



TRIBHUWAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS

Proposal

On

Collect game on C

Submitted by:

[Dipika Joshi] [THA081BCT009]

[Shreya Maharjan] [THA081BCT042]

[Sristi Rani] [THA081BCT044]

Submitted to:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

February, 2025

ABSTRACT

This proposal outlines the design and development of a "Collect Game" implemented in the C programming language. The primary goal is to create an interactive, console-based game where the player controls a player to collect objects. This project aims to enhance understanding of basic programming concepts such as input handling, real-time interaction, game loops, and collision detection. The game utilizes a simple terminal interface, providing an educational platform for beginners to learn essential C programming principles, including memory management and logic structuring. The proposal discusses the methodology of the game's creation, including its architecture, game design, and feasibility analysis, which confirms the project's technical, economical, and operational viability. Through this project, the developers seek to gain practical experience in system-level programming, providing a solid foundation for future, more complex programming endeavors.

Table of Contents

ABSTRACT.....	i
Table of Contents.....	ii
List of abbreviations.....	iv
List of figures.....	v
1. INTRODUCTION.....	1
1.1 Background Introduction.....	1
1.2 Motivation.....	2
1.3 Problem Definition.....	2
1.4 Objectives.....	2
2. LITERATURE REVIEW.....	4
2.1 Focusing on Educational Benefits.....	4
2.2 Emphasizing Challenges and Improvements.....	5
2.3 Performance Optimization Perspective.....	5
2.4 Highlighting Broader Applications.....	6
2.5 C as a Language Choice.....	6
3. PROPOSED SYSTEM ARCHITECTURE.....	7
3.1 Block Diagram of System Architecture.....	7
3.2 Data Flow Diagram.....	7
4. METHODOLOGY.....	8
4.1 Game concept and design.....	8
4.2 Developing environment.....	8
4.3 Implementing approach.....	8
4.3.1 Structure of the game.....	8

4.3.2	Input handling.....	8
4.3.3	Game logic and loop.....	8
5.	SCOPE AND APPLICATIONS.....	9
6.	TIME ESTIMATION.....	9
7.	FEASIBILITY ANALYSIS	10
	References.....	11

List of Abbreviations

OS - Operating System

VS - Visual Studio

AI - Artificial Intelligence

SDL - Simple Direct MediaLayer

OpenGL - Open Graphics Library

List of Figures

Figure 1: System Architecture.....	7
Figure 2: Data Flow Diagram.....	7
Figure 3: Gnatt Chart.....	9

1. INTRODUCTION

1.1 Background Introduction

A programming language is a formal set of instructions that allows human to interact and communicate with computers to perform specific tasks. These languages enable developers to write software and systems that control hardware or process data. Programming languages are used in various areas including web development, game development, artificial intelligence, system programming and database management. Programming languages play a crucial role in software and technological development. The choice of language depends on factors like performance, application domain and ease of use.

1.1.1 Types of Programming Languages

Programming languages can be categorized based on their functionality, abstraction level, and execution style. Below are the main types:

i. Low-Level Language

Programming is done in this language by using binary code.

a. Machine Language (Binary Code)

They are machine dependent language and programming is done by using the combination of 0s and 1s. It is hard and tedious for programmers using it and is hardware dependent but the execution is fast in machine language.

b. Assembly Language

In this language, programming is done by using binary code and mnemonics. It is a little easier for programmers to use assembly language. It requires assembler to convert code into machine language .

This language is a little easier for programmers to use compare to machine language.

ii. High-Level Language

High level language is a user friendly and machine independent type of language. They are similar to Natural language. High level language must be translated by interpreter or compiler to understand by the computer. For example: C, C++, Java etc.

The C programming language is a general-purpose and procedural language that supports structured programming and provides low-level access to the system memory.

It is a powerful and widely-used programming language that has played a crucial role in the development of modern computing. It was originally developed in the early 1970s and has since become the foundation for many other programming languages. Its efficiency, flexibility, and portability have made it a preferred choice for system programming, software development, and embedded systems

It was created by Dennis Ritchie at Bell Labs in 1972 as an improvement over the B programming language. It was designed to develop the UNIX operating system, which remains one of the most influential operating systems today. C remains a fundamental programming language due to its performance, control, and widespread use. Understanding C is crucial for anyone interested in computer science, software development, or system programming.

1.2 Motivation

While still being puzzled about which type of project should we do, a new idea emerged in simple game development. Researching a little more about the game on YouTube and other AI tools, we decided to move forward with this one. We ultimately thought it might be a good mini project for us to start with. Though it is a little simple game, we believe this project will be a big learning experience in our early phase of programming which will further build our confidence level for upcoming new projects in later year.

