

**Autonomous Resource Detection and Retrieval**  
**Fisk University,**  
**Nashville Tennessee**

Student team members:

Kahlil Andrews

Novell Carter

Binula Illukpitiya

Andrew Maltz

***Abstract—*** For our virtual robotics competition we tried to look at ways to get the Swarmies to go to an area of the map and essentially claim it as their territory. No Swarmies would be in the same area and each Swarmie would traverse the area until all targets were collected. Through our time in the competition we ran into various problems with the software but despite this we became better code readers and were able to at least observe the normal actions of the Swarmies and develop some code that could possibly make them more efficient.

## ***I.*** Introduction

As a team of four people comprised of two local high school students and members with little to no coding experience, our goal was to gain valuable coding experience as well as experience working with a team to create an efficient way for the Swarmies to traverse the map and gather all targets as quick as possible. Starting out, the team was faced with a multitude of problems and given goals that were beyond our current scope of knowledge. As a starting point, we made an effort to discuss ideas during our first couple meetings and research code and general robotics. We also looked through the webinars and downloaded the software we would need for the competition. When we first started the competition the team was only comprised of two individuals so we focused on brainstorming and recruiting. The competition served as an excellent way for the neighboring high school students to gain coding experience and work with a college team. It also provided precious experience for a college student who had just become a computer science major to get

better at coding and learn to conquer other computer related problems.

## **II.** Related Work

Our team overall had little to no experience with coding and virtual robotics before Swarmathon resulting in the climb of a steep learning curve. As such we gained experience with programming, while working with the rocket team at Fisk University. The rocket team had been working on a rocket that had a camera on the payload capable of detecting various colors and identifying them. At the start of the year, we worked on this project in order to gain supervised experience and assistance from our colleagues. During our time installing the necessary software for Swarmathon, we looked at various online resources to help with the installation as well as the webinars and directions provided on the site. Once we had the simulator and Qt creator on some of our laptops, we took the time out to rewatch webinars and watch tutorials for Qt creator. Through these sources we gained a basic understanding of the programs so that we could alter and write our own code. One of our members was also an essential resource as he had years of experience in robotics competitions. His experience came in handy in understanding what the Swarmies were supposed to do and formulating a method to efficiently get the Swarmies to perform their task. He also had an understanding of machine error and imperfections in sensors that guided our sensor use.

### III. Methods

The team spent the first few days figuring out how to install the NASA Swarmathon program. The first couple of meetings were also spent simply watching the webinars and understanding what the goals and objectives of the competition were as well as how we could potentially go about completing them. As our understanding of the goal increased we began to brainstorm on the best way to accomplish this task. Our group eventually came up with a sort of territorial approach. Essentially, the Swarmies would go to various parts of the map and explore that part of the map. The sensor readings of the Swarmies would be interconnected so that an area being swept through by one Swarmie would not be swept over by another. This method would succeed in dividing the work needed to collect all the targets and do so in the most efficient manner. We also thought of using an ants approach which is to say that when targets were found the Swarmies would converge on that spot and clear all targets then go to other areas of the map. As such we knew that we would first need to observe the Swarmies without our code and then go through the code and read it piece by piece. Knowing that this would be part of solving the problem we decided as a group that first we would simply watch the Swarmies in action and understand the actual simulation, then we would dive into Qtcreator and explore the code. After we did these two steps, we would share the code as a group

and edit it through the week. Unfortunately this method was not achievable due to an array of obstacles that hindered our progress.

#### A. *Problems Faced*

Throughout our experience with Swarmathon, we faced an array of problems. The first problem we faced was a lack of members on the team. At the very beginning of the competition our team was composed of two people, one junior in college and a high school senior. Working with only two people meant that we would have a fairly limited flow of ideas which could be detrimental if we had failed to notice a flaw. The lack of members also meant that the two of us would be burdened with a large amount of work. The next problem we faced was our overall lack of experience. We had one computer science major and one member who regularly competed in robotics competitions but our other members did not have much experience with either of these. Not to mention none of us had dealt with Ubuntu, Qt Creator, or the other software that came with being in the Swarmathon competition. As such we had a fairly large learning curve to go through. Our greatest issue by far was overcoming the software problems we had.

For starters our whole group had problems simply installing Ubuntu on our laptops. The most notable problems faced were partitioning issues, improper drivers and hardware bottlenecks. Fortunately, we successfully installed Ubuntu on four laptops despite the setbacks. From there we

moved to install ROS Indigo and the software for the simulator. At this point we lost another laptop because the simulator just would not work on that laptop. As for the other three laptops, the installation process was very problematic. There were times when we would get to a step and it would give an error that we had to research and ultimately, through researching online forums and documentation, resolve in a nonconventional way. Eventually the simulator was installed with varying results on the output. One laptop managed to run the simulation effectively while the other two ran at a pace that was impractical. The installation of Qt Creator limited the population even more which resulting in only one computer being able to run both a simulator and Qt Creator. The simulation, however, quickly stopped working. The simulator would load the Swarmies, map, and target, but they could not be switched to automatic or manual as the user dashboard simply froze up. This problem left us with no way of coding and simulating on the same computer resulting in stagnant progress for a couple of weeks. However, the team was able to push forward and resolve the issue just as we had done when faced with past issues.

