

UI Tests -

User experience is an essential component to any game. This is why we paid closed attention to making this game as user friendly and accessible as possible. To test this, we made sure that the menu was working and the rules were laid out clearly.

- Rules test

In this test ,we made sure that the menu appeared without any errors or warnings and worked when any of the options were selected.

The code for the test was -

```
// implementation of display_rules()
void Menu::display_rules(){
    std::string aim = "AIM:\n Help our dragon rescue as many eggs as possible! Watch out for im
    std::string points = "-> +2 points when the dragon collects the blue egg \n-> +1 point when
    std::string control = "GAME CONTROLS:\nMove dragon using 'W', 'S', 'A', 'D' keys\n";

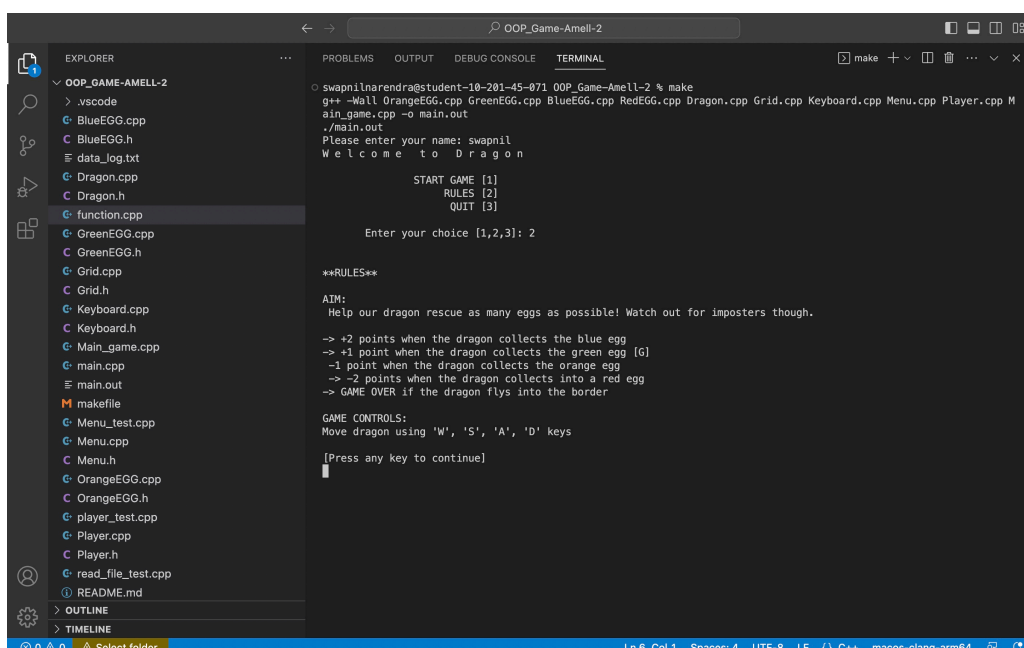
    //cout << "\033[31;42m";
    std::cout << std::endl << "**RULES**\n" << std::endl << aim << std::endl;
    usleep(50000);
    std::cout << points << std::endl;
    usleep(50000);
    std::cout << control << std::endl;

    std::cout << "[Press any key to continue]" << std::endl;
    std::string key;
    std::cin >> key;
}
```

The expected results -

- The menu should be displayed without any errors.

The result was -



```
g++ -Wall OrangeEGG.cpp GreenEGG.cpp BlueEGG.cpp RedEGG.cpp Dragon.cpp Grid.cpp Keyboard.cpp Menu.cpp Player.cpp M
ain_game.cpp -o main.out
./main.out
Please enter your name: swapnil
Welcome to Dragon

START GAME [1]
RULES [2]
QUIT [3]

Enter your choice [1,2,3]: 2

**RULES**

AIM:
Help our dragon rescue as many eggs as possible! Watch out for imposters though.

-> +2 points when the dragon collects the blue egg
-> +1 point when the dragon collects the green egg [G]
-1 point when the dragon collects the orange egg
-> -2 points when the dragon collects into a red egg
-> GAME OVER if the dragon flies into the border

GAME CONTROLS:
Move dragon using 'W', 'S', 'A', 'D' keys

[Press any key to continue]
```

- Main test

In this test ,we made sure that the menu appeared without any errors or warnings and worked when any of the options were selected.

