Test Plan

Versioning

node.js	v16.13.2
npm	v8.1.2

Ideal Test Plan

- Unit Testing
  - In each new algorithm we've integrated, we'd test basic and edge cases to thoroughly ensure that each one functions properly. Additionally, we should make sure it will fail when it's supposed to, and properly fail without breaking completely.
- Integration and System Testing (with UI)
  - Ensuring each algorithm visualization is correct
  - Ensuring new code doesn't break any of the existing algorithms
- Descriptions of Tools Used
  - Jest is a commonly used testing framework for JavaScript code, and would be helpful to use for unity and integration testing we will do with this application.
- Descriptions of Types of End Users
  - Primary Users: Robotics students at a college level, robotics instructors
- Performance, Reliability, etc. Testing
  - Stress testing by having many people use the tool at once
- Acceptance Testing
  - Making sure it is easy to use, with an intuitive UI and clear instructions

Realistic Test Plan

- Unit Testing
  - Testing basic and edge cases in each new algorithm we've integrated
  - Making sure it fails when it should
- Integration and System Testing (with UI)
  - Ensuring each algorithm visualization is correct
  - Ensuring new code doesn't break any of the existing algorithms
- Descriptions of Types of End Users
  - Primary Users: Robotics students at a college level, robotics instructors
- Acceptance Testing
  - Making sure it is easy to use, with an intuitive UI and clear instructions