Project Status Report

Project Name: Team Cerebral
Team Members: Nick Spina, Matthew Frosini, Conor Ahern, Connor Schultz
Date: 3/25/18 **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:

- Initial functioning graphics system is working
  - Restructured entire program, moving all original ATR2 functions into a new "atr2" class, and variables/robot structures into "atr2var" class
  - Main loop now just spawns a thread for ATR2 to run in so graphics can keep running live, without being blocked by ATR2 functions
  - Only arena is drawn for now
  - drawing functions from draw_robot (which includes robot triangles and scan lines/arcs) and do_missile (which includes both "bullets" and "hit made" circles) implemented
    - robot drawing utilizes Qt's Pixmap feature: each robot is being drawn off-screen to a Pixmap in a Pixmap array, and then each individual robot Pixmap is drawn all into one Pixmap, which is then drawn onto the on-screen arena.
    - missile drawing is still only drawing directly to the arena window in the Qt paintEvent(), and some flicker is present. missile shooting also sometimes causes robots to flicker as it takes too long to draw them live.

Obstacles encountered since the last status report:

- Our team has successfully gotten "live" graphics by spawning a new thread for the ATR2 main program, and reduced flicker and/or artifacts in the robots by utilizing a Pixmap array. However, to create a Pixmap array for missiles using the same system/method, a Pixmap array of size 1023 would need to be made (the number of max_missiles allowed in ATR2), which would use an unacceptable amount of system resources, and cause frequent blocks in the painting

thread as it tries to paint all Pixmaps in the array to one single Pixmap for on-screen drawing. We need to create a new, more effective system for missiles instead.

Risks facing the project:

- From last week: By implementing Qt libraries, we are forcing users to install the massive (~2-4GB) 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.
- This week: Our team tried compiling Qt on Windows in order to statically link the libraries needed for the program to run, so the program could be shared and run on other computers that do not have Qt. However, Qt compilation fails on Windows systems.

Objectives for the next week:

- Re-do missile's graphics implementation to be more efficient and reduce or eliminate flicker
- Plan/create a feature to include other graphics items found in original ATR2
    - Right column of robot info boxes including Armor, Heat, Robot names
        - End-of-match stats popup with "press any key to continue" to go to next match
        - End-of-game stats popup that gives summary of all matches in the game
        - Create design artifact/tests
- From last week: Test non-graphical matches and figure out why the results are not similar to original ATRobots
    - Graphics are now in a usable-enough state to begin tests on robots
    - Create test constraints, prepare to test every robot bundled with the original ATR2 and ensure all robots behave identically as they did in the original program
        - Debug and fix ATR2 functions/instructions that aren't functioning properly
        - For next week, only focus on a few select robots at first to test and closely inspect differences between
            - CIRCLES.AT2
            - RANDMAN3.AT2
            - SNIPER2.AT2

User Features:

		Planned			Actual		
#	User Feature <***Short Name: Short Description***>	Cycle planned for completion	Total planned hours	Planned hours this cycle	Status (completed, discarded, in progress, unstarted, 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 (did not restart yet)	0	6
3	Graphics for robot matches (robot arena only)	2	70	70	Initial graphics system implemented, need to test and optimize	20	20

Team Actions:

	User Feature <***# only***>			Planned	Actual							
Name	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	14	26	0	0	14	26
Nick Spina	1, 2	1a, 3	1, 1a, 2, 3	48	0	0	8	12	0	0	8	12
Matt Frosini	1a	1, 2	1, 1a, 2	48	0	2	4	4	0	0	4	6
Connor Schultz	1, 1a, 3	2	1, 1a, 2, 3	48	0	3	10	10	0	0	10	13