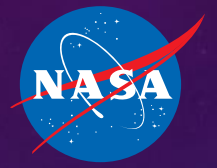
National Aeronautics and Space Administration



Improvements And Bug Fixes to the Astrobee Control Station

Kalie Ching

JOSE BENAVIDES, AMES RESEARCH CENTER - TI

SUMMER 2021

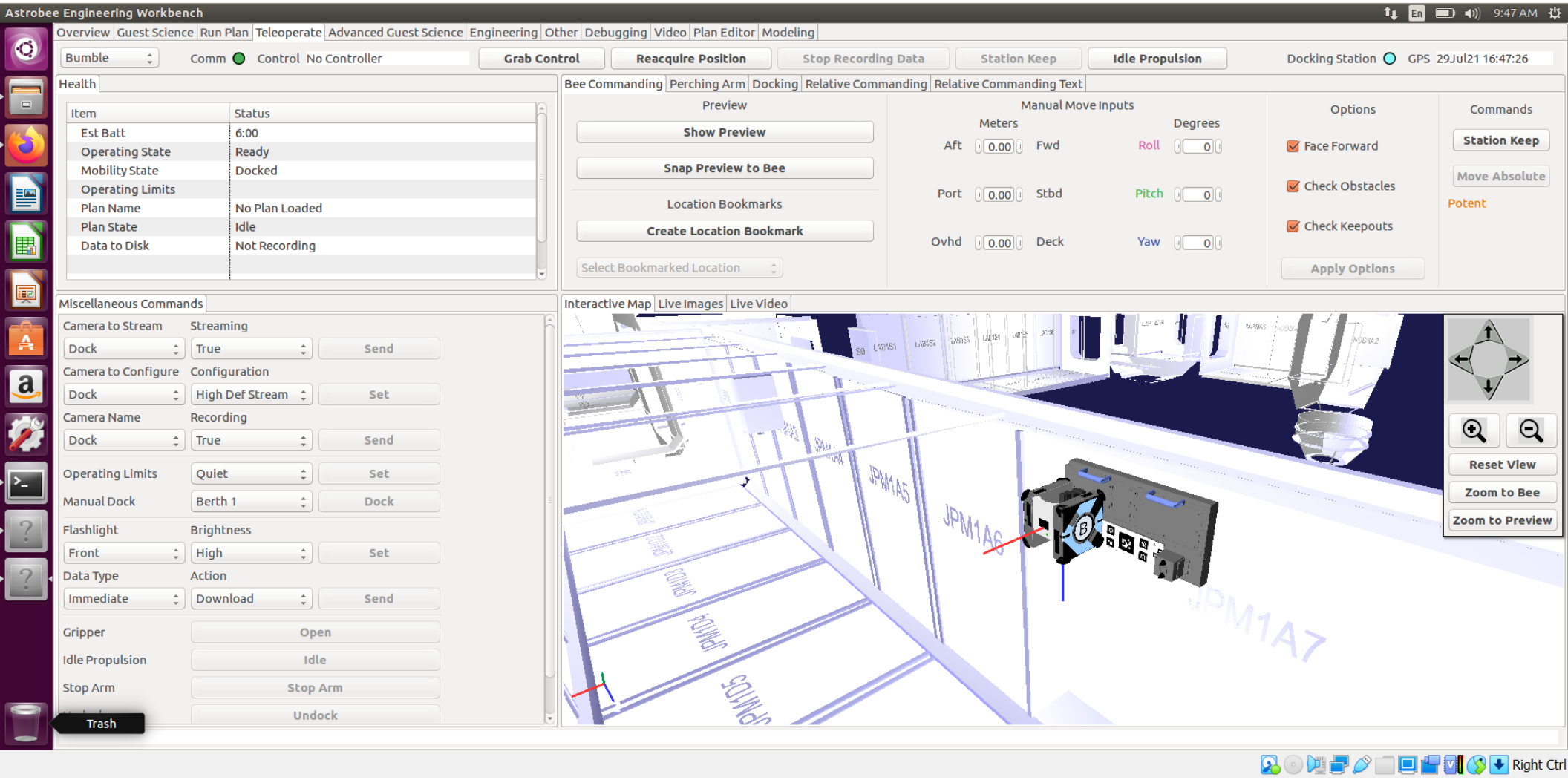

Thank you for this opportunity.



