
Feature Documentation: UI and Route Calculation Update for Two Users

for

NASA EVA Path Phase 3

Version 1.1

Developer

Tenadam Weldesemayat

Product Owner

Daren Welsh

Test Team

Lincoln Powell

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

August 8, 2018

Table of Contents

i. Revision History	2
1. Introduction	3
1.1 Background	3
1.2 Intent	3
1.3 Agreed Change(s)	3
2. Development	4
2.1 Code Additions or Modifications	4
3. Functional Testing	7
3.1 Proposed Functional Test Cases	7
3.1.1 Test Case 1: Find possible routes for Crew Member 1	7
3.1.2 Test Case 2: Select Crew Member 2 tab from the left navigation under Controls.	8
3.1.3 Test Case 3: Find possible routes for Crew Member 2	8
3.1.4 Test Case 4: Toggle between the Crew Member 1 and 2 tabs to see route paths on ISS model.	9
3.1.5 Test Case 5: Check Path Results are shown for Crew Member 1 and 2.	10
3.2 Assumptions and Constraints	10
3.2.1 Assumptions	10
3.2.2 Constraints	11
3.3 Findings	11
References	11

i. Revision History

Revision	Author	Date	Description
1.0	Tenadam Weldesemayat	8/9/18	Initial document.
1.1	Lincoln Powell	8/12/18	Reviewed document.

1. Introduction

1.1 Background

The NASA EVA Navigator web application delivered from Phase 2 completed tailored backlog items designed by Daren to meet existing objectives for the product. One of the desired functionality remained in the backlog was the addition of user interface and route calculation accounting for two crew members. The application delivered from Phase 2 allowed only a single user to choose start and end handrails, wingspan and shows path results with three routes as shown below:

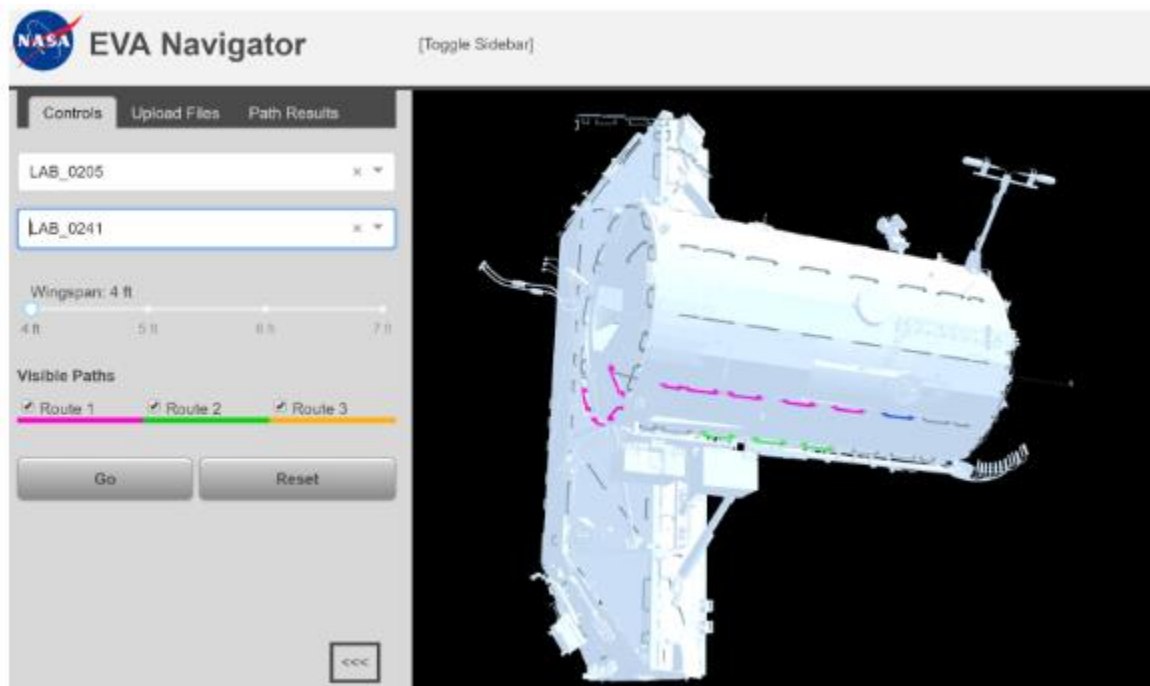


Figure 1. EVA Navigation User Interface for designed for a single crew from Phase 2.

