The Trading Game

A Team Building Activity

By Jessica Douma & Fernando Pacheco

Abstract

The Trading Game is an iOS application that facilitates an adaptation of a popular team-building exercise where players trade information and a variety of items. The application was built using Apple's Swift programming language and the server, hosted on an Intel Edison, was implemented using the popular networking framework, Twisted Python. The application currently supports 12 clients and properly handles transaction requests between any two clients.

Motivation

We wanted to build a mobile application that facilitated team building exercises. The exercise we have adapted was a multi-component activity and designed for a large group of participants which made it difficult to moderate. Additionally, the participants had to keep track of physical items and records pertinent to the in-game transactions. Our goal was to reduce supply costs and automate logistical components of the activity while retaining the person-to-person interactions.

Expected Functionality

Server

Act as game moderator

Enable/disable trading rounds

Sends game updates at the end of each round

Support up to 12 iOS clients

iOS App

Player interface

Support transactions between clients

Displays and updates client balance

Display leaderboard at the end of each round

Accomplished Functionality

Server

Act as game moderator

Enable/disable trading rounds

Sends game updates at the end of each round

Support up to 12 iOS clients

iOS App

Player interface

Support transactions between clients

Displays and updates client balance

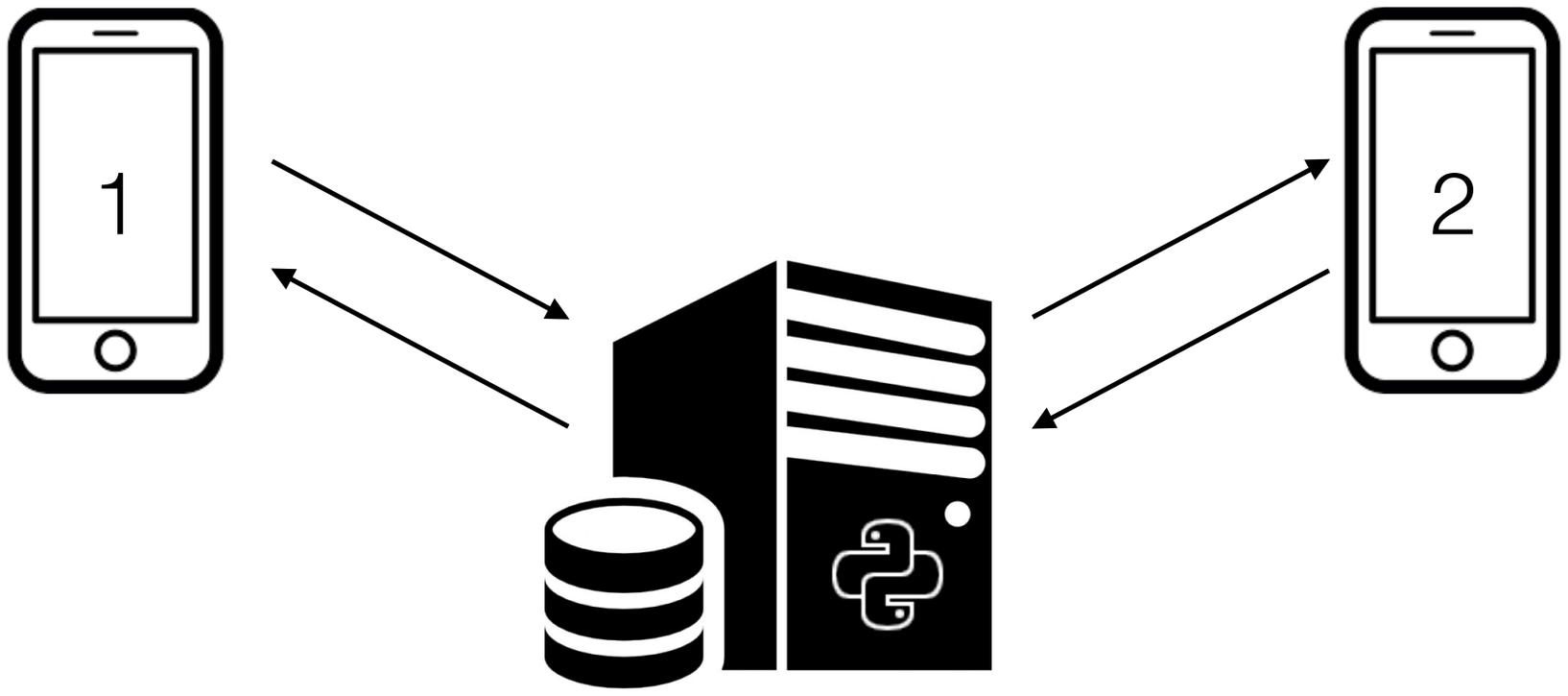
Display leaderboard at the end of each round

