes and what would be the criteria for including the changes to the current product. If the changes will affect existing programs, these modules need to be identified.

7.0 FEATURES TO BE TESTED

Identify all software features and combinations of software features that will be tested.

8.0 FEATURES NOT TO BE TESTED

Identify all features and significant combinations of features which will not be tested and the reasons.

9.0 RESOURCES/ROLES & RESPONSIBILITIES

Specify the staff members who are involved in the test project and what their roles are going to be (for example, Mary Brown (User) compile Test Cases for Acceptance Testing). Identify groups responsible for managing, designing, preparing, executing, and resolving the test activities as well as related issues. Also identify groups responsible for providing the test environment. These groups may include developers, testers, operations staff, testing services, etc.

10.0 SCHEDULES

Identify the deliverable documents. You can list the following documents:

- Test Plan
- Test Cases
- Test Incident Reports
- Test Summary Reports

11.0 RISKS/ASSUMPTIONS

Identify the high-risk assumptions of the test plan. Specify contingency plans for each (for example, delay in delivery of test items might require increased night shift scheduling to meet the delivery date).

12.0 TOOLS

List the Automation tools you are going to use. List also the Bug tracking tool here.