**CS110: Fundamentals of Computer Programming**

**SEMESTER PROJECT**



**Date: 26-12-2023**

**CLASS: BESE-14-A**

**Submitted By:**

**Abdul Hadi Saqib (467626)**

**Moazam Saif (464994)**

**Submitted to:**

**Dr. Momina Moetesum**

**School of Electrical Engineering and Computer Science (SEECS)**

**National University of Science and Techno**

**SUBJECT: SEMESTER PROJECT  
  
PHASE III: PROJECT REPORT SUBMISSION  
  
Game Name: EL LABERINTO DE NUST**

**1. Introduction:  
  
1.1 Project Overview:  
  
This Game named as “El Laberinto De NUST” is a text-based game made as a Semester Project for the course: Fundamentals of Computer Programing. This project is made after implementing all the knowledge gained by this course. This game is programmed and designed by Abdul Hadi Saqib and Moazam Saif.  
  
1.2 Brief Description:**

This game is a Text-Based Adventure Game where the player spawns at NUST with the objective to find a way out. There are 5 levels and 5 main items to be collectable required to win the game. This is a choice-based game with each having a lose case and a win case. Confidence meter is also there which increases or decreases depending upon nature of choice made. If the player finishes the game within the choices limit and with confidence, he wins. If any of the mentioned thing breaches its limit. The game is over at the spot.

**1.3 Objectives:**The objective of this game is to find 5 different items across different locations of NUST, required to exit NUST.  
  
The following items are required to finish the game, which the player collects in each level:

* 1. Permission Letter
* 2. NUST ID Card
* 3. Ticket
* 4. Code
* 5. 10 Coins

**1.4 Target Audience:**

This game is for all. People of any age can play this game. However, as this game is based upon NUST, and it presents a story of a usual day at NUST. So, the intended audience is slightly towards NUSTIANS, and the student aiming for NUST.

**2. Introduction:**

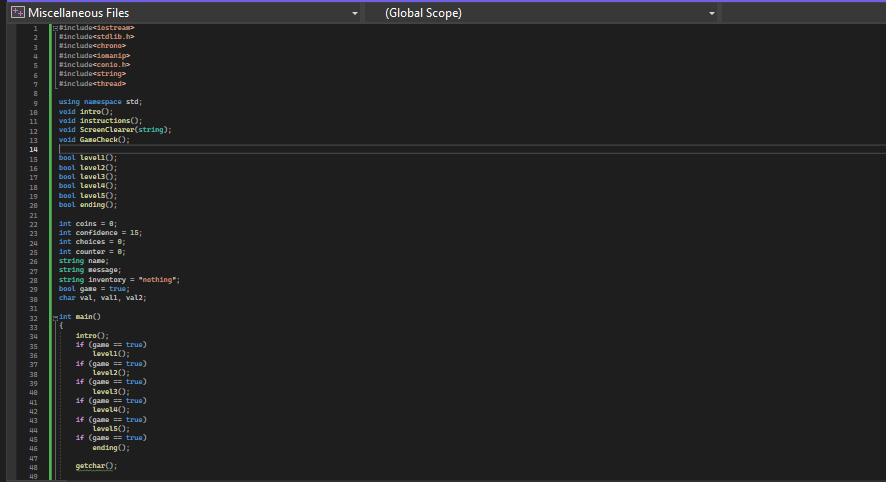
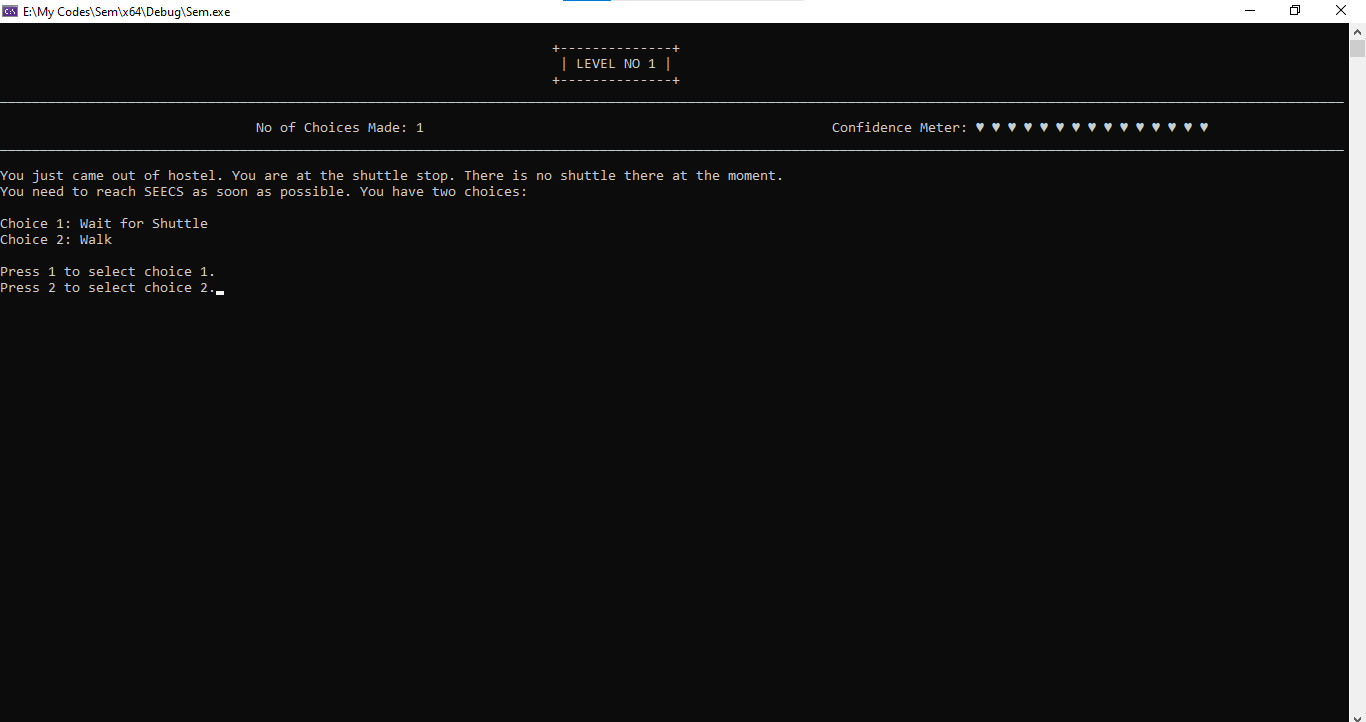
**2.1 Storyline:  
2.1.1 Game Concept:**