Theoretical Background

Wireless LANs transmit data through the air using radio transmission and are used to connect a set of wireless computers to a wired network. To enable the use of a remote sever, WiFi was chosen over Bluetooth due to its reliability and long range.

TCP is a reliable connection-oriented protocol that provides ordered and error checked delivery of byte streams. TCP was chosen over UDP because correct functionality of the application depends on the ordering of - and receipt of all - incoming packets.

Implementation

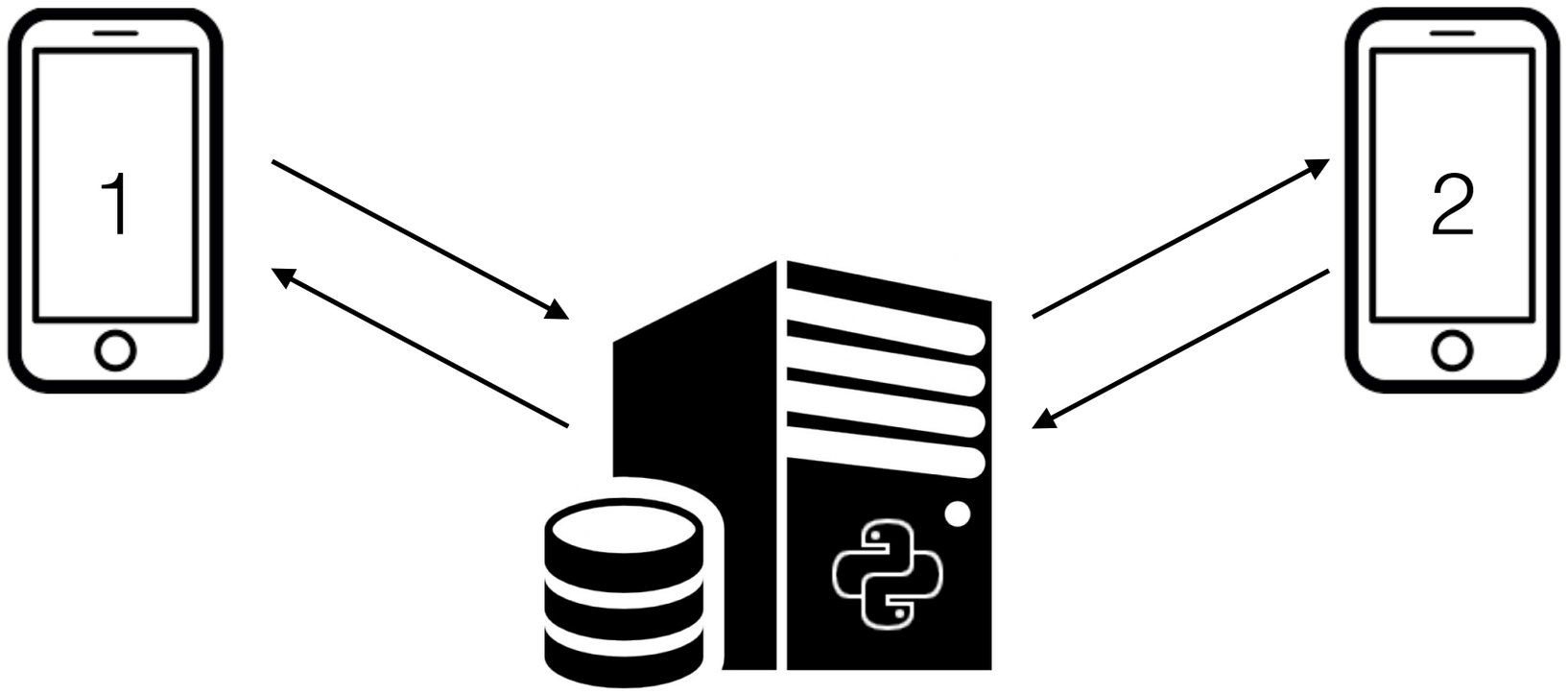


Step 1: Upon successful connection to the server, each client is initialized with an item inventory

