University of Dhaka

Department of Computer Science and Engineering

Project Report:

Fundamentals of Programming Lab (CSE-1211)

Project Name:

R.I.O.T. (Running Is Our Therapy)

Team Members:

Ahaj Mahhin

Roll: 01

Sayed Md. Waki Assami

Roll: 11

Zisan Mahmud

Roll: 23

Introduction

Running Is Our Therapy (R.I.O.T.) is an SDL-based survival game written in C/C++ language. Simple, clear, and easily customizable code made this game distinctive. This game has attractive features and a good user interface. Use of keyboard and mouse interaction has made the game more enjoyable. The game is not an infinite game, so we decided to make 2 levels of the game and completing all the levels with maximum collection of point will be awarded a "Champion Crown". Based on the concept of “Jurassic Survival” like avoiding obstacles, hunting for food, dodging the enemies etc. we add fully new theme that made the game distinctive.

Objectives

1. Improving C/C++ programming language skills
2. Applying C/C++ logic in real life problem
3. Becoming familiar with app/game development
4. Improving problem solving skill
5. Learning how to implement structured programming in practical field
6. Finally, this game is a source of entertainment. So, giving people thrilling experiences by the gameplay is also an objective of the project

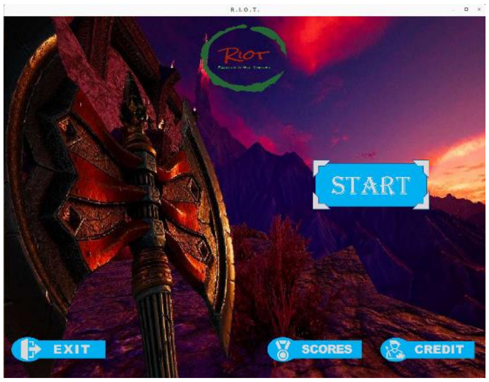
Project Features

1. Simple and structured code, easy to understand and flexible to change.

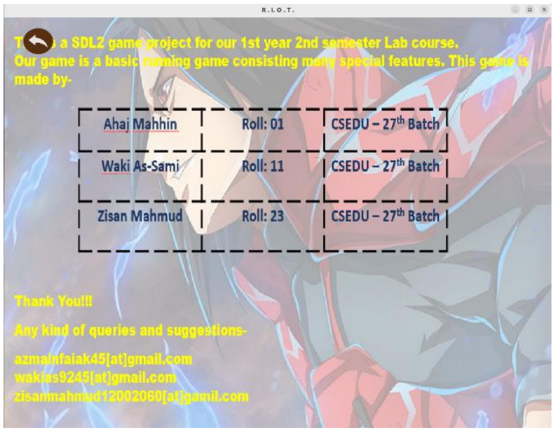
2. Easy installation in Debian-Based Computers and flexible frame rate on every device.

3. Memory efficient and smooth player interface.

4. Dynamic menu option to control the game.



5. Introduction to the owner of the game



R.I.O.T.

This is a SDL2 game project for our 1st year 2nd semester Lab course.
Our game is a basic running game consisting many special features. This game is made by-

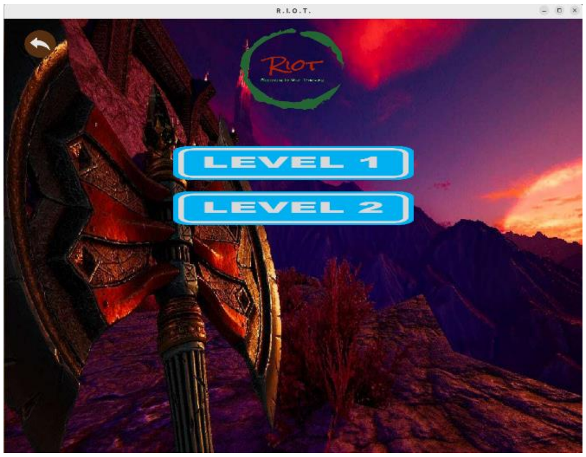
Ahaj Mahhin	Roll: 01	CSEDU – 27th Batch
Waki As-Sami	Roll: 11	CSEDU – 27th Batch
Zisan Mahmud	Roll: 23	CSEDU – 27th Batch

Thank You!!!

