

VIETNAM NATIONAL UNIVERSITY OF HO
CHI MINH CITY THE INTERNATIONAL
UNIVERSITY SCHOOL OF COMPUTER
SCIENCE AND ENGINEERING



FINAL REPORT

BLUE BOY ADVENTURE

GAME - GROUP 6

Course: Algorithms & Data Structures

ABSTRACT

'Blue Boy Adventure' is a 2D action role-playing game (RPG) developed in Java using Swing as the Graphical User Interface (GUI). It offers a visually stunning world map composed of different tiles, each depicting unique terrains and environments. Players can interact with NPCs, objects, monsters, and items, engaging in conversations, battles, puzzle-solving, and exploration. The game features a variety of events that add excitement and unpredictability, such as damage pits, healing pools, teleport tiles, and switches. Players can equip and use items strategically to enhance their abilities and overcome challenges. Items can be obtained through defeating monsters, cutting down trees, or discovering treasures, rewarding thorough exploration. 'Blue Boy Adventure' provides an immersive and rewarding gameplay experience, combining visual aesthetics, interactive elements, and strategic decision-making.

Project's Member

Table 1: The contribution and information of all project members

No	Full Name	ID	Contribution
1	Phùng Quốc Tân	ITITIU21102	20%
2	Nguyễn Minh Đức	ITITIU21045	20%
3	Bùi Đức Mạnh	ITITIU21076	20%
4	Phạm Quang Vinh	ITITIU21347	20%
5	Lê Minh Phụng	ITITIU20281	20%

TABLE OF CONTENT

Project's Member	2
Chapter 1. INTRODUCTION	4
I. Purpose and scope of the report	4
II. Project's objective	4
Chapter 2. GAME RULE AND HOW-TO-PLAY	4
I. Rules of Game	4
II. How-To-Play	6
Chapter 3. GAME ANALYTIC	7
I. Game Feature	7
II. User Interface	9
III. UML Diagram	9
IV. Game Functionality	9
Chapter 4. CONCLUSION	9
I. Summary	9
II. Future Improvement	9
REFERENCE	10

CHAPTER 1. INTRODUCTION

I. Purpose and scope of the report

The purpose of the report for the game is to provide a comprehensive overview of its development, features, and performance. The report aims to document the entire development process, including the design choices, programming techniques, and challenges faced during the creation of the game. It will also outline the scope of the game, describing its intended audience, gameplay mechanics, and overall objectives. Additionally, the report will analyze the game's performance, including bug reports, and areas for improvement. Overall, the report serves as a valuable resource for evaluating the game's success, identifying areas of achievement, and suggesting future enhancements and updates.

II. Project's objective

The main objective of 'Blue Boy Adventure' is to ignite the player's enthusiasm by immersing them in a thrilling and mysterious adventure. As they delve into the game's richly crafted world, they will uncover a captivating backstory that fuels their curiosity and desire to explore further. The game entices players to embark on a journey filled with unknown challenges, hidden treasures, and intriguing characters. Through their exploration and interactions, players will gradually unravel the secrets of the game's narrative, keeping them engaged and eager to discover what lies ahead. The objective is to create a sense of wonder and anticipation, drawing players deeper into the game's world and ensuring an unforgettable and exhilarating gaming experience.