Step 2: Server notifies the client that trading rounds have begun

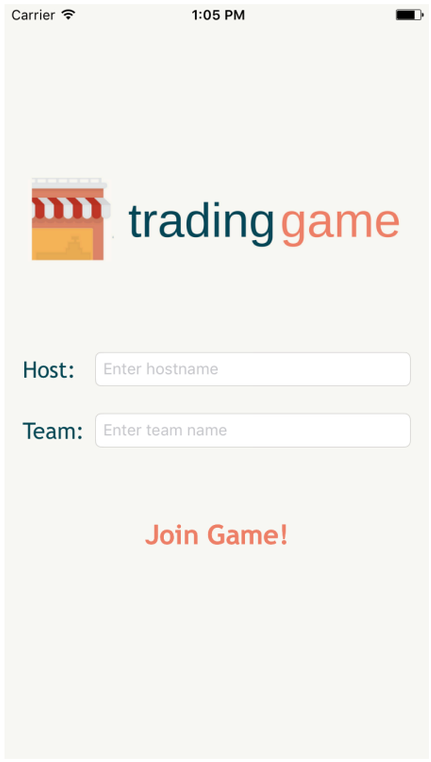
Step 3: Client can perform a transaction i.e. client 1 sends a transaction to client 2 through the server, that the server then forwards to client 2