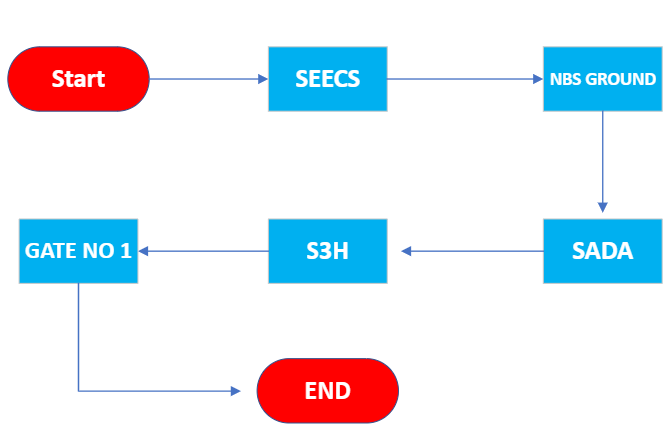
* The game is a text-based game in which the player is given multiple choices to complete the objective.
* This game consists of 5 levels.
* This game is set in NUST, and each level is based upon some location of NUST.
* In order to complete the game, the player must have specific items which the player attains after the completion of each level.
* The player is given multiple routes via choices to reach its destination and each route has its own level of complexity.
* There is a time limit in our game, and time is regarded as the number of choices made in our game.
* The Least possible choices to win the game are 25.
* The Most possible choices to win the game are 49.
* While keeping the complexities in mind, the games provides the player a maximum number of 37 choices to win the game.
* There is a confidence meter as well, which will increase or decerase depending upon the choices made by the user.
* The player is given 15 confidence bars as a default, In case the confidence meter runs out, the game ends.

**2.1.2 Game Story  
  
The player opens his eyes at Zakaria Hostel, NUST. He faces multiple challenges and go through making a couple of choices to reach SEECS where he gets a permission Letter issued required to exit. Level 2 covers his adventure from SEECS to NBS Ground where he makes a series of choices to find his NUST ID Card there. Level 3 covers his adventure from NBS Ground to SADA. In SADA, he goes through multiple challenges such as a brain Stromer riddle and much more. He gets a ticket from SADA required to exit. Level 4 includes the challenges the player faces to reach S3H and in S3H as well. This player ends when the player has successfully generated a code required to exit. Level 5 is a Mini-Game Level which consists of 5 Mini-games. Each Mini game will give 1-4 coins. A total of 10 consists of required from this level. At the End all 5 items are checked if player has these within the game restrictions. He wins and successfully exits NUST.   
  
