

3D Gun Range

CS352: Computer Graphics & Visualization Lab

Project Report

Course Instructor:
Dr. Somnath Dey

Submitted By:

Ramakrishna Raju Alluri – 200001065

Satwik Vankayalapati – 200001077

Siddhartha Chennareddy – 200002070

Introduction

Our project is to create a 3D Gun Range using OpenGL Library. A gun range is essentially a specific space created for competitions, training, qualifying, or practice with firearms. As they give people the chance to learn about weapon safety and practice shooting methods in a controlled setting, online 3D firing range may also be a great tool for firearms education and safety training.

As seen in the image, we attempted to recreate an outdoor shooting range. Our 3D firing range offers a more immersive and realistic training environment. Without the need for an actual shooting range, it offers users a virtual shooting experience that enables them to hone their shooting techniques in a safe and regulated setting. To improve the user's game experience, we developed a firing range with a variety of weaponry, maps, landscapes, and level settings. The user can choose from a variety of choices and play the game.



Specifications

Libraries used are:

- GNU C++ Library
- Open Graphics Library (OpenGL) – Graphics rendering API
- OpenGL Extension Wrangler Library (GLEW)
- OpenGL Utility Toolkit Library (GLUT)
- irrKlang Library – Audio Effects
- Simple Open GL (SOIL) Library – Texture Loading



Steps to run the project:

- Clone the GitHub repository.
- Download the necessary libraries such as SOIL, irrKlang
- Open the “src” folder in the terminal and execute the make file for the code using syntax – “make”
- Next, execute the compiled code using syntax – “./main”
- A window will pop up and you can start playing the game.

How to play the game:

- Use the mouse to select the options to play the game
- ‘W’ key – for moving forward
- ‘A’ key – for moving left
- ‘S’ key – for moving backward
- ‘D’ key – for moving right
- Mouse is used to change the direction of motion (field of vision of the person) and shoot just like in other games
- After playing the game, ‘F1’ to restart the game and press ‘F2’ to exit the game



Functionalities Implemented

The key objective of this project is to develop a shooting game (similar to a 3D gun range) in which users must hit a number of targets within a finite amount of time using limited ammo.

Main functionalities:

- 1) On opening the game, we get a start menu where the user gets to select the:
 - Difficulty
 - Atmospheric conditions of game play (day/cloudy evening (night sky))
 - Gun used to shoot the targets.
- 2) On starting the game, a first-person shooter's perspective of a firing range is created. There is a shooter (first- person view) in a shooting range. He has a certain range of movement beyond which he cannot move. He needs to eliminate five targets that are moving at a specific speed using his range of mobility. Two hits are required to knock each target to the ground.
- 3) Various other stationary obstacles and objects are placed in the location to enhance the look of the map – bringing it closer to reality.
- 4) Once the targets are down within the given time and ammunition, the shooter moves to the next level where the range of movement is reduced and the shooter is placed far away from the targets. Level to level also brings about changes to the map. In total, there are 4 levels – 2 maps (used alternatively). On completing all the levels or failing one (time limit exceeded or ammo completed), the scorecard is displayed and the user can either restart or exit the game.
- 5) As the difficulty increases, the targets move faster and the ammo provided at the beginning is reduced.

How each functionality works:

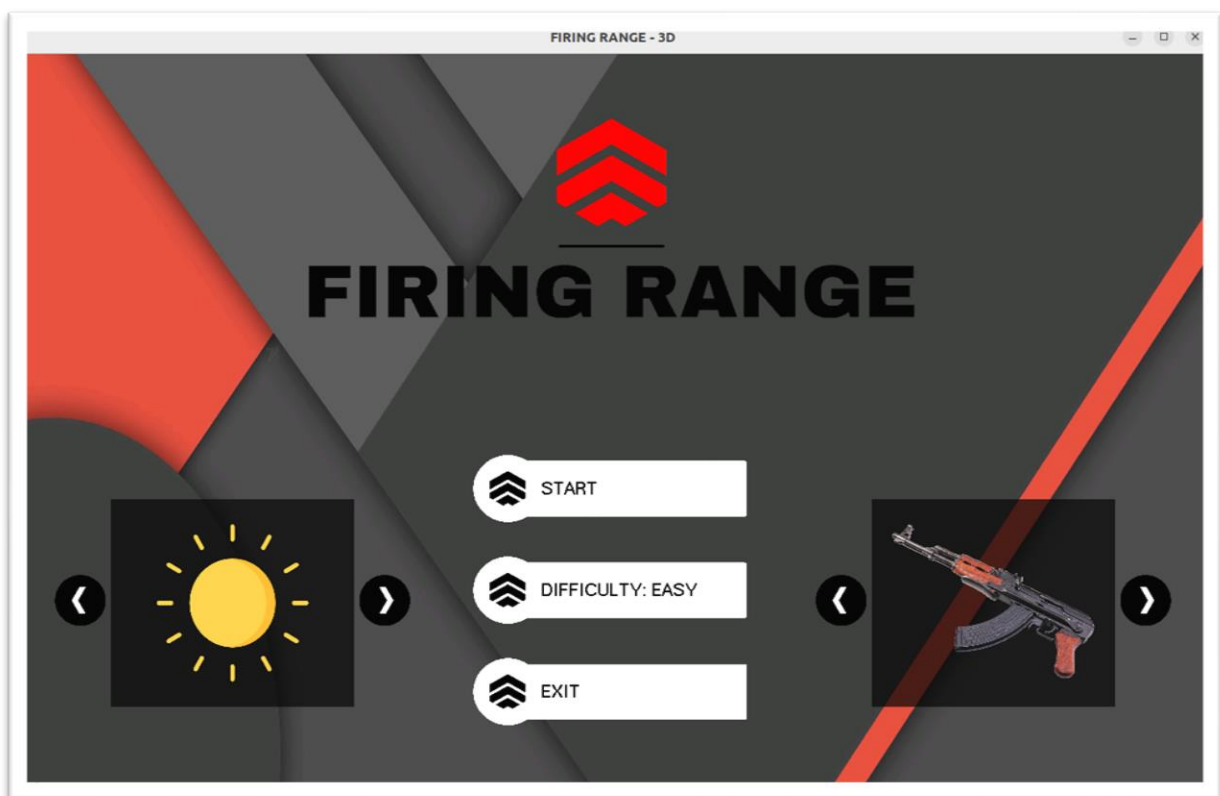
- 1) **The user selecting his choices** - This functionality works on the principle of picking up the background color on the screen and matching it with the allotted colors in the backend. The user clicks on the option which has a predefined background color which is different from the other regions on the screen. Each button/region has a unique background color. Depending on the color selected, the corresponding task is carried out.
- 2) **Rendering the images and 3D objects (generating the maps)** - 3D models are created using .mtl and .obj files of the object from their corresponding .png/.jpg files. 2D images are recursively rendered within an area using the “glBindTexture” functionality.
- 3) **Lighting effects** – As we are rendering the maps using images, changing the images as per the required lighting effects does the work. So, for shooting during the day time, we render an image with sufficient brightness. Whereas for an cloudy evening atmosphere, we render the same image by reducing its brightness.
- 4) **Movement of the shooter (changing the view)** - The shooter can move in four directions – forward, backward, left and right. All of the motions are with respect to the XZ plane. There is no motion in the Y- direction. Depending on the key pressed, we change the X and Z coordinates using translation and also calculate the yaw (rotation around vertical axis) and pitch (rotation around side-to-side axis) to rotate the field of view of the shooter. Checking out the coordinates of the obstacles, we restrict the movement of the shooter.
- 5) **Firing (Shooting)** – The field of view (screen) is changed in such a way that the user always shoots at the location present at the center of the screen. On clicking the mouse, the color of the center of

the screen is taken. If it matches with the predefined color of the targets, their life count is reduced. If their life count reaches zero, the target is no more rendered. The ammo count is reduced depending on the bullets fired. Additionally, we coordinate with the other functionalities while keeping an eye on the passing time.