### ***B. Solutions to our problems***

Despite the problems we faced, we were able to progress and gain crucial experience. We worked past the lack of members by recruiting and getting help from the rocket team. We were able to get two more members on our team and by working with

the rocket team, we were also able to get many useful ideas and input to move past the problems we faced. The webinars and actively going online to research coding and robotics greatly helped us with inexperience and our learning curve. This was definitely the case with the actual code. Not all of our members were familiar with coding languages but this was definitely the case with C++. We all had essentially no experience with C++ so we had to study the language before we started dealing with the code. By studying the language, we were able to gain a better understanding of what the code was doing and where we would need to alter things. The installation process required a great deal of patience, perseverance, and preparation. From the beginning, we had decided to install everything on our laptops not just so we could work individually, but because we wanted back up laptops for emergency. This came in handy as we got a working laptop up till a certain point. Even when the simulator stopped working on that laptop we were able to watch the simulation on another laptop and code based on what we saw.

## **IV. Experiments**

At the beginning of the competition we thought that testing the code and our alterations would be an easy process but this was not the case due to our predicament. Due to us not having a computer with all the software working on it, we had to be

creative with the way we would test the code. At the start of this problem, we were a little hesitant to code because we did not have a way to test it. So if we changed something based solely on speculation and by some chance we found out that it lead to unexpected negative results, we would not know what to change it back to get the original results, as only one computer could code. As such, we ran the simulator on a separate computer and observed what the Swarmie would do under various circumstances. It does not seem like much but running the simulation while viewing the base code yielded more understanding than we imagined. During those simulations we saw that when multiple targets were in one space the Swarmie would just continuously charge into them until it either picked one up or gave up. By isolating a target, we found that the Swarmie would pick it up in most cases. When it did not pick it up it was because it would either not open the claw or it would not lower the wrist to properly pick up the target. Once the target was picked up, the Swarmie was coded to head towards the center but one Swarmie actually held the target for up to five minutes whilst driving around in search for the center. Eventually, we dragged the Swarmie to the center to see if it would actually drop off the target which it did. It should have gone back to the cluster of targets but it simply wandered around aimlessly. Based on this we went back to the code and altered the parts that we knew were not working such as telling the Swarmie to lower its wrist before trying to grab the target and actually opening its claw when it encounters a target. Unfortunately, we

haven't quite figured out how to transfer the code to the simulation computer for testing.

## *V. Results*

The results of our work as it pertains to coding the Swarmies in an efficient manner to gather the targets around the map was inconclusive. However, our results as it pertains to using the Swarmathon software shows that the software can be difficult to use and sometimes does not work at all. At the start of this competition, we all had different laptops from different companies that all had varying amounts of memory and processing power yet none of them were able to use Qt creator and the simulator at the same time. Qt creator seemed to be a huge problem as only one laptop was even able to install qt creator successfully. Then after Qt creator was installed the simulator completely shut down on that laptop. Even downloading and installing these applications proved difficult. In some instances a step would not work at all then a minute later it would unexplainably work with no actual changes to how we did it. Everything seemed really finicky especially the simulator. At times the connection to the Swarmies was absolutely perfect and it ran smoothly but then a couple seconds later it completely disconnected. This was a rather weird glitch that seemed to occur with or without internet connection. On the bright side each of us got a lot better at problem solving and code reading. We faced many problems throughout and the code was comprehensive and long yet we were all still

able to solve some of the issues together and accurately predict what the Swarmies would do at times solely based on reading the code.

## **VI. Conclusion**

In conclusion, Swarmathon was altogether a great learning experience that helped each of us not just with coding and robotics but also problem solving and working as a group. We did not necessarily concretely discover much about the Swarmies and how to make them better but we do have ideas based on what we observed and read. For starters, having an algorithm in place for the Swarmie to remember the location of targets and come back would be a good idea. There could also be added code so that the Swarmies could notify one another of their findings. Another idea would be to have this algorithm in place but instead of calling other Swarmies over the Swarmie that discovered the target would be in charge of that area while other Swarmies would have other sections of the map. This method would also avoid confusion should two Swarmies discovering targets at the same time. Also making the simulation smoother and faster would greatly help future competitors. This change will allow them to run more simulations and test their code without having to worry about the simulation crashing.