2.2 Gameplay Mechanism**

This game follows a **“Choice-Based Mechanism”** where players make decisions that affect the narrative and outcomes of the game. It is often referred to as a "branching narrative" or **"interactive storytelling."** In this game, the story unfolds based on the choices made by the player, leading to different paths, plot twists, and different stories. The choices made by the player create branches in the narrative, influencing the overall experience and shaping the direction of the game. This mechanism adds an element of player agency and engagement, as the decisions made impact the unfolding story and the ultimate resolution of the game.

Player is mostly given to choose between 2 choices. One being the win case and one being the lose case. Making the lose case choice will lead to making more choices, after making those choices the player will eventually reach to the win case of the main choice.

**3. Design:  
  
3.1 Architecture:**The game requires 7 different header files to run. Along with that 10 user-defined functions have been used. Other than that 11 global variables have been declared which are used throughout different functions in the programs. The main function just calls the functions of each level and the whole program gets executed. **  
  
3.2 User Interface:  
  
The Game Explain the whole scenario and gives 2 choices to the player to make. The whole game can majorly be played by only 2 keys. (1 and 2 ). While in some events the no of choices gets increased to 4 also.  
  
The player is asked to Press 1 to make the first choice and Press 2 to make the second choice.  
  
  
  
  
Sample:  
  
**

**3.3 Flow Chart:  
  
**

**4. Technical Requirements:**

**4.1 Programming Language  
  
This game is programmed in C++. This program includes various C++ features like Chrono and Indentation iomanip etc.  
  
4.2 Libraries and Tools  
  
More than Half a Dozen libraries are used in our game. These are the one’s used:**#include<iostream>

#include<stdlib.h>

#include<chrono>

#include<iomanip>

#include<conio.h>

#include<string>

#include<thread>  
  
**Noticeable libraries are chrono, thread and iomanip. iomanip is used for enhancing the user interface. The thread library is used to create a typing effect. Chrono is used to get a function of time to create randomness. While conio.h is the major one as user’s input is the most important.  
  
4.3 Platform Compatibility**This game can be run in a system having a compatible C++ compiler. But having Visual Studio 2022 is preferred as it is developed on this actual IDE. **5. Code:**

**5.1 Code Organization**

**In the code, first all the functions are declared and then later defined after the main function.**

**All the function definitions are well separated and written after the main function so that program is readable and the functions can be used again and again.**

**As far as the direction of the program is concerned, it is very simple. The main function calls the function for level 1 then so on upto level 5 as long as the required conditions are satisfied.**

**5.2 Key Code Snippets**  
This is the code preset that only takes a valid input. Majorly used in the game:  
  
**Code:**  
  
do

{

val = \_getche(); // Gets the user input

switch (val)

{

case '1':

{

ScreenClearer("LEVEL NO 3"); // Displays UI

message = "";

typewrite(message);

break;

}

case '2':

{

ScreenClearer("LEVEL NO 3");

message = "";

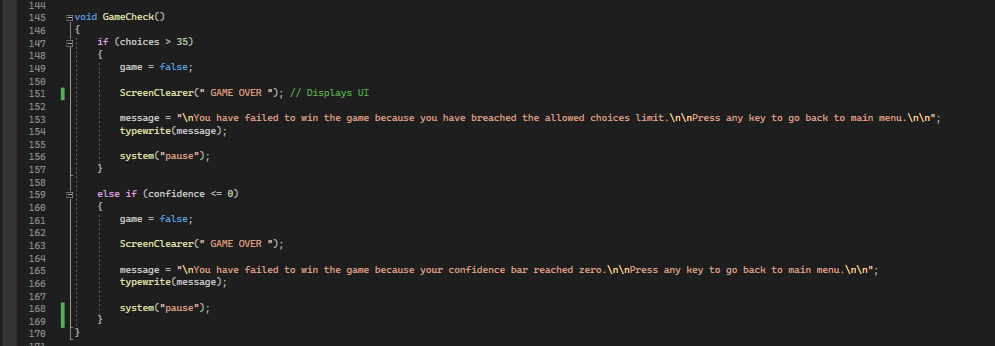
typewrite(message);

break;

