Feature Documentation:
Full Camera Orientation/Rotation

for

NASA EVA Path Phase 3

Version 1.1

Developer
Lincoln Powell

Product Owner
Daren Welsh

Test Team
Tenadam Weldesemayat

University of Maryland University College
SWEN 670 9040 (2185)
Software Engineering Project
Dr. Michael Brown

August 11, 2018

Table of Contents

i. Revision History

Revision	Author	Date	Description
1.0	Lincoln Powell	8/10/18	Initial document.
1.1	Tenadam Weldesamayat	8/11/18	Reviewed document and added as part of test team.

1. Introduction

1.1 Background

The finalized NASA EVA Navigator web application delivered from Phase 2 completed tailored backlog items designed by Daren to meet existing objectives for the product, one being to embed and allow manipulation of the International Space Station (ISS) 3D model on the webpage. The model facilitated basic interactions (e.g. rotation and zoom); however, it was identified by Daren that the model could not switch x-/-y axes to reorient the model nor could it fully rotate. Since end users would be using the application as a tool to simulate space navigation on the ISS, Daren needs the application to allow users to fully orient and rotate the ISS model.

1.2 Intent

Daren's intent for this change is to investigate if constraints exists which prevent full orientation and rotation of the ISS model. Additionally, based on research results, if a solution can be identified and incorporated, make the needed changes to allow users to fully orient and rotate the ISS model.

1.3 Agreed Change(s)

Upon code review, Lincoln discovered that the ISS model was using a ThreeJS camera orientation, called OrbitControls, which allows the camera to orbit around a target; however, has the intended limitation to restrict camera orientation to the poles of the target.

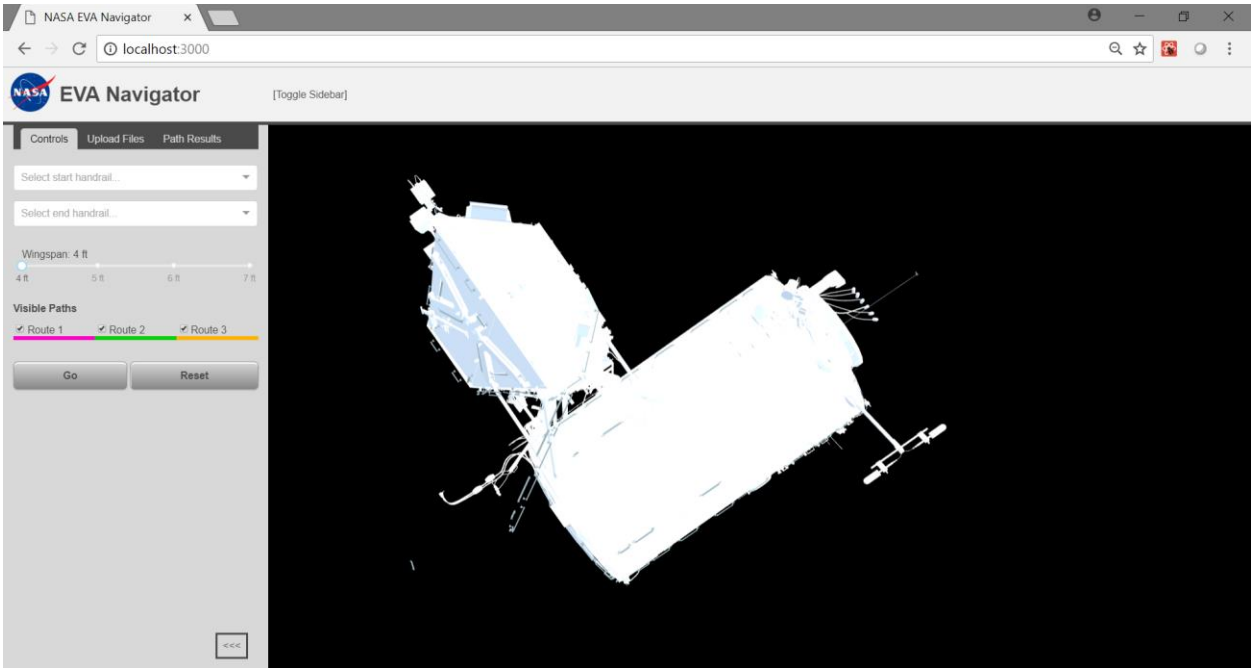

Figure 1. Attempting to orient camera beyond the north pole of the ISS model.