1.2 Intent

Daren's intent for this change is to allow two crew members to be able to use the NASA EVA application to calculate shortest paths with de-conflicting routes. Toward this goal, he desired to extend the existing functionality accounting two users and adding logic so that the second crew will get route results that are not taken by the first crew member.

1.3 Agreed Change(s)

The addition of UI to the current interface to allow a second user to interact with the system as well as fins routes that do not conflict with the route results from the first user.

2. Development

2.1 Code Additions or Modifications

The changes made to EVA Navigator project to accomplish the UI and Route Calculation changes to accommodate for two users were mainly to the Container.js, Render.js, Controls.js and the layout.scss files. The screenshots below show the code changes made to the project. The pink highlights demonstrate the lines removed and the light green once show additions (see Figure 2 and 3 below). In the sections below a small portion of the code updated are shown as an example. The entire repository can be found [here](#).

In the Container.js file of the EVA Navigator project, states are defined for different controls under the second user UI. The below snippet shows an example of these changes.

```
206 -   const { routes } = this.state;
208 +   debugger;
209 +   const { routesSecond } = this.state;

207 210   fetch(window.location.protocol + '//' + window.location.hostname + ':8080', {
208 211     method: 'post',
209 212     body: JSON.stringify({
210 213       startHandrail: data.startHandrailSecond ? data.startHandrailSecond : "",
211 214       endHandrail: data.endHandrailSecond ? data.endHandrailSecond : "",
212 215       nodes: this.handrails,
213 216       wingspan: data.wingspanSecond.toString(),
214 -     previousRoutes: this.state.routes //passing previous routes
217 +     previousRoutes: this.state.routesSecond //passing previous routes
```

In the Container.js file of the EVA Navigator project, the path results section is updated to render path results for both crew members. The UI is also updated so that results are shown side by side. The changes are shown in the below snippet.

```

302 +         <table>
303 +             <tbody>
304 +                 <tr>
305 +                     <td className='td-crew1-results'>
306 +                         <div className='crew1-results'>
307 +                             <h3 className='crew1-results-header'>1. Crew 1 Route Results</h3>
308 +                             {routes.map((route, routeI) =>
309 +                                 <div key={routeI}>
310 +                                     <div>Route {routeI + 1}</div>
311 +                                     <div>Total distance: {route.distanceTotal} inches</div>
312 +                                     <ul>
313 +                                         {route.nodes.map((node, nodeI) =>
314 +                                             <li key={nodeI}>{node}</li>
315 +                                         )}
316 +                                     </ul>
317 +                                 </div>
318 +                             )}
319 +                         </div>
320 +                     </td>
321 +                     <td className='td-crew2-results'>
322 +                         <div className='crew2-results'>
323 +                             <h3 className='crew2-results-header'>2. Crew 2 Route Results</h3>
324 +                             {routesSecond.map((route, routeI) =>
325 +                                 <div key={routeI}>
326 +                                     <div>Route {routeI + 1}</div>
327 +                                     <div>Total distance: {route.distanceTotal} inches</div>
328 +                                     <ul>
329 +                                         {route.nodes.map((node, nodeI) =>
330 +                                             <li key={nodeI}>{node}</li>
331 +                                         )}
332 +                                     </ul>
333 +                                 </div>
334 +                             )}
335 +                         </div>
336 +                     </td>
337 +                 </tr>
338 +             </tbody>
339 +         </table>

```

In the Controls.js file of the EVA Navigator project, navigation UI and onClick updates are done as shown in the example below.

```

233 233         <Tab>Controls</Tab>
234 234         <Tab>Upload Files</Tab>
235 235         <Tab>Path Results</Tab>
236 236     </TabList>
237 237     <TabPanel>
238 238         <Tabs forceRenderTabPanel>
239 239             <TabList>
240 240                 <Tab onClick={this.changeSelectedTabIndex.bind(this, 'CrewOne')} >Crew Member 1</Tab>
241 241                 <Tab onClick={this.changeSelectedTabIndex.bind(this, 'CrewTwo')} >Crew Member 2</Tab>
242 242             </TabList>
243 243             <TabPanel>
244 244                 <div className='handrails-selector'>
245 245                     <Select
246 246                         name='startHandrail'
247 247                         placeholder='Select start handrail...'
248 248                         value={startHandrail}
249 249                         options={this.createHandrailOptions()}
250 250                         onChange={option => onStartEndHandrailsChange('start', option)}