1.3 Problem Definition

The main objective of this project is to design and code a collecting game using the C programming language. The player controls a character and must collect randomly appearing targets by moving the character to the same location. The game ends after a certain number of targets are caught or a certain amount of time elapses or when the moves are over.

1.4 Objectives

Our aim is to create a collecting game utilizing the C programming language, where players can control a character and must collect randomly appearing targets by moving the character to the same location.

Throughout this project, we intend to accomplish the following goals:

- Effectively design and implement a collecting game using the C programming language, operational within a terminal setting
- Gain experience in structuring a simple game using C.

The collecting game in C provides a strong foundation for game programming by integrating logic, real time input handling, graphics and performance optimization. It is an excellent beginner to intermediate project that enhances both technical and problem solving skills.

2. LITERATURE REVIEW

As a practical application development tool, C has a very strong foundation of being capable of performing practical tasks efficiently. Being a programmer in C, it is really important how we carry out practical problems using C. It not only it builds your capacity to handle and understand practical issues, it also helps enhance the innovative designs and applications.

The collect game in C programming is a great project for beginners looking to understand game mechanics, user input, and basic graphics rendering. While the simplicity of the game makes it suitable for learning, the techniques and concepts used can serve as the foundation for more complex game development projects. They teach how to handle user input, create game rules and display the game. These games can be simple as collecting ball or more complex like two players, obstacles, etc. When building one, programmers often use libraries like SDL (for better graphics) or ncurses (for text-based games).

In teaching learning process in C, we get to understand a lot of things. The gaming environment helps us to interact with the game properly where we can challenge as well as practice for our mental betterment.

2.1 Focusing on Educational Benefits

The development of the Collect game in C serves as an excellent foundation for students and beginners in programming. Through this project, learners gain hands-on experience with fundamental programming concepts such as loops, conditionals, and real-time input handling. It provides an opportunity to understand how a game loop operates and how to implement object movement. Additionally, working on such a project encourages problem-solving and logical thinking, as students must design efficient ways to detect object interactions and ensure smooth gameplay. Despite C's lack of built-in graphical support, this project helps in understanding low-level programming concepts such as memory management, pointer operations, and direct hardware interactions. Completing the game successfully gives students the confidence to explore more complex game development techniques and prepares them for learning higher-level programming languages with advanced game development frameworks. Ultimately, this project demonstrates that even a simple game can provide valuable insights into computer science principles and software development.

2.2 Emphasizing Challenges and Improvements

Although this game appears to be a simple project, implementing it in C presents several challenges that require creative solutions. One of the main difficulties is managing smooth graphics rendering as C does not provide built-in graphical support, requiring external libraries such as SDL or OpenGL. Additionally, handling real-time user input efficiently without causing delays or glitches can be difficult, especially in text-based implementations. Collision detection and boundary handling must be carefully designed to ensure an accurate and fair gaming experience. Moreover, maintaining an appropriate difficulty curve, where the game gradually becomes more challenging, requires thoughtful algorithm design. These challenges highlight the importance of structured programming and efficient resource management. Future improvements to the game could include power-ups, multiple difficulty levels, and multiplayer functionality, making it more engaging. By incorporating better graphical libraries and optimizing algorithms, the game could be transformed into a more professional and visually appealing experience.

2.3 Performance Optimization Perspective

Performance optimization is a crucial aspect of game development, and the Collect game in C offers an opportunity to explore how efficient coding practices impact execution speed and responsiveness. Since the game relies on a continuous loop to update the screen and process user input, optimizing the loop to minimize unnecessary computations is essential. One of the key performance bottlenecks in simple games is collision detection; optimizing it using efficient algorithms can significantly improve gameplay smoothness. Additionally, controlling frame rates is necessary to ensure that object movements appear natural and consistent across different hardware configurations. Another area of optimization is input handling, where event-driven programming techniques can prevent unnecessary delays caused by polling methods. Implementing multithreading for handling different tasks, such as rendering and input processing, could further enhance performance, though it requires careful synchronization. Overall, this project serves as an excellent example on how small-scale games can help developers understand the importance of efficiency and resource management in programming.

2.4 Highlighting Broader Applications

While this game is primarily a beginner-level programming project, its underlying logic has numerous real-world applications beyond entertainment. The core mechanics of tracking an object's movement and ensuring it is "caught" at the right moment can be applied to robotics, automation, and even AI. For instance, object-tracking systems used in robotics rely on similar principles of detecting an object's position and responding to its movement in real time. The concepts explored in this project can also be adapted to industrial automation, such as conveyor belt systems where robotic arms must accurately pick up objects. Additionally, the logic of detecting collision and movement can be extended to AI-based vision systems, where cameras and sensors track and interact with moving objects. By incorporating advanced techniques such as machine learning or sensor integration, this simple game could be the foundation for more sophisticated technological applications. Therefore, beyond being a fun project, this game illustrates how fundamental programming concepts can have practical applications in various industries.