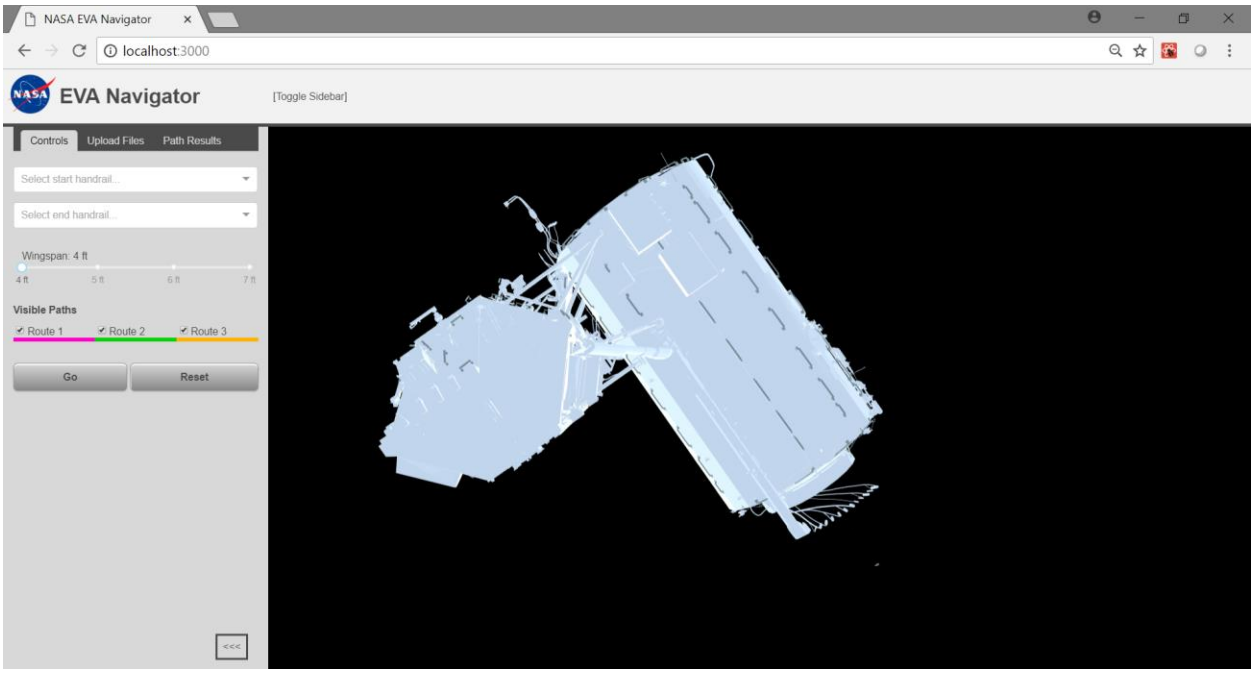

Figure 2. Attempting to orient camera beyond the south pole of the ISS model.

More, OrbitControls prevents target tilting, keeping the target locked on the y-axis.