```

In the Renderer.js file the logic for selecting a handrail from the ISS model is updated accounting two crew members.

```

    if (this.props.crewMemberSelected === 1) {
      if ((!this.props.startHandrail || !this.props.endHandrail)
        && this.props.startHandrail !== intersects[0].object.name.replace('.st1', ''))
        && this.props.endHandrail !== intersects[0].object.name.replace('.st1', '')) {
        this.props.handleStartEndHandrailsChanged(this.props.startHandrail ? 'end' : 'start', intersects[0].object, true);
      }

      // Else if the chosen handrail is the preselected start handrail, clear the start handrail.
      else if (this.props.startHandrail === intersects[0].object.name.replace('.st1', '')) {
        this.props.handleStartEndHandrailsChanged('start', null, false, this.props.crewMemberSelected);
      }

      // Else if the chosen handrail is the preselected end handrail, clear the end handrail.
      else if (this.props.endHandrail === intersects[0].object.name.replace('.st1', '')) {
        this.props.handleStartEndHandrailsChanged('end', null, false, this.props.crewMemberSelected);
      }
    }
    if (this.props.crewMemberSelected === 2) {
      // If the start handrailsecond or the end handrail equals null (meaning neither handrail has been chosen)
      // or if the chosen handrail is not the preselected start or end handrail, mark the handrail respectively.
      if ((!this.props.startHandrailSecond || !this.props.endHandrailSecond)
        && this.props.startHandrailSecond !== intersects[0].object.name.replace('.st1', ''))
        && this.props.endHandrailSecond !== intersects[0].object.name.replace('.st1', '')) {
        //debugger;
        this.props.handleStartEndHandrailsChanged(this.props.startHandrailSecond ? 'end' : 'start', intersects[0].object, true);
      }

      else if (this.props.startHandrailSecond === intersects[0].object.name.replace('.st1', '')) {
        this.props.handleStartEndHandrailsChanged('start', null, false, this.props.crewMemberSelected);
      }

      else if (this.props.endHandrailSecond === intersects[0].object.name.replace('.st1', '')) {
        this.props.handleStartEndHandrailsChanged('end', null, false, this.props.crewMemberSelected);
      }
    }
  }
}

```

In the layout.scss file of the EVA Navigator project, css attributes are added to align the path results for both crew members side by side.

6 ■■■■■ src/_layout.scss		View ▼	
	幸	@@ -118,4 +118,10 @@ main {	
118	118	padding-left: 1em;	
119	119	overflow: auto;	
120	120	height: 50%;	
121	+	width: 400px;	
121	122	}	
123	+		
124	+	.td-crew1-results, .td-crew2-results {	
125	+	vertical-align: top;	
126	+	padding-right: 10px;	
127	+	} @■	

3. Functional Testing

3.1 Proposed Functional Test Cases

3.1.1 Test Case 1: Find possible routes for Crew Member 1

Description: Crew Member 1 should be able to view possible routes after choosing start and end handrails from the dropdown.

Requirements:

1. Routes should be visible for Crew member 1 after choosing start and end handrails from the dropdown and clicking on the Go button.

Prerequisites:

1. Start handrail dropdown work as intended.
2. End handrail dropdown works as intended.
3. Wingspan changes work as intended
4. Route calculation works as intended.

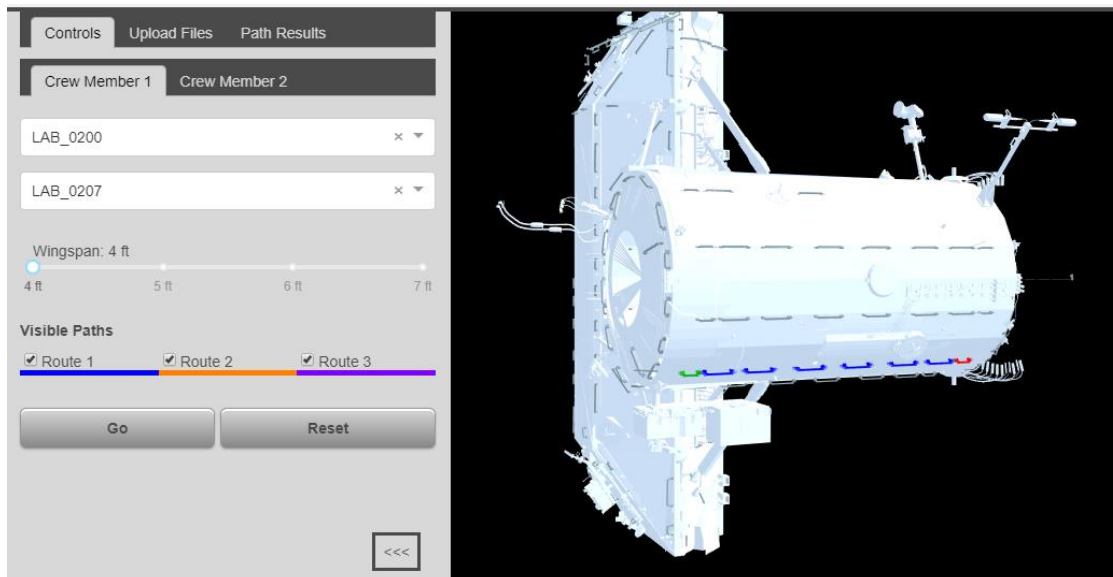
Steps:

