

Open e2e Testing Initiative with Open MCT

John Hill

About Me



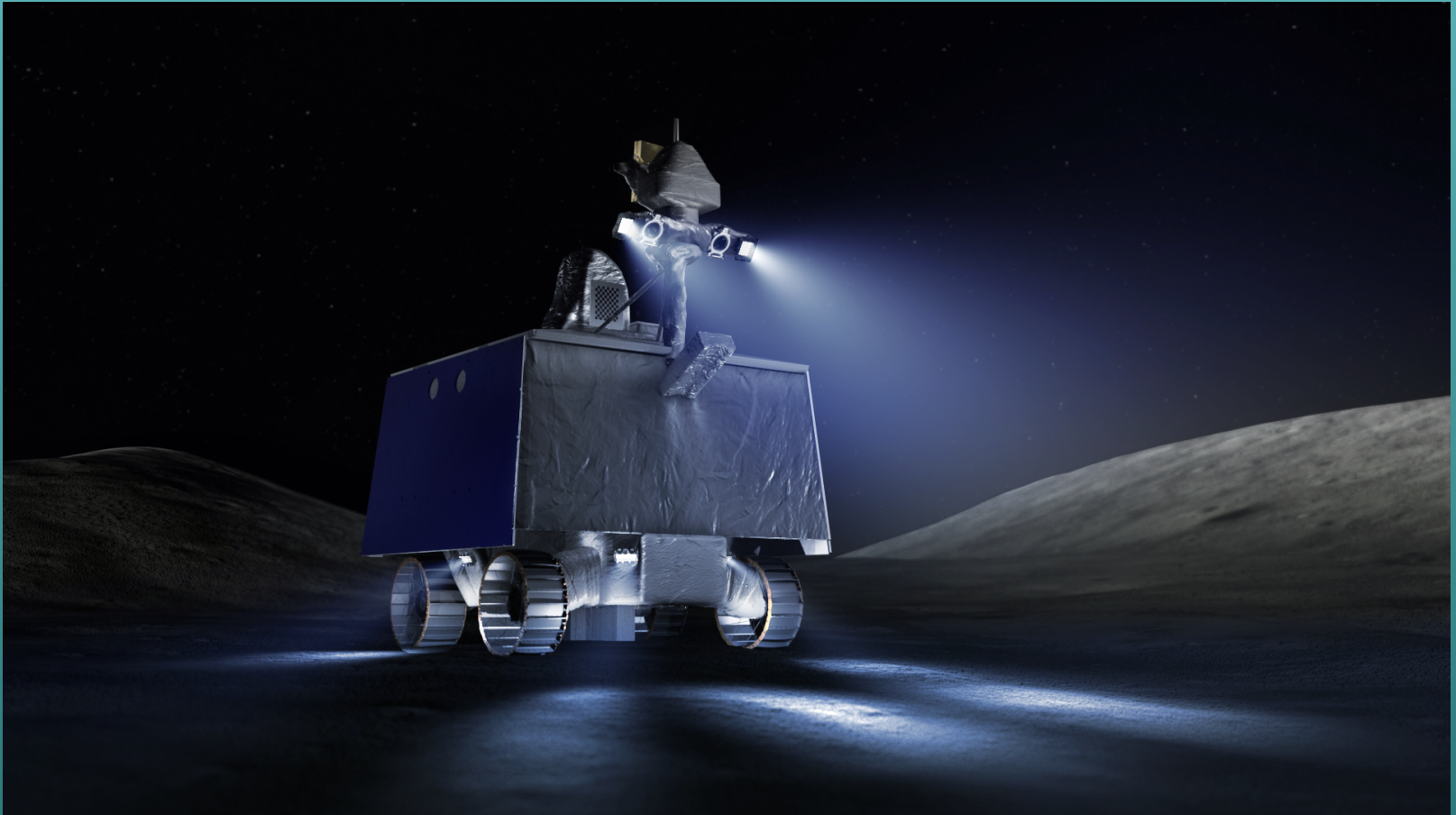
Web UI Test Engineer,
Space Mission Control Software, KBR, Inc
@ NASA Ames Research Center