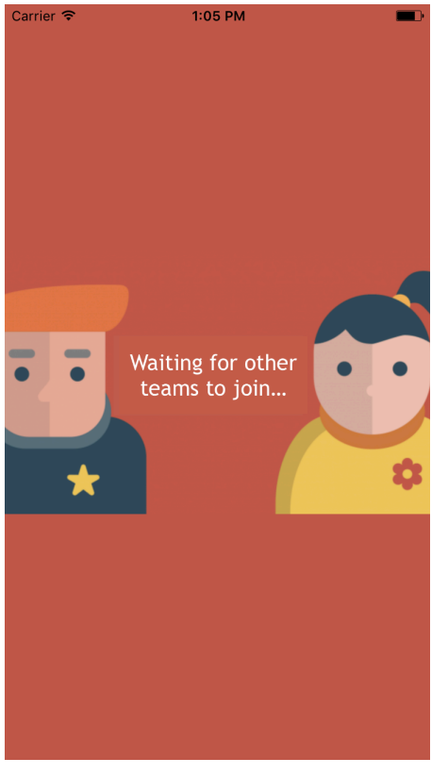
Implementation

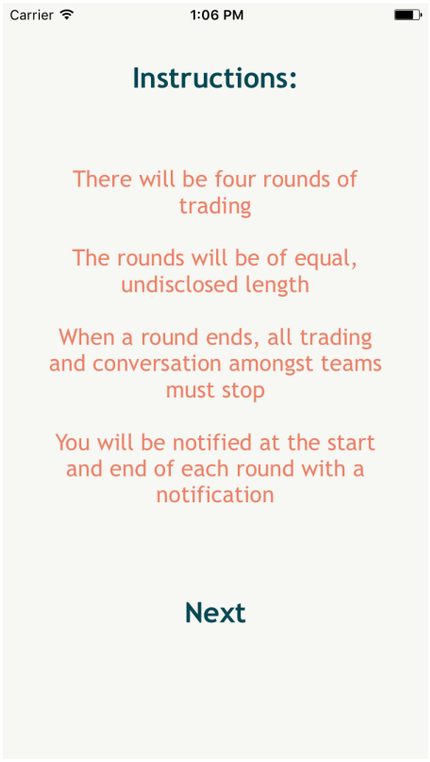


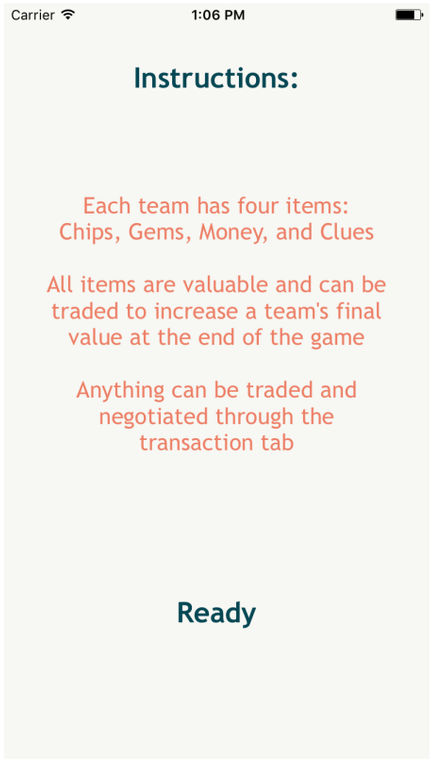
iOS application was implemented using Swift 3 and Swift Sockets

