Senior Design Final Report

Operationalize Networked Collaboration Features for Moon Trek

Jet Propulsion Laboratory
California Institute of Technology

Version 1.0 04/27/2022

<u>Team Members:</u>

Sean Chung, Aldo Gil I, Tommy Lay, Allen Marquez, Tam Nguyen, Alex Sahakian, Andy Tsan, Srivats Venkataraman, Jian Wu, Anna Yesayan

<u>Faculty Advisor:</u>

Dr. David Krum

<u>Liaisons:</u>

Emily Law, Shan Malhotra, Richard Kim, George Chang, Eddie Arevalo

Table of contents

1. Introduction

1.1. Background

NASA JPL's Moontrek is a public website that allows users to access and visualize data of various planets within our solar system. The platform also provides the user with a variety of tools and visual aids, it also allows them to easily navigate and move around the planet. While the application was powerful by itself, there was a lack of collaboration. If the pandemic has taught us anything it's how much we must rely on technologies for getting things done.

While last year's team was capable of developing and integrating the collaborative features, NASA JPL was working on migrating the platform to a new framework. What this meant was the implementation of last year's project wasn't fully compatible with the new framework. This year's team had the responsibility to take on last year's work and migrate it for compatibility with the new framework.

We also had to implement the following collaborative features: States which would allow user(s) to switch between different data layers. Collaborative rooms would allow users to invite new users to collaborate on a session. Annotation would allow users to create markups on the planet's surface, some of the tools were a freehand drawing, multiline tool, and shapes and users could also choose the color and width of each tool. Waypoints would allow users to pinpoint important coordinates on the map. Chat would allow users to communicate with each other using text without leaving the platform. Fly to would allow the users to easily pan to a latitude, longitude position on the map. In order to build this application, JPL required Angular JS for the frontend and Java along with Jersey for the backend.

1.2. Design Principal

We initially worked on getting all the components on the front-end ready and ensured it worked for one user. After this task was completed we started working on the backend, we wanted to get the collaborative session working first before we tackled the other parts of the collaborative features. The hardest part of this was figuring out how to send the data between each user in a collab session without delay.

1.3. Design Benefits

Being a web application it's easy to be integrated into the newly developed framework. It is also easy for people to get access to since you would just need a modern-day browse and access to the internet. This means you could access the website from anywhere in the world. The barrier to entry is also relatively low.

1.4. Achievements

During this academic year, the team achieved a lot of stuff. The most important is the team being able to add collaborative features. As mentioned above collaborative features allowed the user to share a session id for other users to join. Draw on the terrain which would update for other users in the same session. Switch between different data points called states. Chat with each other using text, mark important coordinates using waypoints, and lastly easily navigate the map using Fly To.

The biggest achievement of this year's team was

After working on the Moontrek application for one academic year, our team completed a multitude of features; the first being- changing the entire backend codebase from Javascript to Java. The code upgrade ultimately allows project members to implement more rigorous features into the system since Java is faster and more powerful. With the basis of the backend developed, the team then went on to create a few core features which include: "Fly to", sessions, collaboration rooms, chatbox, annotation tool, and "LatLong".

To start, the "Fly To" function allows users to redirect other participants' screens to a location on the surface map with the use of a waypoint. Next, users can host rooms using the collaborative rooms feature. After sharing a generated link, other clients can join the host's room to interact with each other. The chatbox feature allows users in the same room to communicate with each other via text. On the other hand, the draw tool allows users to create sketches on the surface map for everyone on the server to see.

Moving on, if a user wishes to save the surface map at a certain state (ex. there is a drawing on the map which coordinates a mission plan), they could save it with the sessions feature under the session tab. Finally, users can visit specific locations on the surface map using the "LatLong" feature. They can do this by entering the x, y coordinate into the textbox located on the right-bottom corner of the web application.

2. Related Works and Technologies

2.1. Existing Solutions

We didn’t find any existing solutions. The only codebase we had was what JPL provided us during the start of the scroll year. We made use of this codebase to understand how last year made the components work and what we could reuse from last year's project in our own project.

2.2. Reused Product

Since our codebase was completely different from last year's project we couldn't reuse any of the components.

3. System Architecture

3.1. Overview

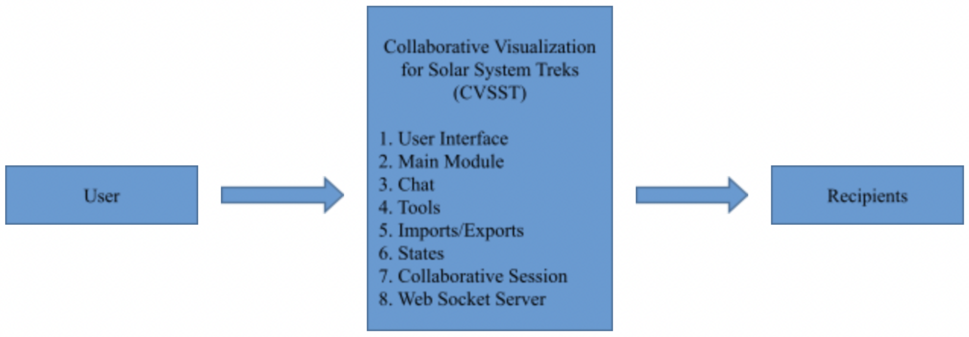

Figure 1: Overview

Since our project was built on top of an existing software we didn't want to change the underlying code too much. Figure 1: Overview shows how data flows between a user and the intended recipients. All the models in Moontrek are initialized after the main module has been initialized.

