

Project Status Report

Project Name: Team Cerebral

Team Members: Nick Spina, Matthew Frosini, Conor Ahern, Connor Schultz

Date: 3/18/18

Cycle Number: 2

System Intent: “Port the classic competitive robot game ATRobots to modern operating systems with a more advanced and evolved interface.”

Cycle Intent: Get ATRobots games to run accurately compared to original ATRobots, with graphics

Accomplishments since the last status report:

- Ported last remaining non-graphical functions from ATRobots
 - init, get_from_ram, get_val, put_val, push, pop, find_label, init_mine, count_missiles, init_missile, damage, scan, com_transmit, com_receive, in_port, out_port, call_int, jump, execute_instruction, do_robot, do_mine, do_missile, show_statistics, init_bout, bout
 - Compiles without error, **matches run successfully**, but match results are not consistent with original ATRobots match results
- Began work on porting graphical functions
 - Started with implementing “graphix” portions of code from do_robot
 - Started researching Qt graphical functions and classes

Obstacles encountered since the last status report:

- Qt graphics functions and classes provide an easy way to “paint” objects such as shapes and lines on different types of canvases, however there doesn’t seem to be a way to have lines drawn continuously while the program runs as the original ATRobots. So far, all that has worked is drawing, and then the drawn objects are printed when the program ends, which is not useful for our project.

Risks facing the project:

- By implementing Qt libraries, we are forcing users to install the massive (~2GB) Qt sources if they wish to compile the program themselves. It’s also very confusing to figure out how to compile with Qt sources, especially on Windows systems.

Objectives for the next week:

- Get live updating/rendering graphics to work
 - Decipher differences between Qt classes like QPixmap, QPainter, QImage, decide which would be best for porting code as close as possible to original code
- Test non-graphical matches and figure out why the results are not similar to original ATRobots
 - This may need to wait until graphical functions begin working, as it would be easier to see what's going wrong with robots with a visual aid

User Features:

#	User Feature < Short Name: <i>Short Description</i> >	Planned			Actual		
		Cycle planned for completion	Total planned hours	Planned hours this cycle	Status (completed, discarded, in progress, unstated, etc.)	Actual hours this cycle	Total actual hours this project
1	Working non-graphical matches between robots	2	110	40	Testing and identifying errors	6	53
1a	Decoding locked robots	2	10	5	Testing	0	2
2	ATRLock with GUI interface	2	45	30	Restarting	0	6
3	Graphics for robot matches	2	70	70	Researching, initial coding	6	0

Team Actions:

Name	User Feature < # only >			Planned	Actual							
	Coder(s)	Tester(s)	Reviewer(s)	Planned hours this cycle	Process hours		Product hours		Customer hours		Total hours	
					Week	Cycle	Week	Cycle	Week	Cycle	Week	Cycle
Conor Ahern	1, 2, 3	1a	1, 1a, 2, 3	48	0	0	12	12	0	0	12	12
Nick Spina	1, 2	1a, 3	1, 1a, 2, 3	48	0	0	4	4	0	0	4	4
Matt Frosini	1a	1, 2	1, 1a, 2	48	2	2	0	0	0	0	2	2
Connor Schultz	1, 1a, 3	2	1, 1a, 2, 3	48	3	3	0	0	0	0	3	3