PROJECT OVERVIEW, ROMAN - S19B

INTRODUCTION

This document contains an overview of the machine program project of **Isaac Nathan Roman [ID 12208397]** from the section **S19B**.

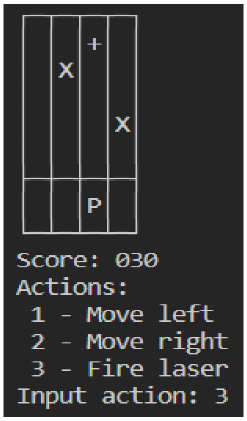
This is to certify that this project is my own work, based on my personal efforts in studying and applying the concepts learned. I have constructed the functions and their respective algorithms and corresponding code by myself. The program was run, tested, and debugged by my own efforts. I further certify that I have not copied in part or whole or otherwise plagiarized the work of other students and/or persons.

The zip file contains the following files / programs:

1. ***.C FILE:* Roman, Shooter [WITH BONUS]**
    - The .c file contains the machine program source code with the additional bonuses applied. It was programmed on-top of the original machine program source code, meaning that both .c files are similar with the exception of the bonus cases applied.
2. ***PDF FILE:* Test Script [WITH BONUS]**
    - The pdf file contains the function test cases. Note that some of the expected/actual output contain screenshots of the program for better visualization.
3. ***PDF FILE:* Test Run Sample [WITH BONUS]**
    - The pdf file contains a sample test run of the 'Roman, Shooter [WITH BONUS].' It details the enemy and player position per turn and contains descriptions on how the program functions.
4. ***EXECUTABLE:* WITH_BONUS.exe**
    - Executable of the .c file, 'Roman, Shooter [WITH BONUS].'
5. ***.C FILE:* Roman, Shooter [ORIGINAL]**
    - The .c file contains the original machine program source code. The program exactly follows the test cases provided in the provided pdf file prompt. I made sure to check other test cases (such as spawn point overlaps, etc.) before adding the bonus cases. Moreover, this serves as reference before I added the bonus cases.
6. ***EXECUTABLE:* ORIGINAL.exe**
    - Executable of the .c file, 'Roman, Shooter [ORIGINAL].'
7. ***PDF FILE:* Project Overview**
    - This file that you are currently reading.

HOW THE GAME WORKS

As described in the provided ‘CCPROG1 Machine Project’ pdf file, we were tasked to create a turn-based tiled shooter game. The player’s goal is to reach 100 points to win the game but if an enemy reaches the player’s field, the player loses.



```
Score: 030
Actions:
 1 - Move left
 2 - Move right
 3 - Fire laser
Input action: 3
```

Figure 1: Game preview

The screenshot provided shows a preview of the WITH_BONUS.exe. Note that the explanation below covers the game elements of the WITH_BONUS.exe file.

The game elements are:

1.) **“P”** - Represents the player. The player has three actions: move left, move right, and fire laser. If the player decides to move out of bounds, an error message will appear and the turn will not count. Moreover, if the player decides to input an invalid action, an error message will appear and the turn will not count.

2.) “X” - Represents the enemy, there are a maximum of 3 enemies per game. They have a set algorithm which is ‘right -> down -> left -> down’. They spawn randomly when they die. If an enemy is in grid borders and their algorithm tells them to move past borders, they move downwards instead. There are scenarios wherein the enemies overlap with one another causing it to seem like there are missing enemies, when in fact there is just an overlap. Once an enemy moves down in 5th row, they disappear and the player loses.

3.) **“+”** - Represents the power-up. It spawns/relocates every 5 rounds and gives the player a x2 score multiplier if shot. It never spawns on-top of an enemy but it can be destroyed by one. If an enemy moves to the location of the power-up, it breaks and disappears. If a player decides to shoot a power-up and an enemy is behind it, the player gains the 2x multiplier and kills the enemy behind it.