Multi-agent USV Integration and Testing

By: Amanda Moffitt
Mt San Jacinto College
2016 CURE Summer Internship
Jet Propulsion Laboratory,
California Institute of Technology

Mentor: Gail Woodward, Jet Propulsion Laboratory,
California Institute of Technology

Project Title: 6282 Multi-mission, multi-agent robotics and computer vision
SWARM II Overview

• The aim of the project is to further build up the system originally developed in Swarm I that allowed a swarm of autonomous boats to work together to complete specific missions while following certain guidelines in respect to the way they behave (COLREGS).

• The Swarm II project is specifically focused on developing a system that allows the swarm to dynamically identify and respond to unknown vessels that enter a patrol zone while still continuing towards the mission goal which is usually to patrol a given area.

• The Project is nearing it’s end with a demo scheduled for the end of September and the team has successfully integrated many new developments that are going to WOW (cooperative behaviors, velocity-control, etc.).
Overview:

• Test Scenarios
  - Creating a scenario
  - Developing a tutorial on the process

• GERBIL Testing Environment

• GIST – Gerbil Instance Search Test
Building Test Scenarios

• A Scenario is basically a blueprint of a future on-water test that has been translated into code so that it can be used in simulation testing beforehand. A scenario is required to run a test.

• We use the WAVES program to perform this translation. (Web-based Adaptable Visualization for Extendable Simulations)

• Scenarios describe start positions for all the vessels as well as paths that the unknown vessels will follow during the test. They also specify other vessel parameters like speed, heading, etc.

• To the right you can see what a scenario outline looks like.
A Scenario created in WAVES
(Once saved, Waves produces a ready to use .json file)
WAVES Tutorial

- Creating scenarios was one of the first things I learned this summer. At that point there were no tutorials in existence explaining how to do so. Therefore, I was tasked with creating one since I was having to learn it with a fresh set of eyes anyways.

- The first few versions were too vague. I needed to consider all use cases and explain all the options.

- Presented and got great feedback which helped to improve the tutorial once more.

- I was able to suggest improvements to the WAVES program throughout this process. Some have been implemented already which is really neat!!

- Creating this tutorial not only allowed me the chance to leave a helping hand behind for someone in the future, but also gave me a thorough understanding of one of the basic components dealt with in the testing process.
Gerbil Simulation Testing

• We start a test using postman:
When a test is started, an instance is launched for each autonomous boat in the scenario as well as a swarmsim instance.

Each boat instance is representative of a simulated boat running its own software.

We monitor all instances on the web via WAVES until all of the unknown vessels or Contacts of Interest have completed their paths.

If an issue is noticed during a test we report the issue via github and tag the appropriate developers.

Anyone can access a running instance if they have the active link.
Gerbil Simulation Testing cont..

• Typical expected Behaviors:
  - Swarm should continuously patrol the specified area in an efficient manner.
  - When an unknown vessel enters the scenario, it should be quickly recognized and tracked until it has been classified.
  - Each unknown vessel needs to be classified as being either hostile or friendly.
  - The swarm needs to respond appropriately to a vessel once it has been classified.

• The performance of how the software tells the boats to act out these behaviors is what we run tests to observe.
Visualization development

The stone ages
Visualization development

Much easier to see behaviors
Visualization development

Current Version
Gerbil Simulation Testing cont..

• When the scenario has run its course, the test is over and each instance can be stopped.
• Stopping a test results in a flood of e-mails containing all of the test data for each instance.
• Thus begins the documentation process.
• Log files are very important in determining whether or not code ran successfully on all fronts and not just in the simulation viewer.
• Developer access to these should be quick and easy, but this is not always the case.
Development of GIST

Aim: Provide an easy way for the user to view the records/logs of a specific instance that was run on either the Kraken or Blackpearl server.

Plan of Attack:

– Implement a filtering method that allows the user to arrive at the instance(s) matching their desired search criteria in a quick and simple way.

