Robot Exploration
Progress Evaluation
Milestone 1

Team Members:
Bryce Altomare (baltomare2017@my.fit.edu)
Austin Gaudio (agaudio2017@my.fit.edu)
Kyle Sergio (ksergio2017@my.fit.edu)

Faculty Advisor & Client

Dr. Marius Silaghi (msilaghi@fit.edu)

Advisor Meetings for Milestone 1:
Monday 9/14/2020
Friday 9/25/2020

Progress Matrix for Milestone 1:

Task	Completion %	Bryce Altomare	Austin Gaudio	Kyle Sergio	To Do
1. Compare and Select technical tools	100%	34%	33%	33%	nothing
2. Provide demos using tools to show	100%	0%	75%	25%	nothing

3. Resolve technical challenges	100%	34%	33%	33%	nothing
4. Compare and select collaboration tools for software development	100%	34%	33%	33%	nothing
5. Create Requirement Document	100%	70%	15%	15%	nothing
6. Create a Design Document	100%	33%	34%	33%	nothing
7. Create Test Plan	100%	34%	33%	33%	nothing

Discussion of each task for Milestone 1:

- **Task 1:Compare and Select Technical tools**

For text editors we decided on using both sublime and notepad++. For Compiling the code we will be using both cygwin using g++ and cmake. The simulator we will be using for Nao(Robot) is called webots. And for the c++ sdk the program is called qibuild.

- **Task 2:**

For demos, we have made simple hello world programs with c++, in cygwin, using Nao's C++ SDK using qibuild and Cmake, and also a demo with Webots, Nao's simulator.

- **Task 3:**

All technical challenges have been resolved. As path finding algorithms go we have four algorithms we plan to implement into Nao including: BFS,DFS,A* OR Dijkstra`s, and D*. Each algorithm has a significant amount of research done in them but more research was done for BFS and DFS because those two are the first we play on implementing. Software to Hardware integration was Resolved as we began working and reading the documentation on Naos drivers and API. Not as much is needed on our part because the Robot is fairly set up to where we don`t need to have an extensive knowledge on the subject. Naos API's and Drivers are accessible using the C++ SDK. The SDK uses Qibuild which utilizes toolchains to connect to Nao's API's from our worktree then into the SDK. After our application is complete it is then packaged and sent to either the physical robot or the simulator. Multithreading is a method of a central processing unit to have multiple threads of execution concurrently. Multithreading shares the resources amongst the multiple threads and attempts to act like a multi-core processor and boost performance and utilization of a single core with parallelism.

Multi-threading is available from the threads library in C and has built-in support in C++11.

- **Task 4:**

Collaboration tools were figured out immediately at the start of milestone one. For software development github will be used to have the code easily accessible for all teammates. For Documents and presentations we are using google drive which includes google docs and slides. Along with a cloud based file system to store any related documentation that is related to the project. For communication we are using the whatsapp to communicate with other teams working on Nao and to communicate with the faculty advisor/ client. For communication among the team we will be using Discord. Lastly, our task calendar is being done on google calendar.

- **Task 5:**

We layed out what will be required for this project for better communication and understanding of what needs to be done for the success of the project and satisfaction of our client.

- **Task 6:**

The team worked together to create the design document in order to understand how the project would be successful. Austin worked on the Modules and UML diagram sections, Kyle worked on the Approach using the path finding algorithms, and Bryce worked on the overall design and overview of the document.

- **Task 7:**

The test plan is to help us better understand how we will test our programs and possible solutions to the project. Such as using the simulator as well as the physical robot to test our path finding algorithms on multiple user defined mazes.

Discussion of each team member:

- **Bryce Altomare:**

Worked on Resolving both technical challenges of hardware to software integration and multi threading programming in c++/c giving the team a great understanding on what we need to do. Along with creating the outlines for each Document needed for milestone 1 to make it easier for the rest of the team to include their portions of work. Along with doing his own portion of the documents.

- **Austin Gaudio:**

Worked extensively on gaining a better understanding and building Naos API and simulator. Along with doing hello world examples on both the c++ sdk (qibuild) and Naos simulator (webots). Furthermore, he worked together with the rest of the team on the documents needed to submit for Milestone 1.

- **Kyle Sergio:**

Worked on Researching potential pathfinding algorithms for Nao as well

as implementing one of the algorithms into a smaller project to help the group acquire a better understanding of the functionality of the algorithm. Also, created the hello world example for the cygwin compiler. As well as worked together with the rest of the team on the documents needed to submit for MIlestone 1

Plan for Milestone 2:

Task	Bryce	Austin	Kyle
Research Path finding algorithms	34%	33%	33%
Implement a Path finding algorithm	33%	33%	34%
Test on Both Physical Robot and Simulation	33%	34%	33%

Discussion of each Task for Milestone 2:

- **Research Path finding algorithms:**

  Along with the research gathered in milestone further research will be needed to implement the various pathing finding algorithms to work alongside Naos API. Currently, we plan on using DFS as our first path finding algorithm to implement and extensive research has been done to know the time complexity, algorithm, and its background.However, we will need to implement it into smaller projects first. Also, to implement alongside the Hardware further research will be undergone.

- **Implement a Path finding algorithm:**

  As stated previously our first path finding algorithm we plan to implement is DFS. We believe this will provide a good foundation to work off of and implement other algorithms because we do have prior experience with DFS and since the graph we plan on using has unweighted edges it will fit perfectly. Later on we may find it easy to use weighted edges which will lead us into a better suited algorithm.

- **Test the path finding on Nao and on Simulation:**

  During the testing phase we plan on testing Nao using DFS on a variety of different mazes. Starting with simple easy cases and then building into unique cases that may occur. We believe Nao`s first case on any maze will be the slowest because it has to figure out the shortest path while it is traversing the maze since each maze will not be pre programmed into Nao with the correct amount of vertices and edges in the graph which Nao will construct. Furthermore, the simulation testing will occur first before any physical testing can begin.

Faculty Sponsor Evaluation

- Faculty Sponsor: detach and return this page to Dr. Chan (HC 322)
- Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Austin Gaudio	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Bryce Altomare	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Kyle Sergio	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

- Faculty Sponsor Signature: _____
  Date: _____