Introduction

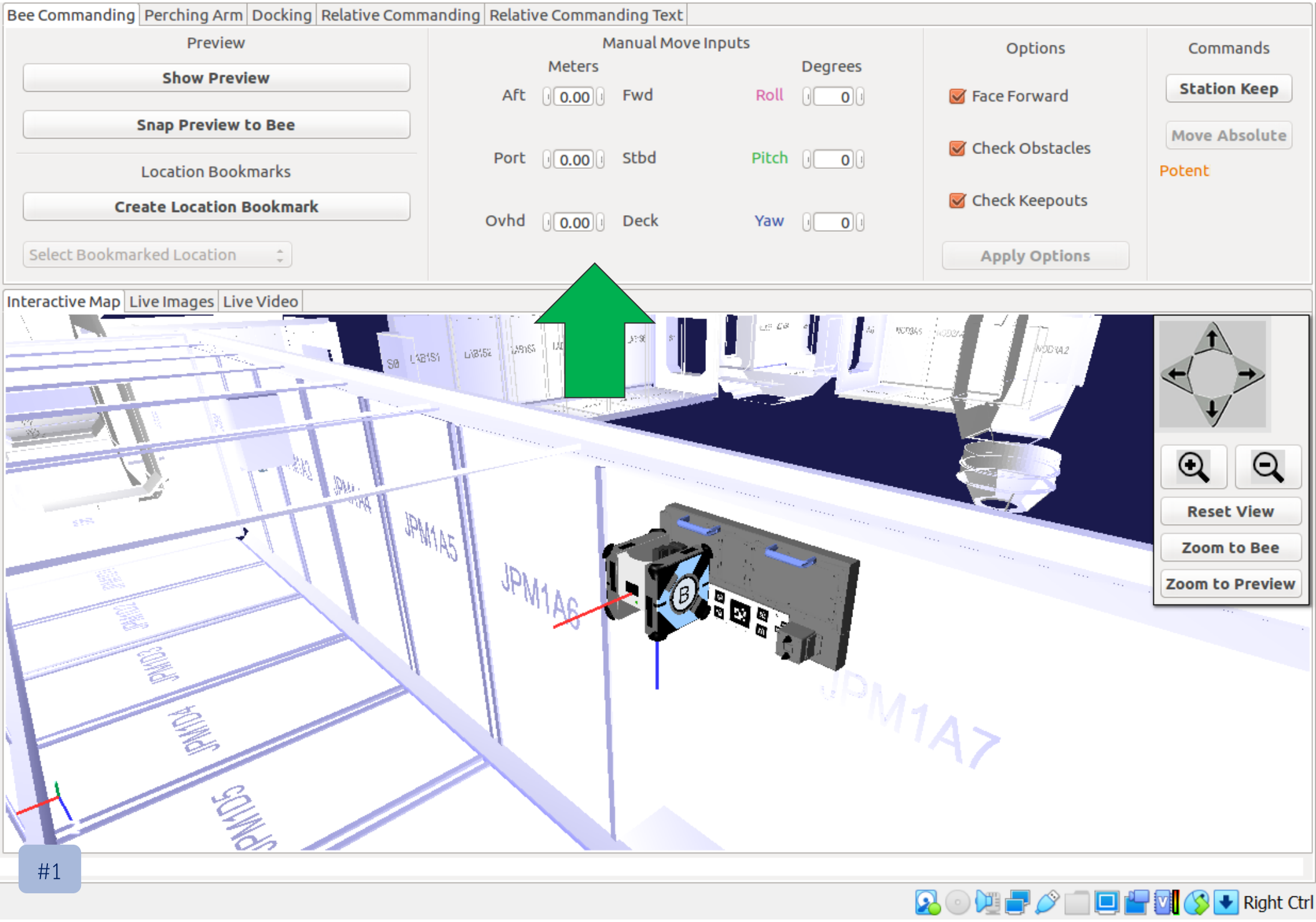
Ground Data System (GDS)

