University of Dhaka

Department of Computer Science and Engineering

Project Report:

Fundamentals of Programming Lab (CSE-1211)

Project Name:

Jack The Adventurer

Team Members:

04-Dip Durlov Halder
(Reg. No: 2020915614)
63-Maisha Noor
(Reg. No: 2020415673)

INTRODUCTION:

‘JACK THE ADVENTURER’ is an SDL-based puzzle game written in C. The code is written in a simple, modular, understandable way with the advantage of easily updating and customizing the code. It has ten manually designed levels. The game is based on the idea that a boy named ‘Jack’ has to reach his destination on each level passing many obstacles at the destination, he gets a scroll to go on to the next level, and on each level, the difficulty of the game increases.

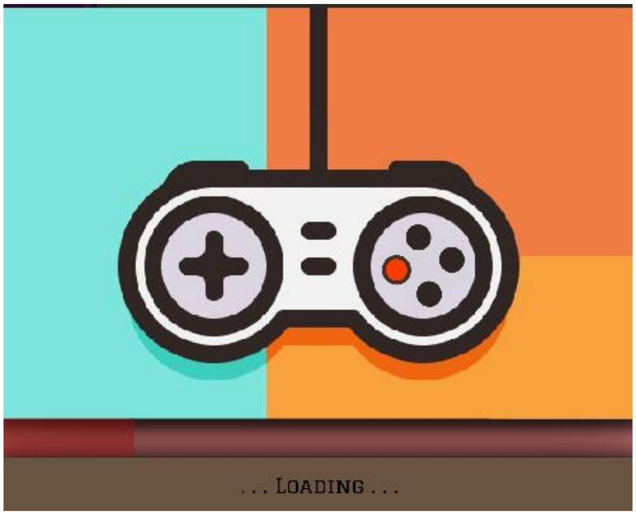
OBJECTIVES:

1. To apply the structured C language and make a real-life project.
2. Make the game attractive for the gamer with graphics and design.
3. Addictive game with extra features (restart button, easy accessing to each level) and increase of difficulty in levels.
4. Easily customizable code and room for further development.

PROJECT FEATURES:

1. Modularized, straightforward code that is easily modifiable, with comments where required.

2. An amazing loading page with extra features.

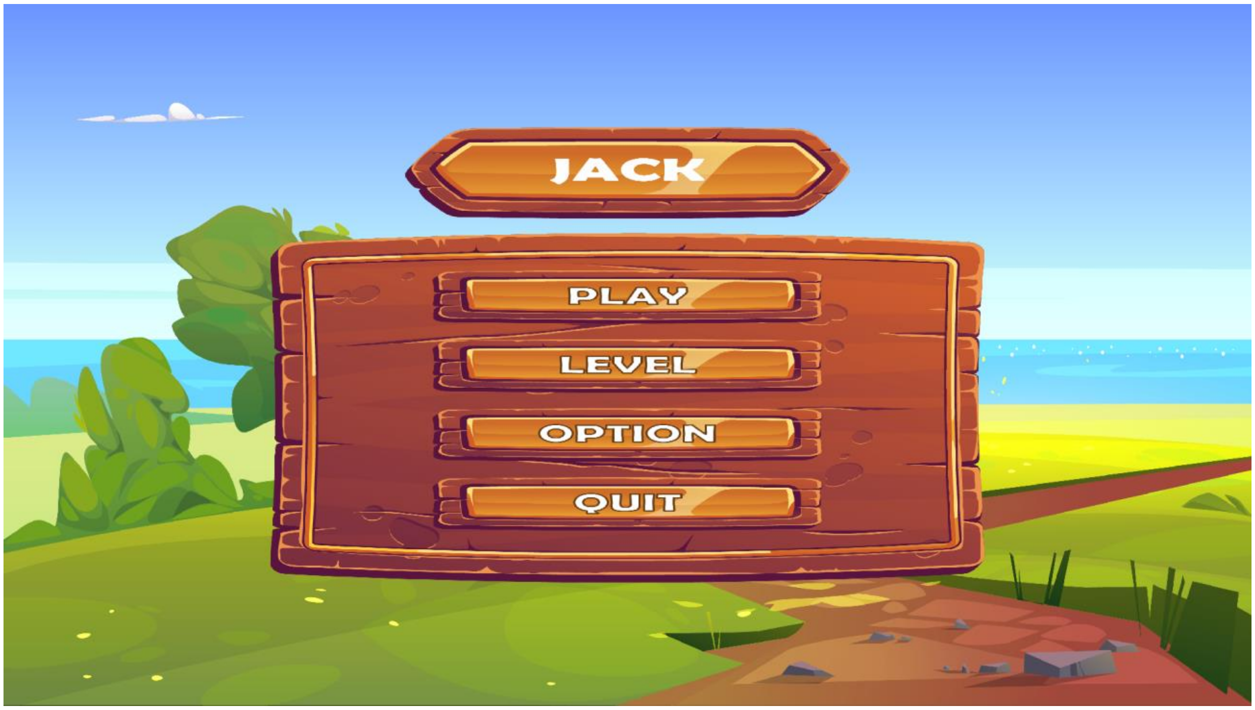


... LOADING ...