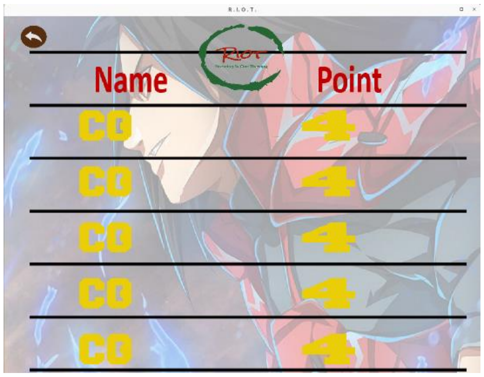
Any kind of queries and suggestions-

azmainfaiak45[at]gmail.com
wakias9245[at]gmail.com
zisanmahmud12002060[at]gamil.com

6. Separately display each levels High Score

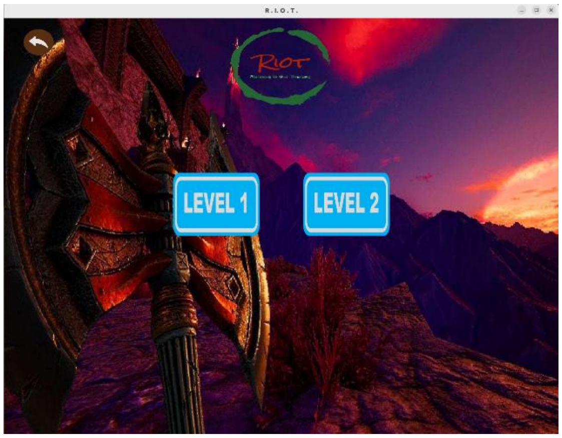


7. Displaying high score on the created score board.

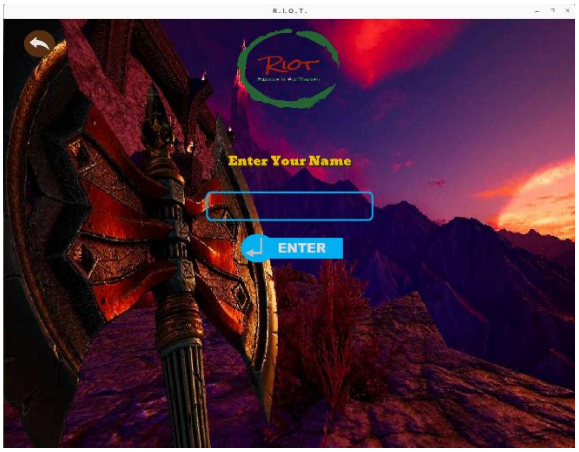


Name	Point
CO	4
CO	4
CO	4
CO	4
CO	4

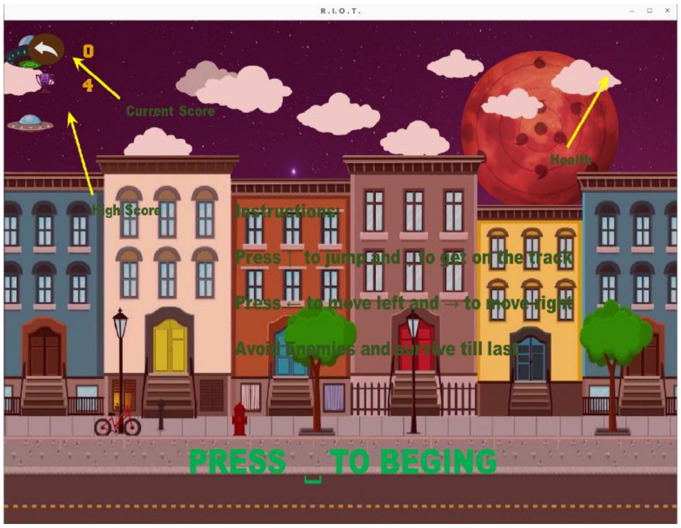
8. This game consists of two different level with attractive feature. Each level can be selected separately.



9. All players can input their name and their score is saved with respect to their name.



10. Level One has many attractive features. On the beginning, players are provided with instructions. After pressing the “SPACE” key, the game begins. In this level players can jump, move left and right to avoid enemies and to get reward. If the players can survive till last with at least amount of health, then they win, or a game over message is shown on the screen.



