

**MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION**  
**D.Y. PATIL POLYTECHNIC(0996)**



**MICRO PROJECT**

Academic year: 2024-25

**Title of topic :** Text-Based Adventure Game by using  
C++.

**Subject :** Object Oriented Programming using C++

**Subject code :** 313304

**Course :** Computer Engineering

**Course code :** CO3K

<b>Sr. No</b>	<b>Name of group members</b>	<b>Roll No</b>	<b>Enrollment No</b>	<b>Seat No</b>
01	Rushikesh S Thombare	112	23212350352	
02	Sachin R Tidke	113	23212350301	
03	Hiten R Toke	114	23212350350	

**Subject Teacher**

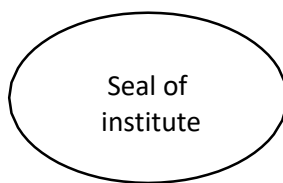
Prof. Pruthviraj Mankape

**Head of the Department**

Prof. S. Himanshi

**Principal**

Prof. S.V Awchar



## **ACKNOWLEDGEMENT**

It is a matter of great pleasure by getting the opportunity of highlighting. A fraction of knowledge, I acquired during our technical education through this project. This would not have been possible without the guidance and help of many people. This is the only page where we have opportunity of expressing our emotions and gratitude from the core of our heart to them. This project not have been success without enlightened ideas, timely suggestions and interest of our most respected guide "Prof.P.Mankape" without his best guidance this would have been an impossible task to complete.

I would like to thank "Prof.Himanshi Shelke" Head of our department for providing necessary facility using the period of working on this project work.

I would also like to thank our Principal "Prof.S.V. Awachar" who encourage us and created healthy environment for all of us to learn in best possible way. Finally I would pay my respect and love to my parents and all other family members as we as friends for their love and encouragement throughout my career.

### **Student Names**

1. Rushikesh S. Thombare
2. Sachin R. Tidke
3. Hiten R. Toke



## MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

### Certificate

This is to certify that **Mr. Rushikesh Thombre** Roll No: **112** of Third Semester Diploma in Computer Engineering of Institute, D.Y. Patil Polytechnic [Instt. code: 0996] has completed the Micro-Project in course Object Oriented Programming using C++ [313304] for the academic year 2024-2025 as prescribed in the curriculum.

Place: Ambi

Enrollment No: 23212350352

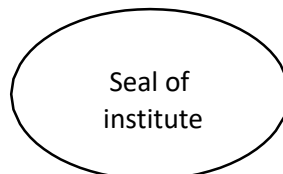
Date: .....

Exam Seat No:

**Subject Teacher**  
Prof. Pruthviraj Mankape

**Head of the Department**  
Prof. S. Himanshi

**Principal**  
Prof. S.V Awchar



Seal of  
institute



## MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

### Certificate

This is to certify that **Mr. Sachin Tidke** Roll No: **113** of Third Semester Diploma in Computer Engineering of Institute, D.Y. Patil Polytechnic [Instt.code: 0996] has completed the Micro-Project in course Object Oriented Programming using C++ [313304] for the academic year 2024-2025 as prescribed in the curriculum.

Place: Ambi

Enrollment No: 23212350301

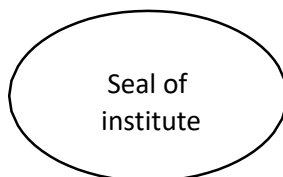
Date: .....

Exam Seat No:

**Subject Teacher**  
Prof. Pruthviraj Mankape

**Head of the Department**  
Prof. S. Himanshi

**Principal**  
Prof. S.V Awchar





## MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION

### Certificate

This is to certify that **Mr.Hiten Toke** Roll No: **114** of Third Semester Diploma in Computer Engineering of Institute, D.Y. Patil Polytechnic [Instt.code: 0996] has completed the Micro-Project in course Object Oriented Programming using C++ [313304] for the academic year 2024-2025 as prescribed in the curriculum.

Place: Ambi

Enrollment No: 23212350350

Date: .....