2.5 C as a Language Choice

The choice of C as the programming language for developing the Collect game highlights both the strengths and limitations of C in game development. On one hand, C provides direct control over memory management, making it an excellent language for performance-critical applications. It allows developers to write efficient, lightweight code that runs with minimal overhead, which is especially important for embedded systems and low-level programming. Learning to develop a game in C teaches valuable programming concepts, such as handling input/output operations, working with loops, and managing system resources efficiently. While C may not be the first choice for modern game development, mastering it provides a solid foundation for understanding the core principles of software engineering and performance optimization.

3.PROPOSED SYSTEM ARCHITECTURE

3.1 Block Diagram of System Architecture

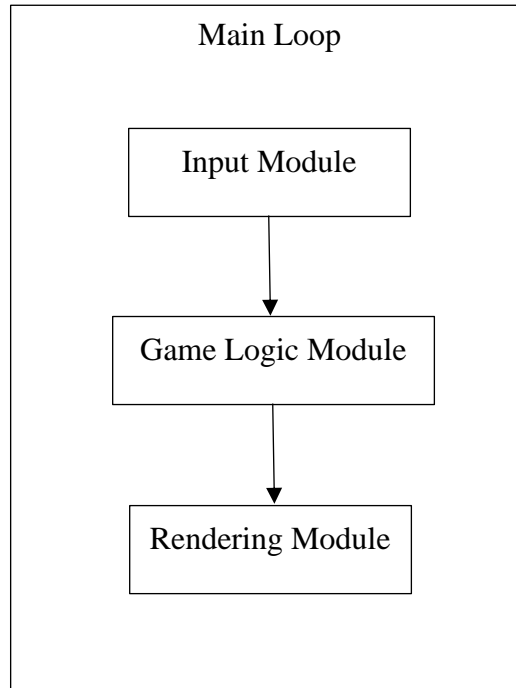


Figure 1: System Architecture

The input module takes input from the keyboard and updates the player's position left and right according to the pressed key. Game logic module manages game states and rules; object appearing, player moving, etc and the rendering module displays the final output i.e. it shows the score. This loop continues till the player presses a reserved key for the end of the game.

3.2 Data Flow Diagram

The following diagram visualizes the flow of data through the system and hence represents how the code works in a graphical manner.

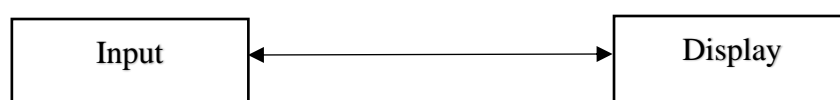


Figure 2: Data Flow Diagram

4.METHODOLOGY

The development of the Collect Game in C follows a structured approach to ensure smooth gameplay, efficient coding, and an engaging user experience. The game involves a player-controlled character that moves left and right to collect object. The methodology includes the game design process, implementation techniques, and testing strategies.

4.1 Game concept and design

The game development process began with brainstorming ideas and defining the core mechanics, objectives, and rules. A simple yet engaging gameplay loop can be designed to ensure user interaction and enjoyment.

4.2 Developing environment

We use C programming for the development of this game, using different libraries like `stdio.h`, `conio.h`, etc. text editors like Visual Studio Code, code blocks can be used and can be compiled using GCC.

4.3 Implementing approach

4.3.1 Structure of the game

We have to initialize the variables, game screen and define the objects. The game screen can be enclosed in a rectangular wall, the object and the player can be defined as different variables.

4.3.2 Input handling

The user's input can be carried out by the WASD keys which move top, right, down and bottom respectively. To update the position of the player, we can use the `getch()` function to read the keyboard input.

4.3.3 Game logic and loop

The objects are at random positions. As the player moves to collect the object, the score will be updated as the player touches the object. The game continues till the player presses a button to stop the game.

5. SCOPE AND APPLICATIONS

This project focuses on learning fundamentals of the C programming language. Rather than focusing on advanced features, it provides an excellent platform for learning C syntax, file handling and event driven programming.

It enhances our fundamental skills before stepping into the vast world of game development. It serves as a beginner friendly project for both learning and real-world experiences.