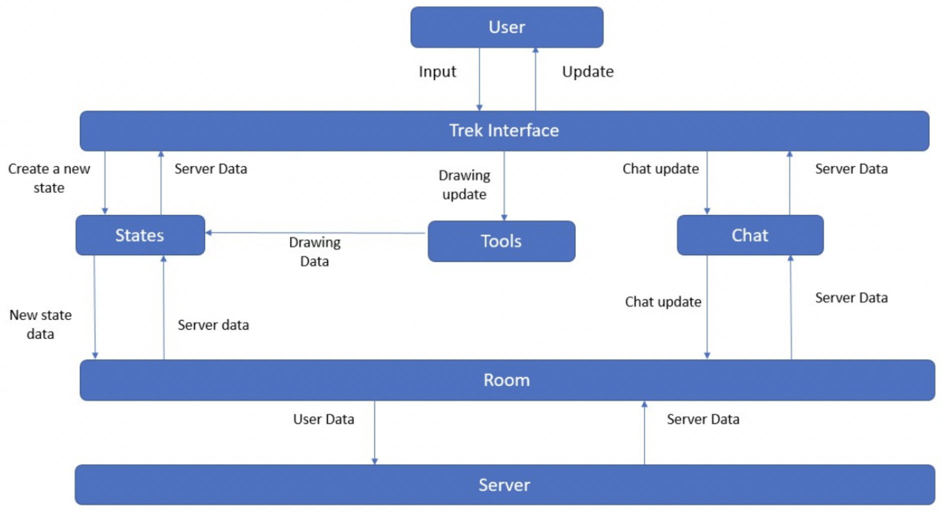

Figure 2: Dataflow

3.2. DataFlow

The dataflow for the application is shown in Figure 2: Dataflow

3.3. Implementation

User Interface Module

- Provides a Graphical User Interface (GUI) for the user to interact with the underlying system
- Implemented using mostly Angular JS
- The user interacts with button clicks, text and maybe scrolling and dragging the mouse.

Main Module

- This is where all the modules for the application are initialized

Chat

- The chat module is responsible for sending and showing new chat messages
- The user interacts with this module by entering text into the text box and hitting send button to send the data via the WebSocket.

Tools

- Tools consist of the various markup tools provided in the system
- The user selects a tool and then can adjust the width and the color of the tool
- The data is then captured and transmitted over the WebSocket

Import/Exports

- Not applicable at the moment.

States

- Allows users to choose between different saved datapoint or start on a new layer
- The data points are then sent over a WebSocket connection for other users to see the updated data.

Collaborative Session

- This is an online session where one user creates a room and shares the room id with other users
- This allows everyone to share ideas via text or by drawing on the terrains

Web socket server

- The web socket server is responsible for creating the collaborative session and for also sending data between users in the same room

We made use of an Agile Methodology for our development as it allowed us to tackle more in less time. Some of the features were harder to implement so there would be times when more people might be working on one feature.

4. Results and Conclusions

4.1. Results

Throughout this semester we were able to create collaborative sessions which allowed users to create and share collaborative rooms. We were also able to add annotation features which allowed users to create markups on the terrains. We were also able to allow users to communicate with one another using text chats. We were also able to give users the features to switch between different data points called states and also create new states. We were also able to let people easily navigate the terrain by making use of the fly to feature which allowed users to enter latitude and longitude positions.

There are some improvements and features that still need to be completed. They are listed below:

- State Management:
  - Currently, to instantiate our collaborative features we are making use of a brute force method which isn't ideal
  - Setting up state management would be ideal
- Waypoints
  - Currently, waypoints only allow users to name waypoints and nothing else
  - The ideal solution/fix for this would be to figure out how to make waypoints useful
- More tools
  - Currently, there are only 4 tools that are working
  - Maybe adding on to the toolsets would be nice.

While we were able to complete most of the collaborative features, they are not fully perfect. Learning a new framework like Angular was a bit of a challenge for most of us. And the lack of good documentation for ESRI ArcGis also delayed the work we could put in for some of the features.

4.2. Future

A future team could look at implementing state management, fixing waypoints, adding more tools, and also maybe adding 3D annotations. Maybe also adding host privileges and the ability to remove participants.

5. Reference

1. https://angular.io/
2. https://v7.material.angular.io/
3. https://developers.arcgis.com/javascript/latest/api-reference/