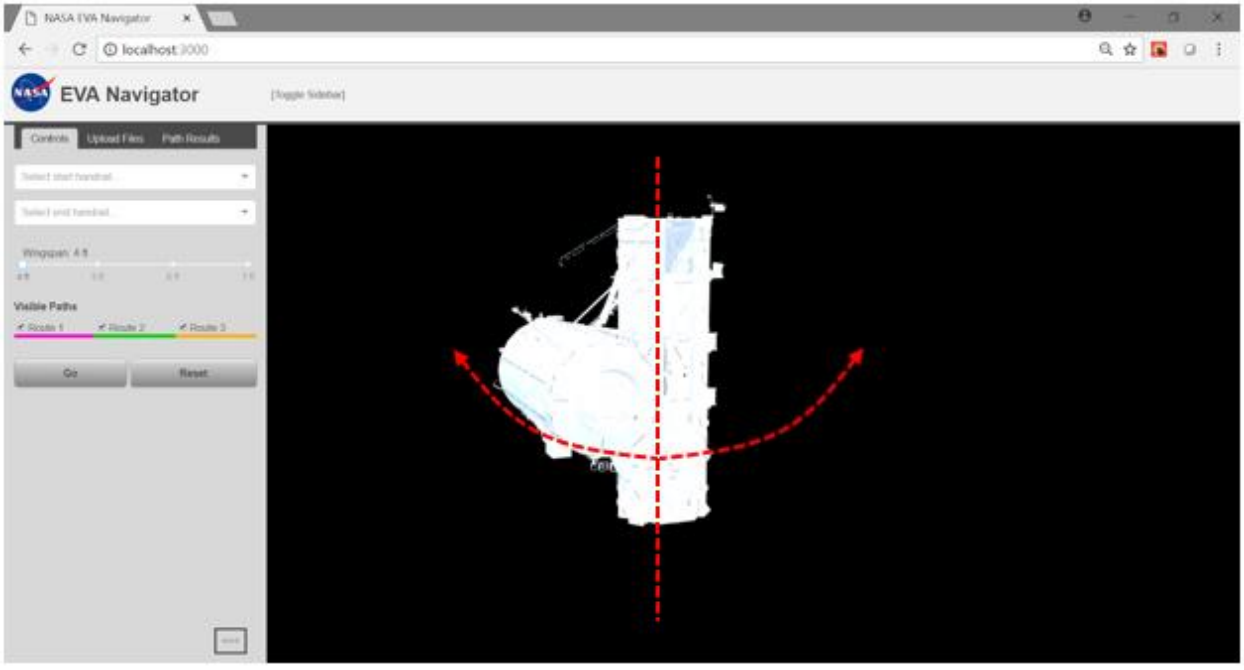

Figure 3. Displaying ISS model y-axis locking (dashed red vertical line) allowing full rotation around y-axis.

In order to circumvent the limitations of OrbitControls, Lincoln suggested using TrackballControls, which is the appropriate camera orientation Daren is seeking that is meant for rotating the camera fully around a target.

2. Development

2.1 Code Additions or Modifications

Similar to how the OrbitControls functionality existed via import, in order to change the camera orientation to use TrackballControls, adding a new, open-source JavaScript file was required. As such, the file, TrackballControls.js, was added in /src/utils/ from GitHub repository, three.js (https://github.com/mrdoob/three.js/blob/master/examples/js/controls/TrackballControls.js) by mrdoob, then imported to the Renderer.js file. Next, two new global variables were declared and initialized:

- TrackballControls *controls*
- Clock *clock*

Then, the TrackballControls *controls* object was configured to allow for panning and a more responsive rotate, zoom and pan speed.

```
106     //PHASE 3 MOD Lincoln Powell/lpowell25@student.umuc.edu 8/10/2018
107     // mouse controls to rotate/zoom the model
108     controls.rotateSpeed = 3.0;
109     controls.zoomSpeed = 3.0;
110     controls.panSpeed = 3.0;
111     controls.staticMoving = true;
```

Figure 4. TrackballControls *controls* configuration.

Finally, the Renderer.js *animate* function was modified for using the TrackballControls object.

```
440   //animate scene movement
441   animate() {
442       this.stats.update();
443       var delta = clock.getDelta();
444       controls.update(delta);
445       requestAnimationFrame(this.animate);
446       this.renderer.render(this.scene, camera);
```

Figure 5. *animate* function.