– Once the user finds the desired instance(s), clicking on the instance should open either an active http link if it’s running, or the logs link if it’s dead. Either link should be opened in a new window.

– Provide the following Filters:
  • By User
  • By Test ID
  • By a ‘Running’ status
  • By Date
  • By Instance Number
Development of GIST cont..

• I chose to build a webpage from scratch and then add the desired functionality to it.

• I used HTML, CSS and JavaScript to develop my original prototype. I had no previous experience with any of these languages prior to this project, so this task has been an awesome learning experience for me.

• The Original prototype was rough in the looks department and lacked a practical design, but working on it helped me learn what kind of functionality I could work with as I continued to develop.

• Had a homepage that linked to a different page depending on which filter you wanted to run.

• Site was functional 😊
Gerbil Instance Search

How would you like to view the instances?

- **By User**
  (Click above to sort by user)

- **By Test ID**
  (Click above to sort by Test ID)

- **By "Running" Status**
  (Click above to see all of the currently running instances)

- **By Date**
  (Click above to sort by date)

- **By Instance #**
  Enter an instance Number: [Enter]
Whose history would you like to look at?

(Choose a user to continue)

- AMANDA.MOFFITT@JPL.NASA.GOV (11 tests run, 0 currently running)
- CSORICE@JPL.NASA.GOV (10 tests run, 1 currently running)
- DAVID.BROWN@JPL.NASA.GOV (13 tests run, 0 currently running)
- EPPERSON@JPL.NASA.GOV (2 tests run, 0 currently running)
- SCOTTPT@JPL.NASA.GOV (22 tests run, 4 currently running)
  - Instance #43, Tested on KRAKEN August 17, 2016 --- swarmsim
  - Instance #47, Tested on KRAKEN August 17, 2016 --- swarmsim
  - Instance #64, Tested on KRAKEN August 17, 2016 --- swarmsim
  - Instance #65, Tested on KRAKEN August 17, 2016
  - Instance #66, Tested on KRAKEN August 17, 2016
  - Instance #78, Tested on KRAKEN August 17, 2016 --- swarmsim
  - Instance #79, currently RUNNING on KRAKEN
  - Instance #80, Tested on KRAKEN August 17, 2016

Example of the page the User Filter would open
Development of GIST cont..

• After sharing my first version, I got some major feedback:
  - Site should have only one page where everything takes place. (Huge design change!!)
  - User should be able to choose multiple filters to run at once. (Huge code change!!)
  - Functionality of the site will be very useful 😊

• So, I developed a new design and worked on just that for as little while.
New Design
Development of GIST cont..

- New design was approved as being a nice step up from its predecessor, so I began developing the functionality of the page.
- A couple new suggestions were made to add some other related functionalities on the site such as stopping instances and displaying current statistics while I was adding functionality.
GIST
Gerbil Instance Search Tool

Server 1 instances (Ex: Kraken)
Server 2 instances (Ex: Blackpearl)
Server N instances (Ex: …)

USER Input

Filters Run
List is created that will be displayed on the home page
Final Product
Summary of what I learned:

- Basic project handling skills (task allocation)
- Basic software development and the testing process involved with it
- Teamwork (The Swarm II team is awesome!!)
- GitHub and version control
- HTML, CSS, JavaScript, Python
- Basic web development
- Scripting
- Linux
- Virtual Machines and Virtual Box
- The importance of Documenting everything
Acknowledgements

JPL mentors and the entire Swarm II team:
• Michael Wolf
• Gail Woodward****
• Cristina and Gigi
• The I&T team (Viet, Josh, Phil, David, Chris, Scott, Michael)

Paul McCudden and Heath Rhoades

And Faculty Sponsors from:
• Mt San Jacinto College
• LACC
• LAVC
• ELAC
• PCC
• Victor Valley College
• Moorpark College

Include the phrase “This work supported by NSF grant #1460538 to Los Angeles City College”
Thank you all for your time!!