Final Project Report

Karly Peckham

Neftali Rodriguez

Andreas Janzen

Senior Project

May 3, 2018

**Original Project Proposal**

Problem/Solution:

Knowledge of the internet and its various parts is becoming more and more essential to basic human life. Unfortunately, networking can be dry and very confusing. It is simply not fun to learn about networking, and if one does not have to, then there is very little reason to seek out the information. A mobile application that addresses networking topics in a fun and engaging game can provide a platform for people to learn in a fun and productive way.

The proposed app consists of three main parts: character design and dialogue, learning mini-games, and social participation. Each is equally important, though not necessarily

equally time-intensive. The success of the application requires the success of each of the three parts – for example, an app that has excellent mini-games but subpar interaction with the user outside the game will likely be discarded.

Backlog:

A.1 Non-Game Work

A.1.1 Writing and Art

\_ Dialogue/exposition writing

– Game introduction - 2 hours

– Networking overview - 2 hours

– Checksum introduction - 2 hours

– Routing introduction - 2 hours

– Segmentation introduction - 2 hours

– Miscellaneous dialogue - 2 hours

\_ Character design

– Instructor 1 art - 5 hours

– Instructor 2 art - 5 hours

– Instructor 3 art - 5 hours

\_ Mini-game art

– Checksum game

\_ Binary background art - 3 hours

\_ Check button icon art - 3 hours

– Routing game

\_ Post office icon design - 3 hours

\_ Map design - 3 hours

– Segmentation game

\_ Background layout - 3 hours

\_ Package design - 3 hours

A.1.2 General Programming

\_ Main menu

– Game selection buttons - 2 hours

– Layout decisions - 2 hours

– Settings button - 2 hours

\_ Level selection

– Level selection buttons - 2 hours

– Level selection recyclerview - 4 hours

– General buttons - 2 hours

\_ Settings menu

– Instructor switch - 2 hours

– Music toggle - 2 hours

– Sound effects toggle - 2 hours

– Google Play account connectivity - 2 hours

\_ Loading screen - 2 hours

\_ Leaderboard

– Get points from games - 2 hours

– Player recyclerview - 4 hours

– Top 3 designation - 2 hours

– Sharing button - 2 hours

– Header - 2 hours

– Google Play connection - 4 hours

– Monthly Leaderboard submission - 2 hours

– Weekly Leaderboard submission - 2 hours

– Daily Leaderboard submission - 2 hours

A.2 Learning Games

A.2.1 Checksum Game

\_ Animation of incoming numbers - 3 hours

\_ Textviews for numbers - 2 hours

\_ Buttons for input - 2 hours

\_ General design - 2 hours

\_ Random sum generation - 2 hours

\_ Checksum validation - 2 hours

\_ Points assignment - 2 hours

\_ Add score to total - 2 hours

\_ Transition animations - 3 hours

A.2.2 Routing Game

\_ Line weight textviews - 2 hours

\_ Router textviews - 2 hours

\_ Buttons to allow for selection - 2 hours

\_ Transition animations - 3 hours

\_ Backend

– Random path generation - 2 hours

– Random path weight generation - 2 hours

– Best path calculation - 2 hours

– Points assignment - 2 hours

– Add score to total - 2 hours

– Best path verification - 2 hours

A.2.3 Segmentation Game

\_ Animation of boxes - 4 hours

\_ Buttons for packages - 2 hours

\_ Transition animations - 3 hours

\_ Backend

– Job randomization - 2 hours

– Package generation/randomization - 2 hours

– Game timer - 2 hours

– Points assignment - 2 hours

– Add score to total - 2 hours

**Revisions to Project Proposal**

As we worked in Unity, we realized that it was not feasible to create an entire, fully-functional game application in one semester. We decided to create a proof-of-concept application with three example levels for each of the three games, as well as example learning dialog.

Unfortunately, we were not able to completely finish the last game, Segmentation. It is the most complicated of the three games to execute, and we did not allocate enough time to finish it completely. However, we have completed some of the game, as is shown in the project demonstration video.

Additionally, we realized that it would more than we had time for to connect this game to Google Play for internet leaderboards. This will need to happen before the game is fully useable, as competition is an aspect of the gamification that we wish to utilize to engage children in playing our game.

Other small project changes include shrinking the number of instructors from three to two and getting rid of background music and sound effects (nobody likes game music anyway).

**Division of Labor**

Karly was in charge of creating the graphics used throughout the game, the pathfind backend, and the checksum backend. She also wrote all of the teaching language and generally micromanaged the group.

Andreas was in charge of creating the front end of the checksum game, as well as the back end of the segmentation game. He also created a pause menu. Andreas also did work on the segmentation front end.

Nef was in charge of creating the pathfind front end and the segmentation front end. He also created all of the in-betweeny connection scenes and game intro/termination scenes.

**How to Run Tests**

We did not create tests for this project (I know, I know, it’s bad form). In the future, we plan to write unit tests as well as overall tests to ensure that the app continues to function after future changes.

**How to Use Product**

This is an Android application, and must be run on an Android emulator or Android phone. To build it, one must have both Unity and the Android SDK and Java JDK, and an Android device or emulator that is running Android 8.0 Oreo or higher. The project must be built in Unity and deployed to an Android device. Assuming that those looking through these project files do not want to go to that hassle, it is easy enough to look through the games and play them in the Unity editor (no Android messiness necessary).

Because Unity is very strange about screen resolutions, we had quite a bit of trouble getting our scenes to play nicely with each other. This is not noticeable in the Unity editor, but when built on a phone, there is a bug that causes some screens to be smaller or larger than expected.

**Project Demo Video**

Please see video that has been turned in alongside this report.