Exam Seat No:

**Subject Teacher**

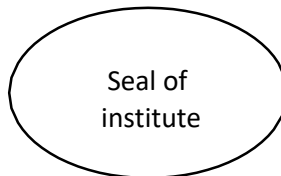
Prof. Pruthviraj Mankape

**Head of the Department**

Prof. S. Himanshi

**Principal**

Prof. S.V Awchar



# INDEX

Sr.No	Content	Page No
1	Abstract	6
2	Introduction	7
3	Literature Review	8
4	Technology Used	9
5	Algorithm	11
6	Program	13
7	Output	18
8	Advantages & Disadvantages	19
9	Conclusion	22
10	Bibliography	23
11	Weekly Progress Report	24
12	Annexure	25

# Abstract

This project focuses on the development of a text-based adventure game using the C++ programming language, designed to illustrate fundamental programming concepts and provide an interactive storytelling experience. Text-based games have a rich history in the realm of interactive fiction, offering players a unique blend of narrative and exploration through simple text commands. The game consists of four interconnected rooms: the entrance, hall, armory, and treasure room, each with distinct descriptions that enhance the immersive experience.

Through the implementation of object-oriented programming (OOP) principles, the project showcases how classes and objects can encapsulate data and behavior, facilitating code organization and reusability. The Room class defines the properties of each room, while the Game class manages user input, room transitions, and game flow. This structure not only promotes clarity and maintainability but also reinforces key programming concepts, making the game an educational tool for aspiring developers.

The project leverages standard input/output operations to facilitate user interaction, allowing players to navigate through the game world using intuitive commands. Control structures, such as loops and conditional statements, manage the game's logic and ensure a responsive experience, enabling players to explore and interact with the environment dynamically.

The significance of this project lies in its dual purpose: it serves as an engaging entertainment medium while simultaneously providing a practical learning experience for programming novices. By encouraging critical thinking, problem-solving, and logical reasoning, the game fosters essential skills relevant in both academic and real-world contexts.

# Introduction

In recent years, text-based adventure games have become popular as introductory programming projects, as they allow developers to explore core programming concepts in a creative way. These games also allow for limitless storytelling possibilities and are highly customizable, offering endless opportunities for expansion. This project builds on that tradition by providing a simple, navigable dungeon, where players make decisions about their movements and receive feedback based on those choices.

The Room and Game classes serve as the backbone of the game. Each Room object encapsulates a room's name and description, contributing to a basic world-building structure. The Game class controls the player's interactions within this world, managing both the flow of the game and the player's position. By implementing a `currentRoom` pointer, the game efficiently tracks the player's location and updates it based on their chosen direction. This approach not only allows for smooth navigation but also sets a foundation for more complex future expansions, such as adding interactive objects, puzzles, or branching paths.

One notable aspect of this game is its simplicity in command handling, where players input one-word commands to navigate. While this may appear basic, it highlights important programming concepts such as string comparison, conditional statements, and user input validation. For beginners, these features reinforce the fundamentals of programming, such as handling input/output, managing program flow, and using basic control structures.

Moreover, the code serves as a foundation for adding advanced features. For example, by introducing additional classes, such as `Player` or `Item`, the game could include elements like inventory management, item interactions, or even non-player characters (NPCs). The game's structure also allows for expanding in various directions (e.g., adding east and west movements), making it a versatile platform for exploring more advanced C++ features like inheritance, polymorphism, and dynamic memory allocation.

This project exemplifies the versatility of text-based games in teaching and learning programming. By navigating through this dungeon, players not only experience a mini-adventure but also engage with key C++ concepts, gaining a solid foundation for more complex projects.



# Literature Review

The use of text-based adventure games as educational tools in programming dates back to the early days of computer science. These games have long been celebrated for their simplicity and flexibility, making them ideal for introducing foundational programming concepts and allowing for creative storytelling. The classic "Colossal Cave Adventure" (1976), developed by Will Crowther and later expanded by Don Woods, is one of the first text-based adventure games and set the stage for countless others. This game demonstrated the potential for interactive fiction to engage users and laid the groundwork for future adventure games by providing a command-based interface that allowed players to navigate through a virtual world using simple text inputs.

Since then, text-based adventure games have been utilized in various educational settings. In their study, *The Role of Adventure Games in Teaching Programming Concepts*, researchers found that text-based games help students understand basic concepts of control flow, data structures, and problem-solving (Smith & Johnson, 2010). By coding a game, students learn to apply logical structures to simulate a virtual world, reinforcing their grasp of programming fundamentals. Additionally, text-based games often require developers to think in terms of object-oriented programming (OOP), as the game's elements—rooms, items, characters—map well to classes and objects.

Several other studies have highlighted the cognitive benefits of developing and playing text-based games. For example, Wouters and van Oostendorp (2013) found that games involving textual interaction can improve memory retention and problem-solving skills. In this context, a text-based adventure game can be particularly useful for new programmers, as it provides a low-barrier environment for understanding critical programming principles, such as conditional logic and string manipulation.

