Test Specification

To The Sky

By: Maui Waui
Alyssa Faiferlick
Jocelyn Hinojosa
Tom Lee
Ian Dos Santos
Pouya Tavakoli

CECS 491A Section 2
April 5. 2022
Version 1.0

Fei Hoffman

Table of Contents

Test Specification	**1**
Table of Contents	**2**
Abstract	**3**
Unit Level	**4**
Module Level	**9**
Integration Level	**X**
System Level	**X**
Acceptance Level	**X**

Abstract

The following document's purpose is to provide a map for our test cases. The five levels of testing will be tested and discussed: Unit, Module, Integration,System, and Acceptance Testing. During these tests, we will ensure that every part of our project works accordingly. Each test will follow each of the following below:

1.Test Level : The level that the software is being tested at.
2. Quality Criterion : The metrics of success.
3. Description of test : A general description of the test
4. Requirements Reference : Use cases that the tests refer to and verify.
5. Steps of the Test Case : A step by step description of the test.
6. Expected outcome : The outcome that tells us the test has succeeded

Unit Level

Unit Testing refers to the individual parts of the software that can be tested for their effectiveness with other Unit components. Various Units are within the software and work together to ensure that the software runs appropriately.

Test Level	Unit Level
Quality Criterion	Functionality
Description of Test	Players are able to unlock and then access all worlds and their corresponding levels.
Requirements Reference	Use Case #1 - 4 Worlds (3 levels per World) Use Case #8 - World/Level Selection System
Steps of the Test Case	1. Player starts up the game and accesses the first level of the first world. 2. After completing the level, the player then has access to the next level. 3. The player exits to hub world. 4. Player still has access to the level that follows the previous level they completed. 5. Repeat steps 3-4 until the player has completed the game. 6. The player exits to hub world.
Expected Outcome	The player should have access to play all worlds and their corresponding levels.

Test Level	Unit Level
Quality Criterion	Reliability
Description of Test	Players are able to save data when given the option. Sava data can be overwritten and accessed correctly.
Requirements Reference	Use Case #6 - Instant Respawn on Puzzles (Fast Level Rebuild on Death) Use Case #22 - Optional Checkpoints
Steps of the Test Case	1. Player encounters a checkpoint or completes a level. 2. Prompt is given to the player if they reach a checkpoint 3. Correctly save data, data is overwritten if necessary. 4. Player exits to the hub world or has their health points depleted to zero. 5. Player is successfully placed in the game state that was previously saved.
Expected Outcome	Player is able to access the most recently saved checkpoint in the game.

Test Level	Unit Level
Quality Criterion	Functionality
Description of Test	Each world has its own unique element. Tiles within that world have unique trait that must correctly implement its unique physical
Requirements Reference	Use Case #11 - Unique Tiles Use Case #12 - Basic Physics Implemented
Steps of the Test Case	1. Player encounters or interacts with a unique tile. 2. Tile reacts with its intended physical attribute i.e. ice block has no friction, causes player to slide 3. Player stops its interaction with a unique tile. 4. The player is no longer affected by the unique tile.
Expected Outcome	Tiles demonstrate that they correctly implement their unique physical attribute.

Test Level	Unit Level
Quality Criterion	Usability
Description of Test	There are a variety of collectibles in the game that give the user access to complete tasks. Such as, the key collectible allows access to new levels. Coins allow exchange in trade shops for items. Power-ups give the player an advantage in a level.
Requirements Reference	Use Case #16 - Lots of Collectibles
Steps of the Test Case	1. Player collects a unique collectible i.e. a key 2. Player navigate to area in which the collectible can be used or activated i.e. a door 3. Player accepts a prompt to use the collectible. 4. The collectible successfully completes its purpose i.e. key unlocks door 5. The collectible is no longer accessible to the player after it is used.
Expected Outcome	Collectibles demonstrate they correctly and consistently implement their expected use i.e. coins are able to be collected and exchanged for items at shops.

