TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING THAPATHALI CAMPUS

A Major Project Report
On
Collection of Games

Submitted By:

Ajay Kumar Pasi (THA081BEI003)

Bishal Upadhayay (THA081BEI010)

Devendra Yadav (THA081BEI011)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

March, 2025

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING THAPATHALI CAMPUS

A Major Project Report

On

Collection of Games

Submitted By:

Ajay Kumar Pasi (THA081BEI003)
Bishal Upadhayay (THA081BEI010)
Devendra Yadav(THA081BEI011)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

In partial fulfillment for the requirement of final project of C-Programming

Under the Supervision of

Er. Dinesh Baniya Kshatri

March, 2025

DECLARATION

We hereby declare that the report of the project entitled **“Collection of Games”** which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, In partial fulfillment for the requirement of
the final project of C-Programming, which contains a list of 5 games which are

1) No. guessing game
2) Rock, Paper, scissor
3) Tic-Tac-Toe
4) Hangman
5) Room escaping game (Riddle based)

These games have been coded completely by the member where a source like youtube is taken. The coders of the programs are

Ajay Kumar Pasi (THA081BEI003)

Bishal Upadhayay (THA081BEI010)

Devendra Yadav (THA081BEI011)

Date: March, 2025

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a major project work entitled "**Collection of Games**" submitted by **Ajay kumar Pasi, Bishal Upadhayay, Devendra Yadav** in partial fulfillment for the final project in C-Programming. The project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled, Good at teamwork and ready to face similar kind of work in C language and hence we recommend the award of partial fulfillment of the final project of the C-Programming.

____

Project Supervisor
Er. Dinesh Baniya Kshatri
Department of Electronics and Computer Engineering, Thapathali Campus

____

External Examiner

____

Project Coordinator
Er. Umesh Kanta Ghimire
Department of Electronics and Computer Engineering, Thapathali Campus

____

Head of Department
Er. Kiran Chandra Dahal
Department of Electronics and Computer Engineering, Thapathali Campus

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards the Institute of Engineering, Thapathali Campus for the inclusion of the final project in C-Programming. We are also thankful to our Supervisor Er. Dinesh Baniya Kshatri and the Department of Electronics and Computer Engineering, Thapathali Campus for providing us with the resources and support which is needed for this project.

Ajay Kumar Pasi (THA081BEI003)

Bishal Upadhayay (THA081BEI010)

Devendra Yadav (THA081BEI011)

ABSTRACT

This project involves the collection of five fun, exciting, challenging and educational game (“No guessing game”, “Rock, Paper, Scissor”, “Tic-Tac-Toe”, “Hangman”, “Room Escaping Game (Riddle Based)”). The games are menu driven i.e., the game can be selected and a random game can played automatically by the program using rand() function. After completion of game the user provide Y/y if he/she wants to play again and need to provide N/n to terminate the program.

Contents

1. INTRODUCTION

The **Multi-Game Program** is an interactive console-based application designed to provide users with a variety of games to enjoy. Upon launching the program, users are greeted with an option to either play a random game or select from a list of five different games: **Number Guessing**, **Rock-Paper-Scissors**, **Tic-Tac-Toe**, **Hangman**, and a **Room Escape Game** involving riddles.

The program offers a fun and engaging experience, where users can challenge themselves with different game types:

- **Number Guessing** allows players to guess a randomly chosen number within a range.
- **Rock-Paper-Scissors** is a classic game where the player competes against the computer.
- **Tic-Tac-Toe** is a two-player game where players try to get three marks in a row.
- **Hangman** challenges players to guess a word by suggesting letters within a limited number of attempts.
- The **Room Escape Game** provides a series of riddles, where the player must answer correctly to "escape."

With its flexible game selection and random choice option, this program ensures a diverse and enjoyable experience, allowing users to test their skills and have fun with various challenges.

Additionally, the program is designed to offer a seamless user experience by prompting the player to continue or exit the game based on their preferences, making it a complete and enjoyable gaming package.

