Project Status Report

Project Name: Team Cerebral
Team Members: Nick Spina, Matthew Frosini, Conor Ahern, Connor Schultz
Date: 4/8/18 **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:

- Missile drawing (graphics)
  - Missiles now use Qt's Pixmap function like the robot drawing does, allowing us to implement missile drawing into our graphics system without flicker.
- Tested ATR2Func functions
  - ucase, lcase, ltrim, rtrim, btrim, lstr, rstr, find_angle, find_anglei
  - All tests completed successfully
- By testing our ATR2Func functions, we found find_angle was the source of our malfunctioning scans.
  - Find_angle had integer division where the results were being truncated, even when a double was expected. In Pascal, integer / integer always resulted in a double, however in C++, this results in a truncated integer.
- We tested the following robots:
  - CIRCLES.AT2 - works like original
  - RANDMAN3.AT2 - works like original
  - SNIPER2.AT2 - does not function properly
  - INDIRECT.AT2 - "works" but we have no robot info printing yet
  - OVERHEAT.AT2 - does not function properly
  - PEASHOOT.AT2 - does not function properly
  - RAMMER.AT2 - does not function properly
  - SDUCK.AT2 - works like original

    - SNIPER.AT2 - does not function properly
    - STRAIGHT.AT2 - does not function properly
- Tested robots
    - We developed a testing process for robots where we construct a “checklist” of how the robot worked in the original ATR2, and then compare to how the robot runs in our ATR2 port.
    - For example, for the CIRCLES robot, we identified these functions to look for:
        - Starts in a random position
        - Hits boundary without going off the screen
        - Moves in a circular pattern
        - Scan lines being drawn
        - Shoots when a robot is within scan lines

Obstacles encountered since the last status report:

- Erasing scan lines
    - Scan lines stay on screen if robots die while scanning
    - Our graphics implementation with Qt is slightly different than ATR2 (for example, we clear the entire screen instead of drawing black lines over the robots’ previous location like in the original ATR2 between cycles)
    - Once we get robot explosions working, this problem will also be solved, as we will modify our system to more closely match the original (when robot is dead, erase scan lines)
- Exploding robots
    - Need to draw explosion when robots die
    - Right now, robots simply stop drawing themselves, but their “dead body” stays where it was for the remainder of the match.
- SNIPER2.AT2, one of the robots we decided to test this week, still does not function.
    - We added a backend debug option to print every instruction as the robot tries to execute them, allowing us to see where the robot is getting “stuck” in a loop, or which instructions it’s executing incorrectly, as the program runs in real time.

Risks facing the project:

- From previous weeks: 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.
- From previous weeks: 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:

- Plan/create a feature to include other graphics items found in original ATR2 (Feature #4)
    - 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
- Continue testing robots and determine which ATR2 instructions are not implemented correctly
    - 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
    - Robots left to test/fix:
        - SNIPER2.AT2 - tested, need to fix
        - SNIPER.AT2 - tested, need to fix
        - STACK.AT2 - need to test
        - STRAIGHT.AT2 - need to test
        - RECEIVE.AT2 - need to test
        - SCANTEST.AT2 - need to test
        - SUICIDE.AT2 - need to test
        - SWEEPER.AT2 - need to test
        - TRACKER.AT2 - need to test
        - TRACON.AT2 - need to test
        - WALLBOMB.AT2 - need to test
        - WEAVE.AT2 - need to test
        - WEAVER.AT2 - need to test
        - ZITGUN.AT2 - need to test

- OVERHEAT.AT2 - tested, need to fix
- PEASHOOT.AT2 - tested, need to fix
- RAMMER.AT2 - tested, need to fix

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	8	55
1a	Decoding locked robots	2	10	5	Testing	0	2
2	ATRLock with GUI interface	3	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	42	42
4	Side info bar for graphical matches	3	15	0	Unstarted	0	0

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	4	4	12	50	0	0	16	54
Nick Spina	1, 2	1a, 3	1, 1a, 2, 3	48	4	4	8	32	0	0	12	36
Matt Frosini	1a	1, 2	1, 1a, 2	48	2	4	0	4	0	0	2	8
Connor Schultz	1, 1a, 3	2	1, 1a, 2, 3	48	4	7	8	24	0	0	12	31