



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS**

Proposal

On

Development of Hangman Game in C Programming

Submitted by:

Arpita Dhakal (THA080BEI008)

Bindu Pandeya (THA080BEI016)

Submitted to:

Department of Electronics and Computer Engineering
Thapathali Campus

Kathmandu, Nepal

March, 2024

ABSTRACT

The Hangman project aims to make a fun word guessing game that's easy to play on a computer screen using the C programming language. This proposal gives an overview of why we want to make this game, what we aim to accomplish, and how we plan to do it. It also talks about the research we've done so far, describes how the game will be set up on the computer, outlines the steps we'll follow to finish the project, talks about different ways the game could be used, estimates how long it will take to finish, and looks at whether it's possible to do.

Table of Contents

ABSTRACT.....	i
Table of Contents	ii
List of Figures.....	iii
List of Abbreviations	iv
1.INTRODUCTION.....	1
1.1 Background Introduction.....	1
1.2 Motivation	1
1.3 Problem Definition	1
1.4 Objectives	1
2.LITERATURE REVIEW	2
2.1 History	2
2.2 How is it done?.....	2
2.3 Applications	2
2.4 Drawbacks and Limitations.....	3
3. PROPOSED SYSTEM ARCHITECTURE.....	3
3.1 Flowchart Of Simple Gameplay.....	4
3.2 System Architecture with block diagram	5
3.3 Data flow diagram	6
3.4 Tools and environment.....	6
4. METHODOLOGY	7
4.1 Requirements Gathering.....	8
4.2 Design.....	8
4.3 Implementation.....	8
4.4 Testing	8

4.5 Deployment	8
5. SCOPE AND APPLICATIONS	8
4. TIME ESTIMATION.....	9
4. FEASIBILITY ANALYSIS.....	9
References	10

List of Figures

Fig: 3.1.1 Simple Gameplay Flow.....	4
Fig: 3.1.2 System Architecture.....	5
Fig: 3.2 Data Flow Diagram.....	6
Fig: 4.1 Methodology.....	7

List of Abbreviations

UI: User Interface

I/O: Input/output

GUI: Graphical User Interface

CPU: Central Processing Unit

RAM: Random Access Memory

OS: Operating System

IDE: Integrated Development Environment

CLI: Command Line Interface

API: Application Programming Interface

UI Module: Responsible for handling user interactions, displaying game information, and receiving input from the player.

Game Logic Module: Manages the game state, including word selection, guessing, and tracking player progress.

Data Management Module: Handles data storage and retrieval, including word lists, player scores, and game settings.

Console Output Module: Displays game graphics, including the hangman figure, current word progress, and game status messages.

1. INTRODUCTION

1.1 Background Introduction

The Hangman game is a classic word guessing game where players attempt to guess a hidden word by guessing individual letters. The game is commonly played as a pencil and paper game, but this project aims to implement a digital version using C programming.

1.2 Motivation

The Hangman game project combines programming skills with game development. By creating a Hangman game in C, we can learn programming concepts while enjoying the gameplay. The project also applies program development methodologies learned in class. It will also incorporate some complex functionalities like file handling, data sorting and so on. In short, we find this hangman project slightly challenging and intriguing for us beginners to begin exploring the fields of software development which makes us more motivated for its completion.

1.3 Problem Definition

The proposal aims to address the absence of accessible word guessing games for computer users offline by introducing a Hangman game developed in the C programming language. Recognizing the educational and entertainment potential of such games, particularly in improving vocabulary and spelling skills, the project targets a diverse audience, including students and casual gamers. With the efficiency and portability of C, the game will offer a lightweight yet engaging experience, running seamlessly in a console window. Ultimately, the proposal seeks to fill a niche for a simple yet enjoyable word game on the computer, delivering both entertainment and educational value to its users.

1.4 Objectives

The main objectives of the Hangman project include:

- Developing a console-based Hangman game using C programming.
- Displaying a main menu that contains game options.
- Creating leaderboard of the high scores of the players.
- Enabling users to choose their topic of interest and providing a custom word from that topic to begin with.
- Implementing features such as limited attempts, guesses, and displaying the hangman figure.

2. LITERATURE REVIEW

2.1 History

Existing research in the field of console-based gaming has explored the development of various interactive games, including Hangman. The Hangman game is a classic word guessing game where players attempt to uncover a hidden word by guessing individual letters. It has been implemented in different programming languages and platforms, aiming to provide an entertaining and engaging experience for players.