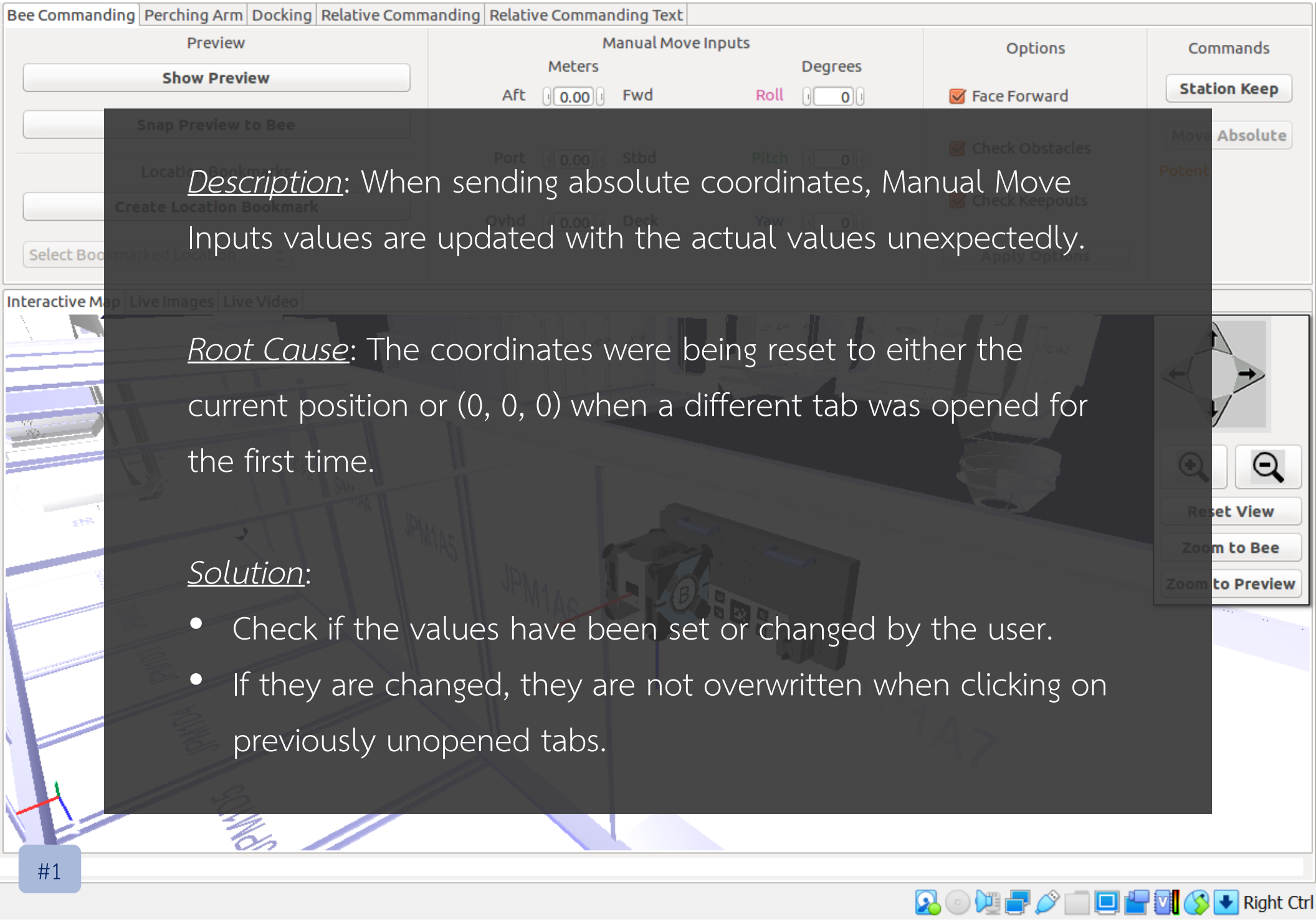
Teleoperation Tab





Bug Fixes + Improvements


Bee Commanding
Perching Arm
Docking
Relative Commanding
Relative Commanding Text
Preview
Show Preview
Snap Preview to Bee
Location Bookmarks
Create Location Bookmark
Select Bookmarked Location
Manual Move Inputs
Meters
Degrees
Aft
0.00
Fwd
Roll
0
Port
0.00
Stbd
Pitch
0
Ovhd
0.00
Deck
Yaw
0
Options
Face Forward
Check Obstacles
Check Keepouts
Apply Options
Commands
Station Keep
Move Absolute
Potent
Interactive Map
Live Images
Live Video
Reset View
Zoom to Bee
Zoom to Preview
JPM1A5
JPM1A6
JPM1A7
#1
Right Ctrl