3. Functional Testing

3.1 Proposed Functional Test Cases

3.1.1 Test Case 1: Orient ISS model by switching x-/y-axes of trust

Description: The user should be able to reorient the ISS model by switching the x-/y- axes of the trust.
Requirements:

1. User should be able to reorient the x-/y- axes of the ISS model.

Prerequisites: None.
Steps:

1. Load "EVA Navigator" web application by navigating browser to http://127.0.0.1:3000 or http://localhost:3000.
2. Left-click and drag the ISS model until the x-/y- axes are swapped.

Input: None.
Expected Output: ISS model is reoriented (x-/y- axes swapped) on trust.
Assumptions: None.

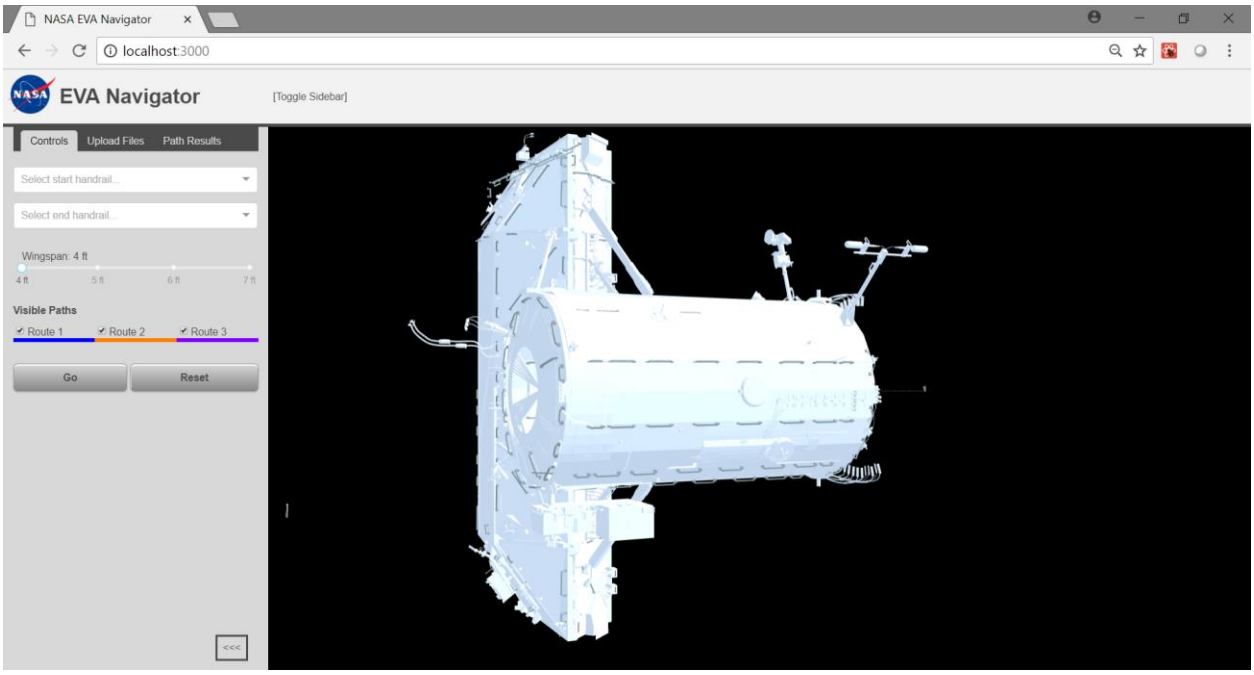

Figure 6. Test case 1 displaying initial orientation of ISS model.