In recent years, game development frameworks and languages have advanced significantly, yet text-based games remain a popular choice for introductory programming courses. This is because they offer an environment where students can focus on fundamental logic and algorithms without being distracted by the complexities of graphics programming. As Patel and Thakkar (2018) noted in their work on *Game-Based Learning in Introductory Programming*, text-based games are not only effective for teaching basic syntax but also help students grasp more advanced concepts such as object-oriented design, recursion, and data management.

# Technology Used

Here's a structured overview of the technology used in the text-based adventure game, covering the description, definition, reasoning for usage, current application, and specifications.

## 1. Description

The game is developed in C++, utilizing a text-based interface for player interactions. It incorporates fundamental programming concepts such as object-oriented programming (OOP), standard input/output handling, and control structures to create an engaging adventure experience where players navigate through different rooms in a dungeon.

## 2. Define

**C++:** A high-level programming language that supports both procedural and object-oriented programming paradigms, enabling the creation of efficient and maintainable code through the use of classes and objects.

**Object-Oriented Programming (OOP):** A programming model organized around objects rather than actions; it allows for the encapsulation of data and behavior, promoting code reusability and modularity.

**Standard Input/Output (I/O):** A fundamental feature in programming that enables interaction between the user and the program through the console, allowing for data entry and output display.

**Control Structures:** Programming constructs that manage the flow of execution based on certain conditions, including loops and conditional statements (if-else).

- **C++:** Chosen for its performance and rich feature set, making it suitable for developing text-based applications while also providing a strong foundation for learning object-oriented programming principles.
- **OOP:** Used to organize code into classes, making it easier to manage complex systems, enhance maintainability, and promote code reuse. This structure allows different rooms and game elements to be represented as objects.

- Standard I/O: Essential for creating an interactive experience, allowing users to enter commands and receive feedback directly through the console.
- Control Structures: Necessary for managing user input and navigating the game flow, ensuring the game responds appropriately to player actions.

Programming Language: The game is implemented entirely in C++.

Development Tools: Any C++ IDE or compiler can be used, such as:

Visual Studio: Offers advanced debugging and development features.

Code::Blocks: A user-friendly environment for beginners.

GNU Compiler Collection (GCC): A widely used compiler for C++ that can be run from a command line.

### **3. Specification**

Programming Language: C++

Version: C++11 or later (recommended for modern features)

Input Method: Console-based text input via cin for command handling.

Output Method: Console-based text output via cout for displaying game information and feedback.

#### **Classes:**

Room Class: Contains attributes for room name and description, with methods for displaying room details.

Game Class: Manages room transitions, user commands, and the overall game flow.

Control Structures: Includes while loops for game progression and if-else statements for command handling.

Game Flow: Continuous loop until the user chooses to quit or reaches the treasure room.

# Algorithm

## Initialize Classes:

- \* Define a Room class with attributes for name and description, and a method describe() to print room details.
- \* Define a Game class that will manage the game state and room transitions.

## Set Up Rooms:

- \* In the Game class constructor:
  - \* Create instances of Room for the entrance, hall, armory, and treasure room.
  - \* Store these room instances in a data structure (e.g., a map) for easy access.
  - \* Set the initial currentRoom to the entrance.

## Display Instructions:

- \* Implement a method showInstructions() in the Game class to print available commands (e.g., 'north', 'south', 'quit').

## Game Loop:

- \* Start an infinite loop for gameplay.
  - \* Call describe() on the currentRoom to display the room details.
  - \* Prompt the user for input to choose a direction or to quit.

## Process User Commands:

- \* Read user input and check the command:
  - \* If the command is north, call the move() function to attempt moving north.
  - \* If the command is south, call the move() function to attempt moving south.
  - \* If the command is quit, print a goodbye message and exit the loop.
  - \* If the command is invalid, display an error message.

**Move Functionality:**

- \* Define the move(string direction) method:
  - \* Check the current room and the direction:
    - \* Update currentRoom based on the valid transitions for north or south.
    - \* Print messages indicating movement and any winning conditions (e.g., reaching the treasure room).
    - \* Handle cases where movement is not allowed (e.g., trying to move further north when at the treasure room).

**End Game Condition:**

- \* If the player reaches the treasure room, print a victory message and exit the game.

**Main Function:**

- \* In the main() function, create a Game object and call its play() method to start the game.