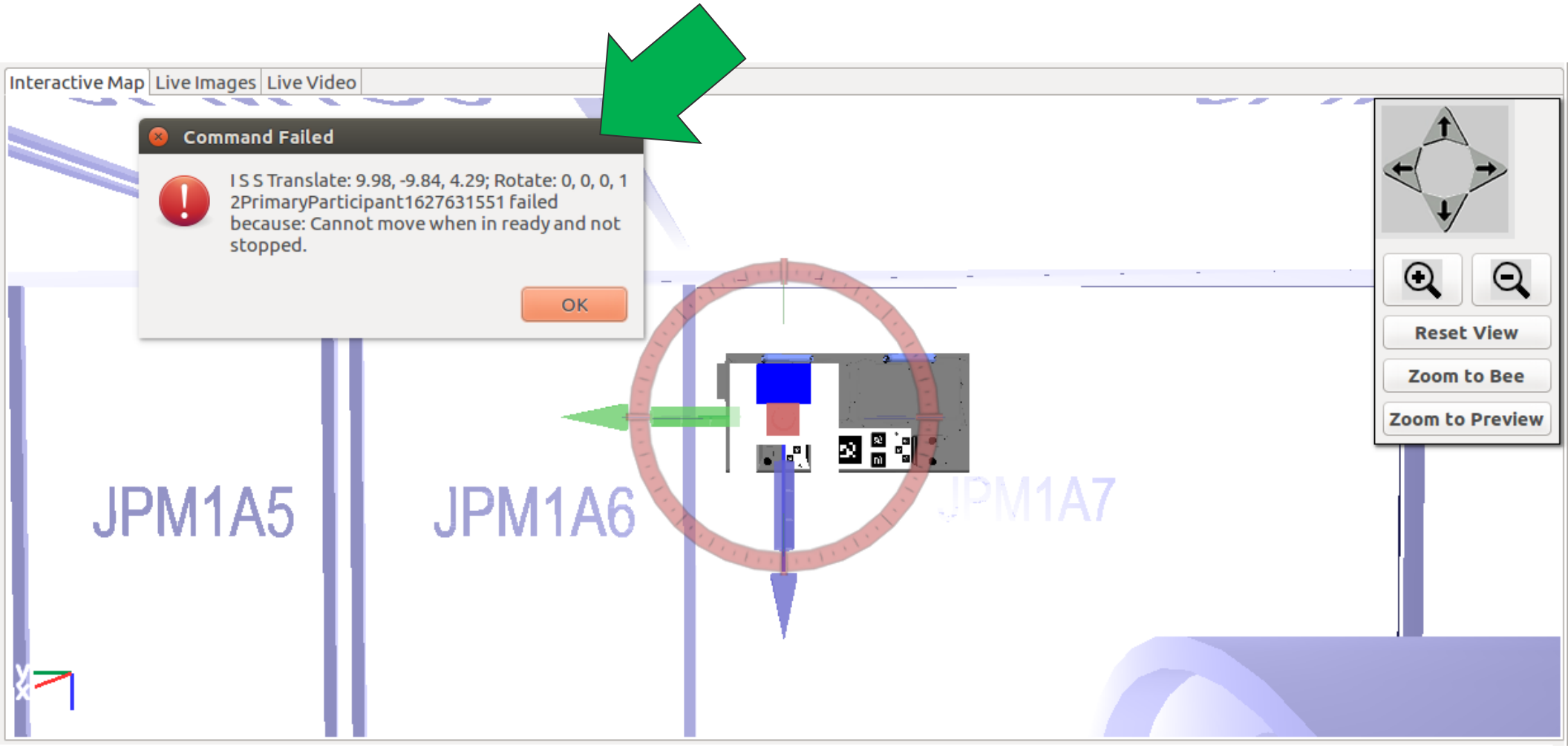


Description: When sending absolute coordinates, Manual Move Inputs values are updated with the actual values unexpectedly.

Root Cause: The coordinates were being reset to either the current position or (0, 0, 0) when a different tab was opened for the first time.

Solution:

- Check if the values have been set or changed by the user.
- If they are changed, they are not overwritten when clicking on previously unopened tabs.


Interactive Map
Live Images
Live Video
Command Failed
I S S Translate: 9.98, -9.84, 4.29; Rotate: 0, 0, 0, 1
2PrimaryParticipant1627631551 failed
because: Cannot move when in ready and not
stopped.
OK
Reset View
Zoom to Bee
Zoom to Preview
JPM1A5
JPM1A6
JPM1A7

#2

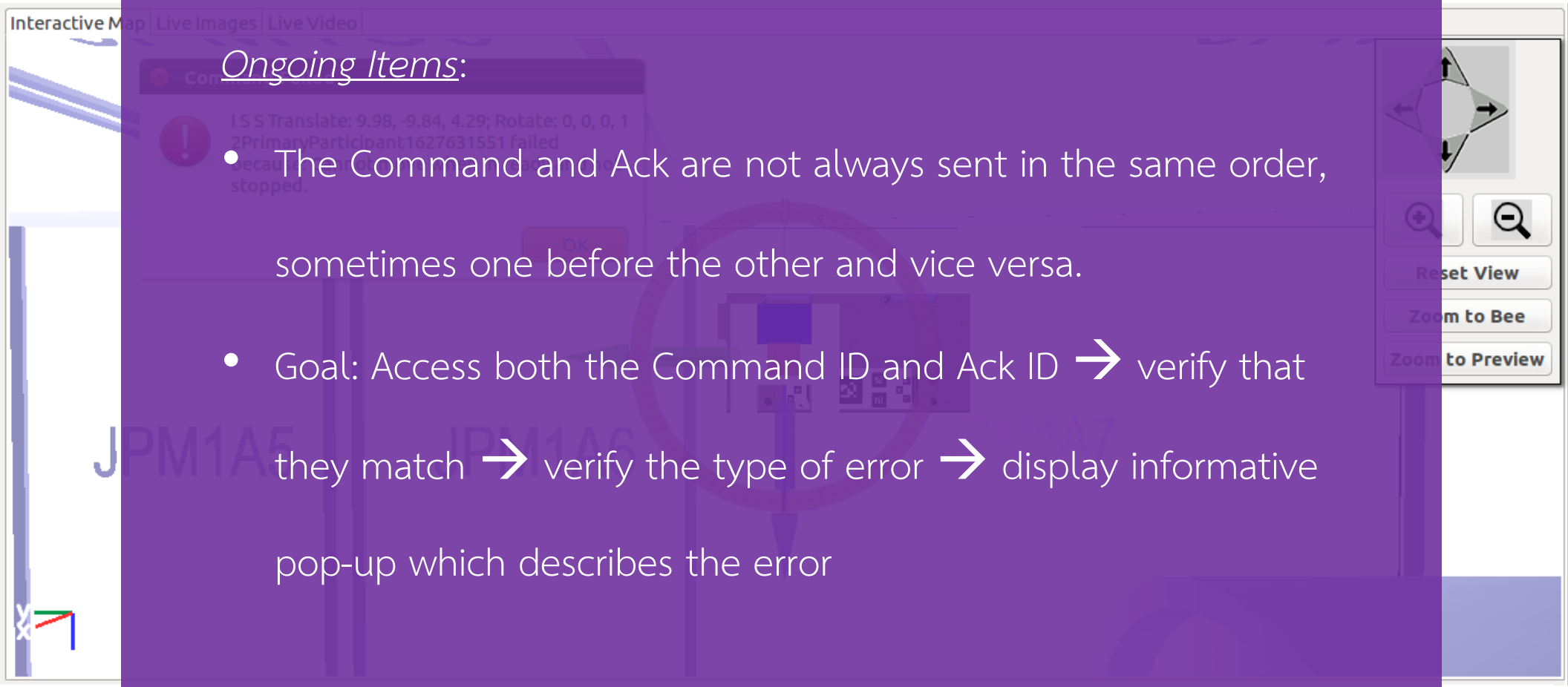
Description: In the instance where errors occurred, too many error pop-up windows would appear at once.

