A PROJECT REPORT

Programming In Java (CSE-310) Project
On

TEXT-BASED ADVENTURE GAME USING JAVA

Submitted by

Shagun Sharma	***12102796***
Dhanesh Verma	***12015129***
Ayush Gupta	***12105138***

Submitted to

Mrs. Chandani Bhasin

Currently Pursuing

Bachelor of Technology
(Computer Science and Engineering)

CONTENTS

DECLARATION

I hereby declare that this submission is our own work and that to the best of our knowledge and beliefs. It contains no material l previously published or written by neither any person nor material which to a substance till extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except here due acknowledgement has been made in the text.

Signature:

Shagun Sharma (12102796)
Dhanesh Verma (12105129)
Ayush Gupta (12105138)

ACKNOWLEDGEMENT

In completing this project, we have been fortunate enough to have support and encouragement from our team. I would like to acknowledge them for their cooperation.

Firstly, we would like to thank Mrs Chandani Bhasin, from the Department of Computer Science & Engineering for guiding us through each and every step of the process with knowledge and support. Her thoughts have been a constant source of inspiration for us.

We would also like to acknowledge the contribution of all group members of their kind assistance, suggestions and cooperation throughout the development of the project.

INTRODUCTION

“You wake up. The room is spinning very gently round your head. Or at least it would be if you could see it which you can’t.”

That whimsical description is the introduction to 1978 ‘The Hitchhiker’s Guide to the Galaxy’ (HG2G), a text-based game rendition of Douglas Adam’s timeless classic. And it only gets better…

The history of text-based games began in 1971 with a programmer, cave explorer, and D&D (Dungeons and Dragons) player William Crowther.

A Text-Based Adventure Game is a type of game in which a player has to make choices in every step of the game.

Based on these choices, game proceeds and at last, we get to know that whether the player wins or loses the game.

Text adventure games are a retro concept. Back then, graphics were much rarer for computers, much less the extremely niche realm of computer games. But as dull as what is essentially playing a book sounds, some are pretty good, even by current standards.

OBJECTIVE

The objective of text-based adventure games is to engage players in an interactive storytelling experience where they can make decisions that affect the outcome of the game. These games typically present players with a series of written descriptions and prompts that describe a fictional world and its characters. Players must use their imaginations to visualize the environment and make choices that affect the plot and outcome of the game.

The ultimate goal of a text-based adventure game is to provide players with a compelling and immersive experience that allows them to explore a fictional world and interact with characters in meaningful ways. These games often require strategic thinking and problem-solving skills, as players must make decisions that affect the outcome of the game and navigate obstacles and challenges along the way. Ultimately, the objective is to create an engaging and memorable gaming experience that leaves players feeling satisfied and eager to play again.

PROJECT OVERVIEW

Text-based adventure games, also known as interactive fiction, are a type of computer game where players navigate through a story using text commands. Instead of relying on graphics or audio, the game describes the environment and characters with written words, and the player types in their actions to progress through the game.

- Random Monsters will be appearing and the player must defeat the monster to proceed further in game .
- The Player [User] must choice strategically between two major options:

  1. Attack
  2. Evade

- The player will also be provided with an lifeline of consuming limited health potions as an option 'Drink Health Potion'.
- the Health Power of the Player will be 100 HP initially and Health Power of the monster will be random.
- The Attack Powers of the Player and the monsters will be random.

* There will be a 50% chance drop rate of health potions from the monsters.

* After every elimination of a monster, The User will be provided with the choice of whether to continue further or to exit the game.