Main Menu test -

The main menu of the game should be displayed with the appropriate options when prompted. The code for this feature is -

```
// Home page displaying options to choose from by the player
std::cout << "                START GAME [1]\n";
std::cout << "                RULES [2]\n";
std::cout << "                QUIT [3]\n";

std::string choice;
std::cout << std::endl << "        Enter your choice [1,2,3]: ";
std::cin >> choice;
const char* str = choice.c_str();

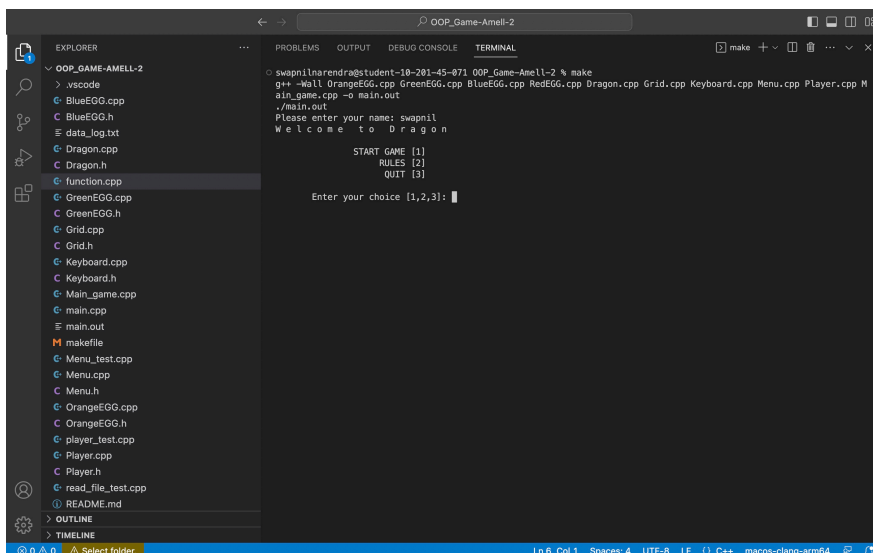
int c;
if (choice.length() > 1){
    c = 0;
}
else if(str[0] > '3' || str[0] < '1'){
    c = 0;
}
else{
    c = stoi(choice);
}

set_choice(c);
```

Expected output -

- The menu should be displayed without any errors.

The output was -

A screenshot of a C++ IDE (Visual Studio Code) showing a project named 'OOP_Game-Amell-2'. The Explorer panel on the left lists various source files including .vscode, BlueEGG.cpp, BlueEGG.h, data_log.txt, Dragon.cpp, Dragon.h, function.cpp, GreenEGG.cpp, GreenEGG.h, Grid.cpp, Grid.h, Keyboard.cpp, Keyboard.h, Main_game.cpp, main.cpp, main.out, makefile, Menu_test.cpp, Menu.cpp, Menu.h, OrangeEGG.cpp, OrangeEGG.h, player_test.cpp, Player.cpp, Player.h, read_file_test.cpp, and README.md. The Output/Debug Console panel on the right shows the execution output: 'Please enter your name: swapnil', 'Welcome to Dragon', a menu with 'START GAME [1]', 'RULES [2]', and 'QUIT [3]', and a prompt 'Enter your choice [1,2,3]:'. The status bar at the bottom indicates 'Ln 6, Col 1', 'Spaces: 4', 'UTF-8', 'LF', 'C++', and 'macos-clang-arm64'.