CHAPTER 2. GAME RULE AND HOW-TO-PLAY

I. Rules of Game

1. **Objective:** The main objective of the game is to guide Blue Boy through various levels, overcome challenges, defeat enemies, and rescue his captured friends.
2. **Controls:** You can control Blue Boy using the arrow keys or designated movement keys on your keyboard. Use the up arrow key to jump, and the left and right arrow keys to move horizontally. Pressing the down arrow key may allow Blue Boy to interact with objects or crouch, depending on the game mechanics.
3. **Level Progression:** The game consists of multiple levels, each with its own unique layout, obstacles, enemies, and secrets. Progress through the levels by reaching the end point, usually indicated by a door or gate.
4. **Obstacles and Enemies:** Blue Boy will encounter various obstacles such as platforms, gaps, spikes, and traps. Timing and precise jumps are crucial to navigate through these challenges successfully. Additionally, enemies will try to impede Blue Boy's progress. Avoid or defeat enemies by jumping on their heads or using power-ups, if available.
5. **Power-ups and Collectibles:** Throughout the game, Blue Boy can collect power-ups and special items to enhance his abilities. These may include extra lives, temporary invincibility, or increased speed. Keep an eye out for collectibles such as coins or gems that can boost your score or unlock bonus content.
6. **Health and Lives:** Blue Boy has a health bar or life count, representing his remaining vitality. Colliding with enemies or hazards will deplete his health or cost him a life. Collect health power-ups or find checkpoints to replenish health and save progress.
7. **Checkpoints and Saving:** Checkpoints are placed strategically within levels, allowing you to respawn at that location if Blue Boy loses a life. Use these checkpoints wisely to avoid repeating significant portions of a level.

8. Time and Scoring: Some levels may include time limits or scoring mechanics. Complete levels as efficiently as possible to achieve higher scores or earn time bonuses.
9. Exploration and Secrets: 'Blue Boy Adventure' encourages exploration. Search for hidden passages, secret areas, or bonus levels. Uncover the game's secrets to enhance your gameplay experience.
10. Enjoyment and Persistence: Remember to have fun! Overcoming challenges may require practice and persistence. Learn from your mistakes, adapt your strategies, and enjoy the exciting adventure that 'Blue Boy Adventure' offers.

II. How-To-Play

1. Start the game by launching the application. You will be greeted with the main menu, where you can select options such as starting a new game, loading a saved game, adjusting settings, or exiting the game.
2. Upon starting a new game, you will assume the role of the main character, Blue Boy. Your objective is to navigate through various levels and complete quests to progress in the game.
3. Use the keyboard arrow keys to move Blue Boy in different directions: up, down, left, and right. Explore the intricately designed 2D world, filled with diverse landscapes, obstacles, and hidden treasures.
4. Encounter enemies throughout your journey. Engage in thrilling real-time battles by pressing the designated attack key. Time your attacks strategically to defeat enemies and earn experience points to level up Blue Boy, enhancing his abilities and unlocking new skills.
5. Interact with non-playable characters (NPCs) and engage in conversations to gather information, receive quests, and uncover the game's storyline. Pay attention to dialogue options and choose responses wisely to influence the outcome of your interactions.
6. Solve puzzles and overcome obstacles to progress further in the game. These challenges may require logical thinking, reflexes, or exploration of the environment. Be observant and use your skills to overcome these hurdles.

7. Collect items, power-ups, and currency scattered throughout the game world. These resources will aid you in your journey, providing health restoration, temporary buffs, or unlocking special abilities.
8. Save your progress frequently using the in-game save feature. This ensures that you can resume your adventure from where you left off in case you need to take a break or exit the game.
9. As you advance, you will encounter boss battles and face tougher adversaries. Prepare yourself by strategizing your approach, utilizing your acquired skills, and exploiting enemy weaknesses to emerge victorious.
10. Explore every nook and cranny of the game world, as secrets, hidden areas, and optional side quests await those who are thorough in their exploration.
11. Immerse yourself in the captivating storyline, rich visuals, and atmospheric sound effects that enhance the gaming experience. Enjoy the satisfying progression, character development, and the sense of accomplishment as you conquer challenges and uncover the mysteries within 'Blue Boy Adventure.'

Remember, the journey is yours to undertake, and the choices you make will shape the outcome of the game. So, brace yourself for an exciting adventure filled with action, exploration, and discovery in this Java-based 2D RPG masterpiece.

CHAPTER 3. GAME ANALYTIC

I. Game Feature

1. World Map with Different Images of Tiles:

The world map in 'Blue Boy Adventure' is a visually diverse and expansive landscape composed of different tiles. Each tile represents a specific terrain or environment, such as forests, mountains, deserts, or villages. These tiles are carefully designed with unique graphics, colors, and details, creating a visually engaging and immersive game world. As you explore the map, you will encounter a variety of tiles that contribute to the overall aesthetics and atmosphere of the game.

