CSE - 1211 Project Report

Project Name : Militia Combat

Group Members :

- Fahim Shakil (Roll : 06)
- Md. Rokonuzaman Rokon (Roll : 08)
- Md. Khairul Mia (Roll : 64)

Department of Computer Science & Engineering
University of Dhaka

Abstract :

The goal of our game project is to design a 2-D graphical computer game using SDL2. For our project, we have tried to design a 2-D shooting game where the objective of the game is to defend the character from the enemy. The player, here is a military man , has to stay alive as long as possible and in this way he can make a high score. The game is made in a Linux environment and written in C/C++ with the use of SDL GUI. For our project, we have implemented modular programming and used functions extensively as taught in our classes. As a result, we have created a 2-D shooting game that is easy to play and entertaining.

Game Description :

This is a single player 2D platform based game. Here our character can move or jump or shoot to shield him from the enemy and to kill the enemy to get more scores. The game goes on as long as the enemies cannot reach our territory. If the enemy can hit our player several times, the health will decay and at one point our character will die and then there will be an option to start again or quit. To kill enemies our

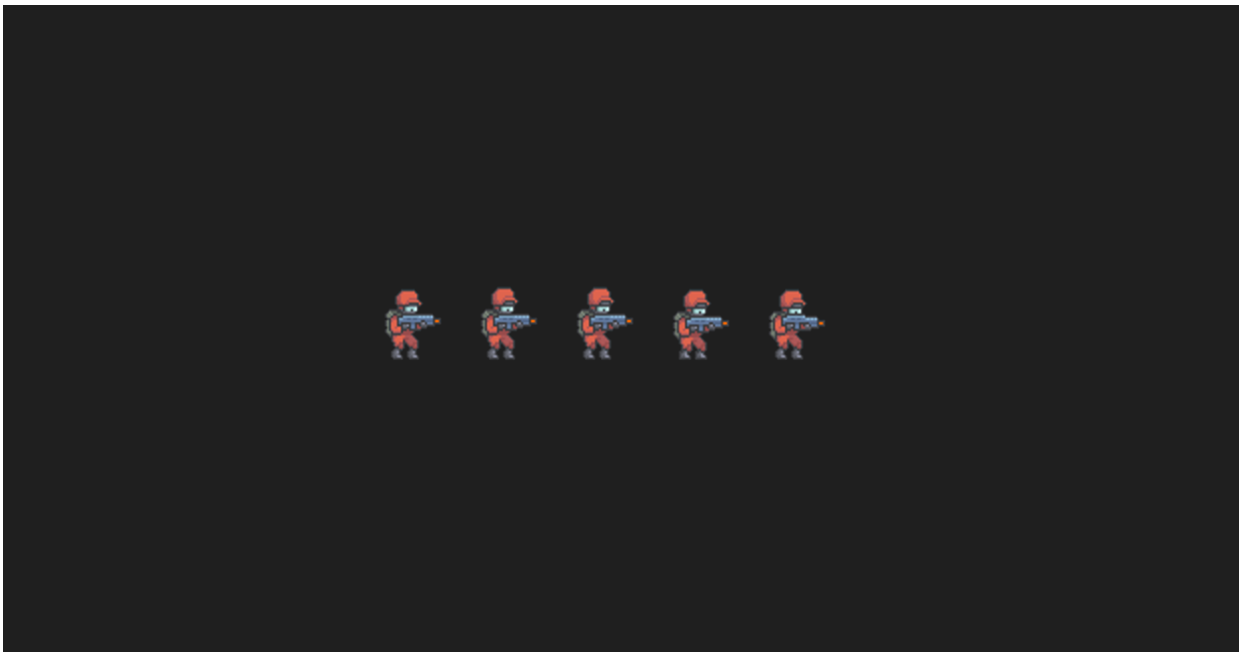
character will have a gun. Our enemy will die after being shot several times. We have used SDL2 libraries as our Graphical User Interference. In this section we will elaborate on the description of the game environment and user controls.

Background :

We have used this image as our background. The theme of this game is set in a jungle where our character will try to save the territory from the animals.

