**CSCE 606 TAMU Software Engineering**

**Team Island Rush V3**

**Team Roles:**

Product Owner: Spencer Adolph

Scrum Master: Minhwan Oh

Backend Developer: Barret Pickett, David Huarng, Jonathan Weishuhn

Frontend Developer: Bozhao Zhang, Shipeng Yang

**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>

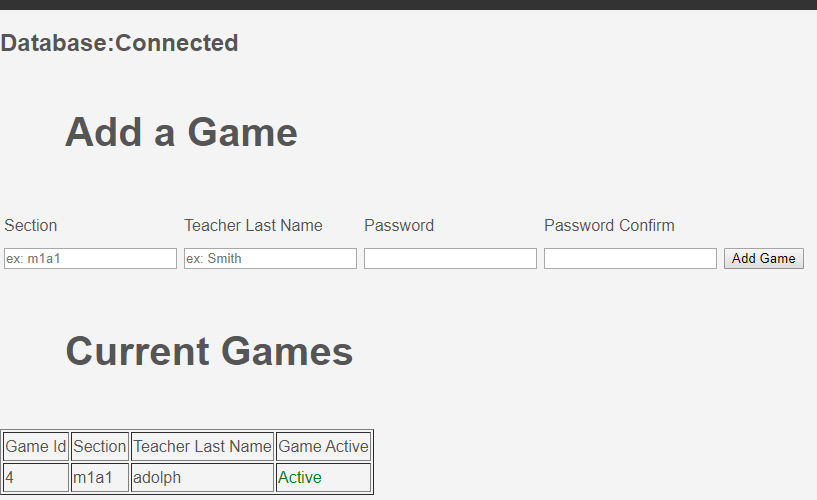
Video: (N/A -> Customer is remote)

**User Stories & Status:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Feature | Description | Tasks | Status | Owner |
| 1 | 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 | Modified Backed Source | Done | Spencer |
| Created Fronted UI | Done | Spencer |
| 2 | Create new sections for unit classification | Currently, there is only one section which include every unit. We'd like to seperate 3 sections such as air, land, ocean section. | Create new section | Done | Bozhao  Shipeng |
| Separate units | Done | Bozhao  Shipeng |
| Organize Components | Done | Bozhao  Shipeng |
| 3 | 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 | UI (components) for displaying aircraft on left, and tankers on the right, as well as some controls for transferring fuel | Done | Spencer |
| refuel event creation (when aircraft and tankers are in same position) | Done | Spencer |
| refuel event handling (sending to client, getting client inputs, ending the event) | Done | Spencer |
| UI controls (clicking to transfer fuel) | Done | Spencer |
| 4 | Minimize popups to view board | Minimize popups (battle, refuel, containers\*) to the left side of the board to a tab underneath the other menus. This enables a person to see the board where the event is taking place, and this event should highlight the position. | Minimize functionality done | Done | Berret  Jonathan |
| UI for minimize done | Done | Berret  Jonathan |
| board highlight done | Done | Berret  Jonathan |
| 5 | Images for capabilities and dice combinations |  |  | Not started | Berret |
| 6 | Capabilities (Warfare Options) | Non-piece purchaseables (effects on gameplay)  Satellite scans, cyber attacks, ....etc | TBD | Not started | TBD |
| 7 | Fuel and Movement Rules | Constraints on how pieces move based on how many moves they have left, how much fuel they have left (handling when fuel reaches 0)  Also preventing plans that exceed moves | TBD | On going | Jonathan  David |
| 8 | Sam's attacking | "SAM Range based on the visibility matrix. When A/C move through the range they are attacked until the A/C are destroyed or out of the SAM range."  Likely to follow V2 Sam logic  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 | TBD | Not started | TBD |
| 9 | Containers | controls / rules / UI / functions for getting pieces into and out of container pieces (fighter on a carrier, SOF team in a C130, tank on a transport) | TBD | Not started | TBD |
| 10 | Satellite scan | This is next step after space area. Customer wants to add new feature; Satellite scan which can scan some part of map to check enemy's units.  See capabilities story, this ties into it. Scan is likely one (or more) of the items available to purchase and use | TBD | Not started | TBD |
| 11 | Space Area | V2, there are only ocean and land area. Customer wants to add space area.  This likely ties into Capabilities story, as these space effects come from using those 'items' or 'capabilities' | TBD | Not started | TBD |
| 12 | 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 | Display the next newsalert (in the order) and stop displaying the old one (when entering phase 0) | Not started | Spencer |
| News Effects are stored when they are encountered, and should expire after the allotted time | Not started | Spencer |
| News Effects have an effect on gameplay (checks occur when making plans / battles / etc....) (how other effects are handled | Not started | Spencer |
| Admins have control over alerts (ordering, adding / deleting, altering?) | Not started | Spencer |