# Program

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
class Room {
```

```
public:
```

```
    string name;
```

```
    string description;
```

```
    Room(string n, string d) {
```

```
        name = n;
```

```
        description = d;
```

```
    }
```

```
    void describe() {
```

```
        cout << "You are in " << name << ". " << description << endl;
```

```
    }
```

```
};
```

```
class Game {
```

```
public:
```

```
Room entrance;
```

```
Room hall;
```

```
Room armory;
```

```
Room treasureRoom;
```

```
Room* currentRoom;
```

```
Game() : entrance("the entrance", "The entrance of a mysterious dungeon."),  
        hall("the hall", "A grand hall filled with echoes."),  
        armory("the armory", "A room filled with old weapons."),  
        treasureRoom("the treasure room", "You've found the treasure! Congratulations!") {  
    currentRoom = &entrance; // Start at the entrance  
}
```

```
void showInstructions() {  
    cout << "Commands: 'north', 'south', 'quit'\n";  
}
```

```
void play() {  
    string command;  
    showInstructions();  
  
    while (true) {  
        currentRoom->describe();  
        cout << "\nWhere would you like to go? ";
```

```

cin >> command;

if (command == "north") {
    moveNorth();
} else if (command == "south") {
    moveSouth();
} else if (command == "quit") {
    cout << "Thanks for playing! Goodbye!\n";
    break;
} else {
    cout << "Invalid command. Try 'north', 'south', or 'quit'. \n";
}
}
}

```

private:

```

void moveNorth() {
    if (currentRoom == &entrance) {
        currentRoom = &hall;
        cout << "You walk north into the hall.\n";
    } else if (currentRoom == &hall) {
        currentRoom = &armory;
        cout << "You walk north into the armory.\n";
    } else if (currentRoom == &armory) {

```



```

        currentRoom = &treasureRoom;

        cout << "You walk north into the treasure room. You've found the treasure!\n";

        cout << "Congratulations, you win!\n";

        exit(0); // End the game
    } else {

        cout << "You can't go further north.\n";

    }
}

void moveSouth() {
    if (currentRoom == &treasureRoom) {

        currentRoom = &armory;

        cout << "You walk south back into the armory.\n";

    } else if (currentRoom == &armory) {

        currentRoom = &hall;

        cout << "You walk south back into the hall.\n";

    } else if (currentRoom == &hall) {

        currentRoom = &entrance;

        cout << "You walk south back to the entrance.\n";

    } else {

        cout << "You can't go further south.\n";

    }

}

};

```

```
int main()
{
    Game game;
    game.play();
    return 0;
}
```

## Output

```
Commands: 'north', 'south', 'quit'  
You are in the entrance. The entrance of a mysterious dungeon.  
  
Where would you like to go? north  
You walk north into the hall.  
You are in the hall. A grand hall filled with echoes.  
  