Related problem found: The time was set incorrectly, causing the function to never complete; therefore, when using the simulation, pop-ups didn't occur.

Solution:

- If the most recent Command ID matches the Ack ID, show the error pop-up; errors occur only if there is bad syntax or if the command failed.

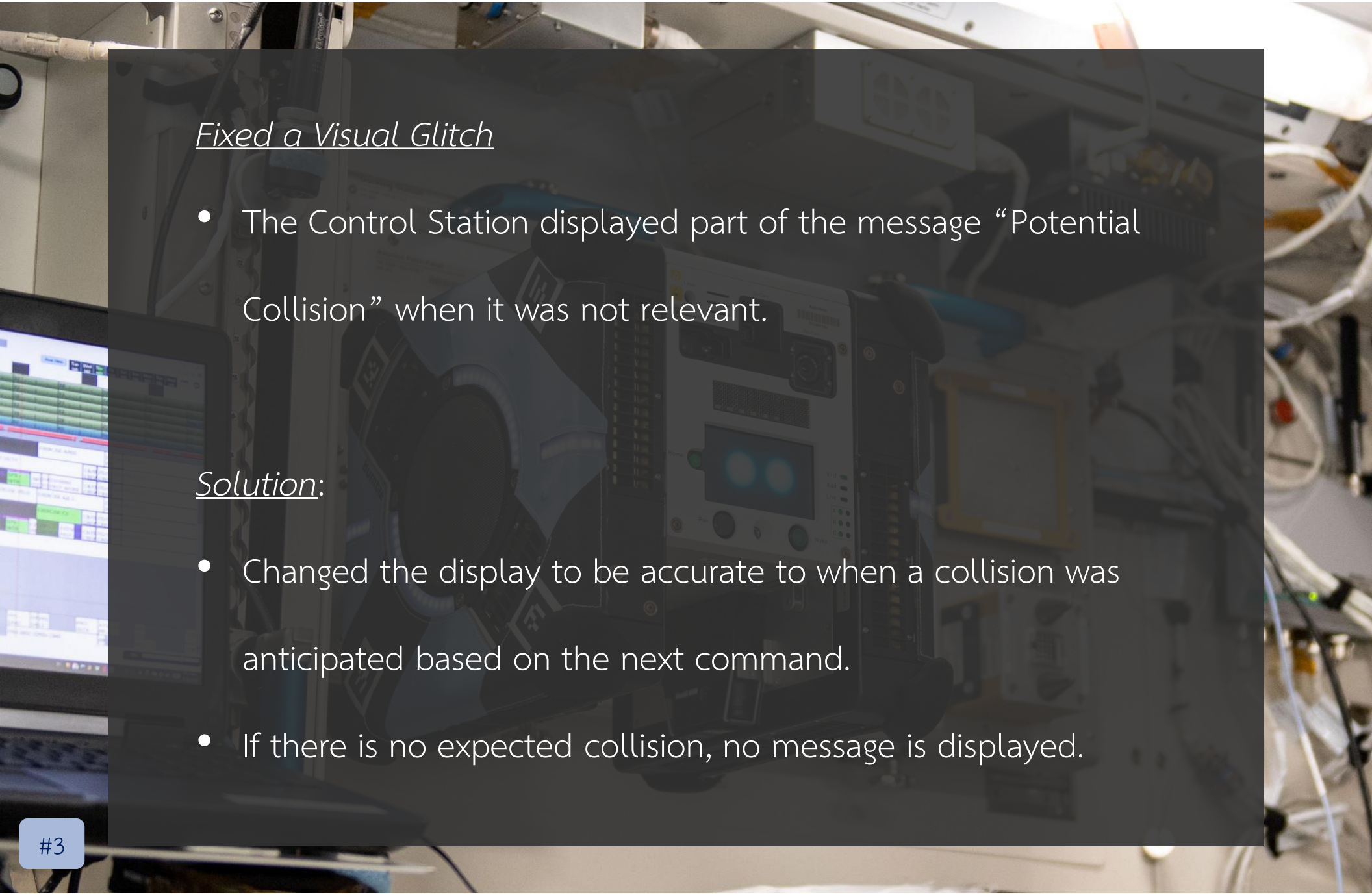
#2



Ongoing Items:

- The Command and Ack are not always sent in the same order, sometimes one before the other and vice versa.
- Goal: Access both the Command ID and Ack ID → verify that they match → verify the type of error → display informative pop-up which describes the error

#2



Fixed a Visual Glitch

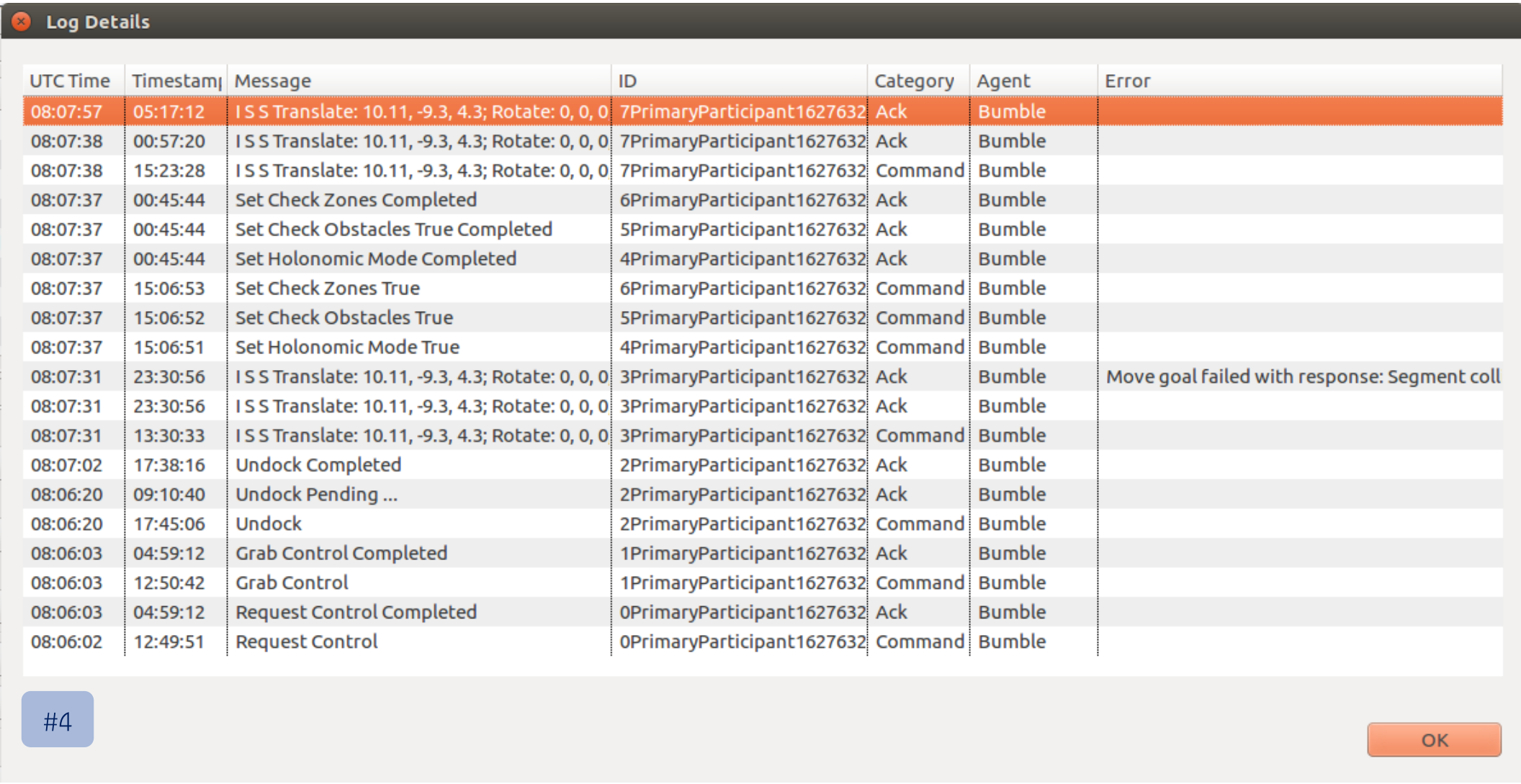
- The Control Station displayed part of the message “Potential Collision” when it was not relevant.

Solution:

- Changed the display to be accurate to when a collision was anticipated based on the next command.
- If there is no expected collision, no message is displayed.

#3

- Timestamps were out of order.
- For Commands, the function that set the time required milliseconds, but time was passed in in microseconds.
- Time was also reported incorrectly from the flight software for Acks.



- A new column was added as a feature of the Command Log.
- The column is labeled “UTC Time" and displays the accurate time the Command or Ack was sent.
- The time zone is UTC, and matches the time displayed in the upper right corner of the Control Station.
- The comparator was updated as well; clicking on the title orders the Command Log entries.