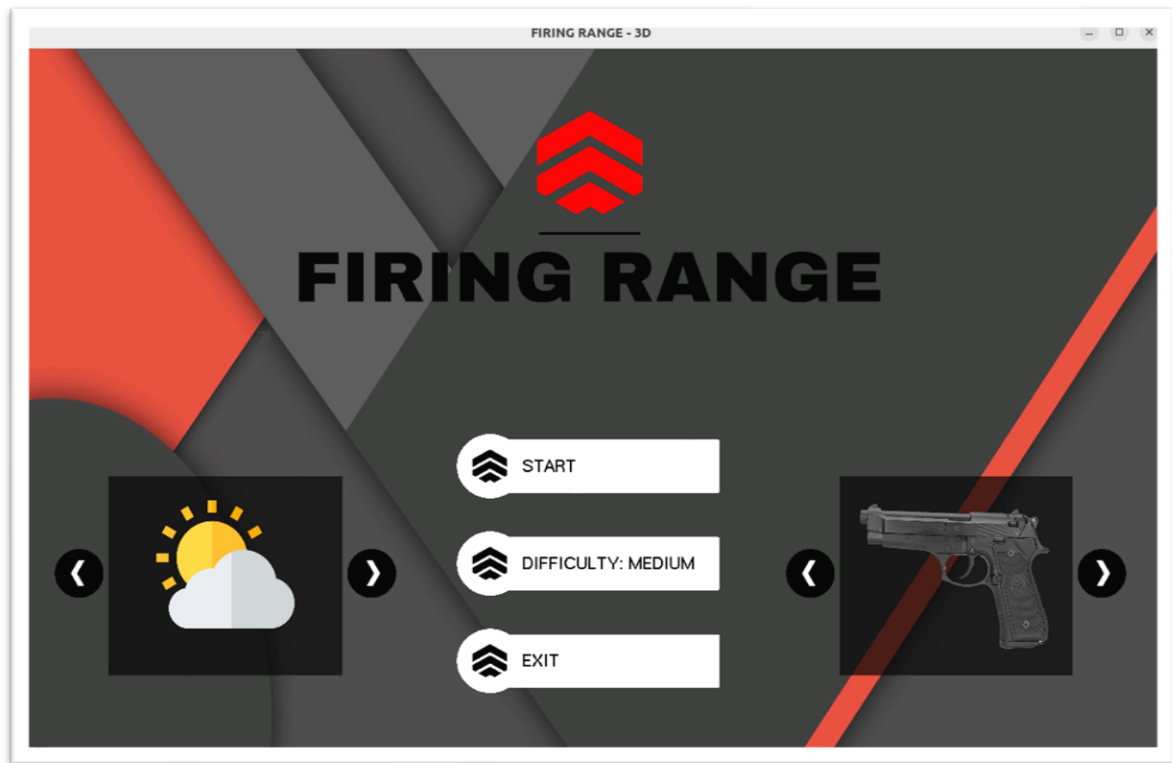
- 6) **Changing levels** – Once the targets are all shot down, the level changes and a new map is rendered. New images replace the previous ones. The minimum distance between the targets and the shooter increases. The location of some of the objects get changed like the table.
- 7) **Scorecard calculation** – Depending on the difficulty, the score gets updated. For a level of higher difficulty, more score gets added if the target is hit. Once the game is completed, the score gets displayed.
- 8) **Restarting the game, Exiting the game** – After the scorecard is displayed, on pressing the key 'F1', the game gets restarted and on pressing 'F2', the user exits.