Instructions:

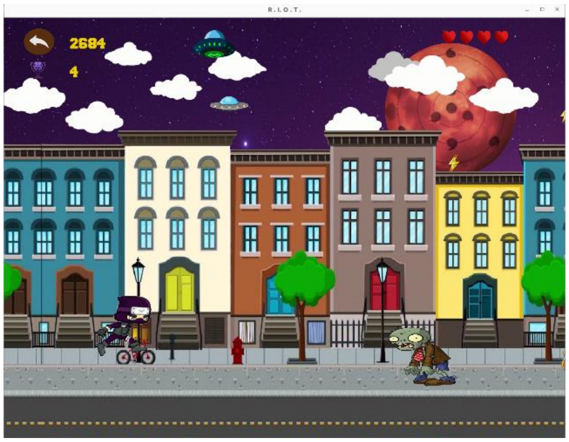
Press ↑ to jump and ↓ to get on the track

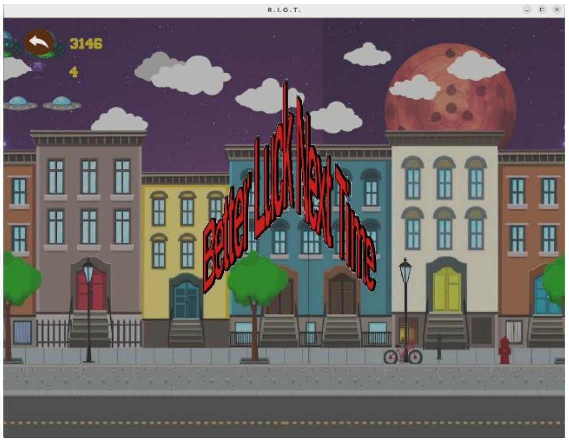
Press ← to move left and → to move right

Avoid Enemies and survive till last

PRESS ␣ TO BEGING

(Instructions)



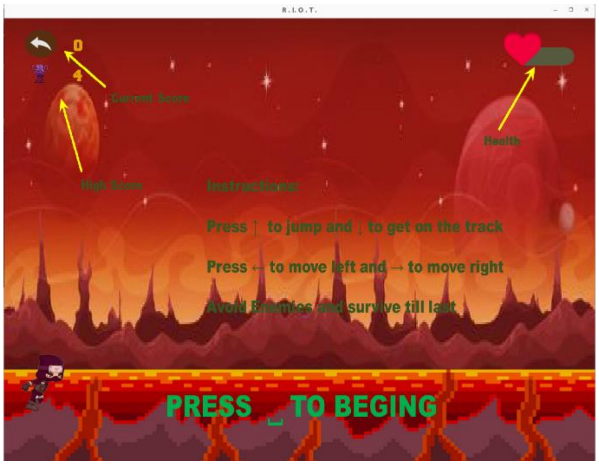


Better Luck Next Time

(Level One)

(Game Over)

11. Level Two has also many attractive features. This level is much more difficult to play in comparison with level one. This level also begins with instructions about how to play and pressing “SPACE” begins the game. Players must jump, move left-right to avoid the attack of the enemy. Movement is much harder that makes this level higher than before. The fruits appear in a different manner and having them increase the point. However, there is a recovery option of health to make this level much adventurous.



Instructions:

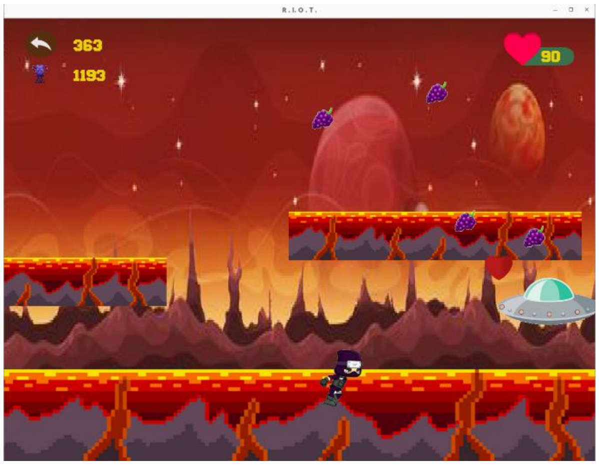
Press ↑ to jump and ↓ to get on the track

