Note: This is derived from the readme, and most parts overlap with other provided documents

Installation

The project depends on the SDL2 graphics library. To simplify setup, it is recommended that you download the most recent release source from the Github repository, however the steps for manual SDL2 installation are provided.

1. Download the MinGW Development Library from https://www.libsdl.org/download-2.0.php
2. Extract the contents (will need program which can handle tar.gz)
3. Navigate to the correct version for your computer (tested with x86_64)
4. Copy bin/SDL2.dll to the project root
5. Copy the following to the SDL2 folder in the project:
   - lib/*.a
   - lib/*.la
   - include/SDL2/*

Compiling

batch script

Included is _run.bat, which will automatically compile if the exe is missing and run the project. Additionally, it will set console size to the ideal for use with the application. This is the recommended way of running.

makefile

There is a makefile in the root directory that can be used to build the executable.

g++/gcc

The program can be built using g++ with the following command from the project root:

```
g++ StrongholdSim.cpp robot.cpp utils/graph.cpp field/fieldnode.cpp game.cpp
utils/utils.cpp field/field.cpp ui/console.cpp ui/interface.cpp -o "main.exe"
-ISDL2 -LSDL2 -lmingw32 -lSDL2main -lSDL2
```

Running

Configuration

Information related to robot performance (such a speed, defenses, etc) is read from a configuration file. This file is a CSV with the below columns. If the values exceed limits which would prevent simulation, errors are thrown. If values exceed limits which would hinder results, warnings are thrown which can be overridden. Multiple robot configurations can be listed, and the user will be prompted to select one by robot number.

Name	Type	Description	Values
team	int	Team number, used in output	0-9999
can_low	boolean	Flag indicating if robot can go under low bar	0 or 1
shot_range	unsigned char	Maximum range for high goal shot [inches]	0-170

centre_shot_time	unsigned char	Time needed for centre high goal shot, 0 indicates unable [seconds]	0-255
side_shot_time	int	Time needed for side high goal shot, 0 indicates unable [seconds]	0-255
centre_angle	unsigned char	Maximum angle for shot at centre goal [degrees]	0-40
side_angle	unsigned char	Angle for shot at side goal [degrees]	0-40
low_time	unsigned char	Time needed to store in low goal 0 indicates unable [seconds]	0-255
defenses	long int (hex)	Hex value representing time to cross defenses , 0 means cannot cross and F means it takes 15s [seconds]	See below
speed	float	Robot's speed [fps]	>0
point_value	float	How many seconds of weight are removed per point scored, higher numbers means more defense crossing	0-9999

Defense Configuration

Category	Defense	ID	Position
A	Portcullis	0	0x0000000**F**
	Cheval de Frise	1	0x000000**F**0
B	Moat	2	0x00000**F**00
	Ramparts	3	0x0000**F**000
C	Drawbridge	4	0x000**F**0000
	Sally Port	5	0x00**F**00000
D	Rock Wall	6	0x0**F**000000
	Rough Terrain	7	0x**F**0000000

Inputs

Upon running the code, you will be asked to provide the following inputs:

- Config File: Path to the config file, defaults to ../robots.csv
- Team number: Number of the team to be used for the simulation. Must be one listed in robots.csv
- Defense configuration: IDs for defenses to use, in order of position. Input should include one defense from each categotry. Defaults to 0 2 4 6
- Simulation length: The number of seconds of match time to be simulated, defaults to 150
- Starting node index: The index of the node where the robot should start. To get indices refer to the generated graphml.

Simulation

After inputs are provided, the robot will be loaded and simulation will begin. This will take a shot amount of time to complete, depending on sim duration.

Event View

Once the simulation is complete, you will be able to see the list of events that occurred during the match. Events include driving between nodes, crossing defense, intaking a ball, and scoring goals.

Console

The console is the primary UI component. It shows the complete event list, with time, robot, location and a description for each event. Events can be navigated using the up/down arrow keys. The currently selected event will be coloured light blue. On the right, the time, score, and defense stats at the selected event can be seen. The user can press escape at any time to exit

Field GUI

The Field GUI is a complimentary window which shows an image of the field with a marker at the robot's currrent location, a line to it's previous, and a line showing the planned path, all with respect to the currently selected node. By pressing g in the console window, the graph will cylce between hidden, nodes only, and nodes + edges.

Test Configurations

There are a variety of robot configurations available for testing the application's behaviour.

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
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

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
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