6. TIME ESTIMATION

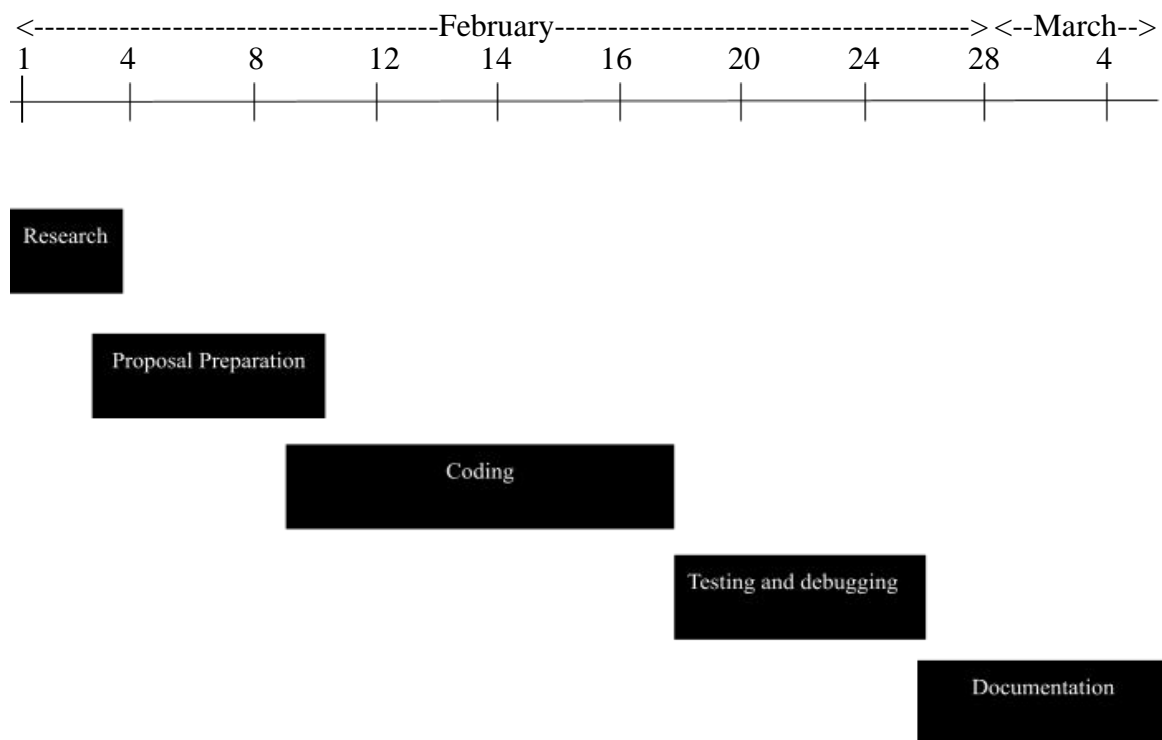


Figure 3: Gantt Chart

7. FEASIBILITY ANALYSIS

The main objective of feasibility study is to establish a software that is acceptable to the users, adaptable to change, and comfortable to establish standards. The feasibility study of the Collect Game in C evaluates its practicality in terms of technical implementation, economic viability, and overall effectiveness as a simple yet engaging game.

7.1 Technical feasibility

This game was developed in C, which is a highly efficient programming language and supported by various OS. Since it is a console-based game, it does not demand high processing power, making it technically feasible for a wide range of devices, including older computers.

7.2 Economical feasibility

The game's development is cost-effective since it does not require expensive software or hardware and open-source tools such as GCC, Code::Blocks, and VS Code can be used for development. The lightweight nature of the game ensures that it runs efficiently on most computers without requiring advanced hardware or high processing power, and since it is a simple project, time was the primary investment rather than the financial resources.

7.3 Operational Feasibility

This game is designed to be simple, and easy to play which ensures smooth operation of this game. It is easy to use, runs smoothly and compatible; works on different devices. Since the game is easy to play, maintain, and function as expected, it is operationally feasible.

Based on the feasibility analysis the Collect Game is both technically and economically feasible. It uses C programming efficiently, requires minimal financial investment, and provides an engaging user experience. While it currently operates as a simple console-based game, its scalability and adaptability make it a promising foundation for future improvements.

References

- [1] Chang, WC., Chou, YM. (2008). Introductory C Programming Language Learning with Game-Based Digital Learning. In: Li, F., Zhao, J., Shih, T.K., Lau, R., Li, Q., McLeod, D. (eds) Advances in Web Based Learning - ICWL 2008. ICWL 2008
- [2] Stephan G. Kochan, “Programming in C”, TDM
- [3] <https://www.simplilearn.com>
- [4] <https://www.tutorialspoint.com>
- [5] Geeks for geeks. Available: <https://www.geeksforgeeks.org/types-of-feasibility-study-in-software-project-development/>. 2024