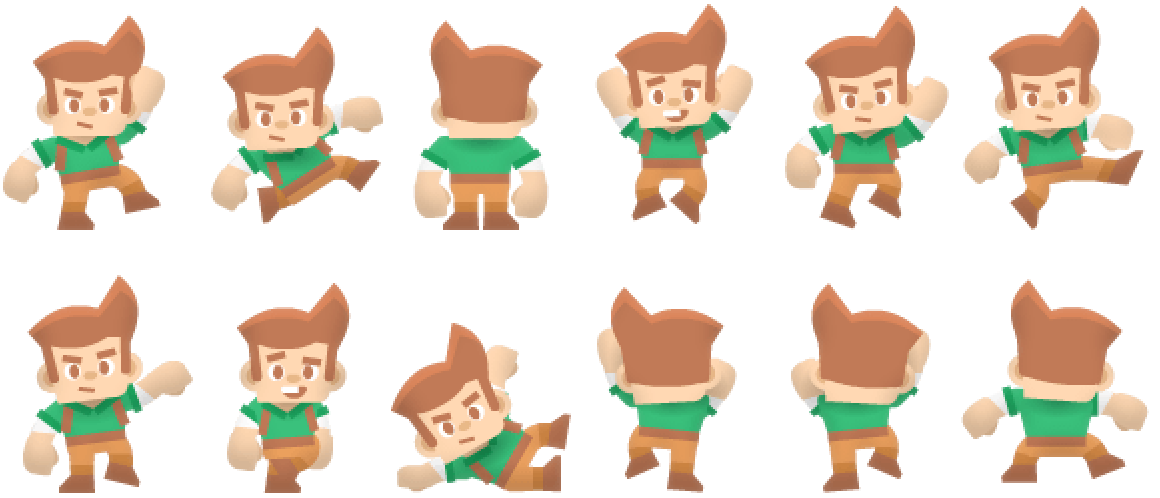
3. UI animation or User Interface animation is used which adds visual effects to UI elements and components to make them interactive.

Warwolf Studio

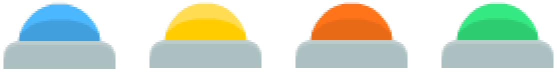
4. Extraordinary Starting Menu with various options, including levels, sound effects, and music.



5. Use of sprite animation makes it more realistic. The different movements have different looking points from climbing on a ladder to falling on the ground while walking on Grass-field.

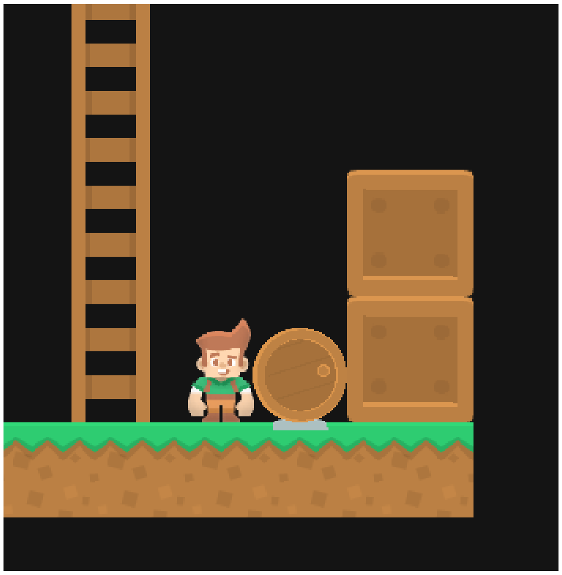


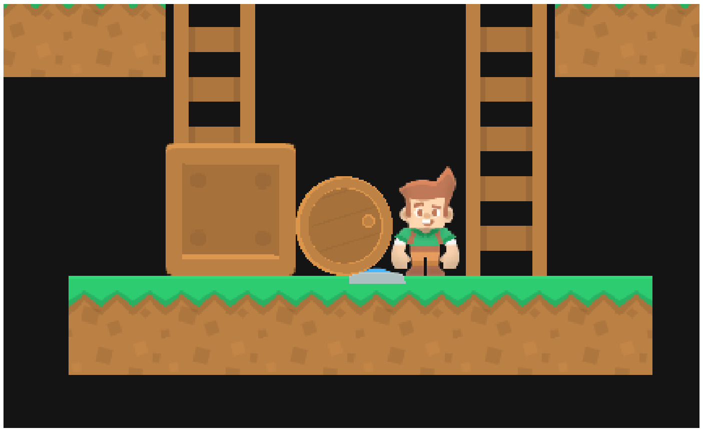
6. The color combination of different switches & doors makes it look more attractive. The Doors get down for a particular period based on the color of the switches.