#4



Learnings + Takeaways



Learned how Eclipse applications are developed and became familiar with a large codebase



Learned and practiced debugging techniques, designed and tested multiple approaches



Gained insight on the importance of communication and collaboration

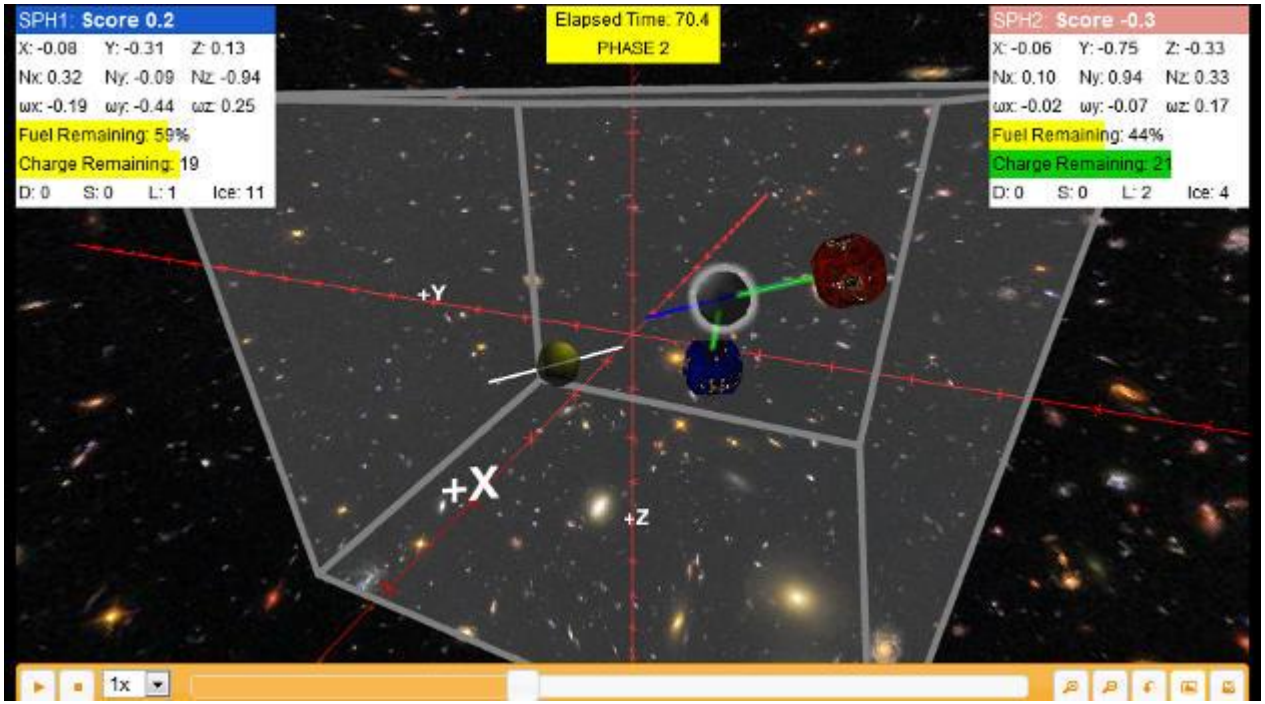


Understood the importance and usefulness of simulations in development



Learned about how the Astrobee robots are used and what operations look like behind the scenes

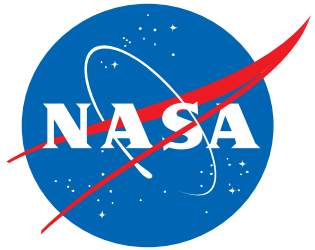

ZERO ROBOTICS
ISS PROGRAMING CHALLENGE
• Working with the Zero Robotics team to transition from using SPHERES to Astrobee
MIT
MIT SSL
Aurora FLIGHT SCIENCES
ilc innovation learning center
NORTHROP GRUMMAN Foundation
CASIS
esa
NASA


SPH1: Score 0.2
X: -0.08 Y: -0.31 Z: 0.13
Nx: 0.32 Ny: -0.09 Nz: -0.94
ωx: -0.19 ωy: -0.44 ωz: 0.25
Fuel Remaining: 59%
Charge Remaining: 19
D: 0 S: 0 L: 1 Ice: 11
Elapsed Time: 70.4
PHASE 2
SPH2: Score -0.3
X: -0.06 Y: -0.75 Z: -0.33
Nx: 0.10 Ny: 0.94 Nz: 0.33
ωx: -0.02 ωy: -0.07 ωz: 0.17
Fuel Remaining: 44%
Charge Remaining: 21
D: 0 S: 0 L: 2 Ice: 4
+Y
+X
+Z
1x


SPHERES


Thank you!


Questions?


