

Capstone Project

1. Dependencies to run locally

cmake >= 3.7

make >= 4.1

SDL2 >= 2.0

gcc/g++ >=5.4

The program was developed locally using Visual Studio Code version 1.60.0.

Ubuntu 20.04 LTS terminal was used on windows.

Libraries version were the following:

cmake 3.16.3

make 4.2.1

SDL2 2.0.10

g++ 9.3.0

2. Compilation

2.1. Clone repository: <https://github.com/guimzago/Capstone.git>

2.2. Make build directory.

2.3. Compile using cmake e make

2.4. Run ./PlayerGame

3. Project

The Project is a multiplayer game based on the snake game from the repository. The players must get the food(yellow square) in order to get a point. Every time one of the players get a point, a maze(blue square) and an enemy(red square) randomly appear on the screen. The maze serves as a wall, blocking the path of the player. The enemy, if hit, will remove one point from the player and disappear from the screen. The first player to reach the score (between 5 and 20 points) determined before the game starts wins. A message is displayed in the console informing the winner. Figure 1 shows a print from the game screen with components identification:

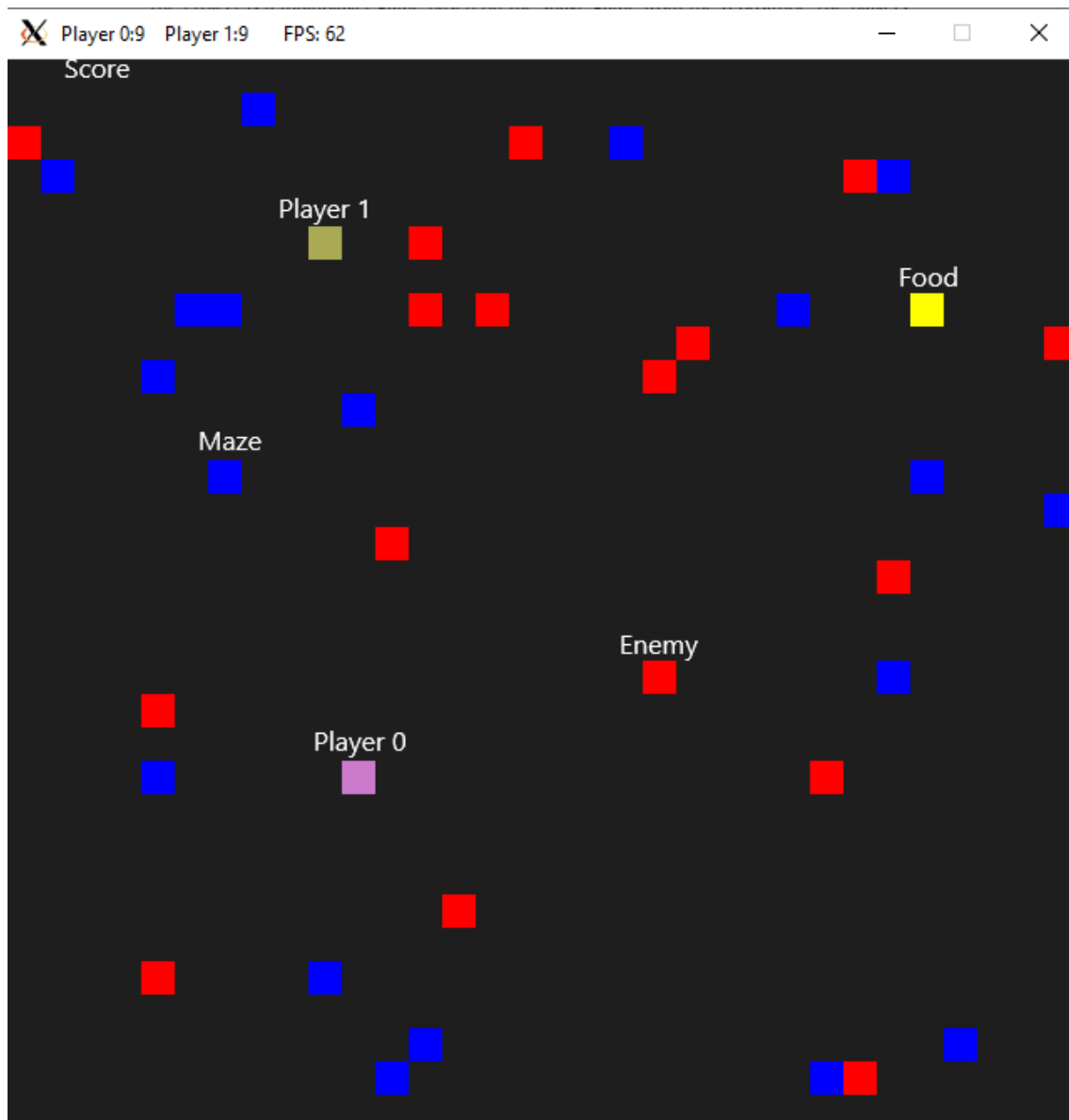


Figure 1 - Game screen

To move, the player has to press the key in the desired direction. To move again, the player has to release the key and press it again. The controls of the players are on table 1:

Table 1- Player Controls

	Player 0	Player 1
UP	Up arrow	W
DOWN	Down arrow	S
LEFT	Left arrow	A
RIGHT	Right arrow	D

3.1. Project files are divided as following:

3.1.1 main.cpp

This file starts the game. The score to win is input here and the renderer, controller and game are instantiated.

3.1.2. **player.cpp and player .h**

These files contain the class player. This class defines the player score, direction headed and updates its location on the screen, if not blocked by a maze square.

3.1.3. **controller.cpp and controller.h**

These files contain the controller class. This class takes the input from the keyboard and passes it to the player.

3.1.4. **renderer.cpp and renderer.h**

These files contain the renderer class. This class renders every component of the game. Players, maze, enemies and food.

3.1.5. **game.cpp and game.h**

The class game controls the main gaming loop and the components of the game. It controls when a player catches food, it places the maze and the enemies. It also controls the players score and if some of the players won.

4. **Rubric points addressed:**

1. *The project demonstrates an understanding of C++ functions and control structures.*

There are loops, conditionals and switches in the program. A few examples are listed:

If	Game.cpp line 75
For	Game.cpp line 99
Switch/case	Controller.cpp line 22

2. *The project accepts user input and processes the input.*

There is an input for the score and controller inputs.

Score input	Main.cpp line 18
Controller input	Controller.cpp line 22

3. *The project uses Object Oriented Programming techniques.*

Every files has classes with attributes and methods. Source: every .h file.

4. *Classes use appropriate access specifiers for class members.*

Every member of every class is specified as public or private. Source: every .h file.

5. *Classes abstract implementation details from their interfaces.*

Every function changes only what it is supposed to. The function name states what it does.

6. *The project makes use of references in function declarations.*

Several functions use pass-by-reference. Examples listed

<code>void Controller::HandleInput(bool &running, std::vector<Player> &player)</code>	Controller.cpp line 16
<code>void Controller::SetDirection (Player &player , Player::Direction direction)</code>	Controller.cpp line 7

7. The project uses multithreading.

The project uses threads for the controller and game update in the game main loop function.

<code>std::thread tController = std::thread(&Controller::HandleInput controller, std::ref(running), std::ref(players));</code>	Game.cpp line 34
<code>std::thread tUpdate = std::thread(&Game::Update, this);</code>	Game.cpp line 37