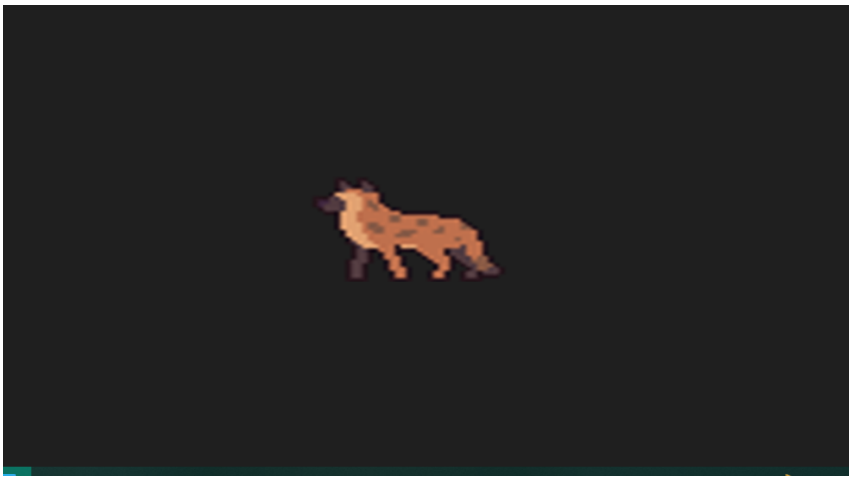
Characters

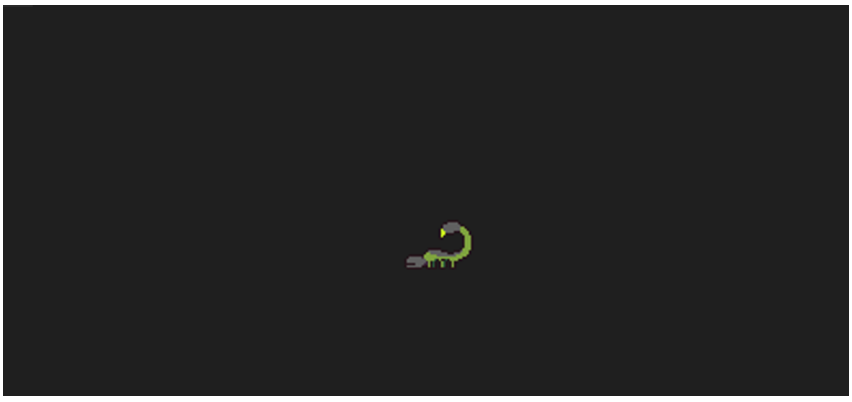
The Military Man :



The major character of the game that represents the player. The character would be visible when the player is playing the game because Militia Combat uses a third person view. The player character is animated when it moves, when it shoots bullets and also when it is shot.

The Enemy Character :





These are the villains of our game. There are multiple enemy characters in our game such as : Scorpions, Snakes, Hyenas and Vultures and they can walk and they will try to reach our territory . The player character should hit and kill as many enemies as possible and keep them away from our territory .

User Manual :

We can control the game play using specific keys on the keyboard and mouse. The different movements of the mouse and keyboard controls available for our player character are listed below :

For the movement :

- Key ‘w’: The player jumps
- Key ‘a’ : The player moves backward
- Key ‘d’ : The player goes forward

For Shooting :

- Mouse - Left Click : Shoots bullet towards the enemy

Other :

- Key ‘ESC’ : Terminates the game
- Key ‘Space’ : Pause or resumes the game

Project Files :

main.cpp, variables.h, initialization.h and initialization.cpp, Load.h and Load.cpp, score.h and score.cpp, highscore.txt, playerMove.h and playerMove.cpp :

The game starts from this file.
Main function is called as soon as the game launches.

Most of the variable declaration and game logic has been implemented in the variables.h file.

Then the init() function is called in the initialize.cpp to initialize the functions that are necessary for rendering images and sounds to the screen. The init function initializes SDL library, SDL_image Library, SDL_mixer Library and our start screen or main menu.
Then we can choose to start our game. The init function also loads game background, player character, enemy character,

and bullet texture. The Background music starts repeatedly on loop as soon as we launch the game.

We have initialized the game window,renderer, music, fonts and everything necessary and then closed them when they are not needed anymore. We have used functions init_SDL, init_Image, init_renderer, init_ttf_font in initEverything.cpp to initialize those things and then SDL_Quit(), SDL_DestroWindow, Mix_Quit(), in the closeEverything.cpp to clear everything from RAM.

We have declared all the variables necessary throughout the whole program at the top in variables.h using extern keyword so that all functions can access the variables.

We have used the file system to save the score and show highscore.

We have used different texture to render the images on the screen. We have used an array of textures to render many enemies at different times. We used the built in rand() function to place the enemies randomly so that the game becomes interesting. We made the enemy movement faster as the game went on to make it harder for us to save our territory.

We have kept a loop going over and over again to detect a collision between the bullet and an enemy. If they collide, we will set the alive state of the enemy to false and increase the player score by 1.

We animated the player character and the enemy characters using SDL_rect. We changed the frame after some milliseconds and set the frame to the first position as soon as we reached the last frame. That’s how the game looks lively.

We used music to entertain the user while playing the game. There is a music when the player shoots a bullet, there is a music also when the bullet hits the enemy and there is also a music as long as the game goes on.

Platform, Library and Tools :

Platform : Linux
Programming Language : C and C++
Libraries : SDL2, SDL2_Image, SDL_ttf, SDL_Mixer.
IDE : VsCode

Limitations :

We have some limitations in our game. The game could be made smoother. We wanted to add many more features also but for some unavoidable reasons we couldn't do so.We should have made the game more modular. We could have added some more levels and made the enemy shoot at us but we couldn't do it here. We faced challenges in terms of coming up with

new and innovative ideas for the entire game and in finding a unique angle to differentiate it from other games in the same genre.

Future Plan :

We will add some more features to make the game more interesting. Like enemies will shoot at the player character, There can be countdown times etc.

We have a plan to Play through the game to identify and fix any bugs or glitches. We will also try to gather feedback from our classmates to make any necessary improvements.

If we get good enough positive feedback from our friends, We might decide on a distribution platform (such as Steam or the App Store) and create a marketing plan to promote the game to potential players.

Conclusion :

Our concept of C and C++ programming got clearer while we were making this game.

We learned about a graphical user interface which is SDL. This was the first time we were doing something with the language and seeing the results on the monitor.

Most importantly, we have worked on real world project and we could implement the knowledge that we gathered throughout the whole semester.

We worked together as a team and it was fun making this game.