Test Level	Unit Level
Quality Criterion	Compatibility
Description of Test	Player is able to deal damage and defeat mobs by bringing their health points to zero.
Requirements Reference	Use Case #10 - Boss Fights Use Case #14 - Enemy AI
Steps of the Test Case	1. Players encounter a mob. 2. Players deal damage to mob by physical attack, magical attack, item, etc. 3. Players continue damage until the mob's health point reaches zero. 4. Mob health points reach zero so it disappears.
Expected Outcome	Player successfully defeats a mob and a boss by depleting its health points to zero.

Module Level

Module Testing refers to the combination of Units and how well the Units work as a whole. For our project this refers to the various gameplay functions and UI Units that work together.

Test Level	Module Level
Quality Criterion	Functional Suitability and Usability
Description of Test	Players navigate around the levels with 2 dimensional tile-based movement
Requirements Reference	Use Case #3 - Tile Based Movement
Steps of the test case	1. Launch the game 2. Start a new game or continue from a save 3. Choose a world, then start a level 4. Test the movement inputs a. W - forward b. S - backward c. A - left d. D - right Ensure the player only moves one game tile per input 5. Move to each edge of the map and ensure movement beyond the edge is blocked 6. Move towards environmental objects like walls and ensure movement through the wall is blocked
Expected Outcome	The player can navigate around a level using the W,A,S,D input scheme. Collisions with walls are restrictive of player movement, and the player cannot move outside of the level area.

Test Level	Module Level
Quality Criterion	Functional Suitability and Usability, User Interface Aesthetics
Description of Test	Unique tiles serve a gameplay function and aesthetically complement their level
Requirements Reference	Use Case #11 - Unique Tiles
Steps of the test case	1. Launch the game 2. Start a new game or continue from a save 3. Choose the Earth world, then start on the first level 4. Interact with each type of unique tile in the level and ensure they function as intended, and complement the level design 5. Exit the level and repeat step 4 for each level in the world 6. Repeat steps 3-5 for the Fire, Water, and Air worlds
Expected Outcome	The player can successfully interact with unique tiles scattered throughout the game. Each unique tile performs as intended, and complements the theme of that world.

Test Level	Module Level
Quality Criterion	Usability, Operability, Functional Completeness, Functional Correctness
Description of Test	Functioning and persistent shopping system
Requirements Reference	Use Case #15 - Trader/Shop System
Steps of the test case	1. Launch the game 2. Start a new game or continue from a save 3. Enter the Hub World 4. Click the trader icon in the top right of the screen 5. Purchase some item(s) and enter one of the levels 6. The purchased items should appear in their appropriate place on the player's HUD 7. Exit and restart the game, continuing from where you were. 8. Check that all purchased items are still available
Expected Outcome	When a player clicks the trader icon they are taken to the trader's page. Items purchased by the player will show up in their inventory. Purchased items persist between game runs.

Test Level	Module Level
Quality Criterion	Usability, User Interface Aesthetics
Description of Test	When a boss is defeated they drop a power up that the player can pick up
Requirements Reference	Use Case #18 - Power ups gained from boss fights
Steps of the test case	1. Launch the game 2. Start a new game or continue from a save 3. Enter the Hub World and choose any world 4. Go to the boss level (level 3) 5. Play the game and defeat the boss 6. Pick up the power up
Expected Outcome	Player has a power up added to their inventory upon defeating a boss

Test Level	Module Level
Quality Criterion	Functional Completeness, Functional Appropriateness
Description of Test	The player can activate a checkpoint to continue from if they fail the level
Requirements Reference	Use Case #22 - Optional checkpoints
Steps of the test case	1. Launch the game 2. Go to Hub World and enter level 2 of any world 3. Play the game until you reach an optional checkpoint 4. Activate the checkpoint, and purposely fail the level
Expected Outcome	When the player fails they respawn at their last checkpoint

Integration Level

Integration Testing refers to how well the system's code functions and integrates with aspects of gameplay and level design.