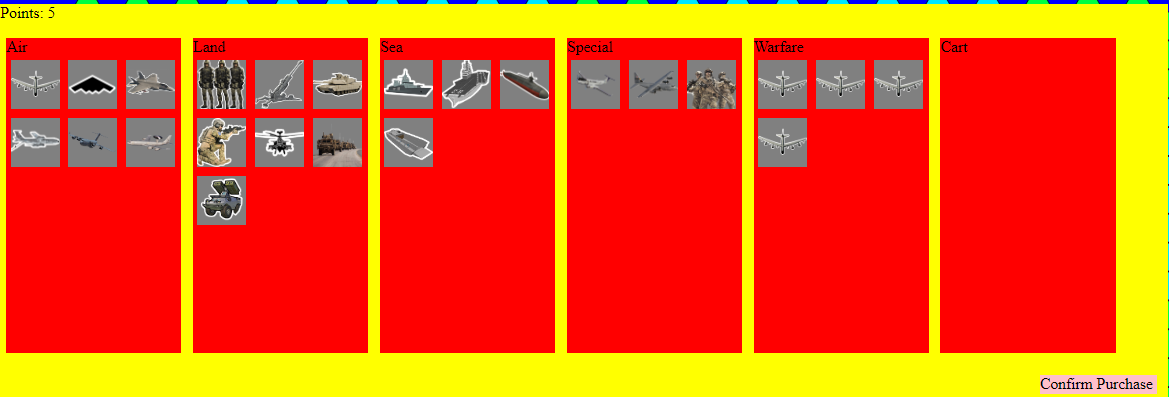
**Work history**

1. Add a game



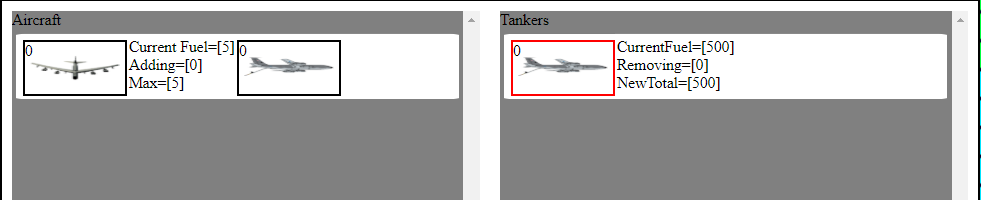
* It can create new game

1. Create new sections for separating units



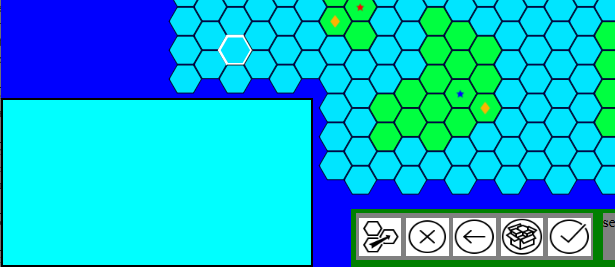
* Previously, units were in one section, so for improving UI, we created new sections, and separate into each section

1. Refuel Air Pieces



* Now, tanker can refuel air unit’s fuel

1. Minimize popup windows



* Previously, popup windows were too big to interfere map. We decided to reduce popup window size optimally

**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.