2. Player Interaction with NPCs, Objects, Monsters, and Items:

In 'Blue Boy Adventure,' the player has the ability to interact with various entities within the game world. This includes non-playable characters (NPCs), objects, monsters, and items. Interacting with NPCs allows you to engage in conversations, receive quests, gather information, or trigger specific events in the game. Objects can be interacted with to solve puzzles, unlock new areas, or discover hidden treasures. Monsters can be engaged in real-time battles, where you can attack, defend, and use skills to defeat them. Items can be picked up, used, or equipped to enhance your abilities, restore health, or provide various advantages in different situations.

3. Different Events:

'Blue Boy Adventure' features a range of dynamic events that can occur throughout the game. These events add excitement, unpredictability, and variety to the gameplay experience. Some examples of events include damage pits that inflict harm on the player upon contact, healing pools that restore health, teleport tiles that transport the player to different locations, or switches that trigger the opening of doors or activation of mechanisms. These events create interactive elements within the game world, requiring the player to adapt and make strategic decisions to progress.

4. Player Equipment and Item Usage:

As you progress in the game, you have the opportunity to acquire and equip various items to enhance your character's abilities. These items

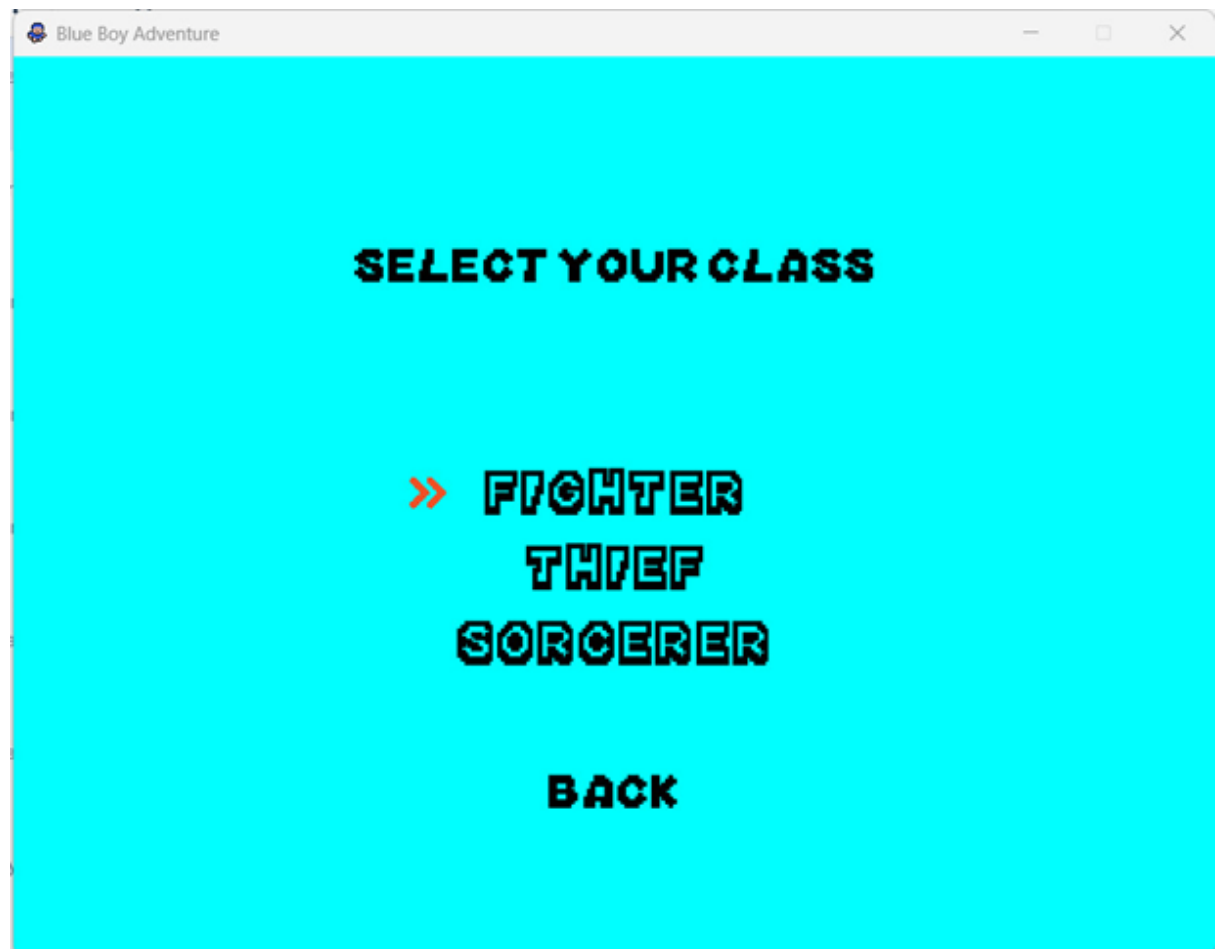
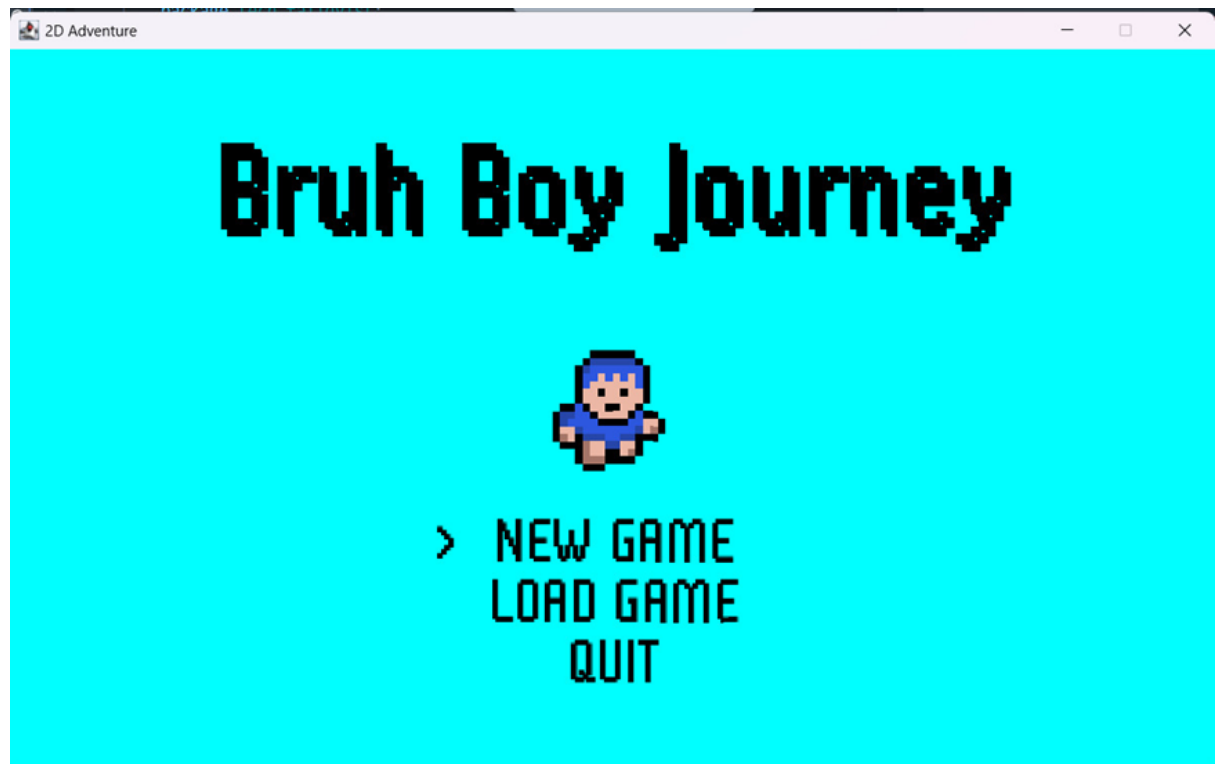
can include weapons, armor, accessories, and consumables. Equipping items provides stat bonuses, improves combat effectiveness, or grants special abilities. Additionally, you can use items in specific situations to overcome challenges or solve puzzles. For example, using a key item to unlock a locked door or utilizing a healing potion to restore health during battles. Managing and utilizing your inventory effectively becomes crucial in maximizing your potential and overcoming obstacles.