Test Level	Integration
Quality Criterion	Functional Suitability, Reliability
Description of Test	Game objects with a physics implementation are not buggy
Requirements Reference	Use case #12 - Basic Physics Implemented
Steps of Test Case	1. Enter any level of the game 2. Play around and observe objects with physics implementations are behaving correctly 3. If object is interactable perform the interaction and observe the physical behavior is correct
Expected Outcome	Objects with physics behave correctly and bug-free

Test Level	Integration
Quality Criterion	Functional Suitability
Description of Test	Bonus level should only be available after completing the optional objective related to it
Requirements Reference	Use case #17 - Bonus Levels
Steps of Test Case	1. Start a new game 2. Try to access the bonus level, access should be denied 3.Grant player the required keys to unlock the level 4. Try to access the bonus level, access should be granted
Expected Outcome	Completing the bonus objectives grants access to the bonus levels

Test Level	Integration
Quality Criterion	Functional Suitability, User Interface Aesthetics
Description of Test	Cosmetic changes are reflected in game
Requirements Reference	Use case #20 - Customizable Player Cosmetics
Steps of Test Case	1. Open player inventory and equip a new cosmetic item 2. Enter a level and see if cosmetic item is being shown 3. Move the player around and ensure the cosmetic is not buggy
Expected Outcome	Players can equip new cosmetic items

Test Level	Integration
Quality Criterion	Functional Suitability, Reliability, Usability
Description of Test	Each level tracks player's stats and gives a star rating based on their performance
Requirements Reference	Use case #21 - Star Rating
Steps of Test Case	1. Enter any level and take note of the star objectives for that level 2. Complete the level and get at least two stars 3. Repeat steps 1 and 2 for a different level 4. Exit and restart the game 5. Observe that your star rating for the levels you played were saved
Expected Outcome	Player receives a persistent star rating based on their performance per level

Test Level	Integration
Quality Criterion	Functional Suitability, Compatibility
Description of Test	Full functioning controller support
Requirements Reference	Use case #2 - Full Controller Support
Steps of Test Case	1. Launch the game 2. From setting, change control scheme to gamepad 3. Navigate the menu using the controller 4. Enter any level 5. Navigate/Play through the level with a controller
Expected Outcome	Player can play the game naturally and comfortably with a controller

System Level

System level testing checks the fully integrated system for any issues involving its different components and how they work alongside the system itself. This level tests for any user experience issues that could happen when the software is running.

Test Level	System Level
Quality Criterion	Usability
Description of Test	Test to see if all players are able to easily understand how to start and play the game/how the controls work.
Requirements Reference	Use case #1 - 4 Worlds (3 Levels per World) Use case #2 - Full Controller Support Use case #3 - Tile based movement Use case #4 - Turn based puzzles and combat
Steps of the Test Case	1. Player starts up the game and accesses main menu 2. Player clicks on “new game” button to start a new empty game 3. Player enters the new hub world and goes through a tutorial of how to play. 4. Player accesses the first level.

Expected Outcome	The player should be able to easily start the game and learn how the controls work/how to navigate through the game when they start a new game.

Test Level	System Level
Quality Criterion	Functionality and reliability
Description of Test	Test to see if player's game progress data is saved to the computer and automatically backed up on SteamCloud.
Requirements Reference	Use case #22 - optional checkpoints
Steps of the Test Case	1. Player saves their progress at a checkpoint or completes a level. 2. Player exits to the hubworld and then exits from the hubworld to the main menu. 3. Player closes the game. 4. Player opens and starts the game again. 5. Player clicks on continue game option
Expected Outcome	When the player saves their progress at a checkpoint they should be able to keep that current progress even when they exit the game entirely.

