**Allsky Camera Network for Detecting Bolides**
**Milestone 1**


| **Members** |
| --- |
| Tyler Turner, tturner2021@my.fit.edu |
| Vincent Quintero, vquintero2021@my.fit.edu |
| Jean-Pierre Derbes, jderbes2021@my.fit.edu |
| Charles Derbes, cderbes2021@my.fit.edu |
| **Faculty Advisor/Client** |
| Csaba Palotai, APSS, cpalotai@fit.edu |


Progress Matrix:

| Task | Completion | Tyler | Vincent | Jean-Pierre | Charles |
| --- | --- | --- | --- | --- | --- |
| Diagnose current issues | 75% | Fix stop camera error | Image composites, Frontend | Fix state (consolidate into a single source of truth) | Fix unknown camera control errors (particularly when changing exposure modes) |
| Create system architecture diagram | 100% | All contribute | All contribute | All contribute | All contribute |
| Storyboard the frontend | 100% | Brainstorm | Create | Brainstorm | Brainstorm |
| API design | 100% | All contribute | All contribute | All contribute | All contribute |
| Design CLI for image processing pipeline | 50% | Create | Brainstorm | Brainstorm | Brainstorm |

Discussion of each accomplished task for the current Milestone:

- Task 1: We have fixed bug after bug but when we fix one, two more appear. We have decided to rewrite a portion of the code that is giving us the most errors. This will take some time, but after it is finished we can start rewriting everything.
- Task 2: The proposed architecture should be much more reliable and reduce the complexity for the researchers. In the past month, we have witnessed the struggles that the researchers must go through to maintain the nodes, and making their work easier is a top priority of this rewrite.
- Task 3: We received feedback from both the researchers and Dr. Palotai and they both loved the idea of the centralized UI. The wireframe created demonstrates how to select a node to manage and how to manage it.
- Task 4: Having designed many APIs in the past, this was not a difficult task but one complication was that we needed to decide how we wanted to handle two-way communication. We ultimately decided to have two RESTful APIs.
- Task 5: We have not made too much progress with the CLI because we do not have all of the information that we need to make the CLI. We have a portion of the CLI designed, but there will be more added later on as we get feedback from Dr. Palotai.

Discussion of contribution of each team member to the current Milestone:

- Tyler Turner:
  - Fixed bugs in current software
  - Designed API
  - Did most of the communication between the team and advisor/researchers
- Vincent Quintero:
  - Input on API design, architecture design
  - Assist on fixing C++ Sentinel communication bugs
  - Create better bolide composite tooling
- Jean-Pierre Derbes:
  - Fixed bugs in sentinel camera state
  - Fixed bugs in communication between sentinel camera C++ and Python
  - Provided input on API design for server and node database
- Charles Derbes:
  - Fixed unknown camera control bug
  - Provided input on API design
  - Worked on ground up architecture redesign

Task Matrix for Milestone 2:

| Task | Tyler | Vincent | Jean-Pierre | Charles |
|---|---|---|---|---|
| Continuously fix endless stream of issues | All contribute | All contribute | All contribute | All contribute |
| Add logs for easier diagnosis of issues | 0% | 90% | 0% | 10% |
| Replace current C++ camera code | All contribute | All contribute | All contribute | All contribute |
| Implement Server API | 70% | 5% | 25% | 0% |
| Implement Client API | 0% | 0% | 30% | 70% |
| Begin writing CLI | 33% | 33% | 33% | 0% |
| IoT style setup | 20% | 0% | 30% | 50% |

Discussion (at least a few sentences, ie a paragraph) of each planned task for the next Milestone:

- Continuously fix endless stream of issues:

  A constant problem, certain boxes have certain issues and as these issues pop up we have to fix them. Typically users wonder why certain pieces of hardware are failing, why the UI breaks a certain way etc. At the moment we have a lot of poll timeout issues.

- Add logs for easier diagnosis of issues:

  Proper logs need to be added so we know which buttons were clicked, in which order, and what the corresponding effect on the system was. Currently our “logs” are very high level and only record the “end”, ie an error or a success message, but nothing before.

- Replace current C++ camera code:

  Cherrypy calls C++ functions but we would like to change it to call functions that use OpenCV. This essentially means we will be reimplementing all of the camera code from scratch. This will involve lots of testing to ensure that functionality works as intended.

- Implement Server API:

  Implement the server API in Golang. The server is mainly responsible for managing events (video, light curve data, composite image), and also managing the health of nodes. The server also manages communication with the user through the notification endpoint.

- Implement Client API:

  Implement the client API using FastApi. The client (a node) will be able to provide the server with information on its status and the local config. The config can also be updated by the server.

- Begin writing CLI:

  The CLI is a tool for researchers to manually do certain things such as generate composites, get light curve data, etc. The CLI exposes internal functionality to researchers giving them more resources and flexibility for analysis.

- IoT Style Setup

  Nodes will connect to the local network using an IoT style setup rather than having to tediously plug in a keyboard and monitor in order to connect. Notifications will be sent to the user to guide them through the setup and alert them if something goes wrong.

Dates of meetings with Client during the current milestone:

see Faculty Advisor Meeting Dates below

Client feedback on the current milestone:

see Faculty Advisor Feedback below

Dates of meetings with Faculty Advisor during the current milestone:

- Sep 11, 2024
- Sep 13, 2024
- Sep 16, 2024
- Sep 18, 2024
- Sep 25, 2024
- Sep 26, 2024
- Sep 27, 2024

Faculty Advisor feedback on each task for the current Milestone:

- Task 1: Dr. Palotai recognizes the improvements that we have made even though it is not finished.
- Task 2: Dr. Palotai approves of the system diagram.
- Task 3: Dr. Palotai believes that the proposed user interface will benefit the researchers in managing the nodes.
- Task 4: Dr. Palotai approves of the API design.
- Task 5: Dr. Palotai likes that the data processing will be easier with a command line interface.

Faculty Advisor Signature: _____ Date: _____