Quality Assurance Plan

COMP2160 – Game Development

Dylan Neilson

47004029

Takaaki Chi

47887729

Table Of Contents

[Introduction: 3](#_Toc150537427)

[Scope and Objectives: 3](#_Toc150537428)

[Quality Standards and Metrics: 3](#_Toc150537429)

[Roles and Responsibilities: 3](#_Toc150537430)

[Testing Strategy: 3](#_Toc150537431)

[Test Plan for Each Feature: 3](#_Toc150537432)

[Test Environment and Tools: 3](#_Toc150537433)

[Test Data: 4](#_Toc150537434)

[Test Schedule: 4](#_Toc150537435)

# Introduction:

The purpose of this document is to provide an overview and understanding of the procedures and practices taken place to ensure we meet our quality objectives and standards for our Game Design Task 2 project.

# Scope and Objectives:

Our QA plan covers every aspect outline in the brief for this project, so that we can ensure the specified game functions are developed effectively and are operating as intended.

# Quality Standards and Metrics:

Our quality standards will be measured using the following criteria:

* Functionality
* Performance
* User Experience
* Adherence to the provided project requirements

# Roles and Responsibilities:

Our QA team consist of the following team members:

* Dylan Neilson
* Takaaki Chi

They will both be responsible for:

* Setting up tests
* Verifying tests
* Executing tests
* Reporting results

# Testing Strategy:

Our testing methodology will primarily be focused on manual testing to ensure a comprehensive look into each aspect of the game. These tests will verify the functionality, usability and performance.

# Test Plan for Each Feature:

Feature 1: Level Maps

* Objective: Verify the implementation of three different maps, grid integration, and obstacle placement.
* Criteria: Maps must be at least 10x10 tiles with appropriate terrain heights and obstacle placement.
* Approach: Manual testing.
* Test cases:
  + Verify that each of the three maps are constructed with a minimum dimension of 10x10 tiles.
  + Verify the placement and integration of obstacles (trees, rocks, barriers) on each map, ensuring they do not obstruct gameplay.
  + Verify that the player can move across all areas of the map without any clipping or collision issues.
  + Verify that the maps adhere to the provided 3D map tiles and obstacle meshes from the asset pack.
  + Verify that the player's interaction with the terrain and obstacles are consistent.

Feature 2: Physics-based Ball Movement

* Objective: Verify the physics-based behaviour of the beach ball and its interaction with the map and obstacles.
* Criteria: The beach ball should bounce and roll realistically and remain within the map boundaries.
* Approach: Manual testing.
* Test cases:
  + Verify that the beach ball responds to physics, showing realistic bouncing and rolling behaviour.
  + Verify that the beach ball does not leave the map area and is prevented from moving outside.
  + Verify that the beach ball interacts with the terrain and obstacles, bouncing off them and responding to collisions.

Feature 3: Player Movement

* Objective: Ensure the player's movement mechanics meet the specified requirements, including speed variations, terrain interaction, and collision handling.
* Criteria: The player should move smoothly and interact with the environment.
* Approach: Manual testing.
* Test cases:
  + Verify that the player moves in all required directions (up, down, left, right) with appropriate speed using the WASD keys.
  + Verify the player's ability to switch between walking and running speeds using the shift key, with smooth transitions.
  + Verify the player avatar smoothly turns to face the direction of movement instead of instantaneously.
  + Verify that the player always remains in contact with the ground, even when navigating ramps.
  + Verify that the player cannot accidentally step off the edge of a cliff.
  + Verify that the player is confined within the map area and cannot move outside its boundaries.

Feature 4: Kicking the Ball

* Objective: Validate the player's ability to grab and kick the ball with specified controls and physics.
* Criteria: The grabbing, aiming, and kicking mechanics should be intuitive, with a visible trajectory for the ball's path.
* Approach: Manual testing.
* Test cases:
  + Verify that holding the space key activates "grab mode" for the player.
  + Verify that the player avatar turns to face the ball when it is within the grabbing radius.
  + Verify that the player can control the direction of the kick using WASD keys.
  + Verify the display of a dotted trajectory line to visualize the expected path of the ball after kicking.
  + Verify that releasing the space key performs the kick, launching the ball in the chosen direction with the specified impulse.