Server implemented with Twisted Python and hosted on an Intel Edison

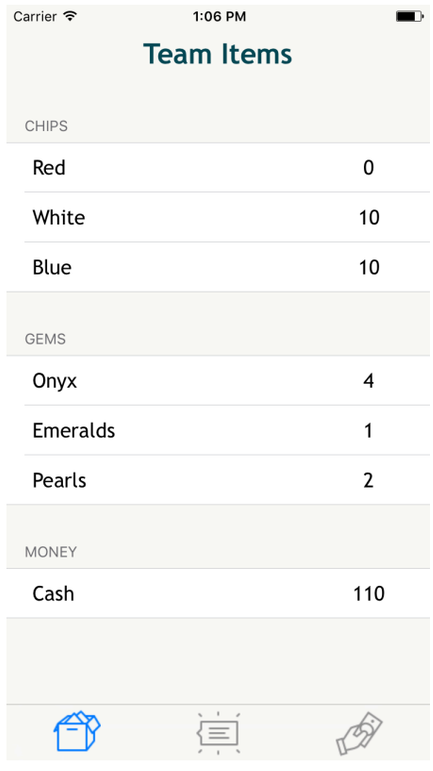


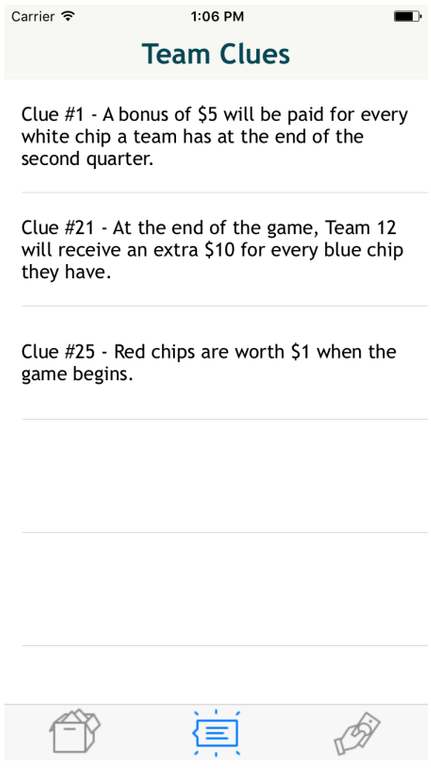


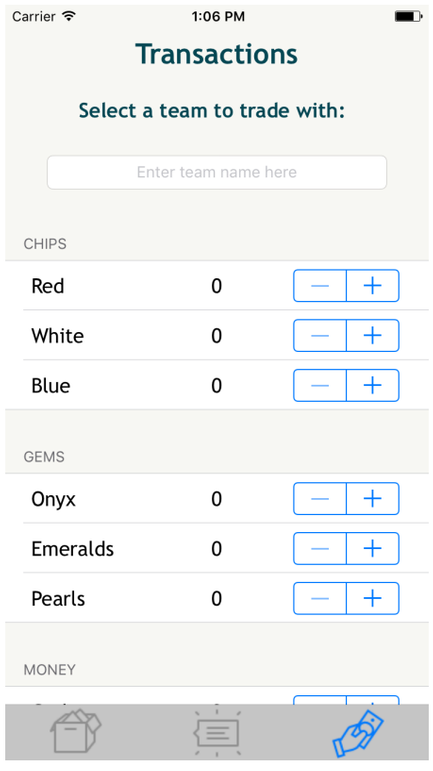


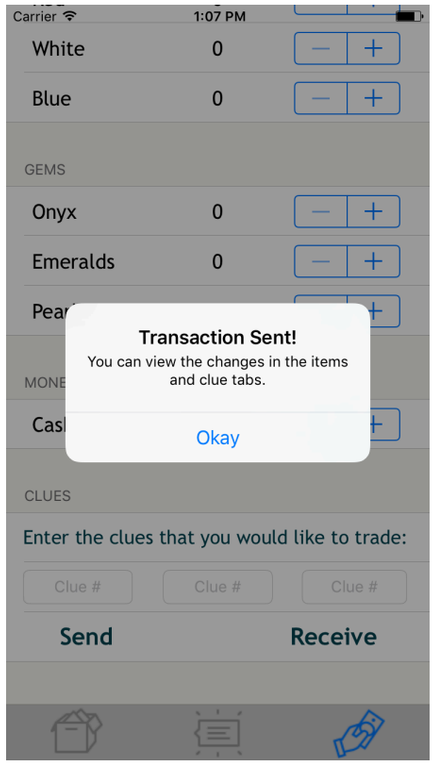


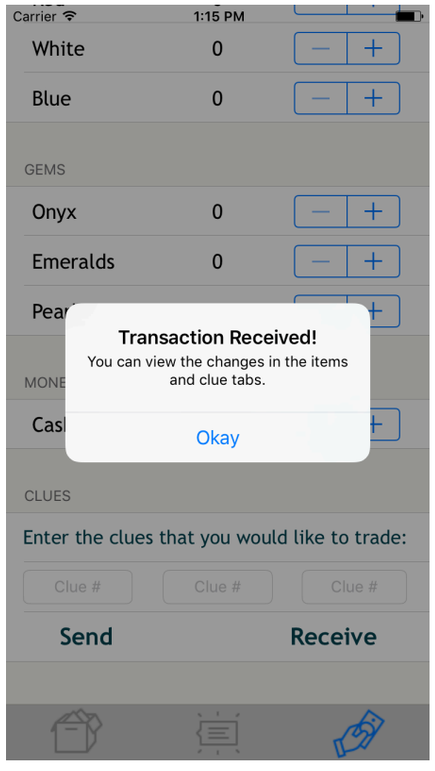
User Interface











Results & Evaluation

- Successfully connect and support communication for up to 12 clients
- Client is properly initialized with all relevant information provided by the server
- Clients are able to make transactions with other clients as long as they have items to trade
- Our choice of socket API made it difficult to implement game moderation

Future Work

- We would like to host the server on a service such as Google Cloud to eliminate the need for users to enter the host name

- We would like to transition to a non-blocking socket API to allow the client application to remain responsive

- We would like to incorporate bluetooth so that users can connect to a bluetooth speaker so that voice commands could be incorporated into the game moderation

Member Contributions

Fernando Pacheco

Implemented the server side logic using Twisted Python

Implemented iOS model - client side logic using Swift 3 and Swift Sockets

Implemented the game logic using provided pseudocode

Jessica Douma

Implemented the iOS view and user interface

Implemented the iOS controller - the link between the model and the view

Provided the pseudocode for the game logic

Conclusion

We were both very pleased with the outcome of our project. We wanted to create an application that we could use within our student organization to be used for one of our team-building activities. We not only accomplished this, but learned so many skills to help us keep developing network based applications for our organization. We both learned relevant skills like iOS development, twisted python, TCP client-server model, asynchronous/synchronous i/o, and how to analyze our application needs and determine which networking method is best suited for our needs.

Please find our source code on our GitHub

https://github.com/jessicadouma/csm117-proj