Source code

```
import java.util.Random;
import java.util.Scanner;

public class TxtRPG
{
    public static void main(String[] args)
    {
        //System objects
        Scanner in = new Scanner(System.in);
        Random rand = new Random();

        //Game variables
        String[] enemies = {"Skeleton", "Zombie", "Lich", "Vampire", "Lamia"};
        int maxEnemyHealth =  75;
        int enemyAttackDamage = 25;

        //Player variables
        String name = "Player";
        int health = 100;
        int score = 0;
        int attackDamage = 50;
        int numHealthPotions = 3;
        int healthPotionHealAmount = 30;
        int healthPotionDropChance = 50;        //Percentage

        boolean play = true;
        while (play)
        {

            boolean running = true;     //Declaring a boolean value for main loop

            //Introduction Line
            System.out.println("\n\t\t!!! Welcome to the Dungeon !!!\n");

            //Start of the Game
            System.out.println("-------------------------------------------------------
------\n"); //Added dashed line for better UI
            System.out.print("Enter Player's Name: \t");
            name = in.nextLine();

            //Main loop
            GAME:               //This is a 'GAME' label named to the while loop. So
that we can iterate back when needed.
            while(running)      //This is the loop which will keep iterating to keep
the game going.
            {
                System.out.println("\n-----------------------------------------------
------------\n"); //Added dashed line for better UI

                //Introducing a random enemy
                int enemyHealth = rand.nextInt(maxEnemyHealth);
```

```
                String enemy = enemies[rand.nextInt(enemies.length)];
                System.out.println("\t#\tA "+ enemy + " has appeared!      #\n");

                //generates random score number
                int genscore = rand.nextInt(1,enemies.length);
                // System.out.println(genscore);

                //Loop till the enemy's HP is 0
                while(enemyHealth > 0)
                {
                    //Platform Details and Choices to play game
                    System.out.println("\n\t\t"+name+"'s HP: " + health);
                    System.out.println("\t\t" + enemy + "'s HP: " + enemyHealth);
                    System.out.println("\n\tWhat would you like to do?");
                    System.out.println("\t 1. ATTACK!!!");
                    System.out.println("\t 2. Drink Health Potion");
                    System.out.println("\t 3. Evade! :| ");

                    //Taking Input from User and implementing accordingly
                    String input = in.nextLine();
                    if (input.equals("1"))
                    {
                        int damageDealt =
rand.nextInt(attackDamage);       //Generating random attack damge of player
                        int damageTaken =
rand.nextInt(enemyAttackDamage);      //Generating random attack damge of enemy

                        //Reducing both HPs accordingly
                        enemyHealth -= damageDealt;
                        health -= damageTaken;

                        //Printing post-fight details
                        System.out.println("-----------------------------------------
------------------"); //Added dashed line for better UI
                        System.out.println("> You striked the "+enemy+" for
"+damageDealt+" damage.");
                        System.out.println("> You received the "+damageTaken+" damage
in retaliation.");
                        System.out.println("-----------------------------------------
------------------"); //Added dashed line for better UI

                        //Breaking loop at Low Health alert
                        if (health < 1)
                        {
                            System.out.print("\n\t\tLOW HEALTH\t\t\n");
                            System.out.println(">You have taken too much damage, you
are too weak to go on!");
                            break;
                        }
                    }
                    else if (input.equals("2"))
                    {
                        //Applying available Health Boost
                        if (numHealthPotions > 0) {
                            health += healthPotionHealAmount;
                            numHealthPotions--;
                            System.out.println("-----------------------------------
----------------------"); //Added dashed line for better UI
```

```
                            System.out.println("\n\t> You drink a health potion ,
healing yourself for "+healthPotionHealAmount+" HP.");      //Printing Received HP
                            System.out.println("\t> You now have "+health+"
HP.");     //Printing Total HP
                            System.out.println("\t> You have "+numHealthPotions+"
health potions left.\n");       //Printing Health Potions Left
                            System.out.println("----------------------------------
----------------------"); //Added dashed line for better UI
                        }
                        //Health Boost Unavailable
                        else
                        {
                            System.out.println("\n\t> You have no health potions left!
