Completeness

While the final product does not achieve all the goals of the proposal (notable simulating multiple robots simultaneously), it does successfully surpass the minimum viable product. It simulates the match as planned, with the only change being the removal of balls on the field due to complexities in accurately simulating the feature, shows the output in a navigable console window, and shows the field map in a GUI. Given these factors, I would deem the project 90% complete per the specification as the simulation backend is capable of multiple robots, and the only limiting factor is the field GUI.

This is a good platform for an eventual distributable application. It allows for a diverse range of customisable parameters such as robot and field configuration and is prepared to handle the vast majority of associated errors. It has a solid backend which is 95% of the way to multi-bot simulation. Finally, is has a simple yet complete user interface which could support multiple robots with a reasonable set of changes.

As the MVP met the requirements in terms of features for course content, the final steps of adding multi-robot simulation are not required to demonstrate the potential or functionality of the application. The only remaining changes would be to track multiple robot's locations in the GUI, which is more tedious than complicated. The only notable incompleteness that may be troublesome is the likely existence of memory leaks in various areas, as this is my first time doing OOP in C++, and I am accustomed to working in languages with garbage collectors.

Testing

While no comprehensive testing has been run both due to time constraints and the difficulty of hand-calculating the complete behaviour of the simulation, a variety of smaller tests were run for various scenarios.

Input Validation

A variety of input files have been included under the tests folder to test various elements of input validation. These files are all derived from the same functional dataset, with deviations added to trigger specific behaviour. Most files only check one error as many result in termination of the program.

File Path	Error	Expected Behaviour	Passed
tests/bd1.csv	shot_range entry missing	csv_parsing, missing values	✓
tests/bd2.csv	Duplicate team names	invalid_parameter, duplicate teams	✓
tests/bd3.csv	Zero speed	invalid_parameter, speed 0	✓
tests/bd4.csv	Cannot cross defenses	invalid_parameter, cannot cross defenses	✓
tests/bd5.csv	Cannot score	invalid_parameter, cannot score	✓
tests/bd6.csv	Centre+side angle, time and range tolerances exceeded	Tolerance warnings for all listed On override, graph generation with interesting properties	✓

Additional testing was run to ensure validation procedures worked for console inputs, however those will not be listed here. They focused on providing values which did not match the expected format or requirements and ensuring proper handling. All inputs behaved as expected, with invalid filenames

causing closure, invalid teams and defenses re-prompting, and duration and location being unhandled for non-numerical values as that is not within expected user behaviour.

Simulation

Similarly, to input validation, configurations have been provided to test various simulation behaviours. These have been provided in the form of teams in the configuration file.

Team	Input variance	Expected Behaviour	Passed
2708	Control data, designed to demo everything	Robot navigates field using most elements	✓
901	Can only score low goal, times boosted to reduce advantage over high	Only scores in low goal, follows general navigation rules	✓
902	Can only score high goal, times boosted to reduce advantage over low	Only scores in high goal, follows general navigation rules	✓
903	Can only cross defense default in pos 5. Cross time boosted to discourage crossing	Only crosses farthest defense (assuming defaults)	✓
904	Can only cross low bar, cannot score centre high	Only crosses low bar, only scores in low or side high	✓
905	Can only use centre goal from straight on and in close	Only scores from specified node close to goal	✓

Roadblocks

The largest roadblock while developing the application was building the graph and executing Dijkstra's algorithm. While for a personal project I would just follow the first implementation I found online (or simply use a library in higher-level languages), this project required a more complete understanding, which took some time to gain without official course materials. This longer time was mainly as I wanted to best match the format the course would likely use, so I had to do more complete research instead of simply following provided resources. Heap removal was also a minor roadblock, as I was working on a general template implementation, which lead to more complex operations with regards to comparing making swaps.

The delay in learning graphs was anticipated as I knew it would be something I had to learn, so I had already done some cursory research to mitigate the effect, notably watching the Computerphile video on Dijkstra's algorithm several times to ensure understanding before beginning. The issues with heaps were not anticipated and did hinder progress for some time. There is little I can do to fix these roadblocks, as they both come down to learning new things, so the best way for me to mitigate similar issues in the future is to start earlier and include more buffer time in the original plan to allow for such roadblocks to occur.