2.2 How it is done?

The implementation of the Hangman game typically involves creating a console-based interface where players can interact with the game through textual input and output. The game logic revolves around selecting a random word from a predefined list, displaying placeholders for each letter of the word, and allowing players to guess letters one at a time. The game updates the display to reveal correctly guessed letters and draws the hangman figure as incorrect guesses accumulate. The game continues until the player guesses the word correctly or exhausts the maximum number of attempts.

2.3 Applications

The Hangman game holds significance as a recreational activity that promotes vocabulary development, cognitive skills, and logical thinking. It provides players

with an opportunity to challenge their word recognition abilities and strategic guessing tactics. Additionally, developing a Hangman game in C programming enhances participants' programming skills, reinforces algorithmic thinking, and introduces them to game development principles.

2.4 Drawbacks and Limitations

While the Hangman game offers educational and entertainment value, it may have some limitations and drawbacks:

- **Lack of complexity:** The Hangman game's simplicity may lead to limited gameplay depth and variation, potentially resulting in repetitive gameplay experiences.
- **Minimal replay value:** Once players become familiar with the game's mechanics and word list, they may lose interest due to a lack of new challenges or content updates.
- **Absence of social features:** Traditional implementations of the Hangman game may lack social features such as multiplayer modes or online leaderboards, limiting opportunities for player interaction and competition.

3. PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture for the Hangman project is designed to align with the project objectives and facilitate the development of a console-based word guessing game using the C programming language. The architecture emphasizes modularity, scalability, and efficiency to ensure the successful implementation of key functionalities.