1. Load “EVA Navigator” web application by navigating browser to <http://127.0.0.1:3000> or <http://localhost:3000>.
2. Select Crew Member 1 tab
3. Select the LAB_0200 handrail from the dropdown.
4. Select the LAB_0207 handrail from the dropdown.
5. Click the Go button.
6. Check that Routes are visible on the ISS model for Crew Member 1.

Input: Click on the LAB_0200 handrail, then click on LAB_0207 and then the Go button.

Expected Output: Routes are displayed on the ISS model for Crew Member 1.

Assumptions: None.



3.1.2 Test Case 2: Select Crew Member 2 tab from the left navigation under Controls.

Description: The user should see the Crew Member 2 tab when clicking on the left navigation.

Requirements:

1. User should be able to see the Crew Member 2 tab under Controls on the left navigation.

Prerequisites: None.

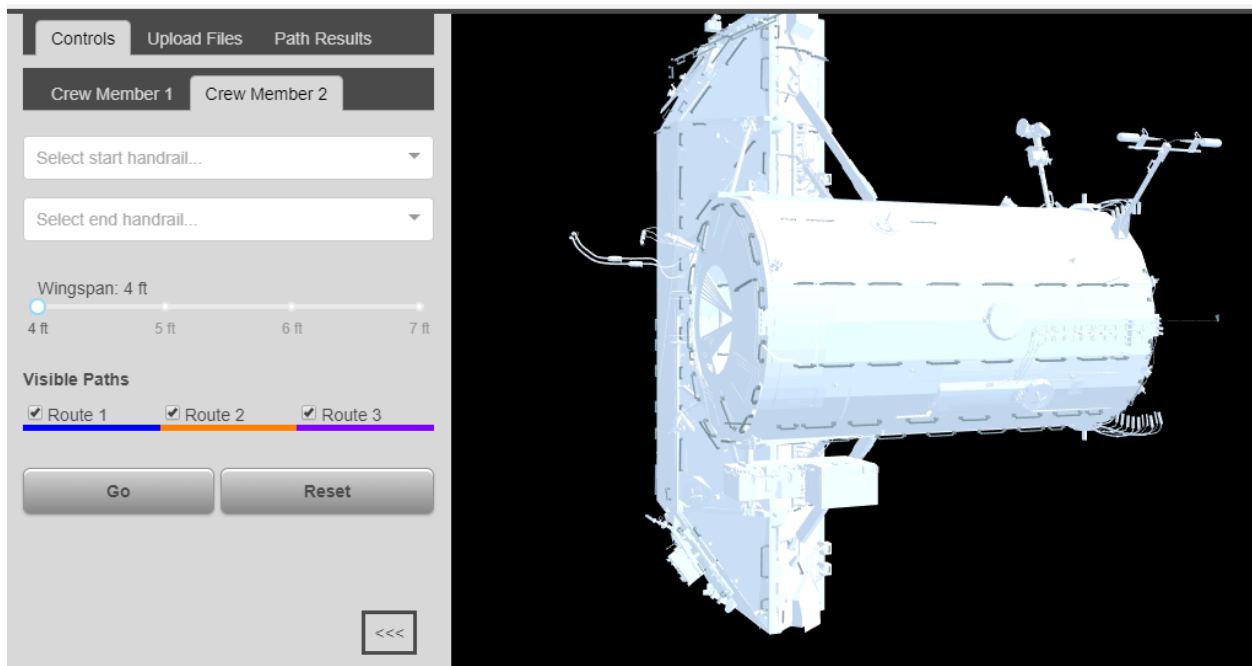
Steps:

1. Load “EVA Navigator” web application by navigating browser to <http://127.0.0.1:3000> or <http://localhost:3000>.
2. Select the Crew Member 2 tab under Controls from the navigation.
3. Check that the tab is selectable.