Reference code - https://github.com/Lombavix-Team/FiringRange3D_OpenGL

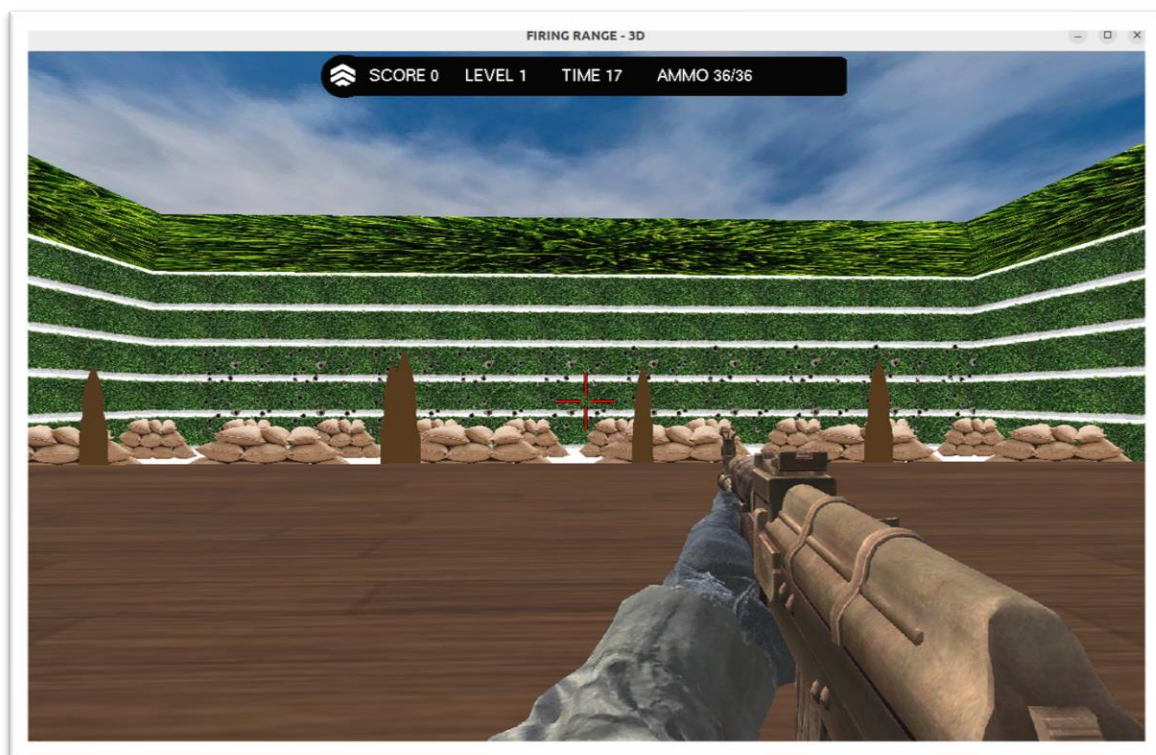
Output



Initial Start Menu of the Game – Choose the weather condition, difficulty and the gun. Start the game or exit.



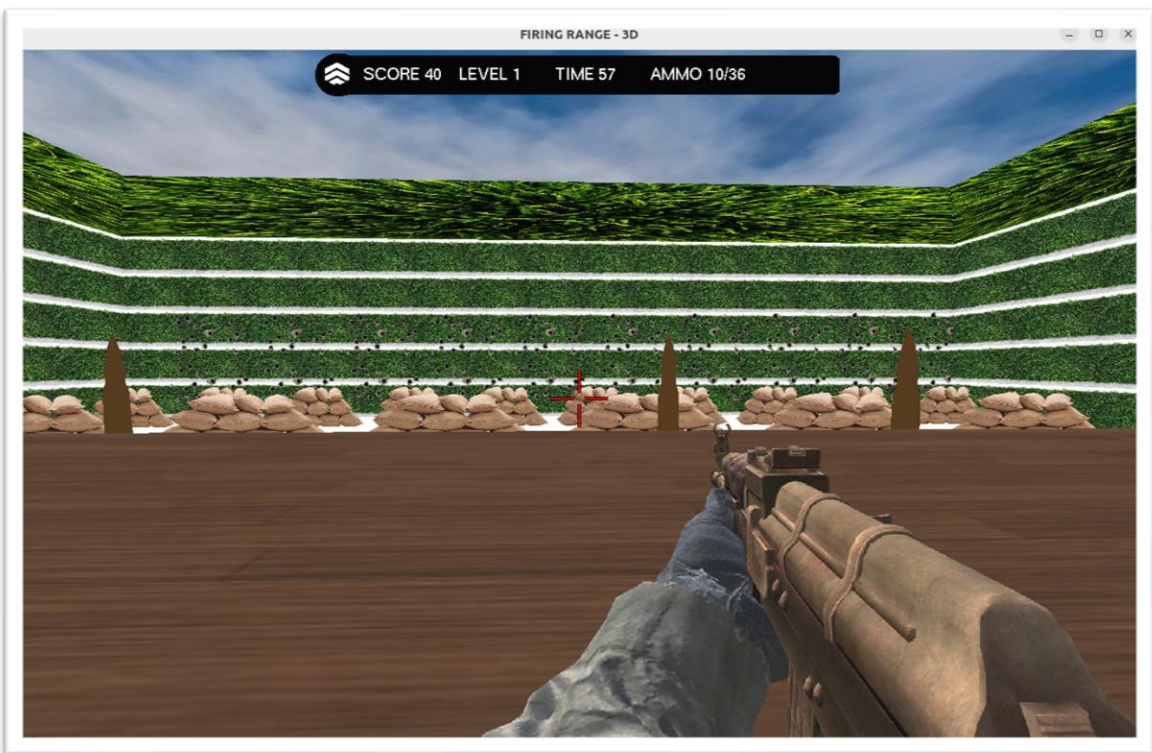
Initial Start Menu of the Game – By changing the selections