3.1 Flowchart of simple gameplay

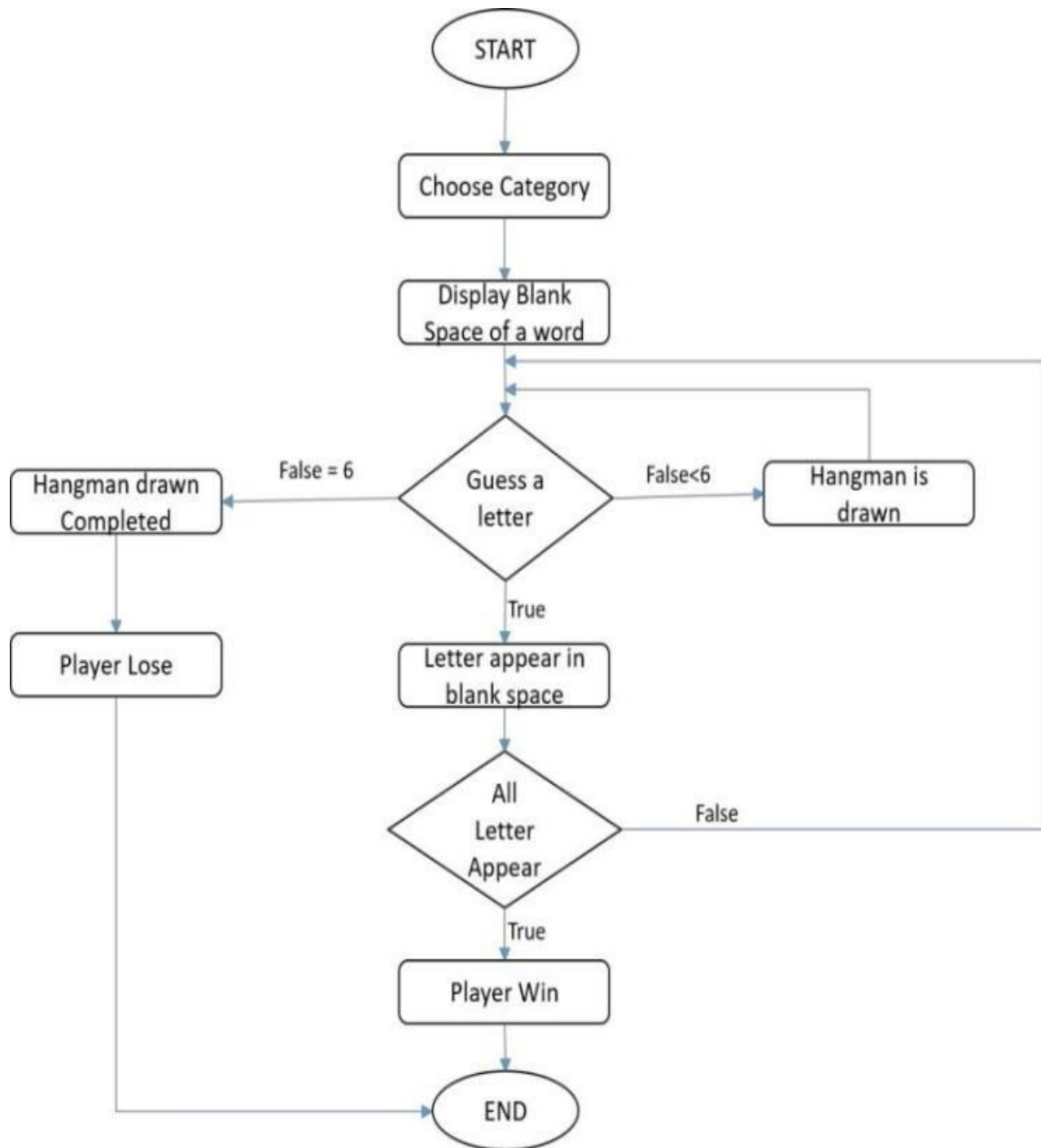


Fig: 3.1.1 Simple Gameplay Flow

3.2 System architecture with block diagram

The system architecture of the Hangman project is composed of the following modules:

1. **Main Menu Module:** Responsible for displaying the main menu options to the player, including starting a new game, viewing the leaderboard, and exiting the game.
2. **Game Engine Module:** Manages the core game logic, including word selection, guessing, hangman figure display, and game state management.
3. **Leaderboard Module:** Handles the storage and retrieval of high scores achieved by players, allowing for the display of leaderboard information.
4. **Input Handling Module:** Processes user input from the console, validates input characters, and communicates with the game engine module.
5. **Output Display Module:** Renders game graphics and textual information to the console, including the hangman figure, current word progress, and game status messages.

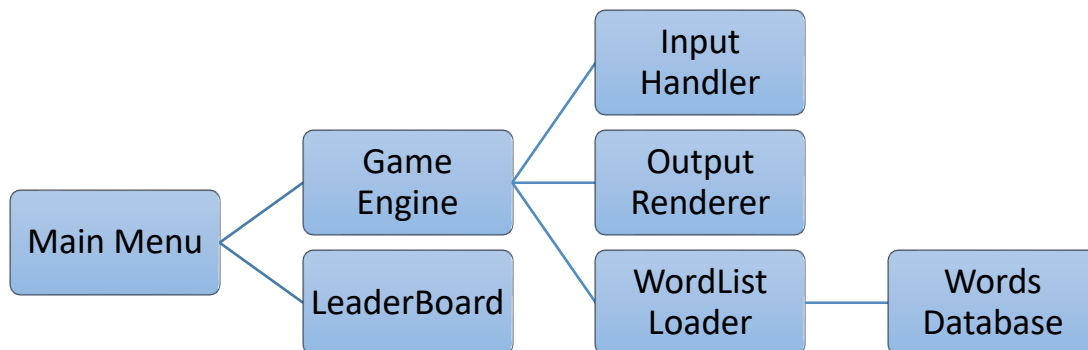


Fig 3.1.2 System Architecture

3.3 Data Flow Diagram

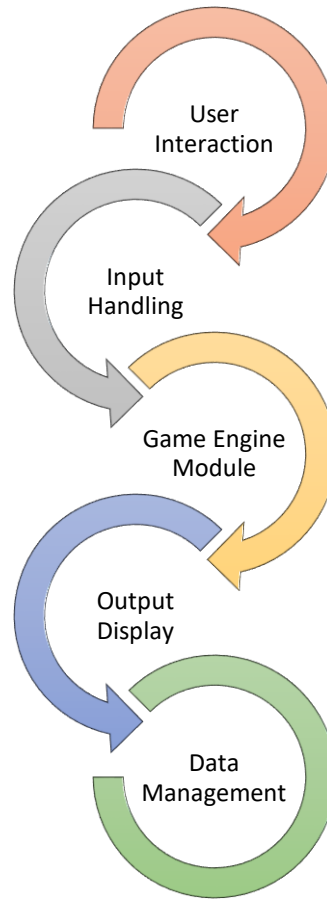


Fig 3.2 Data Flow Diagram

3.4 Tools and Environment

The Hangman project will utilize the following tools and environment for development:

- **Operating System:** Microsoft Windows 10
- **Text Editor/IDE:** Visual Studio Code (VSCode)
 - Visual Studio Code provides a feature-rich environment for writing and editing C code, with support for syntax highlighting, IntelliSense, debugging, and Git integration.