Feature 5: Goal

* Objective: Verify the goal's representation and functionality are as specified.
* Criteria: Completing a level should be triggered when the ball passes through the goal ring.
* Approach: Manual testing.
* Test cases:
  + Verify that each level contains a goal represented by one of the provided ring meshes.
  + Verify that completing a level is achieved when the ball successfully passes through the goal ring from either side.

Feature 6: Player Animation

* Objective: Verify that player animations are correctly triggered and synchronized with player actions.
* Criteria: Animations should play smoothly and correspond to the player's actions.
* Approach: Manual testing.
* Test cases:
  + Verify that the Idle animation plays on a loop when the player is standing still.
  + Verify that the Walk animation plays on a loop while the player is walking.
  + Verify that the Run animation plays on a loop when the player is running.
  + Verify that the Interact animation plays once when the player grabs the ball.
  + Verify that the Throw animation plays once when the player kicks the ball.
  + Verify that the Cheer animation plays on a loop when the player successfully scores a goal.

Feature 7: Camera Control

* Objective: Verify the behaviour of the game's camera, including its positioning and transitions.
* Criteria: The camera should provide a clear view of the game environment and smoothly transition between different states.
* Approach: Manual testing.
* Test cases:
  + Verify that the game maintains a 16:10 aspect ratio.
  + Verify that the camera is orthographic and positioned at a 45° angle to create an isometric effect.
  + Verify that the player is at the centre of the camera when the player is moving.
  + Verify that the camera adjusts to show more of the world in the direction the player is aiming when preparing to kick.
  + Verify that the camera transitions between these views are smooth.
  + Verify that an arrow sprite is displayed when the ball is off-screen, pointing in the direction of the ball.

Feature 8: User Interface

* Objective: Verify the user interface elements are functional and adhere to design specifications.
* Criteria: The UI should display relevant information and enable player interaction.
* Approach: Manual testing.
* Test cases:
  + Verify that the specified font is used for the game's text elements.
  + Verify that the initial level dialog box displays the level number, "par" for the level, and a Start button.
  + Verify that the "level complete" dialog shows the par for the level, the number of kicks taken, and the total score for all completed levels.
  + Verify that the Retry button allows the player to replay the current level without resetting the score.
  + Verify that the Next button loads the next level, displaying the starting dialog.
  + Verify that the top left of the screen shows the number of kicks made so far and the par for the level.

Feature 9: Analytics

* Objective: Verify the collection of playtesting analytics data and log file generation.
* Criteria: The system should record relevant data, and log files should be correctly formatted.
* Approach: Manual testing.
* Test cases:
  + Verify that the system records the system username using Environment.UserName property.
  + Verify that the IP address of the computer is recorded using Dns.GetHostEntry method.
  + Verify that the system records the date and time when the player starts and quits the game using DateTime.Now property.
  + Verify that, after each level completion, the scene name, time taken, total number of kicks, and the player's decision to retry or proceed are logged.
  + Verify that log files are correctly formatted with one record per line and fields delimited by tabs.

# Test Environment and Tools:

Testing will be done through the Unity 3D game engine on both a Windows and Mac OS device.

# Test Data:

Test data will consist of sample levels, player profiles, and expected outcomes, as well as any additional test data needed for specific test cases.

# Test Schedule:

QA activities will be conducted once all development branches are merged. Following this will allow us to integrate all changes and additions made by team members and begin testing to see that everything is compatible and functioning as specified. This will be conducted the day before submission.

Reporting will be recorded in a separate document where team members can add entries and record their findings and results for further analysis.

Once all test cases have been verified, any fixes will be conducted, and a final build will be merged into the main branch.