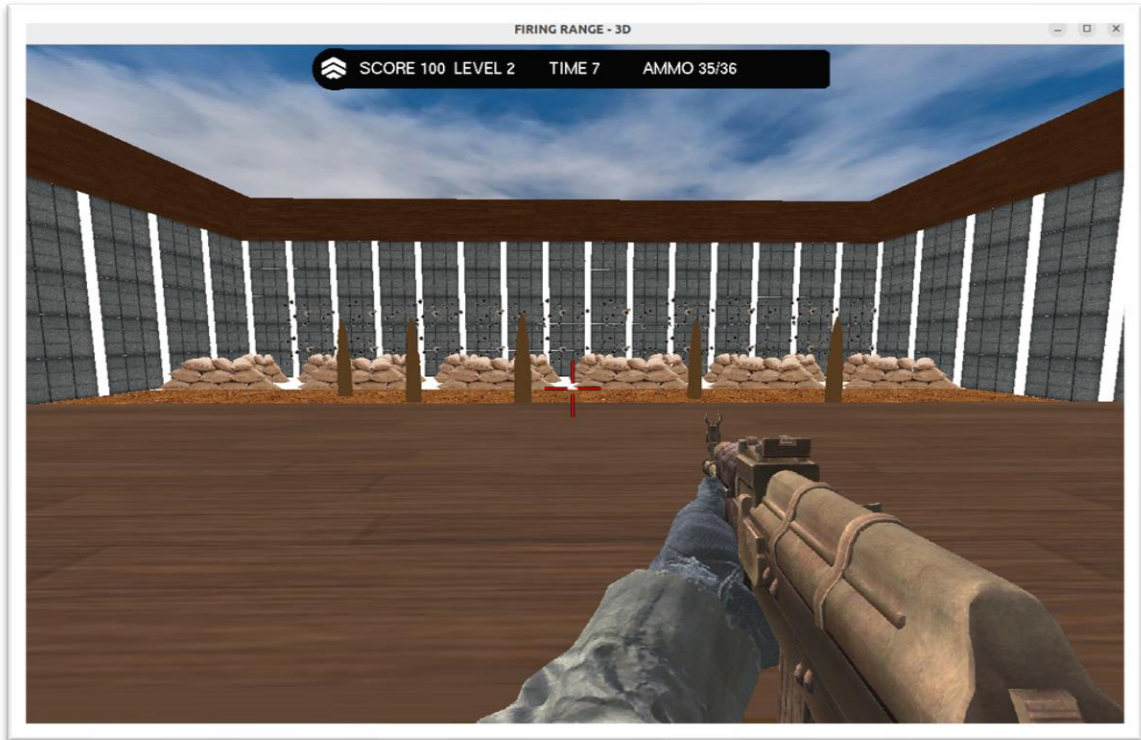
First Map : Difficulty – Easy : Level-1



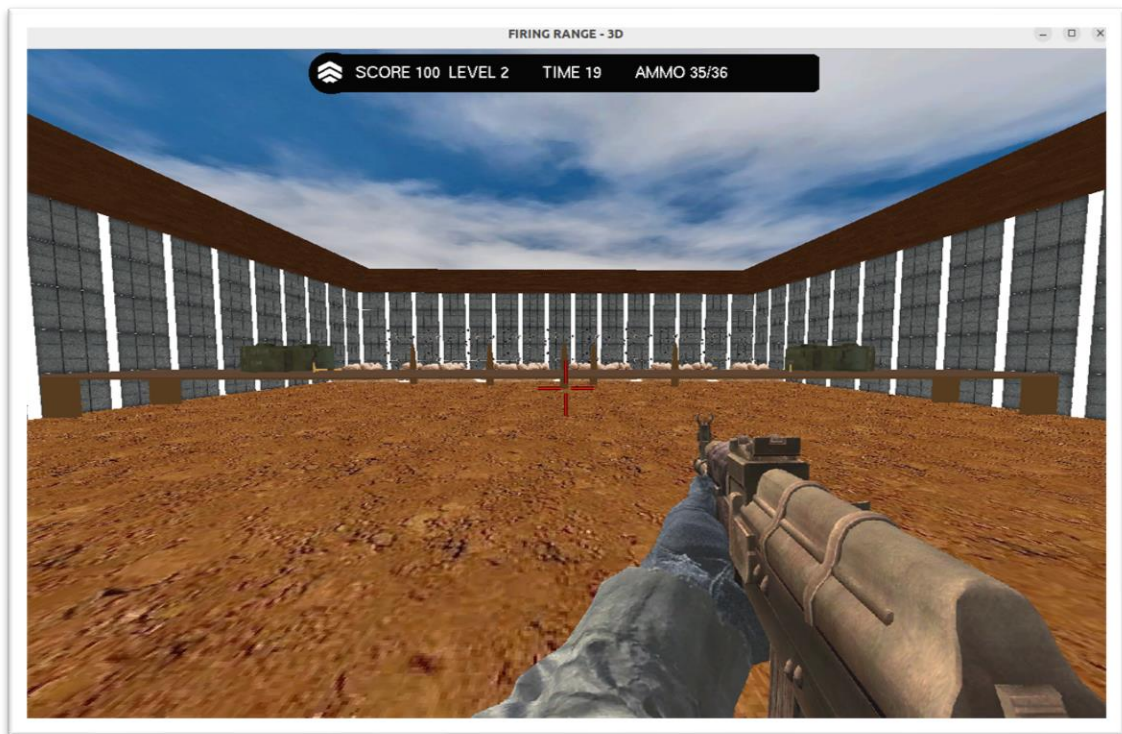
