

Landslider: A Depth Detection Game

Test Plan Document

Team 5

Michael Fadem

Cameron Chinena

Christopher Gonzales

Eduardo Lash

Jeremy Keys

Table of Contents

[Table of Contents](#)

[1 Introduction](#)

[1.1 Major Testing Issues](#)

[2 Testing Strategy](#)

[3 Test Environment](#)

[4 Performance and Stress Testing](#)

[5 Project Results](#)

1 Introduction

This document will outline the test plan that will be used to test the functionality of the system based on the team's requirements and the customer's requirements.

1.1 Major Testing Issues

The most important and difficult aspect of testing this system is the user's place in the environment relative to the RealSense camera and computer monitor itself. While the game will allow the user to calibrate the camera and game space it will be easy for the user to move out of the acceptable play area of the game and for the RealSense camera to become nonresponsive making the game unplayable. Setting up the test environment for the system will be simple since it will only require that the RealSense camera is pointed in front of the computer monitor and that the user is within the camera's visual range for hand and gesture tracking which is seven to thirty one inches. Since the camera has such a range of recognition area it is important that the user is also not that close to the monitor as to help with detection but also to prevent any harm to the monitor from the user extending or moving their arms. Due to these issues with the RealSense camera it will be important to ensure the sliding actions and collision detection within the game are not affected from any issues that could arise from the RealSense camera.

2 Testing Strategy

The overall approach to testing this system is a bottom-up testing method. This approach will allow the team to determine the functionality of each component outside the context of the total game. Since the system is highly interactive it is important that individual components work on each of their individual functions since that will provide an indication that the next step up in the testing approach will not fail due to an individual component but rather the two components interacting with each other. Once it is determined that individual components are working as specified the functionality that is required from the components working together will be tested next by utilizing the functionality of all of the components previously tested to achieve the desired outcome from the combined components.

For the fully integrated system testing we will test the system as if the tester is the end user. This tester will be both from the team and outside of it to ensure that the team is not testing the system in such a way to avoid known bugs and that all aspects of the system are tested equality. This process means that the tester will play the game and test each component, meaning that they will start at the main menu screen and proceed to each menu accessible from that scene and finally proceed to the game space. Once the tester is in the game space they will ensure that all of the gesture and hand tracking components are functioning and that the non-playable character and tile objects are making the correct modifications to the user interface. If the game is performing as specified in the design and requirement specifications and has successfully passed all of the test set for it then the system will be ready for release. If any bugs or failures arise during testing the tester will log the error or bug and the circumstances within the game that caused them. Following a bug fix the team will log the bug/error and the fix that done to prevent the bug/error from occurring again.

3 Test Environment

Due to the highly interactive nature of our system our test environments will differ from traditional test environments, such as databases or websites. The primary setup for our test environment is a sequence of levels that will contain a tile layout that will allow for the system to test each aspect of the user interface interactivity (i.e. health, score, and timer) as well as player/gesture interactivity. Since the game has multiple themes of tiles, the team will create a sequence of tiles testing each themes tiles followed by a level combining all of the tile types to ensure they are working together properly. To achieve regression testing on any modifications we make we will provide the system with a series of test functions that will change the tiles in the game space to test tile functionality and log the user interface changes to ensure that the system is still doing the correct changes. To assist with the unit testing the will be running the Unity Test Tools suite which allows the team to test attributes of the game and any effects that should be occurring to both the NPC and player attributes.

4 Performance and Stress Testing

To ensure that the performance of the system is satisfactory for users, the testing sessions will try to maintain tracking of the user's hand throughout a full game. Despite how well the camera tracks, it can sometimes stop tracking, or not find the user, as the range of tracking is sensitive, and some slight movement can cause tracking errors. Testing also needs to see if the camera can keep up with the user's movement, the user at some point may try to rapidly move tiles by moving their hands in a quick motion, so testing to see if multiple tiles will move in the correct way in rapid succession is possible. It is also important that we ensure that the user cannot modify the tile objects in such a way that they are unable to continue using the gesture based moving. To test the performance of the tile interaction with the NPC the team will test the interaction at all speed intervals of the NPC as well as all tiles that are detrimental to the NPC's attributes. If the system is able to regularly track and continue tracking the user's hands under all of the specified conditions and the user interface is changing then the performance of the system will be acceptable.

5 Project Results

The first area that was tested in the game was the interactivity of the RealSense hand data with the the tile moving system. With the help of the other testers outside of the team it was determined that transitioning to a one hand only game state increased playability and reduced the amount of accidental collisions. During this testing it was discovered that the timer and score functioned correctly on their own but encountered errors when integrated with the game space. These bugs were addressed and corrected resulting in fully functional user interface components. The feedback from the testers noted that while the menu system is not intuitive at first interaction this was due to the unfamiliar nature of the type of interaction (RealSense). After some time with the system the testers became use to the design and simplicity. Also it was noted that the inverted row controls increased the learning curve of the game but after some play time it was easily adjusted to. Due to the highly interactivity of the game and time

constraints the team was unable to complete any automated testing, however large parts of our system are unable to be tested automatically.