Input: Click on the Crew Member 2 tab.

Expected Output: Crew Member 2 tab is visible.

Assumptions: None.



3.1.3 Test Case 3: Find possible routes for Crew Member 2

Description: Crew Member 2 should be able to view possible routes after choosing start and end handrails from the dropdown.

Requirements:

1. Routes should be visible for Crew member 2 after choosing start and end handrails from the dropdown and clicking on the Go button.

Prerequisites:

1. Start handrail dropdown work as intended.
2. End handrail dropdown works as intended.
3. Wingspan changes work as intended
4. Route calculation works as intended.

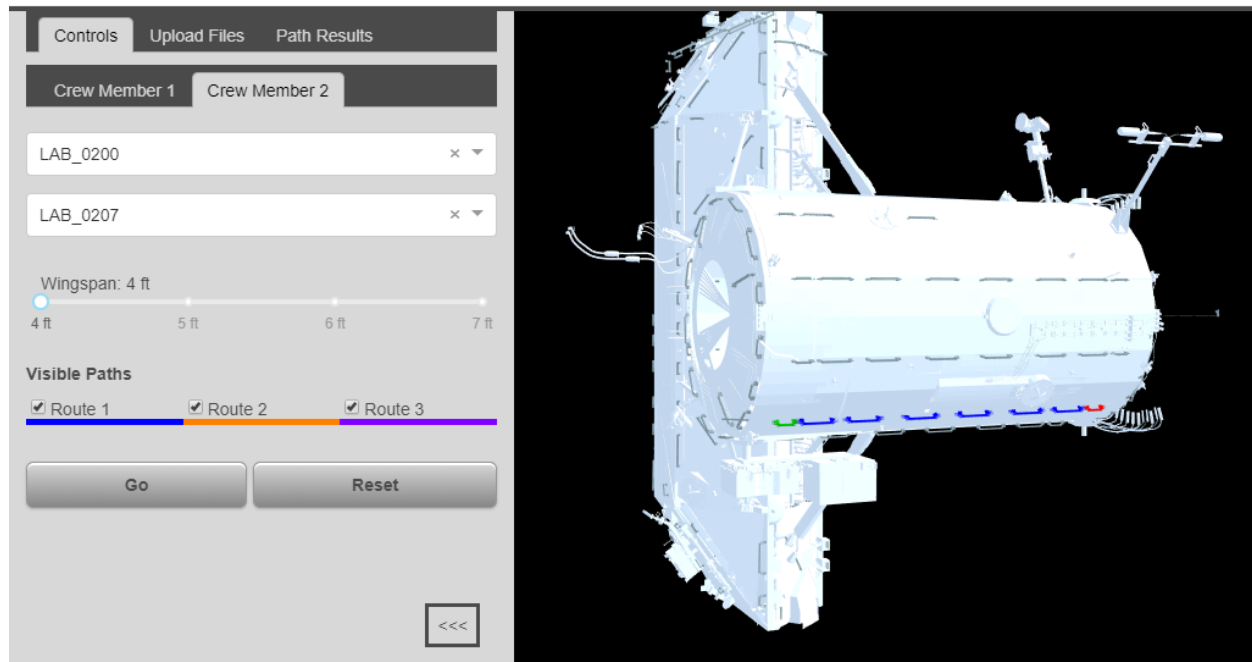
Steps:

1. Load “EVA Navigator” web application by navigating browser to <http://127.0.0.1:3000> or <http://localhost:3000>.
2. Select Crew Member 2 tab
3. Select the LAB_0200 handrail from the dropdown.
4. Select the LAB_0207 handrail from the dropdown.
5. Click the Go button.
6. Check that Routes are visible on the ISS model for Crew Member 2.

Input: Click on the LAB_0200 handrail, then click on LAB_0207 and then the Go button.

Expected Output: Routes are displayed on the ISS model for Crew Member 2.

Assumptions: None.



3.1.4 Test Case 4: Toggle between the Crew Member 1 and 2 tabs to see route paths on ISS model.

Description: The user should be able to toggle between the Crew Member 1 and Crew Member 2 tab and see route results for both users.