Wide range view of the Map



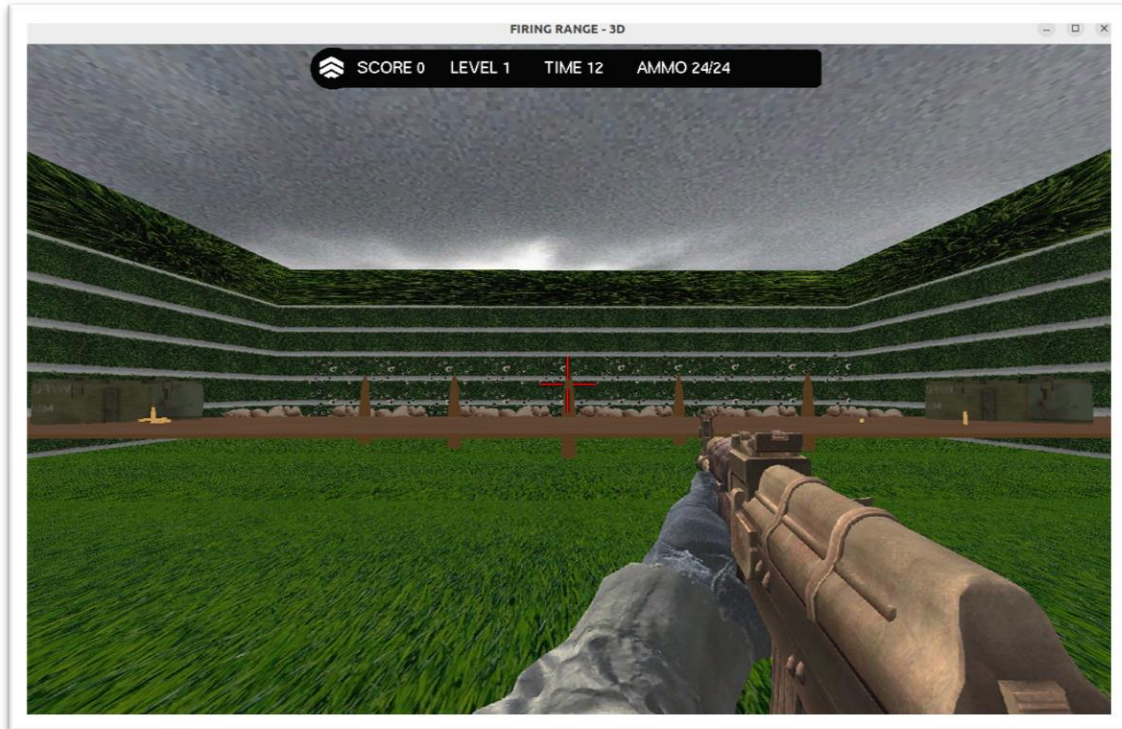
After shooting few targets – Ammo decreased, Time elapsed, Score increased