- **Compiler:** GCC (GNU Compiler Collection)
 - GCC is a powerful compiler collection that supports the compilation of C code on Windows. It offers robust optimization capabilities and generates efficient machine code for optimal performance.
- **Version Control System:** GitHub
 - GitHub serves as the repository hosting platform for the Hangman project, enabling version control, collaboration, code review, and issue tracking.
- **Command Line Interface (CLI):** Windows Terminal
 - Windows Terminal provides a modern and efficient command-line interface for executing and interacting with the Hangman game. It offers customizable features such as multiple tabs, panes, and profiles, enhancing the development experience.

4. METHODOLOGY

The methodology for the Hangman project will involve iterative development, following the stages of:

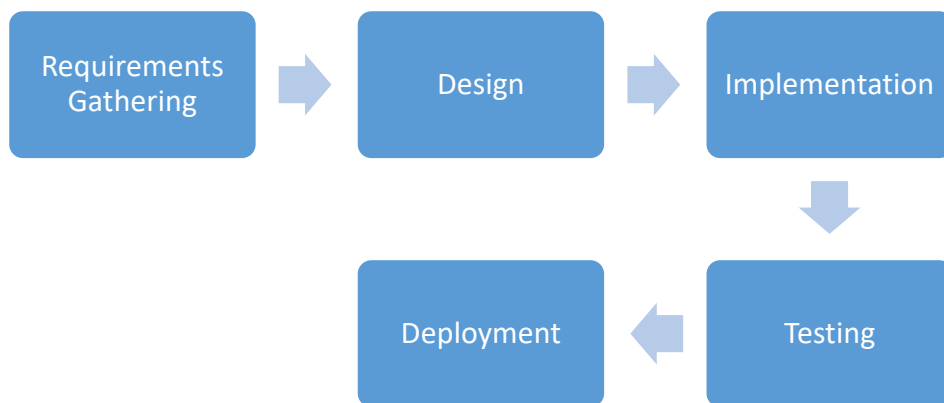


Fig 4.1 Methodology

4.1 Requirements Gathering

Gathering requirements for the Hangman game, including defining gameplay mechanics, user interface elements, and technical specifications.

4.2 Design

Designing the Hangman game architecture, user interface layout, and game mechanics. This stage will involve creating mockups, flowcharts, and data structures to support the game's functionality.

4.3 Implementation

Implementing the Hangman game logic, user interface elements, and supporting functions in the C programming language. Participants will write code to handle word selection, guessing, hangman figure display, and game state management.

4.4 Testing

Testing the Hangman game for functionality, usability, and performance. This stage will involve unit testing, integration testing, and user acceptance testing to ensure the game meets the defined requirements and performs as expected.

4.5 Deployment

Deploying the Hangman game for use by players. This may involve uploading the game online or sharing it with peers for feedback and testing.

5. SCOPE AND APPLICATIONS

The scope of the Hangman project includes developing a functional console-based Hangman game in C programming. The application of the project is primarily educational and can be used to:

- Demonstrate programming concepts and principles.

- Provide hands-on experience in game development.
- Enhance problem-solving skills and logical thinking.

The Hangman game can also serve as a fun and engaging activity for individuals interested in programming and game development.

6. TIME ESTIMATION

A rough estimation of the time required for the project including requirements gathering, design, implementation, testing and deployment is expected to be 3 weeks.

7. FEASIBILITY ANALYSIS

A feasibility analysis will be conducted to assess the technical, operational, and economic feasibility of the Hangman project. This analysis will consider factors such as:

- Technical capabilities: Assessing whether the project can be implemented with the available resources and technology.
- Operational feasibility: Evaluating the practicality and usability of the Hangman game for target users.

The feasibility analysis will help determine the viability of the project and identify any potential challenges or constraints.

References

Block diagram of hang man

https://www.researchgate.net/figure/Flowchart-of-Hangman-Game-IV-CASE-STUDY-HANGAROO-Below-Fig-2-3-4-and-5-are-the_fig1_351865346