Playwright Ambassador

<https://playwright.dev/community/ambassadors>

VIPER

Open MCT



In this talk

- Open MCT e2e Test Initiative
- Getting Started with Playwright and Open MCT
- Challenges with e2e Testing Projects
- Challenges unique to *this* e2e testing project
- Call to Action!



github.com/nasa/openmct

Open e2e Test Initiative Goals

1. Increase our test coverage
2. Provide an opportunity for the testing community to directly contribute to *this* NASA Open Source project
3. Learn from folks who demonstrate some new approaches to old testing challenges with Playwright.
4. Ensure that Open MCT testing is accessible to everyone

History

- The project already has a healthy open source developer community
- The project already had an existing Open Source contribution process and CLA
- e2e testing came online July 21
- Open e2e Testing Initiative came online October 21

Getting Started

- ~~We leverage official playwright.dev/docs/next/intro~~
- Getting started with Open MCT's implementation of Playwright
- Let's run through this together!



Getting Started with Open MCT e2e Documentation

What did we do?

- Utilized Playwright to mimic Mission Control Operator Behavior to interact with Open MCT
- We demo'd how easy it is to get going with the codegen and --debug.
- How Open MCT reenforces evolving best practices into the onboarding process.
- Test Stubs and test.fixme()
- Invoked failure and utilized the triage tooling

What to contribute?

- Test stubs -> Real Tests
- Originate from Open MCT dev team
- Bugs and enhancements

```
1 //If you run out of time to write new tests, please stub in the missing tests in place
2 //with a test.fixme and BDD-style test steps. Someone will carry the baton!
3 test.fixme('Can Rename Timer Object from Tree', async ({ page }) => {
4   //Create a new object
5   //Copy this object
6   //Delete first object
7   //Expect copied object to persist
8 });
```

Challenges with any e2e project

- Onboarding
 - AKA How to get from 0 to 1 test contributed.
 - Solution: Re-enforcing example template, lint, and JSDoc
 - ~~Solution 2: Test Promotion~~
- Web-dev-isms
 - Generally, JS web devs focus on clever, elegant, and concise implementations of test frameworks
 - QA is more interested in saving 3 hours of triage time than 3 lines of code
 - Solution: Failure must be obvious in tracefile and error logs

Unique Opportunities (Challenges)

- Open Source tests are re-used in downstream projects
 - ExampleTemplate suite needs to run against closed source projects.
 - More abstraction and fewer assumptions
 - No Page Objects?! (Solution: AppActions)
- "UI Contract Tests" and Performance
 - No assertions, just Contracts
 - Similar to Pact API Contracts
- Dev and Test (me) Time is shared
 - Easier to make application more testable

Please, no drive-bys 🙏

```
src/plugins/flexibleLayout/components/flexibleLayout.vue
```

↑	@@ -91,9 +91,9 @@
91	91 <script>
92	92 import ContainerComponent from './container.vue';
93	93 import Container from '../utils/container';
94	94 + import DropHint from './dropHint.vue';
94	95 import Frame from '../utils/frame';
95	96 import ResizeHandle from './resizeHandle.vue';
96	96 - import DropHint from './dropHint.vue';
97	97
98	98 const MIN_CONTAINER_SIZE = 5;
99	99
↓	

App Actions and a11y

- No Common DOM = No Common POM (Page Objects)
- App Action Pattern combines repeated user behavior into common functions (not fixtures)
- Instead, we leverage a11y locators
 - Aria and roles for screen readers
 - Now the preferred way of interacting with Chrome Dev Tools Recorder
- If dev and test time is shared, where would you spend that time to make the app more testable?

Call to Action!

*“ I am here [Playwright Slack] because of your Open MCT project and I liked what I was doing so much I brought it to my employer. Now I code Playwright all day!
~ Open MCT e2e contributor*



Summary

- Open MCT e2e Test Initiative
- Getting Started with Playwright and Open MCT
- Challenges with any e2e Project
- Challenges unique to this e2e Project
- Call to Action?!



github.com/nasa/openmct

Cutting Room Floor

Why Playwright?

- First-class support for Automated Performance Testing
- "Official" Chrome, Chrome Canary, and iPad Capabilities
- Support for Browserless.io to run tests in a "hermetically sealed" environment
- Ability to generate code coverage reports