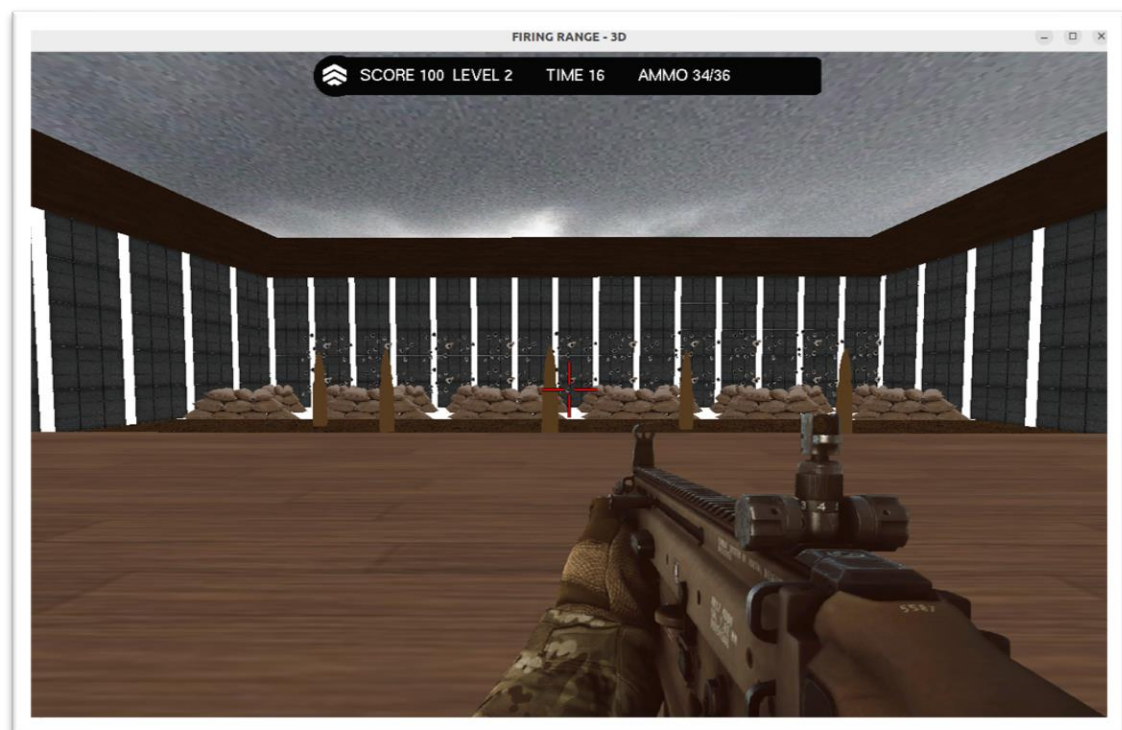
Second Map : Difficulty – Easy : Level-2