1.1 Motivation

The motivation behind the **Multi-Game Program** is to provide users with a fun, engaging way to enjoy classic games like **Number Guessing**, **Tic-Tac-Toe**, **Rock-Paper-Scissors**, and **Hangman**. These games not only entertain but also help improve important skills such as **problem-solving**, **strategic thinking**, and **vocabulary**. The variety of games ensures that users experience different types of challenges, appealing to diverse interests. The program also includes a **random game option**, adding excitement and keeping the gaming experience fresh. It is designed to be accessible and enjoyable for users of all ages. For the developer, this project provides an opportunity to practice and enhance programming skills such as **user input handling** and **randomization**. It also encourages the creation of an **interactive and user-friendly experience**. The program can easily be expanded with more games and features in the future, making it versatile. Additionally, it allows the developer to refine their understanding of **game logic** and **program design**. Ultimately, the **Multi-Game Program** combines fun and skill development, making it a rewarding project for both users and developers.

1.2 Project Objectives

The specific objectives of this project are listed below**:**

- To create a **multi-game program** that provides entertainment and challenges for users.
- To demonstrate **proficiency in C programming** and problem-solving.
- To explore **game development concepts** such as randomization, user input handling, and game logic.
- To incorporate **educational elements** (e.g., riddles in the Room Escape Game).
- To showcase **teamwork and creativity** in designing and implementing the games.

1.4 Project Applications

There are many applications "**Collection of Game**" program. Some of the applications of the project are listed below.

- **Educational Tool**:
    - Improves cognitive skills like logical thinking, vocabulary, and problem-solving.
    - Can be used in schools to engage students in learning through interactive games.
- **Entertainment and Leisure**:
    - Provides variety for individuals or groups to play during breaks or leisure time.
- **Programming Practice**:
    - A valuable project for developers to improve coding skills, including randomization, user input handling, and game logic.
- **Excitement and Variety**:
    - The random game feature adds excitement, making the program appealing to game enthusiasts.

1.5 Scope of Project

This project can be used for the kids and adult for entertainment. After adding the graphics to each game would bring charm to the game. The game like Hangman and Room escaping game can be used in the schools to entertain by playing these brain-teasing games. This would also encourage the student to code.

1.6 Report Organization

This report is divided into 5 parts. The five parts are the 5 games of the project. Each parts contains the introduction of the game, gameplay of the game, technical details, challenges of the game and finally the objectives of the game. In the objective part the objective of the game is provided. In gameplay game the way of playing game is provided, in technical details the important details of the game for execution is provided and finally in challenges the challenges that need to be faced while coding and execution is provided.

2) CONCEPTS USED IN THE PROJECT:

I. Functions**:**

- The program uses functions like `numberGuessing()`, `rockPaperScissors()`, `ticTacToe()`, `hangman()`, etc., to modularize the code, making it more organized and reusable.

II. Arrays**:**

- Arrays are used to store multiple values, such as `wordList[]` for the words in the Hangman game and `board[][]` for the Tic-Tac-Toe grid.

III. Structures**:**

- A `struct` is used for riddles in the Room Escape game, where the `Riddle` structure stores `question` and `answer` pairs.

IV. Randomization**:**

- The `rand()` function is used to generate random numbers, such as for random game selection and generating a random word for the Hangman game.

V. Loops**:**

- Loops like `for` and `do-while` are used to repeat game actions like taking turns, making guesses, and checking the conditions until the game ends.

VI. Conditionals**:**

- Conditional statements like `if`, `else`, and `switch` are used to handle different scenarios, such as checking if the user has guessed the number in the Number Guessing game or comparing choices in Rock-Paper-Scissors.

VII. Input/Output Handling**:**

- The program uses `scanf()` for user input and `printf()` for displaying information to the user, making the interaction with the user dynamic and responsive.

VIII. String Manipulation**:**

- Functions like `strcpy()`, `strchr()`, `strncat()`, and `strcmp()` are used to manipulate strings for comparing answers and managing user guesses in the Hangman and Room Escape games.

IX. File Handling**:**

a. The `fopen()` and `fclose()` functions are used to read riddles from a file (`riddle.txt`) in the Room Escape game, demonstrating file handling.

X. Error Handling**:**

a. The program checks for invalid inputs (e.g., out-of-range choices in games) and handles errors like failing to open the riddle file.

XI. Case Sensitivity**:**

a. In the Room Escape game, user input and correct answers are converted to lowercase for case-insensitive comparison using `to lower().`

XII. Game Logic**:**

a. Each game has its specific logic implemented, like checking for winning conditions (Tic-Tac-Toe), counting guesses (Number Guessing), or checking answers (Hangman and Room Escape).

XIII. Random Shuffling**:**

a. The `rand()` function is used to shuffle the riddles for added variety in the Room Escape game.