Requirements:

1. User should be able to toggle between the Crew Member 1 and Crew Member 2 tab and see route results for both users.

Prerequisites:

- 3.1.1 Test Case 1: Find possible routes for Crew Member 1
- 3.1.3 Test Case 1: Find possible routes for Crew Member 2

Steps:

1. Toggle between Crew Member 1 and Crew Member 2 tabs.
2. Check route results for both users are visible on the ISS model.

Input: Click on the LAB_0200 handrail, then click on LAB_0207, then the Go button on both tabs and toggle between the tabs.

Expected Output: Route results are visible for both Crew Members on the ISS model.

Assumptions: None.

3.1.5 Test Case 5: Check Path Results are shown for Crew Member 1 and 2.

Description: The user should be able to see the path results for both Crew Members.

Requirements:

1. The user should be able to see the path results for both Crew Members.

Prerequisites:

- 3.1.1 Test Case 1: Find possible routes for Crew Member 1
- 3.1.3 Test Case 1: Find possible routes for Crew Member 2

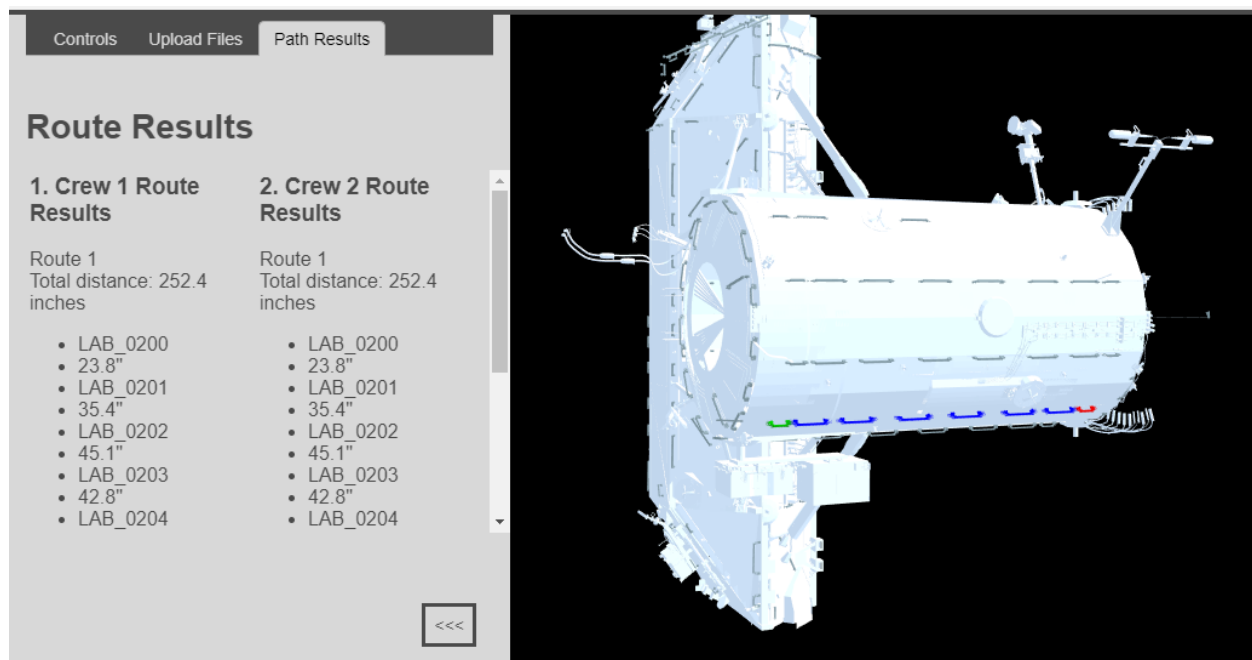
Steps:

1. Click on Path Results tab from the navigation.
2. Check path results are shown for the selected handrails for both crew members

Input: Click on the LAB_0200 handrail, then click on LAB_0207, then the Go button for both crew members and then click on Path Results tab.

Expected Output: Path Results for the selected routes are visible side by side for both Crew Members.

Assumptions: None.



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. It is also assumed that existing handrails in the ISS model properly correspond to the drop-down menus.

3.2.2 Constraints

Due to limited familiarity using ReactJS (along with React-Select) and lack of time to experiment, the route calculation to deconflict the paths, accounting for two crew members, remains in progress at this time. The work performed towards this feature request is aimed to give a future developer the ability to continue using the code modifications made.

3.3 Findings

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

References

None.