} // end of case 2

} // end of switch

} while (val != '1' && val != '2'); // end of do-while

**Game Condition Checker:**  
  
  
  
  
  
**6. Use Cases:  
  
6.1 Player Interactions  
  
The player interacts using keyboard. Player chooses his option by pressing either 1, 2 , 3 or 4. As the game uses switch case. It loads the consequences of that choice after getting the user input. In some cases, the player is asked to enter some words or letters as well. The game handles that input as well and proceeds further.**

**6.2 Edge Cases**The game handles invalid input very well. If the player enters anything apart from the required input. The game simply loads the function back in initial cases. As the game progresses, The game follows the loop mechanism where it keeps asking for the input until player enters the valid input and then proceeds further once player enters a valid input.  
  
**For Example:**   
  
do

{

} while (val != '1' && val != '2'); // end of do-while  
  
  
**7. Challenges Faced:  
  
7.1 Technical Challenges**

* **We faced a lot of Challenges while making this game. Halding invalid input was itself a big challenge. After much thinking, we developed a do-while switch case mechanism which functions perfectly and gives exactly what we wanted to make.**
* **Developing a system that terminates the game when user breaches the limit was also one of the key challenges we faced. We overcome this issue by defining the game state, which simply don’t load the further level and terminates the current level by returning false.**
* **Creating a UI while maintaining a Modular Code was also one of the key challenges faced. This led to creating a separate function for UI that displays all the progress quite efficiently.**

**7.2 Design Challenges**

**Making the Game design was challenging as well. The game storyline and the nature of then choices took a lot of time as we wanted to implement maximum creativity.**

* The game follows a branching mechanism. This led to creation of a nested branched system. This made the code trivial. We tried to make it as simple as it was possible to do by modularity.
* Keeping track of players’ progress was also challenging. We initially had only 1 variable to temporarily store input. But had to increase later on as the code became more nested.

**8. Learning Objectives Achieved:  
  
8.1 Programming Skills  
  
This Game development enhanced our skills a lot. We learned how to implement these programming constructs in making a real project.  
  
8.2 Problem solving:**

* **During development we came across various problems. This project certainly increased our problem-solving skills.**
* **We used to consider Do-while loop less beneficial. But while making this game, its use completely changed my point of view.**
* **This project helped us in understanding handling the invalid input.**
* **This gave us a general view of UI.**
* This enhanced out modularity concepts as well.

**8.3 Collaboration and Communication:**This was a group project and we had great Collaboration and communication regarding division of the game.

* We preferred to have real – life meetings instead of having a virtual call.
* The Game Ideas and Story were decided and made mutually.
* The General overview and game basics were also made in a meeting together which setup the basics of the game starting from scratch.
* Level no 1 was made together.
* Level no 2 and Level no 3 was made by Abdul Hadi Saqib.
* While Level no 4 and Level no 5 was made by Moazam Saif.
* UI and other things were mutually created.
* Report was also mutually created.

Communication Skills are important while making a big project. We had a good understanding with each other which really helped up developing this game.  
  
**9. Conclusion:  
  
9.1 Summary:**In summary, the report outlines the development of the text-based adventure game "El Laberinto De NUST" as a semester project for the Fundamentals of Computer Programming course. Created by Abdul Hadi Saqib and Moazam Saif, the game incorporates a choice-based mechanism with five levels set in NUST, requiring players to collect specific items for a successful exit. The report covers the game's objectives, gameplay mechanism, design aspects, technical requirements, code snippets, and the challenges faced during development. It concludes with reflections on achieved learning objectives, emphasizing enhanced programming skills, problem-solving abilities, and effective collaboration and communication within the team. "El Laberinto De NUST" serves as both a project milestone and a testament to the team's growth in the field of computer science and game development.  
  
**9.2 Future Enhancement:**

* **This Game’s Story can be improved in future.**
* **We can introduce more riddles and a bit complex one too.**
* **A Difficulty system can also be added.**
* **More levels can be added.**
* **UI can be improved.**

**10: Acknowlegments:**

Internet is a huge help. We had a lot of doubts regarding the functioning of different things. We seek help from various site such as GeeksforGeeks , W3Schools and many more.   
  
**11: References:  
  
Geeks for Geeks:** [**https://www.geeksforgeeks.org/c-plus-plus/**](https://www.geeksforgeeks.org/c-plus-plus/) **W3Schools:** [**https://www.w3schools.com/cpp/default.asp**](https://www.w3schools.com/cpp/default.asp)