3) OVERVIEW:

i. **Number Guessing Game:** A simple game where the player guesses a randomly generated number.

ii. **Rock-Paper-Scissors:** A classic two-player game against the computer.

iii. **Tic-Tac-Toe:** A two-player game where players take turns marking a 3x3 grid.

iv. **Hangman:** A word-guessing game where the player has limited attempts to guess the word.

v. **Room Escape Game:** A riddle-based game where the player must solve riddles to "escape" the room.

4) NUMBER GUESSING GAME:

□ Introduction:

The Number Guessing Game is a straightforward game where the player has to guess a randomly generated number between 1 and 100. The game gives feedback after each guess, indicating if the number is too high or too low to help the player make more informed guesses in the future.

• Objective:

The objective is to guess the randomly chosen number with as few attempts as possible, encouraging users to sharpen their logical thinking and estimation skills.

• Gameplay:

- The computer randomly selects a number between 1 and 100.
- The player enters their guess.
- The game informs the player if the guess is too high, too low, or correct.
- The player continues guessing until the correct number is found.

• Technical Details:

- Uses `rand()` to generate a random number.
- Input is received via `scanf()`, which is used to take the player's guess.
- A loop (`do-while`) repeats the guessing process until the player guesses the correct number.

• Challenges:

- Managing input validation to ensure guesses are within the valid range and preventing invalid entries.
- Ensuring the game can handle multiple rounds or resets properly without crashing or malfunctioning.

ROCK, PAPER , SCISSOR

- **Introduction:**
  Rock-Paper-Scissors is a popular hand game played between two players. In this version, the game is played against the computer, and players need to select one of the three options: rock, paper, or scissors. The goal is to choose the option that defeats the computer's choice based on established rules.

- **Objective:**
  The objective is to pick the shape that beats the computer's randomly chosen shape:
  - Rock beats Scissors.
  - Scissors beat Paper
  - Paper beats Rock.

Gameplay**:**

- The player selects one of the three options (Rock, Paper, or Scissors).
- The computer randomly picks one of the three choices.
- The winner is determined by the rules, and the game will display the result.

Technical Details**:**

- The player's selection is handled using `scanf()`.
- The computer's choice is randomly generated using `rand()`.
- The game outcome is determined using simple conditional checks (`if`, `else`) to compare the player's and computer's selections.

Challenges**:**

- Validating user input to ensure only one of the three options (Rock, Paper, or Scissors) is selected.

Providing clear, understandable feedback to the player regarding the outcome of the game.

TIC-TAC-TOE:

• Introduction:

Tic-Tac-Toe is a classic two-player game where players take turns marking spaces on a 3x3 grid. The objective is to form a line of three marks (either X or O) either horizontally, vertically, or diagonally.

- **Objective:**

The objective is to place three marks in a row, column, or diagonal before the opponent does, creating a line of three identical marks to win the game.

- **Gameplay:**
    - The game is played on a 3x3 grid.
    - Players alternate placing their marks (either X or O) on the grid.
    - The game ends when one player wins by completing a line of three marks, or the grid is filled with no winner, resulting in a draw.

- **Technical Details:**
    - The game grid is represented using a 2D array.
    - The program checks for a winner by scanning all rows, columns, and diagonals.
    - Input is received using `scanf()`, and the game's progress is displayed after every move.

- **Challenges:**
    - Ensuring that players cannot choose a position that is already filled.
    - Efficiently checking for a winner and ending the game when a player wins or the grid is full.

HANGMAN GAME:

- **Introduction:**

Hangman is a word-guessing game where the player attempts to guess a hidden word by suggesting letters. Incorrect guesses result in parts of a figure being drawn, and the player loses if the figure is fully drawn before the word is guessed.

- **Objective:**

The goal is to guess the word by suggesting letters without exceeding the allowed number of incorrect guesses.

- **Gameplay:**
    - The program selects a word from a list.
    - The player guesses letters one at a time.
    - Correct guesses reveal letters in the word, while incorrect guesses reduce the number of allowed attempts.
    - The game ends when the player either guesses the word correctly or runs out of attempts.

- **Technical Details:**
    - The word list is stored in an array, and a random word is selected for each game.

  - Correct guesses are replaced with the actual letters in the word.
  - The game tracks incorrect guesses and the remaining attempts.

