Team Roles:

Product Owner: Spencer Adolph

Scrum Master: Minhwan Oh

Customer Meeting date/time/place:

First meeting was a group-phone call. (They are in Colorado)

Call took place 10/18/2019 @ 1pm

Next call scheduled for 2 weeks from now.

Summary:

Our main customer represents the Military and Strategic Studies department in the Air Force Academy. They are primarily focused on developing their core class (MSS 251) which currently utilizes Island Rush Version 2. The customer had Version 3 created in the beginning of 2019, and we are continuing that development. Included with what we were given is documentation on the expected final product, as well as an already started codebase.

The stakeholders (faculty members teaching MSS 251), have advised us that at the end of our development we will be handing off the project to a new team of contractors. Until then, they have indicated which features (from the documentation) they would like us to focus on developing. Beyond these features, we are to continue to fix the current codebase and make it better.

Github: <https://github.com/island-rush/K3>

Pivotal: <https://www.pivotaltracker.com/n/projects/2400557>

Video: (N/A -> Customer is remote)

User Stories:

Feature: Add a game

As a Course Director

So that my faculty members have their own games to play

I want to add a game to the Island Rush Database

Feature: Refuel Air Pieces

As an Air Commander Player

So that I can best understand refueling and utilizing game resources / strategy

I want to transfer fuel from tanker pieces to other aircraft pieces

Feature: SAM’s Attack Indicator

As a player

So that I know when a SAM attacks and destroys my air pieces

I want to add an indicator or popup to show that it happened, and where it happened

Feature: News Alerts

As a teacher

So that I control which team has an advantage in the game

I want to add & modify news alerts that negatively or positively affect gameplay

User Interfaces:

Most, almost all can be taken from Version 2 directly and re-used. Several are already implemented in Version 3’s existing codebase. Therefore there is no need for us to re-create user interfaces.

Legacy:

Our project can be considered a legacy project because we are taking in an existing codebase and continuing to develop it. Our overall strategy will begin with becoming familiar with the current codebase, getting it working and understanding which features are already implemented. Next, we will begin to work on the features that were emphasized by the customer. While we do this, we will additionally improve the codebase with better documentation and better organization. Finally we will make it easy to transition our development to the next team.

There is an already existing documentation for the final product we are working on. This contains detailed explanations of each feature, and how they are used by each user. These are where we will get our user stories for this project.