Test Level	System Level
Quality Criterion	Compatibility
Description of Test	Test to see if player is able to play the game using different controllers.
Requirements Reference	Use case #2 - full controller support
Steps of the Test Case	1. Player connects a controller to their computer 2. Player starts up the game. 3. Player navigates through the game with the controller. 4. Repeat steps 1-3 with different

	controllers.
Expected Outcome	Player should be able to navigate through the game and manipulate their character with the controller.

Test Level	System Level
Quality Criterion	Portability
Description of Test	Test to see if player can install and run the game easily.
Requirements Reference	Use case #1 - 4 worlds (3 levels per world)
Steps of the Test Case	1. Player purchases game on steam 2. Player installs game on their computer 3. Player opens/runs game from either the steam library or local file on their computer
Expected Outcome	The player should be able to successfully purchase, download, and run the game without any issues.

Test Level	System Level
Quality Criterion	Performance efficiency
Description of Test	Hub world and levels will be tested to see if they are able to load without any issues
Requirements Reference	Use Case #1 - 4 Worlds (3 levels per World) Use Case #8 - World/Level Selection System
Steps of the Test Case	1. Player starts game 2. Player starts new game or continues previous game 3. Player checks how long wait is to load the hubworld 4. After accessing hubworld, player tries to access level and checks how long wait is to load level
Expected Outcome	The player should be able to load and access the hubworld and levels quickly without any

	issues (like the game crashing).

Acceptance Level

Acceptance Testing refers to the ability of the software to effectively run the game and test all gameplay features.

Download Game

Test level	Acceptance
Quality Criterion	Usability
Description of Test	Testing the ability to download the game from the game store
Requirements Reference	Use case #24 Download from game store
Steps of the Test Case	1. Go to the game store 2. Download game 3. Run game
Expected Outcome	Able to successfully download and run the game from the game store

Test Gameplay Features

Test level	Acceptance
Quality Criterion	Efficiency, Functionality
Description of Test	Test the various enemy AI for correct functionality
Requirements Reference	Use case #4 Turn Based Puzzles/Combat Use case #6 Instant Respawn on Puzzles (Fast Level Rebuild on Death) Use case #7 Multiple Level Formats Use case #8 World/Level Selection System Use case #10 Boss Fights Use case #11 Unique tiles Use case #12 Basic Physics Implemented Use case #15 Trader/shop system Use case #16 Lots of Collectibles Use case #18 Power ups gained from boss fights Use case #20 Customizable player cosmetics

	Use case #21 Star rating based on level/stage objectives Use case #22 Optional checkpoints Use case #23 Background music/Sound effects Use case #14 Enemy AI
Steps of the Test Case	1. Run Game 2. Choose level 3. Check enemy for correct AI behavior
Expected Outcome	Game features run correctly and use the correct scripts.

Test Objectives

Test level	Acceptance
Quality Criterion	Functionality
Description of Test	Test objectives given per level and see if the game tracks the player's progress
Requirements Reference	Use Case #21 Star rating based on level/stage objectives
Steps of the Test Case	1. Launch game 2. Select level 3. Play level 4. Do objectives 5. Complete level 6. Check objective completion
Expected Outcome	Objectives completed are correctly tracked and awarded.

Save System

Test level	Acceptance
Quality Criterion	Functionality
Description of Test	Test save system when player completes a level or quits the game
Requirements Reference	Use Case #25 Save System
Steps of the Test Case	1. Launch game 2. Play new level 3. Complete level 4. Quit game 5. Launch game 6. Check save status
Expected Outcome	Game correctly saves when the use completes a level

Controller Support

Test level	Acceptance
Quality Criterion	Functionality
Description of Test	Use the game if controller responds to inputs
Requirements Reference	Use Case #2 Full controller support
Steps of the Test Case	1. Launch Game 2. Use Controller 3. Evaluate Controller inputs
Expected Outcome	Game responds to controller inputs