Press ← to move left and → to move right

Avoid Enemies and survive till last

PRESS ␣ TO BEGING

Instruction



Level Two

12. Modular distribution of code, availability of code in online GitHub with informative Readme file.

Project Modules

1. We used structure.h to make a reference structure that helps to create windows, create texture and render images separately.
2. We declared all the variables in the variable header files (variablesLevelOne.h & variablesLevelTwo.h) that make it simple to use all the variables throughout all the codes.
3. We make inputs separately, so we can change them when needed.
4. Rendering all the images and texture, we use draw header files.
5. Loading all the images and texture in initialize header files.
6. To detect the collisions between character, points, enemy we implement collision header file for each level

Team Member Responsibilities

Ahaj Mahhin (Roll: 01)

1. Developing game idea and features
2. Handling input and output, keyboard – mouse interaction
3. Graphics Designing (Photoshop)
4. Implementing SDL2 (Sound, Collision)

Sayed Md. Waki Assami (Roll: 11)

1. Making the source code structured
2. Implementing SDL2
3. Handling input – text
4. File system (Storing data)

Zisan Mahmud (Roll: 23)

1. Game logic design
2. Font, Instruction Coding
3. Source code writing in C/C++
4. Code testing and error fixing
5. Implementing SDL2

Platform, Library & Tools

- Platform: LINUX Kernel based OS – UBUNTU 22.04 LTS
- Library: SDL2 (Simple DirectMedia Layer is a cross-platform development library designed to provide low level access)
- Language: C, C++
- Tool: VS Code, gedit text editor

Limitations

- Character animation for both level is not too attractive, no extra animation for jump effect
- Crouch is not available

- There are some limitations while jumping, we could not implement a smooth jump effect
- Pausing the game is not implemented, it appeared to be much complex
- Graphics designing is still to develop
- No object-oriented code written or used
- We tried to connect both level one after another, but we couldn't succeed
- There is limited control over the character of the game
- Cannot save the current progress of the game

Conclusions

We gathered a mass amount of knowledge and experience through this project. We now know how to create our own unique header files and use SDL header libraries, have knowledge of the fundamentals of game production. Furthermore, we developed our ability to solve issues on our own. We collaborated as a team, which improved our ability to communicate and plan. Writing correct modular codes is what we learned. Because creating a game is not the same as solving typical C/C++ issues, we had to use our C/C++ expertise in a unique way. Even though it initially looked difficult, it boosted our coding abilities, taught us useful new skills, our capacity for problem-solving also improved.

Apart from learning new languages and technologies, this game project taught us cooperation, stress management, peer communication - We collaborated as a team, which improved our ability to communicate and plan, and a variety of other necessary skills.

Future Plan

As we are still at the beginner level, we couldn't implement all the features we wanted in our game. Also, there are lack of useful components in SDL2/C++ to make a high-quality game. We aspire to make it across all platforms (Android, IOS, Windows) 3D game using the power of Unreal Engine, Unity, Autodesk, Blender etc. later. With all necessary knowledge we will learn in coming days, we will fulfill our desire and make an outstanding game with fantastic features.

Repositories

GitHub Repositories: [*https://github.com/zisan23/CSE-1211-Lab-Project*](https://github.com/zisan23/CSE-1211-Lab-Project)

YouTube Video: [*https://youtu.be/PfBPaJ7uiiw*](https://youtu.be/PfBPaJ7uiiw)

References

Learning References:

1. https://lazyfoo.net/tutorials/SDL/
2. https://wiki.libsdl.org/SDL2/FrontPage
3. https://stackoverflow.com/
4. https://www.geeksforgeeks.org/sdl-library-in-c-c-with-examples/
5. https://www.geeksforgeeks.org/header-files-in-c-cpp-and-its-uses/

Materials and resources:

1. https://youtu.be/QQzAHcojEKg
2. https://youtu.be/KsG6dJlLBDw
3. https://www.youtube.com/watch?v=-CIkR1R3a_A&list=PLUFd3gYWwiYFdSoqZp35RU6mja4N5u4dz
4. https://github.com/Shahriar-Rumel/CSEProject-1201