7. There are barrels and boxes. Barrels move by spinning in a certain direction and boxes move from one side to another through every push.

Also, if two boxes are beside each other in any situation, Jack won’t be able to push the boxes. In the same case, if two barrels are also beside each other or one barrel and one box are beside each other, Jack can’t move any of them.





8. Barrels stop spinning when stuck by any door and keep spinning when pushed between two barriers.

PROJECT MODULES:

1. **gamefunc.h :**

arrayprint() : Used to print an array to generate a random map.

matrixprint() : Used to print matrix to generate a random map

Grass() : Copy every Grass-field having certain X and Y coordination and cubic shape grass field number to render.

Lad() : Copy Ladder to render

Grid() : Draw grids on the screen which help to take perfect measurements of a certain area where objects are going to be placed.

Switch() : Copy switches to render.

Door() : Copy doors to render

Max() : Find the max value of an array. Use to generate a random map.

FirstUp() : Find the upper left grass field number concerning the sequence of a generation grass field.

LastUp() : Find the upper right grass field number.

Jack() : Copy Jack to render.

Scroll() : Copy Scroll to render.

RightValid() : Conditions of Jack icon's right movement.

LeftValid() : Conditions of Jack icon's left movement.

DownValid() : Conditions of Jack icon's down movement.

UpValid() : Conditions of Jack icon's up movement.

FallValid() : Determine when Jack will fall.

SwitchCondition() : Determine whether the switches are ON or OFF.

Box() : Copy Box to render.

BoxRightValid() : Conditions of box's right movement.

BoxLeftValid() : Conditions of the box's left movement.

BoxFallValid() : Determine when boxes will fall.

BARL() :Copy Barrel to render.

BarRightValid() : Conditions of barrel's right movement.

BarLeftValid() : Conditions of barrel's left movement.

BarFallValid() : Determine when Barrel will fall.

BarTerminate() : Determine when Barrel will stop rotating.

Barrier() : Copy the wooden barrier to render.

BarReverse() : Control the barrel's reverse rotation after the collision with the barrier.

UI_restart() : Copy restart and home button to the corner of the screen.

2. level.h:

level_1() : Design the map of level 1

level_2(): Design the map of level 2

level_3() : Design the map of level 3

level_4() : Design the map of level 4

level_5() : Design the map of level 5

level_6() : Design the map of level 6

level_7() : Design the map of level 7

level_8() : Design the map of level 8

level_9() : Design the map of level 9

level_10() : Design the map of level 10

TEAM MEMBER RESPONSIBILITIES:

04- Dip Durlov:

1. Game design ideation and implementation.
2. Design and implementation of the skeleton of the header files.
3. Sprite animation, rendering of different states of Jack & background.
4. Creating functions of each object & logic functions of completing each level.
5. Code debugging and testing.

63- Maisha Noor:

1. Level Designing (Manually) and starting menu designing.
2. Handling mouse and keyboard events to control the main menu page.
3. Graphics designing of the starting menu page.
4. Rendering of various options and states.
5. Code debugging and testing.

PLATFORM, LIBRARY & TOOLS:

1. C/C++: Implementation of the basic code in C++
2. SDL2: SDL2 is a cross-platform development library designed to provide low-level access to audio, video, keyboard, mouse, joystick, and graphics hardware.
3. VS Code: Vs code is used to write the code which is a simple but powerful IDE
4. Git/GitHub: Using git to store the code online and work collectively.

LIMITATIONS:

1. Delay of loading the game after running the code.
2. Smooth Movement of Jack gets disrupted sometimes.
3. For climbing on ladders, Jack has to stand on a certain range, which sometimes shows problems.

CONCLUSIONS:

The main purpose of this project was to apply the learnings of C/C++ language to make a simple yet interesting game that anyone can play. We learned to work collaboratively and also come up with solutions to different problems. The most difficult part of the project was fixing bugs at different stages and implementing the ideas using SDL features.

FUTURE PLANS:

We will be constructing more levels and adding more attractive features on upcoming levels and will try to recreate this game in more advanced programming languages.

Repositories

1. *GitHub Repository: https://github.com/tikly31/Mind-Game/commits/main*
2. *YouTube Video: https://www.youtube.com/watch?v=Xw-C7NHCPjI*

References

https://lazyfoo.net
https://wiki.libsdl.org