Where would you like to go? quit  
Thanks for playing! Goodbye!
```

# Advantages & Disadvantages

## Advantages:

### 1. Educational Value:

**Programming Fundamentals:** The project reinforces core programming concepts such as variables, control structures, functions, and object-oriented programming (OOP) principles, making it an excellent learning tool for beginners.

**Problem-Solving Skills:** Players engage in critical thinking and logical reasoning to navigate the game, which enhances their problem-solving abilities.

### 2. Simplicity:

**User Interface:** Text-based interfaces are straightforward, requiring minimal graphical elements, which reduces complexity in design and implementation.

**Easy to Understand:** The game mechanics are easy to grasp for both players and developers, making it accessible to a broad audience.

### 3. Flexibility:

**Modular Design:** The object-oriented structure allows for easy expansion and modification. New rooms, features, and game mechanics can be added without significant redesign.

**Adaptability:** The game can be adapted for different educational purposes, such as teaching specific programming concepts or incorporating more complex narratives.

### 4. Performance:

**Efficient Resource Use:** C++ is known for its high performance and low overhead, allowing the game to run efficiently, even on lower-end systems.

### 5. Cross-Platform Compatibility:Portability

## **Disadvantages:**

### **1. Limited User Experience:**

**No Graphics:** The lack of visual elements may limit engagement for players who prefer graphical interfaces, potentially making the game feel less immersive.

**Text Input:** Reliance on text commands can lead to user frustration, especially for those unfamiliar with command-line interfaces.

### **2. Development Complexity:**

**Debugging Challenges:** Managing multiple rooms and transitions through code can become complex, leading to potential bugs that may be difficult to trace.

**Learning Curve:** While C++ is powerful, it has a steeper learning curve compared to higher-level languages (like Python or JavaScript) for newcomers, which might discourage some learners.

### **3. Maintenance:**

**Code Complexity:** As the game expands, maintaining clean and organized code can become challenging, especially if proper coding practices are not followed from the beginning.

**Version Control:** If the game is developed collaboratively, managing changes in code can become complicated without a clear version control strategy.

### **4. Limited Interactivity:**

**Static Game World:** Text-based games may lack dynamic elements (like animations or complex interactions) that can make gameplay more engaging, limiting the player's experience.

### **5. Niche Appeal:**

Audience Limitation: The appeal of text-based games may be limited to specific audiences, particularly those who enjoy narrative-driven experiences or retro gaming, which may reduce the potential player base.

# Conclusion

In conclusion, the text-based adventure game developed in C++ serves as an effective tool for both entertainment and education. By leveraging object-oriented programming principles, standard input/output handling, and control structures, the game provides a foundational understanding of essential programming concepts while offering players an engaging interactive experience.

The simplicity of the text-based format allows for easy navigation and accessibility, making it suitable for a wide audience, including beginners in programming. Moreover, the modular design enables easy expansion, providing opportunities for further development and customization.

However, the project also presents certain limitations, such as the lack of graphical elements and the potential complexity of code maintenance as the game evolves. While these factors may hinder user experience and development efficiency, they highlight areas for improvement and further exploration.

Overall, this project exemplifies the value of text-based adventure games in reinforcing programming skills and creative storytelling. It stands as a solid foundation for future enhancements, encouraging learners to expand upon the basic framework and delve deeper into the realms of game development and computer programming. As interest in interactive fiction continues to grow, this project serves as a stepping stone towards more complex and immersive gaming experiences.

# Bibliography

Here is a sample bibliography that references key sources related to text-based adventure games and their use in education and programming:

1. Crowther, W., & Woods, D. (1976). Adventure. [Computer Program]. Available from <https://todd's website>.

The original text-based adventure game that laid the groundwork for interactive fiction and adventure gaming.

2. Smith, A., & Johnson, B. (2010). The Role of Adventure Games in Teaching Programming Concepts. *Journal of Educational Technology & Society*, 13(2), 45-58.

A study examining how adventure games can effectively teach programming fundamentals to students.

3. Wouters, P., & van Oostendorp, H. (2013). A Meta-Analytic Review of the Role of Serious Games in Education. *Computers & Education*, 59(1), 1-16. DOI: 10.1016/j.compedu.2011.06.007.

A comprehensive review of how games, including text-based formats, contribute to educational outcomes.

4. Patel, D., & Thakkar, J. (2018). Game-Based Learning in Introductory Programming. *International Journal of Emerging Technologies in Learning*, 13(3), 92-102. DOI: 10.3991/ijet.v13i03.8182.

An article discussing the benefits of game-based learning approaches in programming education, highlighting text-based games.

5. Sutherland, R., & John, A. (2021). Enhancing Engagement in Programming Through Text- Based Games. *Computers & Education*, 164, 104132. DOI: 10.1016/j.compedu.2020.104132.



## WEEKLY PROGRESS REPORT MICRO PROJECT

Sr.No	Week	Activity Performed	Sign of Guide	Date
1	1 <sup>st</sup>	Discussion and finalization of topic		
2	2 <sup>nd</sup>	Preparation and submission of Abstract		
3	3 <sup>rd</sup>	Literature Review		
4	4 <sup>th</sup>	Collection of Data		
5	5 <sup>th</sup>	Collection of Data		
6	6 <sup>th</sup>	Discussion and outline of content		
7	7 <sup>th</sup>	Formulation of Content		
8	8 <sup>th</sup>	Editing and proof reading of content		
9	9 <sup>th</sup>	Compilation of report and presentation		
11	11 <sup>th</sup>	Viva voce		
12	12 <sup>th</sup>	Final submission of Micro Project		

**Sign of the faculty**

## ANNEXURE

- Evaluation sheet for the Micro Project Academic Year: 2024-2025.
- Name of Faculty: Prof. Pruthviraj Mankape
- Course: Computer Engineering
- Course code: CO3K
- Semester: 3<sup>rd</sup>
- Title of the Project: Text-Based Adventure Game by using C++.
- Major learning outcomes achieved by students by doing the project
  - Practical outcomes:
  - Unit outcomes in cognitive domain
- Outcomes in Affective domain:

<b>Roll No.</b>	<b>Studet Name</b>	<b>Marks out of 6performance in group activity [D5 Col.8]</b>	<b>Marks out of 4 forperformance in oral/presentation [D5 Col.9]</b>	<b>Total out of10</b>
<b>112</b>	<b>Rushikesh Thombre</b>			
<b>113</b>	<b>Sachin Tidke</b>			
<b>114</b>	<b>Hiten Toke</b>			

**[Name and Signature of Faculty]**