5. Item Drops and Treasure Discoveries:

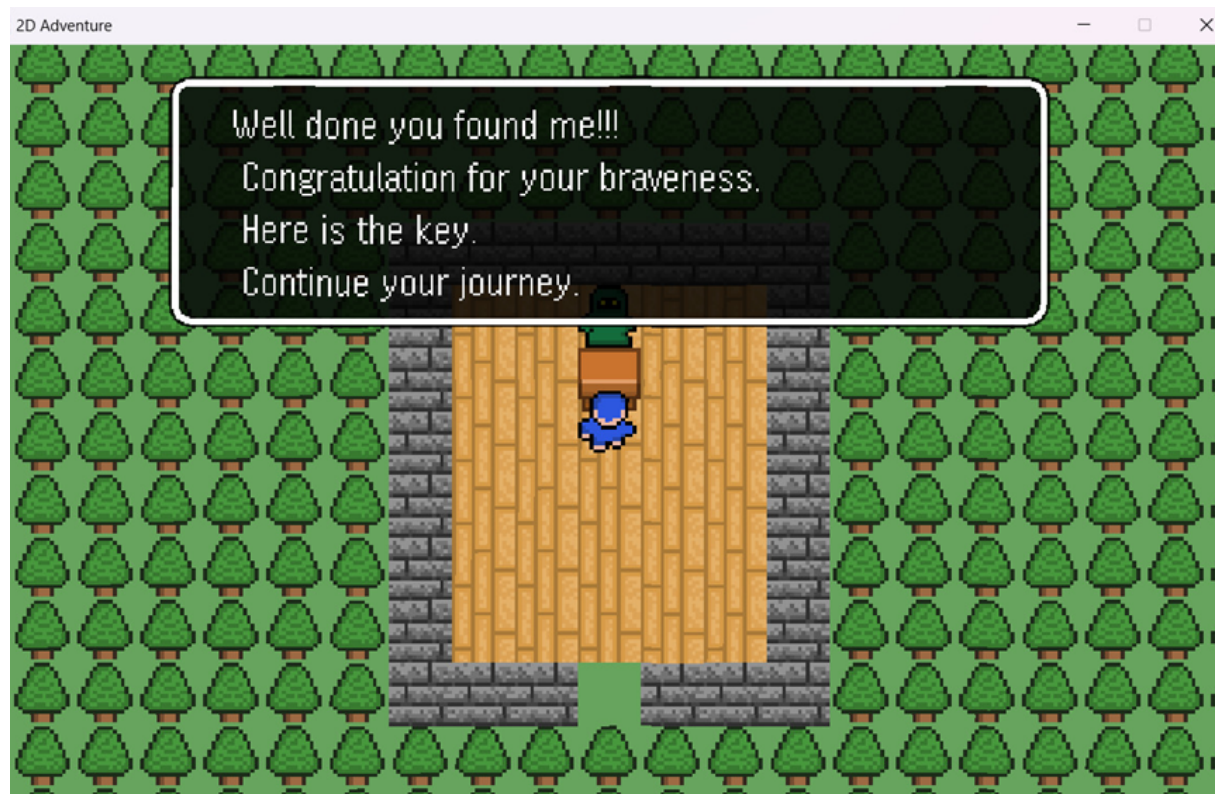
In 'Blue Boy Adventure', items can be obtained through different means such as defeating monsters in battles, cutting down trees, or discovering them in hidden treasures scattered throughout the game world. Defeated monsters may drop valuable loot, including equipment upgrades, crafting materials, or consumables. Cutting down trees or exploring hidden areas can yield items and resources that contribute to your progression. Discovering and collecting these items adds a sense of exploration, achievement, and rewards to your gameplay experience, encouraging thorough exploration of the game world.