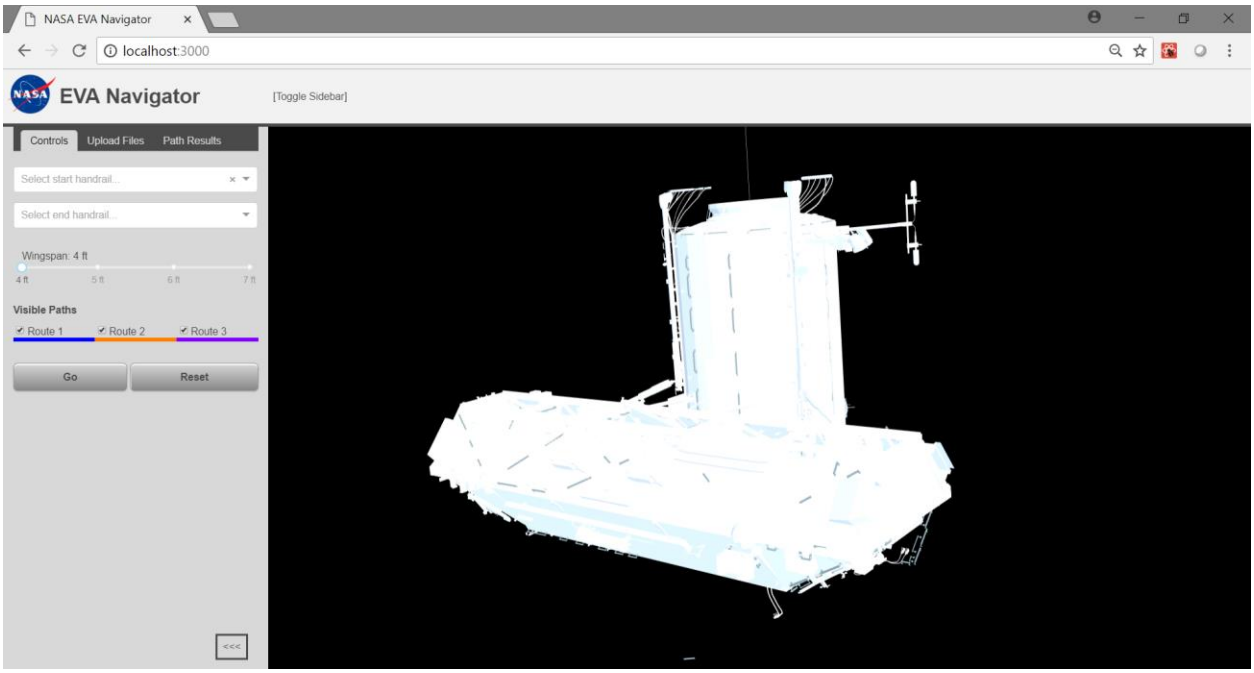

Figure 7. Test case 1 displaying reoriented ISS model.

3.1.2 Test Case 2: Complete full rotation of ISS model on x-axis

Description: The user should be able to fully rotate the ISS model on the x-axis.
Requirements:

1. User should be able to choose an ending handrail by clicking on a handrail on the ISS model.

Prerequisites: None.
Steps:

1. Load "EVA Navigator" web application by navigating browser to http://127.0.0.1:3000 or http://localhost:3000.
2. Left-click and drag the ISS model northbound, rotating on x-axis, until pole is eclipsed.
3. Refresh webpage to reload the ISS model.
4. Left-click and drag the ISS model southbound, rotating on x-axis, until pole is eclipsed.

Input: None.
Expected Output: ISS model is rotated.
Assumptions: None.