NASA

Exit Presentation Script

Kalie Ching

1. Hello all. My name is Kalie Ching, and I am a rising sophomore from the Bay Area studying Computer Science at UC Berkeley. This summer, I got the opportunity to work with the Astrobee Facility team and develop the Ground Data System software, otherwise known as the Control Station, as well as aid the Zero Robotics team in transitioning from using SPHERES to Astrobee for their student competition.
2. First of all, I'd like to say thank you for this opportunity, answering my questions along the way, supporting and encouraging me, and providing an amazing environment to learn and gain new skills. I'm so excited to share with you all what I have learned and have been working on in the past few months.
3. Coming into this project, I knew very little about the software running on and controlling the Astrobee robots, how they functioned within their space on the ISS, and how different operations sessions would play out. In the past three months, I got the chance to work directly with the Astrobee simulation, control software, and attend operations meetings where the facility team, guest scientists, and astronauts all worked together to complete a goal using Astrobee.
4. A bulk of the work I have done throughout this summer is on the Ground Data System, an Eclipse Rich Client Platform application which commands the free flying Astrobee robots on the ISS. GDS has 11 different tabs for members in the operations team as well as guest scientists to send commands, change settings, and view live feedback. I have worked on bug fixes, improvements, and new features to the GDS software for operations to run with fewer hiccups.
5. Now, let's get into the bugs and features I have worked on.
6. The first bug requires us to take a closer look at specifically the teleoperate tab. If you notice, you can see that under the Bee Commanding subtab we are currently on, the Manual Move Inputs section allows for you to type in coordinates and rotation values to move the robot to if they are valid. These values are absolute, which is the location of the robot in the context of the three-dimensional ISS space.
7. The bug was that when you typed in absolute coordinates, the Manual Move Inputs values were updated with the actual values unexpectedly. I found that the coordinates were being reset to the most recent position the robot was at, or 0, 0, 0 every time a new tab was opened. The solution to this was to add a check to see if the values had already been set or changed. If they were, they were not overwritten. The challenges I ran into included recreating the bug consistently and narrowing down the functions and variables that controlled this functionality. For example, the variables were being updated in one location, but being reset elsewhere, so it was particularly difficult to find where they were being reset.
8. This next bug has to do with too many error dialogs popping up on top of each other. An example of an error dialog can be seen in the picture here; this specific error is caused when you try to move the robot when it's not in the right state. The pop-up shows some information about the intended command, the ID, and the error message.
9. When testing for this bug, the simulation was initially unable to reproduce this bug. I found that no error pop-ups would display because the internal time was reported incorrectly, which was a related bug that directly affected this problem. From some discussions on this bug, we

concluded that error pop-ups should only show up for errors that match the command that was just sent, instead of being based on time. The solution to this was to check if the ID of the most recent command sent matched the Ack ID, and only then showing the error dialog.

10. The ongoing items related to this bug include fixing the fact that sometimes, the command and ack are not always sent in the same order, sometimes one before the other. Our goal is to have this process as described on the slide as the final fix. We want to access the command ID and Ack ID, verify that they match, verify the type of error, and then display an informative pop-up which describes the error.
11. I also fixed a visual glitch on the GDS, where previously, the code displayed part of the message "potential collision" when it was not relevant. The solution to this was that I changed the display to be accurate based on whether there was a collision anticipated or not.
12. Because of bugs in timestamps, the timestamps were also displayed out of order in the command log, a feature that allowed the user to see all the commands and related acks that had been sent to the robot and by the robot respectively. The time was also reported incorrectly from the flight software for each ack, which caused problems in the order and timing
13. The solution was to create a new column using the Java Date() class that recorded the time local to the computer to allow the user who is commanding the robot to retrace the exact commands they had sent and their resulting behavior. Since the command log is in the form of a data table, I updated the comparator as well, so that users could click on the title to order the command log entries.
14. Now, I'd love to talk about some of the learnings and takeaways from this experience.
15. First of all, I learned how Eclipse applications are developed and became familiar with a large codebase. By breaking down big tasks into smaller ones, observing which functions called which, and studying past commits, I was able to distinguish which parts of the code needed to be adjusted.
16. I also learned and practiced debugging techniques and tested multiple approaches to each problem. For one bug, I had to test up to 4 or 5 different approaches before one worked out.
17. Communication and collaboration were also vital throughout this entire process. I sent weekly updates to Jose, attended the weekly meetings to stay updated, documented my progress, and asked many questions.
18. This was also the first time I have ever worked with simulations, and I have come to understand the importance and usefulness of simulations when developing. It saved time and allowed for more flexibility in testing.
19. Finally, I was able to get a unique look into how Astrobee robots are used, and what operations sessions look like behind the scenes. It's so exciting to see the experiments the team and external users do with the robot.
20. In the past month, I have been communicating with the Zero Robotics team to help them to transition their platform from using SPHERES to Astrobee. Zero Robotics is a robotics programming competition that allows participating students to program the robot on the ISS to complete a challenge, gaining programming and robotics skills along the way.
21. This work has consisted of helping the team become familiar with the Astrobee simulation, assisting them throughout setup, and pinpointing the differences between SPHERES and Astrobee in order to update the online platform that participating students use. Here you can see the SPHERES interface the students interacted with, as well as an image of the actual

SPHERES robot. I view this as the capstone to my time here, as I can pass on the familiarity I have gained by using the simulation and GDS to an Astrobee collaborator.

22. Thank you again to Jose for being an amazing mentor, thank you to Moeez, Katie, Ruben, Roberto, Jonathan, and everyone on the team who answered my questions and helped me throughout the internship. And thank you all for listening and joining me for this presentation today.
23. Questions?
24. Finishing slide