7.4 Custom Program Design

What I Have Created

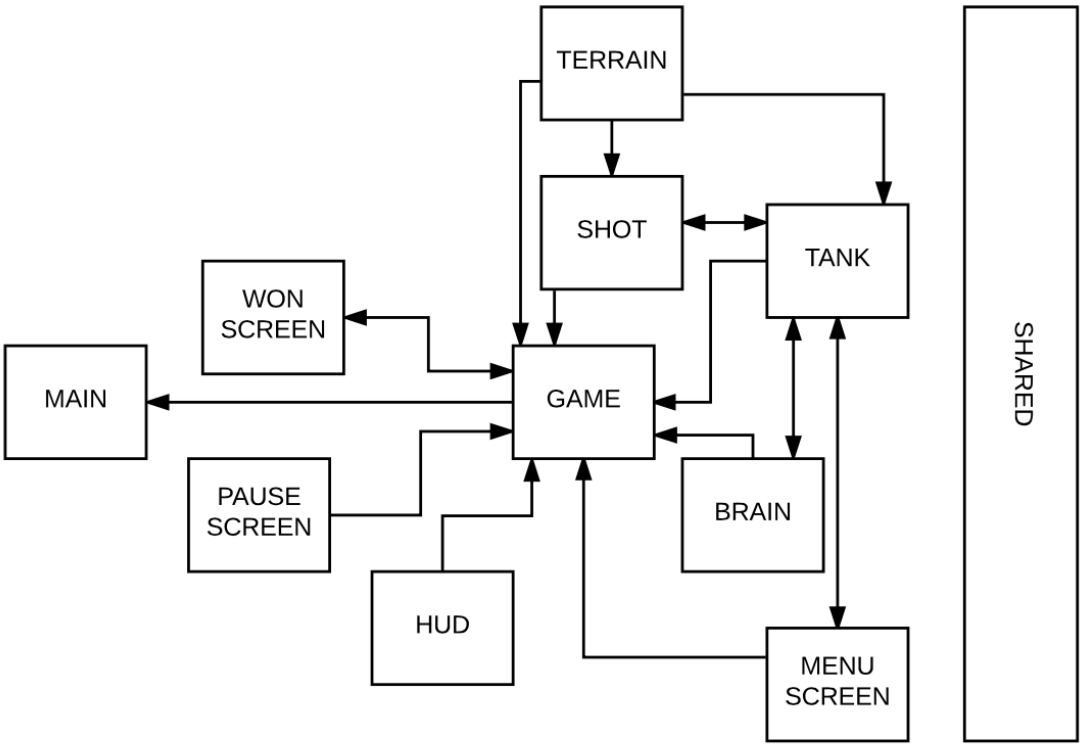
The custom program I have created is a game where tanks battle each other on a destructible terrain. It is a naïve recreation of the core functionality in the classic game Scorched Earth: http://www.abandonia.com/files/games/70/Scorched%20Earth_5.png

I aimed to create a relatively complete application, with the standard game states expected by users, full gameplay including menu controls, game physics, sounds, computer opponents and randomized elements to improve gameplay.

A demonstration video can be found here: https://youtu.be/09NeqEEkkf0

Solution Architecture

Below is a diagram of the models (also “main” and “shared” utilities) comprising the application architecture.



These models each have a .cpp file and a header file.

The following custom types are used in the program:

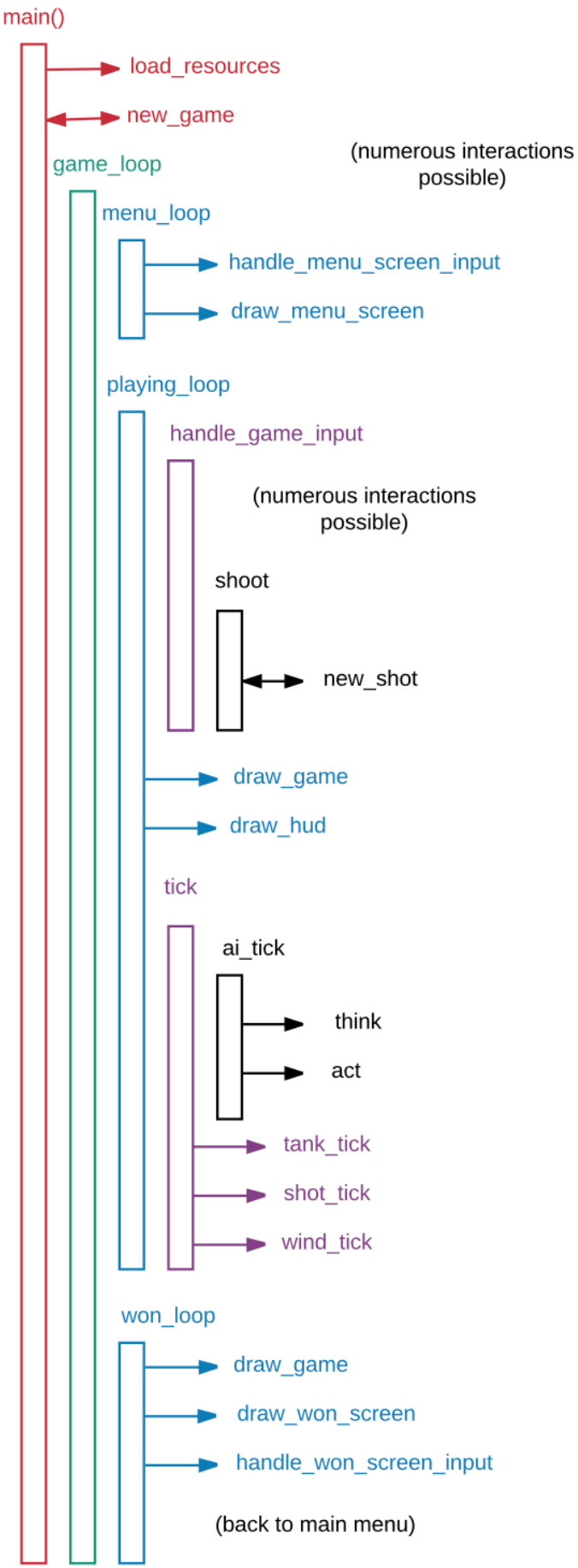
Aggregate Type	Type	Purpose
enum	game_state	States the game can be in
enum	brain_state	States the computer brain can be in
struct	terrain	Attributes of the current game terrain
struct	shot	Attributes of a shot that has been fired
struct	brain	Attributes for a computer player's planning
struct	tank	Represents a player (human or computer)
struct	ui_element	An individual interactible element of ui
struct	player_toggle	The toggle between human and computer
struct	menu_screen	All the ui elements in the menu screen
struct	won_screen	All the ui elements in the won screen
struct	game	The whole game

Flow Through Key/Golden Path

There are many branches and possibilities; a simplified path through the application showing some of the key functions/procedures is shown on the next page.

Each box and textbox with an arrow in the diagram is a function or procedure (it is conceptually similar to a sequence diagram).

Note that many of the arrows point to functions and procedures that actually call more functions and procedures. The subsequent calls have been omitted to present an accessible representation of the program. The application is relatively large, at 2678 lines including both header and C++ files.


main()
load_resources
new_game
game_loop
(numerous interactions possible)
menu_loop
handle_menu_screen_input
draw_menu_screen
playing_loop
handle_game_input
(numerous interactions possible)
shoot
new_shot
draw_game
draw_hud
tick
ai_tick
think
act
tank_tick
shot_tick
wind_tick
won_loop
draw_game
draw_won_screen
handle_won_screen_input
(back to main menu)