These features collectively enhance the depth, interactivity, and immersion of 'Blue Boy Adventure.' By offering a diverse world map, engaging interactions with various entities, dynamic events, equipment customization, and item acquisition, the game provides players with a captivating and rewarding gameplay experience.

II. User Interface








```

classDiagram
    class Entity {
        +BufferedImage_motion
        +BufferedImage_attack
        +BufferedImage_still
        +solidArea
        +attackArea
        +collisionOn
        +invincible
        +COUNTER
        +CHARACTER_ATTRIBUTES
        +ITEM_ATTRIBUTES
        +TYPE
        +constructor()
        +setAction()
        +damageReaction()
        +speak()
        +use()
        +checkDrop()
        +dropItem()
        +update()
        +damagePlayer()
        +dyingAnimation()
        +changeAlpha()
        +setup()
    }
    class NPC_OldMan {
        +solidArea
        +solidAreaDefault
        +setDialogue()
    }
    class AssetSetter {
        +gp
        +AssetSetter(GamePanel)
        +setObject()
        +setNPC()
        +setMonster()
        +setInteractiveTile()
    }
    class GamePanel {
        +Thread
        +FPS
        +originalTileSize
        +maxScreenCol
        +setupGame()
        +startGameThread()
        +update()
        +paintComponent()
        +playMusic()
    }
    class TileManager {
        +mapTileNum
        +tile()
        +setup()
        +loadMap()
        +draw()
    }
    class Tile {
        +BufferedImage
        +collision
    }
    class KeyHandler {
        +shotKeyPressed
        +enterPressed
        +keyPressed()
        +keyReleased()
        +States()
    }
    class UI {
        +messageCounter
        +titleScreenState
        +commandNum
        +draw()
        +drawPlayerLife()
        +drawMessage()
        +drawSubWindow()
        +drawInventory()
        +drawCharacterScreen()
        +drawPauseScreen()
        +drawTitleScreen()
        +addMessage()
    }
    class EventRect {
        +eventRectDefault
        +eventDone
    }
    class EventHandler {
        +canTouchEvent
        +checkEvent()
        +hit()
        +healingPool()
        +damagePit()
        +teleport()
    }
    class CollisionChecker {
        +CollisionChecker(GamePanel)
        +checkTile()
        +checkObject()
        +checkEntity()
        +checkPlayer()
    }

    Entity <|-- NPC_OldMan
    Entity <|-- Player
    Entity "0..*" -- "1" AssetSetter
    AssetSetter "1" -- "0..*" GamePanel
    GamePanel "1" -- "1" TileManager
    GamePanel "1" -- "1" Tile
    GamePanel "1" -- "1" KeyHandler
    GamePanel "1" -- "1" UI
    GamePanel "1" -- "0..1" EventRect
    GamePanel "1" -- "1" CollisionChecker
    TileManager "1" -- "1" Tile
    Tile "1" -- "1..*" KeyHandler
    KeyHandler "1..*" -- "1" UI
    UI "1" -- "1" EventRect
    EventRect "0..1" -- "1" CollisionChecker
    CollisionChecker "1" -- "1" EventHandler
    CollisionChecker "1" -- "1" GamePanel
    
```

The diagram illustrates the architecture of a game engine. It features several core classes: **Entity** (base class for **NPC_OldMan** and **Player**), **AssetSetter**, **GamePanel**, **TileManager**, **Tile**, **KeyHandler**, **UI**, **EventRect**, **EventHandler**, and **CollisionChecker**. The relationships are defined by associations with multiplicity and generalization. **GamePanel** acts as a central hub, managing **TileManager**, **Tile**, **KeyHandler**, **UI**, **EventRect**, and **CollisionChecker**. **AssetSetter** manages **Entity** instances and interacts with **GamePanel**. **TileManager** manages **Tile** objects, which are then rendered by **GamePanel**. **KeyHandler** processes user input and sends commands to **UI**. **UI** manages the game's state and displays information. **EventRect** and **CollisionChecker** handle game events and collisions, respectively. **EventHandler** manages game events and interacts with **CollisionChecker**.