\n\t>Defeat enemies for a chance to get one!!\n");       //Printing Health Potions
Left
                        }
                    }
                    else if (input.equals("3"))
                    {
                        //Printing Evade from enemy
                        System.out.println("\n\t> You evaded away from the "+enemy+"
!");
                        continue GAME;          //recurses back to Main loop
                    }
                    else
                    {
                        //Default for negating Invalid Input
                        System.out.println("\n\t    #       Invalid Command     #");
                    }
            }

            if(health< 1)
            {
                System.out.println("\n\t You limp out of the dungeon, weak from
battle.");
                break;
            }

            // System.out.println("------------------------------------------------
---------\n"); //Added dashed line for better UI

            //Printing Player's Stats
            System.out.println("\n\t#    "+enemy+" was DEFEATED !!\t#");
            System.out.println("\n\t#   You have "+health+" HP left !\t#");
            score += genscore;      //Incrensing Score
            System.out.println("\n\t#   Your Score is : "+score+"\t#\n");

            //Health Potion Drop
            if (rand.nextInt(100) < healthPotionDropChance) {
                numHealthPotions++;
                System.out.println("\t# The "+enemy+" dropped a health potion! #");
                System.out.println("\t# you now have "+numHealthPotions+" health
potion(s)");
            }

            //Continuing the Game
            System.out.println("\n\t#   Your Score is : "+score+"\t#\n");
```

```
            System.out.println("-------------------------------------------------------
------\n"); //Added dashed line for better UI
            System.out.println("\n\tWhat would you like to do now?");
            System.out.println("\t 1. Continue Fighting!");
            System.out.println("\t 2. Exit the Dungeon");

            //Taking User's Choice in Input for continution
            String input = in.nextLine();
            while (!input.equals("1") && !input.equals("2"))        //Loop for
negating Invalid Input
            {
                System.out.println("\n\t# Invalid Command #");
                input = in.nextLine();
            }

            //Implementing according to User's Input
            if (input.equals("1")) {
                System.out.println("> You continue on your adventure.");
            }
            else if (input.equals("2")) {
                System.out.println("> You exit the Dungeon, successfully from your
adventures!!\n");
                System.out.println("\n\t#   Your HighScore is : "+score+"\t#\n");
                break;
            }
            }

            //Closure Line
            System.out.println("\t\t#########################");
            System.out.println("\t\t# THANKS FOR PLAYING ! #");
            System.out.println("\t\t#########################");

            System.out.println("-------------------------------------------------------
------\n"); //Added dashed line for better UI

            //Adding Loop for 'Try Again'
            System.out.println("\tWould you like to try again?\n");
            System.out.println("\t\tTRY AGAIN [Y/N]\t\t\n");
            // "y" will set 'play' to 'true', anything else will set it to 'false'
            play = in.nextLine().trim().equalsIgnoreCase("y");
        }

        if (play == false )
        {
            System.out.println("\t\t######################");
            System.out.println("\t\t# COME BACK AGAIN ! #");
            System.out.println("\t\t######################");

        }
        System.out.println("Press any key to
continue.....................................................................
....");
        in.nextLine();
        in.close();
    }
}
```

Future Trends of Text-Based Games

Text-based adventure games have been around for several decades and have evolved significantly since their inception. In the future, there are several aspects that may shape the development of these games:

- Advancements in natural language processing (NLP): As NLP technology advances, it may become possible to create more sophisticated and nuanced game narratives that can adapt to player input in real-time.
- Integration with other technologies: Text-based adventure games may be integrated with other technologies, such as virtual reality, to create more immersive and interactive experiences.
- Increased emphasis on accessibility: Developers may prioritize making text-based adventure games more accessible to players with disabilities by incorporating features such as text-to-speech and audio descriptions.
- More focus on multiplayer experiences: As online gaming becomes more prevalent, there may be a greater emphasis on creating multiplayer text-based adventure games that allow players to interact and collaborate with each other in real-time.
- Greater diversity in game narratives: As the gaming industry continues to evolve, there may be a greater focus on

creating more diverse and inclusive game narratives that reflect a broader range of perspectives and experiences.

Overall, the future of text-based adventure games is likely to be shaped by a combination of technological advancements, player preferences, and industry trends.