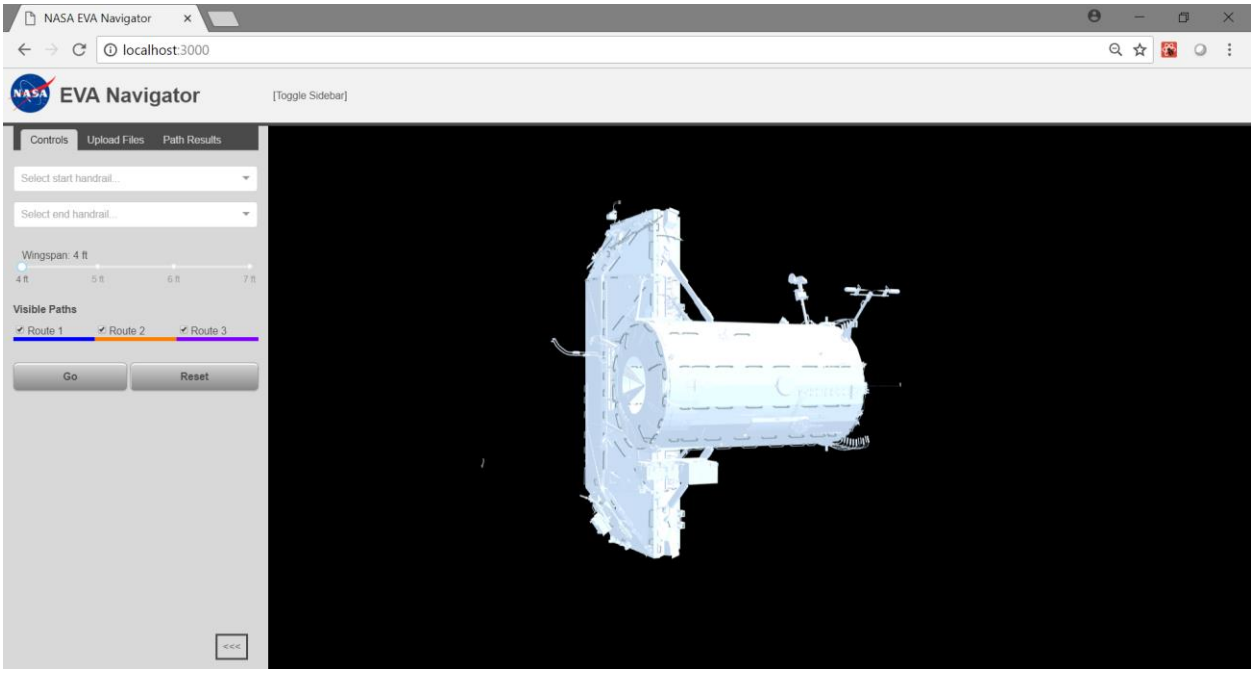

Figure 8. Test case 2 displaying initial orientation of ISS model.