The Tile class represents the game world's tiles, defining the terrain, environment, and boundaries. The Sound class manages audio elements, including background music, sound effects, and voiceovers, enhancing the immersive experience.

The KeyHandler class processes keyboard input, translating player actions into in-game commands. The EventReact class handles event-based reactions triggered by gameplay events, such as quests, achievements, or specific actions.

The EventHandler class manages game events, such as level transitions, cutscenes, and triggers, coordinating the flow and progression of the game. The Player class represents the playable character, encapsulating attributes, abilities, and game-specific functionalities.

Lastly, the AssetSetter class handles asset management, including loading and configuring game assets such as textures, models, sounds, and other resources required for the game.

Together, these components form a comprehensive class diagram for a game, providing a structured representation of the game's architecture and interactions between various elements.

IV. Game Functionality

You wake up in a mysterious forest and notice an abandoned house covered around by old trees. Before starting on your adventure, you seek help and instructions to understand the mechanics of the game. With the knowledge in hand, you confidently start the game, ready to face the challenges ahead. Armed with the ability to move your player using either the WASD keys or the arrow keys, you explore the forest, collecting valuable items along the way. As you progress, you encounter monsters that stand in your path, forcing you to engage in thrilling battles to survive. Finally, after overcoming numerous obstacles, you discover the entrance to the elusive abandoned house. Inside, you encounter a wise man who reveals a hidden treasure key to you, marking the completion of your epic quest.

CHAPTER 4. CONCLUSION

I. Summary

During our project, we faced various obstacles that required resilience and determination. In addition to achieving our primary objectives, this endeavor provided us with invaluable opportunities for expanding our knowledge in DSA and personal growth. We deepened our understanding of DSA through practical application, developed project management and teamwork skills, explored new technologies, and enhanced problem-solving abilities. The project's successful completion instilled a sense of accomplishment and confidence in our abilities.

Overall, this project was a platform for growth, equipping us with valuable skills and experiences for future endeavors in DSA and beyond.

II. Future Improvement

Given an extended time frame for refinement, we would allocate our efforts towards the implementation of the following enhancements:

- **Quest System:** Implement a quest system where players can undertake various quests, complete objectives, and earn rewards. This can add depth and progression to the game.
- **Skill and Leveling System:** Introduce a skill system where players can learn and improve different skills as they level up. This can enhance character customization and provide a sense of growth and specialization.
- **Crafting System:** Create a crafting system that allows players to gather resources and craft items, weapons, and equipment. This can provide additional gameplay options and encourage exploration.
- **Multiplayer Support:** Implement multiplayer functionality to allow players to connect and interact with each other in a shared game world. This can introduce cooperative gameplay, PvP battles, and social interactions.

- Trading System: Introduce a trading system that enables players to buy, sell, and trade items with NPCs or other players. This can create an in-game economy and promote player interaction.
- Mini-games: Include various mini-games or side activities that players can engage in, such as fishing, gambling, or puzzle-solving. This can provide a break from the main gameplay and offer additional challenges and rewards.
- Expanded World and Content: Continuously expand the game world by adding new areas, dungeons, quests, NPCs, and items. This helps to keep the game fresh and provides ongoing content for players.
- Customization Options: Allow players to customize their characters' appearance, such as choosing different outfits, hairstyles, or accessories. This adds personalization and allows players to express their individuality.
- Boss Battles: Design challenging boss encounters that require strategic gameplay and teamwork. Boss battles can provide memorable moments and exciting rewards for players.
- Achievements and Leaderboards: Implement an achievement system to track and reward player accomplishments. Additionally, add leaderboards to showcase the top players in different categories, such as PvP rankings or quest completion times.

REFERENCE

- Series playlist Youtube of RyiSnow "How to Make a 2D Game in Java"
- "Design Patterns in Java - The Catalog of Java Examples" - Refactoring Guru
<https://refactoring.guru/design-patterns/java>
- "Tổng hợp một số Design Pattern trong Java" - Bui Thanh Viet
<https://viblo.asia/p/tong-hop-mot-so-design-pattern-trong-java-qzaGzZWRGyO>