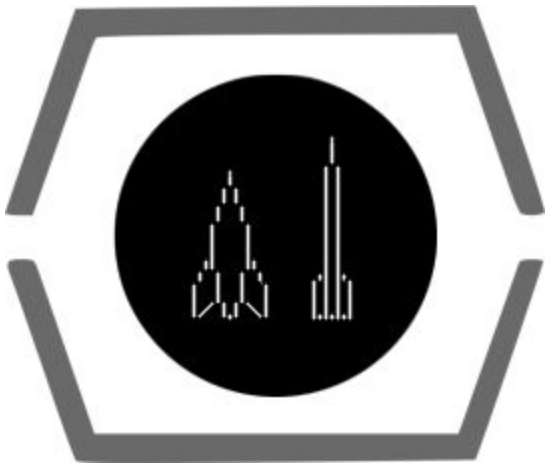
SpaceWar 2020



Final Project

Advanced Programming
Prof. Gilberto Echeverría

Monterrey Institute of Technology and Higher Education
Santa Fe Campus

May 11, 2017

By:
Alonso Iturbe // A01021621
Arthur Alves // A01022593
Jonathan Ginsburg // A01021617

Overview

Our final project is a space shooter-type network game in which each player must attempt to eliminate the other player by means of shooting laser beams while avoiding the opponent's hits.

SpaceWar 2020 takes its name and inspiration from Steve Russell's 1962 *Spacewar!*, a simple game developed for the DEC PDP-1 at MIT. The added twist on our program is that while each player plays on his or her own computer, they believe the other player is the enemy. In reality, they are both ships from the same armada (this bit is taken from *Black Mirror's* Season 3 Episode 5, "Men Against Fire").

Development

The game was developed primarily using Xcode, Atom and SFML. Additional resources such as sprites and fonts that were either taken from free internet archives, while the music file that serves as background music is an original creation.

Specific programming topics and their use in the program

- Inter-process communication: Since the program is a network game, TCP sockets are used to exchange information between the clients and the server. Additionally, a number of mutexes are extensively used to ensure there are no overlaps or errors while processing these messages.
- Thread creation: The server program uses multiple threads to handle the users connected concurrently to its sockets in order to correctly service each player's requests and updates.
- Pointers: Since all the information in the game must be dynamically updated and readily accessible, pointers are used to make all the program elements, structures and objects flexible and updateable by any function.
- Graphics: Since the program is not a terminal program but rather a graphical game, the SFML Library was used to make the user experience as aesthetically pleasing as possible.

Server <-> Client communication protocol

Client to Server Communication (C->S)

Message Type	Encoding	Triggered By	Triggers
Shoot (S)	S:<x-origin>_<y-origin>;	When user presses shoot key and shooting is allowed.	In server, triggers scattering of S->C S messages to other players.
Move (M)	M:<x-pos>_<y-pos>:	Once every other frame is drawn on screen.	In server, broadcasting of S->C P messages to other players.
Hit (H)	H::	Once, when user considers he has hit the enemy. In order to claim his point.	In server, triggers broadcasting of S->C U messages to *all* players (including H message originator).

Server to Client Communication (S->C)

Message Type	Encoding	Triggered By	Triggers
Assign ID (A)	A:<id>:	Start of the game, sent to each player.	Self assignation of player identification.
Shot from enemy (S)	S:<id>_<x-origin>_<y-origin>:	A C->S shot (S) message.	In client, display and animation of enemy projectile.
Enemy position (P)	P:<x-pos>_<y-pos>:	A C->S move (M) message.	In client, display and animation of enemy change in position.
Update scoreboard (U)	U:<id>_<score>[;<id>_<score>...]:	A C->S hit (H) message.	In client, updating of scoreboard, for displaying.
Finish game (F)	F:<id_winner>:	One player reaching the maximum score. Marks the end of game.	In client, a win or lose message and graceful shutdown of executable. Also, socket disconnection, initiated from server.

Instructions

Compile the programs

To compile and run this game, you will need SFML library installed on your computer.

Server program:

- g++ main.cpp -o spaceserver -lsfml-network -lpthread -std=c++11

Client program:

- g++ *.cpp -o spacewar -lsfml-graphics -lsfml-window -lsfml-system -lsfml-audio -lsfml-network -std=c++11 -lpthread

Alternatively, run the shell scripts included in the server and client folders.

Note: make sure that the compiled executable for the client is in the same directory as the provided resources folder.

To play the game

Run one instance of the server program first, and two instances of the client programs.