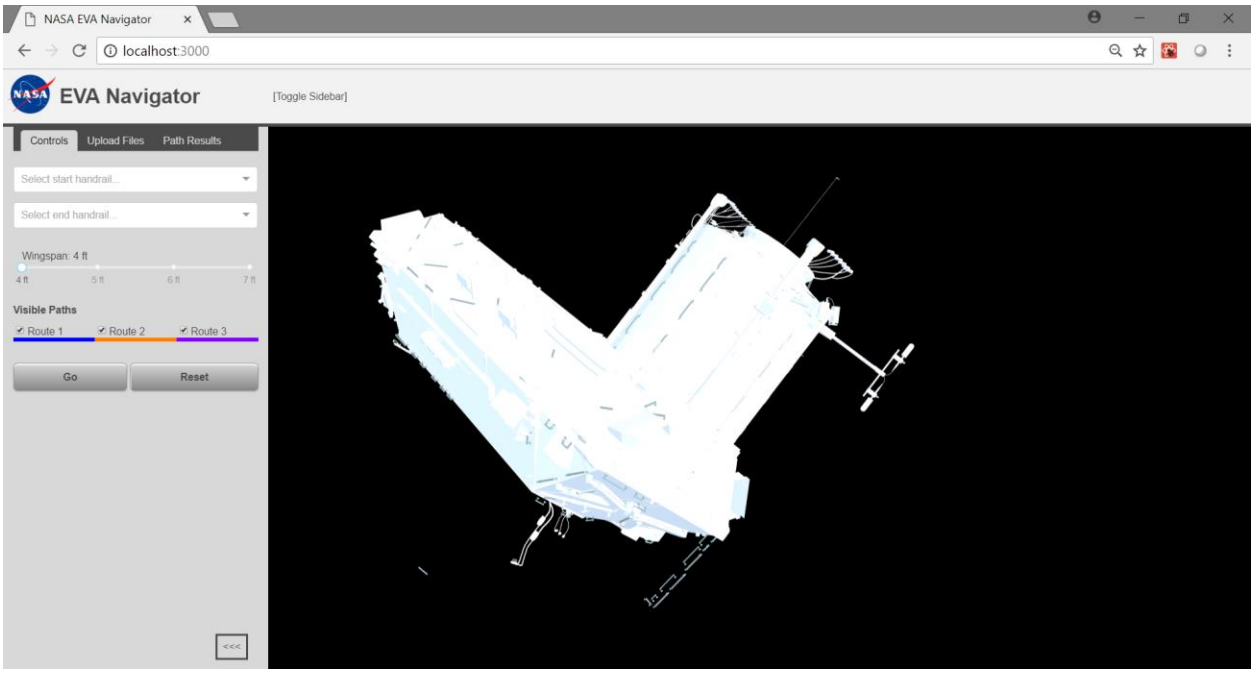

Figure 9. Test case 2 displaying ISS model rotated beyond north pole.

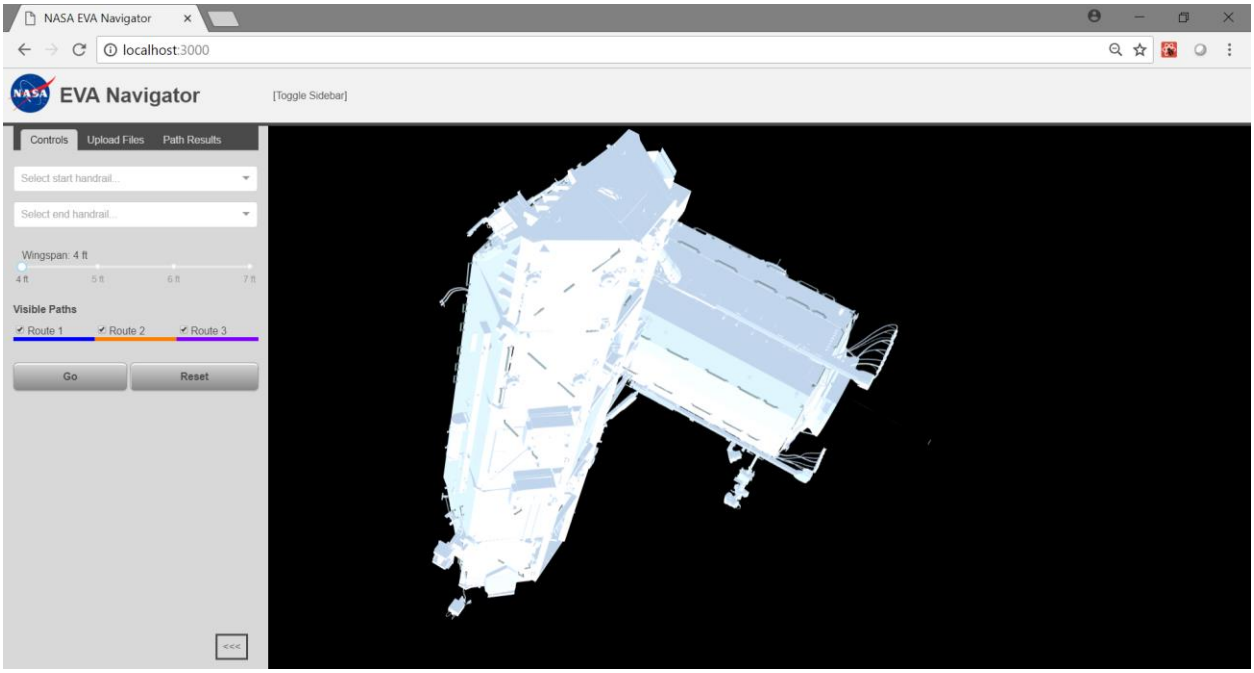

Figure 10. Test case 2 displaying ISS model rotated beyond south pole.

3.2 Assumptions and Constraints

3.2.1 Assumptions

It is assumed that the EVA Navigator web application has been setup and launched correctly, following the User_Manual.docx, section 3, Software Installation based on your operating system.

3.2.2 Constraints

None.

3.3 Findings

All functional tests worked as expected, resulting in the desired behavior.

References

None.