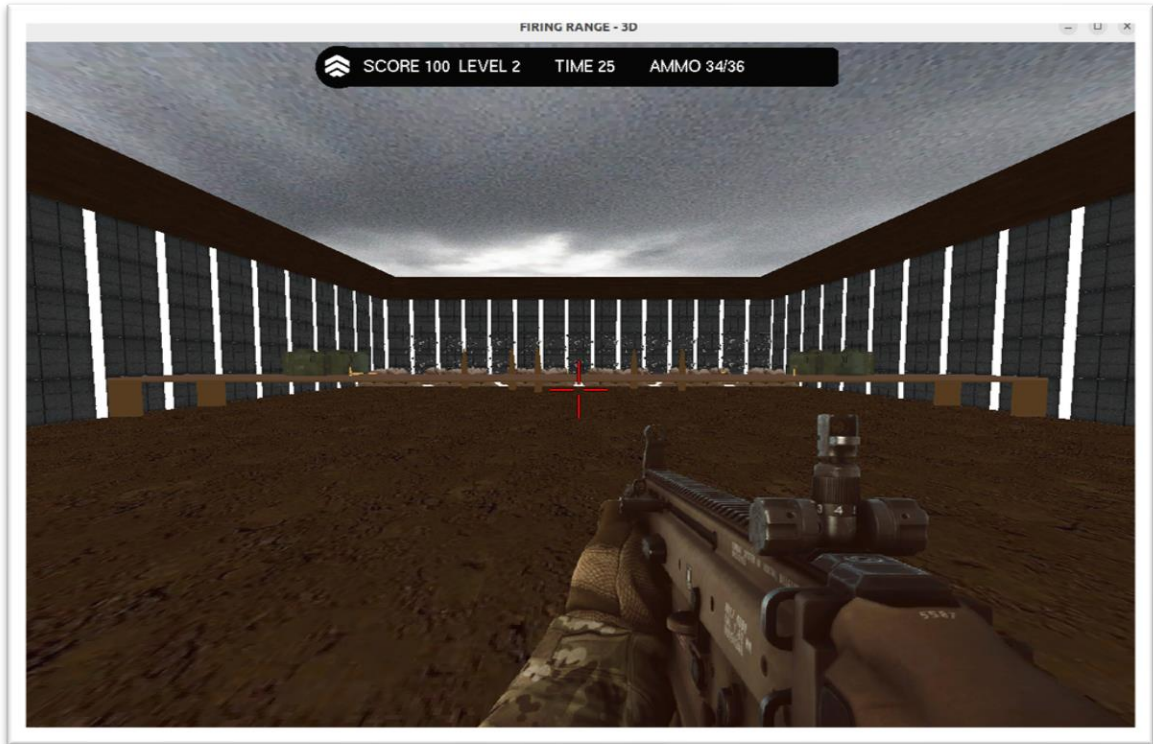
Wide range view of the Map



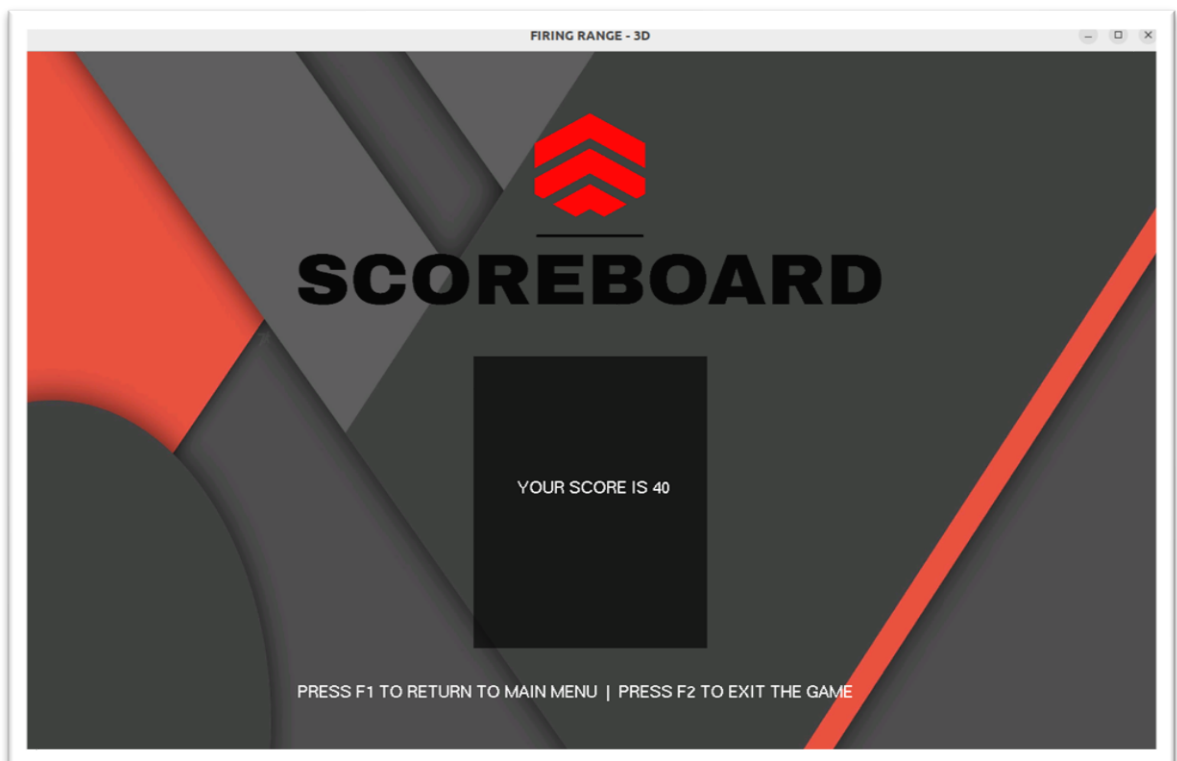
Wide range view of Evening/Night view of the first map (Lighting effect) : Difficulty - Medium



Evening/Night view of the second map (Lighting effect)



Wide range view of Evening/Night view of the second map (Lighting effect)



Scoreboard after the game has ended