- **Challenges:**

  - Handling invalid input and ensuring that the player's guesses are case-insensitive.
  - Managing the word display and updating the screen after each guess.

ROOM ESCAPING GAME(RIDDLE BASED):

- **Introduction:**

The Room Escape Game is a riddle-based game where players need to answer a series of riddles correctly to escape a virtual room. Each game session randomly presents different riddles from a stored list, and the player must solve them to progress.

- **Objective:**

The goal is to correctly solve enough riddles to "escape the room" and complete the game.

- **Gameplay:**

  - The program randomly selects a set of riddles for the player to solve.
  - The player is prompted to answer each riddle.
  - The player needs to answer at least two riddles correctly to escape the room.

- **Technical Details:**

  - The riddles are stored in a file (`riddle.txt`) and read into a structure.
  - The riddles are shuffled to provide variety in each game.
  - The program compares answers by converting both the player’s input and the correct answer to lowercase for case-insensitive matching.

- **Challenges:**

  - Ensuring the riddle file is properly read and errors like missing files are handled.
  - Managing the randomization of riddles and ensuring they are presented in a fun, engaging order.
  - Handling case-insensitive answers and providing meaningful feedback after each guess.

TECHNOLOGIES AND TOOL USED:

Programming language: C-programming

Library/Function used:

- **`stdio.h` (Standard Input/Output Library)**
    - Used for input and output operations, such as `printf()` for displaying messages and `scanf()` for taking user input.
    - Essential for handling user interaction in all games.
- **`stdlib.h` (Standard Library)**
    - Provides functions like `rand()` and `srand()`, which are used for generating random numbers in games like Number Guessing and Rock-Paper-Scissors.
    - Also used for memory management (`malloc()`, `free()`) if dynamic memory allocation is needed.
- **`time.h` (Time Library)**
    - Used for seeding the random number generator with `srand(time(NULL))` to ensure different random numbers are generated in each session.
- **`ctype.h` (Character Type Library)**
    - Contains functions like `tolower()` to handle case-insensitive comparisons, which is useful in games like Hangman and the Room Escape Game.
- **`string.h` (String Handling Library)**
    - Provides functions like `strcmp()` for comparing strings, `strlen()` for checking word lengths, and `strcpy()` for copying strings.
    - Essential for handling words in Hangman and the Room Escape Game.

CHALLENGES AND SOLUTIONS:

1. Handling User Input Errors

- **Challenge:** Users may enter invalid inputs, such as letters instead of numbers or out-of-range choices.
- **Solution:** Input validation using loops and conditional checks to prompt users until valid input is received.

2. Ensuring Fair Randomness in Games

- **Challenge:** Random number generation in games like Number Guessing and Rock-Paper-Scissors may feel repetitive.
- **Solution:** Using `srand(time(NULL))` to seed the random number generator for better unpredictability.

3. Managing Game State and Flow

- **Challenge:** Keeping track of turns, scores, and progress in games like Tic-Tac-Toe and Hangman.
- **Solution:** Using arrays, loops, and functions to manage game state efficiently.

4. Handling String Operations in Hangman

- **Challenge:** Comparing guessed letters with the target word and updating the displayed word dynamically.
- **Solution:** Using `strlen()`, `strcmp()`, and loops to process and update the word correctly.

5. Maintaining Code Readability and Modularity

- **Challenge:** A multi-game program can become complex and difficult to manage.
- **Solution:** Implementing functions for each game, structuring the code properly, and adding comments for clarity.

6. Preventing Infinite Loops and Crashes

- **Challenge:** Errors in loop conditions or input handling could cause the program to freeze or crash.
- **Solution:** Proper loop termination conditions, boundary checks, and error handling mechanisms

CONCLUSION:

The Multi-Game Program successfully integrates multiple classic games into a single interactive platform, offering entertainment, learning, and skill development. It enhances problem-solving, strategic thinking, and user engagement while demonstrating fundamental programming concepts such as loops, conditionals, functions, and randomness. Despite challenges like input validation, game state management, and ensuring fairness, effective solutions were implemented to improve functionality and user experience. The project serves as a great learning opportunity for programming and game development, allowing future enhancements like graphical UI integration or multiplayer features. Overall, the program is a fun, efficient, and educational tool for users of all ages.

REFRENCES

1) Let Us C by Yashavant Kantkar.
2) https://www.youtube.com/results?search_query=code+with+harry