**Table of Contents**

1. Document History
2. Stakeholders
3. Introduction
4. Objectives
5. Scope
6. Tactics
7. Test Strategy
8. Failure Details
9. Failure Mitigation
10. Repairs
11. Takeaway

**Document History**

|  |  |
| --- | --- |
| Authors | Ethan Roush, Alyssa Atkinson, Eddie Collins, Ren Bowers |
| Creation Date | October 8, 2018 |
| Last Change Date | N/A |
| Prior Changes | N/A |

**Stakeholders**

|  |  |
| --- | --- |
| **Name** | **Role** |
| Ethan Roush | Team member |
| Alyssa Atkinson | Team member |
| Eddie Collins | Team member |
| Ren Bowers | Team member |
| Apollo Team | Project Owner |

**Introduction**

The goal of this project is to study the Apollo project, create an extensive test plan, and carry out that test plan. While carrying out the tests, we will be noting which tests passed and which ones failed. For those that fail, a description of what went wrong will be provided. As tests are completed, attempts at fixing the code will be made.

**Objectives**

* Objectives
  + Determine the available features that can be tested
  + Create an extensive array of tests based on the available features
  + Carry out the testing and documenting the results
  + Determine the cause of any failed tests, attempt to fix them, & document those results

**Scope**

The scope of our testing will apply to all available features in the Apollo project. The following tools will be used

* Apollo online simulator – to visualize the tests
* Apollo online task manager – to manage tests
* Microsoft Word – for the test plan
* Doxygen – for code documentation
* Visual Studio – as a code editor, not the compiler
* Xcode – as a code editor, not the compiler
* Atom – as a code editor, not the compiler

**Tactics**

All sub-scenarios (indicated by an arrow) have sub-scenarios. For the sake of this chart, they have been omitted. However, if one or more of those numbered sub-scenarios fail, those numbers will be logged in this chart.

|  |  |
| --- | --- |
| **Name of Worldsim Scenario** | **Pass/Fail with Number(s)** |
| Lane-change in a through lane |  |
| * motor vehicle |  |
| * non-motor vehicle |  |
| * pedestrian |  |
| * unknown or no obstacle |  |
| Left-turn at an intersection |  |
| * motor vehicle |  |
| * pedestrian |  |
| * unknown or no obstacle |  |
| Right-turn at an intersection | Passed |
| * motor vehicle | Passed |
| * pedestrian | Passed |
| * unknown or no obstacle | Passed |
| Traveling straight in a through lane | Failed |
| * motor vehicle |  |
| * pedestrian |  |
| * unknown or no obstacle |  |
| Traveling straight through an intersection | Passed |
| * motor vehicle | Passed |
| * non-motor vehicle | Passed |
| * pedestrian | Passed |
| * unknown or no obstacle | Passed |
| U-turn at an intersection | Passed |
| * motor vehicle | Passed |
| * pedestrian | Passed |
| * unknown or no obstacle | Passed |

|  |  |
| --- | --- |
| **Name of Logsim Scenario** | **Pass/Fail with Number(s)** |
| Lane-change in a through lane |  |
| * motor vehicle |  |
| Left-turn at an intersection |  |
| * motor vehicle |  |
| Right-turn at an intersection |  |
| * motor vehicle |  |
| Traveling straight in a through lane |  |
| * motor vehicle |  |
| * non-motor vehicle |  |
| * unknown or no obstacle |  |
| Traveling straight through an intersection |  |
| * motor vehicle |  |

**Test Strategy**

Each scenario will be tested by at least one of the team members. After the tests are complete, the results will be logged in the charts under the “Tactics” category of this document. Once the tests are complete, the failing numbers (if any) will be reiterated under the “Failure Details” section. Included with the numbers will be what the actual sub-scenario is.

**Failure Details**

Below are the failed sub-scenarios with their respective numbers and names. The strategy regarding how they’ll be mitigated will be outlined in the “Failure Mitigation” section.

|  |  |
| --- | --- |
| **Number** | **Name of Failing Scenario** |
|  |  |

**Failure Mitigation**

Once all the failed scenarios have been logged, each team member will be assigned a test to attempt repairs. Repairs will be divided in the most efficient method possible. The dividing depends on what language the code is written in and how many tests failed. The details on each failure will be logged in the “Repairs” section of this document.

**Repairs**

The chart below states the information regarding each failed scenario. This information includes the scenario number, language the code is written in and what file(s) need repairs, and who will be attempting repairs. It’s important to note that even though a team member will be given one or more failures, multiple or all team members may work on that repair depending on how intensive that repair is. If that’s the case, multiple names will be given in that section of the chart.

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Language | File(s) | Team Member(s) |
|  |  |  |  |

**Takeaway**

While repairing the faulty code will be a rewarding victory, that is **NOT** the goal of this project. The goal of this project is to gain experience in doing such. For example, we’ll be using tools that are new to us. This increases our knowledge on those specific tools while simultaneously improving our ability to learn new tools. This is important for the workforce because we will be expected to learn new tools quickly. In addition, this will give us all experience on improving our coding skills and being able to dive into a project that we didn’t start. When we get into the workforce, whether the project will be code-based or not, we may have to learn a new project that already has significant progress made and be expected to understand and contribute to it. Lastly, this project will give us experience in teamwork. The workforce will expect us to be able to work with others.