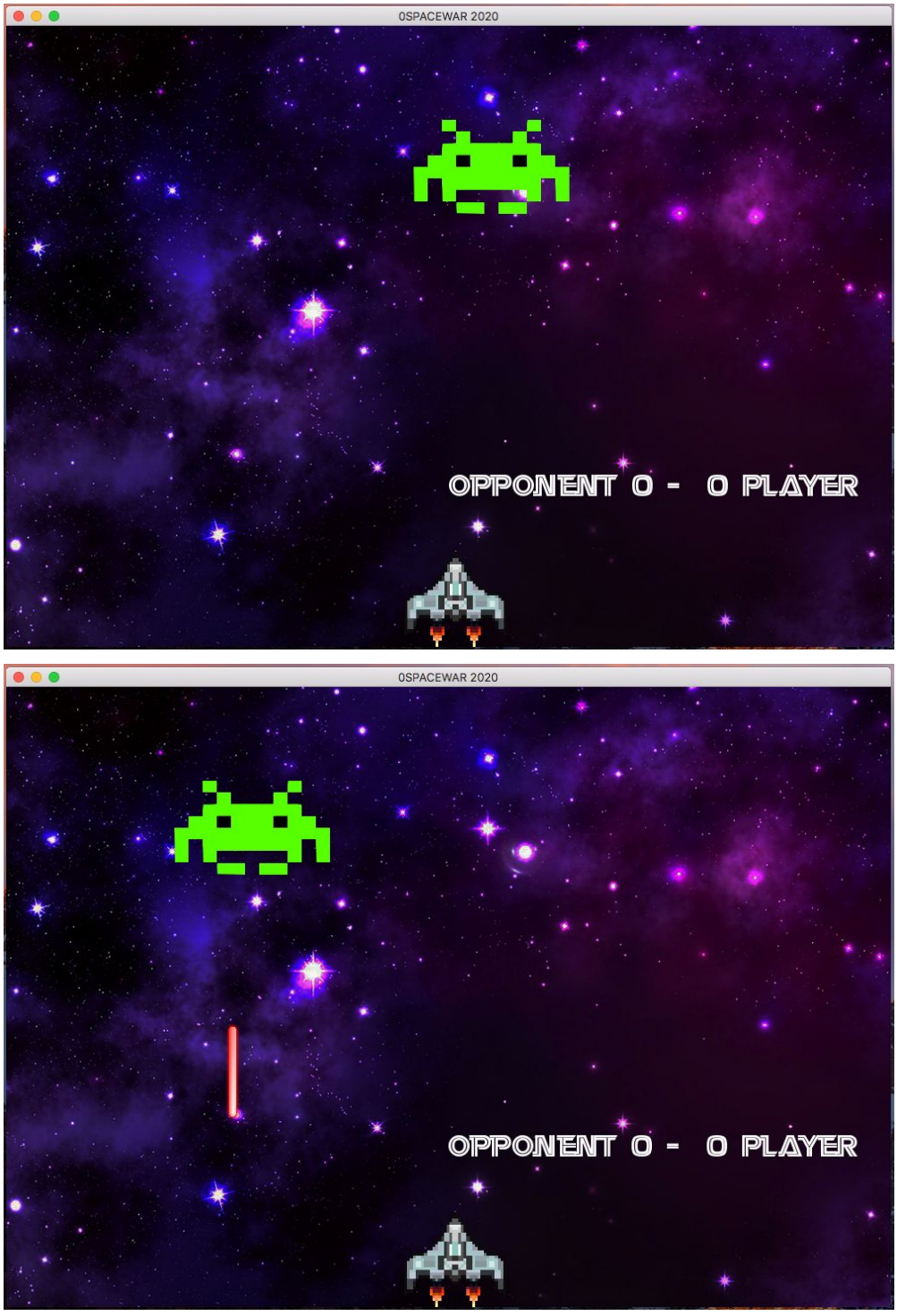
Server program:

- ./spaceserver <port-number>

Client program:

- ./spacewar <ip-address-of-server> <port-number> <input-type>
    - The port number must match the port number of the server program.
    - The input types include:
        1. Arrow keys to move and spacebar to shoot
        2. WASD keys to move and Tab to shoot
        3. Controller support

Screenshots



Images 1 and 2. Gameplay

```
Jonathans-MacBook-Pro-2:client MacBook$ ls
MainFunctions.cpp       MovingSprite.cpp        Resources               main.cpp
MainFunctions.hpp       MovingSprite.hpp        compile.sh
Jonathans-MacBook-Pro-2:client MacBook$ ./compile.sh
Jonathans-MacBook-Pro-2:client MacBook$ ./spacewar 127.0.0.1 30000 1
Waiting for other player to connect to server.
Player id from server: 0
Player's laser hit Opponent!
Player's laser hit Opponent!
Opponent's laser hit Player!
Opponent's laser hit Player!
Opponent's laser hit Player!
Player's laser hit Opponent!
Player's laser hit Opponent!
Player's laser hit Opponent!
Congratulations! You won!
AL lib: (EE) alc_cleanup: 1 device not closed
Jonathans-MacBook-Pro-2:client MacBook$
```

Image 3. Usage client

```
Jonathans-MacBook-Pro-2:server MacBook$ ls
compile.sh      main.cpp
Jonathans-MacBook-Pro-2:server MacBook$ ./compile.sh
Jonathans-MacBook-Pro-2:server MacBook$ ./spaceserver 30000
Player Joined! Players Online: 1
Player Joined! Players Online: 2
Jonathans-MacBook-Pro-2:server MacBook$
```

Image 4. Usage server

References

The main reference used for this project's development was the [SFML Library Documentation](), which aided us throughout the creation of the client-server message passing framework, since we used the native TCP socket functions included in the SFML/Network package.

String split into vector implementation by Stack Overflow user Zunino.
http://stackoverflow.com/questions/236129/split-a-string-in-c