Herriot Watt University

School of Engineering and Physical Sciences

Electrical and Electronics Engineering Department

Robotics and Autonomous Systems Engineering

Game Project

Nihal Shajahan Oravil

H00489071

Dr. Girish Balasubramanian

13th April 2025

Game Project

1. Overview:

- This is a simple, turn-based battle game written in C. The player chooses a class (Warrior, Mage, or Rogue), and battles against a computer-controlled opponent. Both participants take turns attacking, using special abilities, or healing, until one is defeated. The game emphasizes strategy through the use of risk-reward mechanics (e.g., special attacks) and health management.

2. Rules of the game:

Objective:

Defeat the computer by reducing its health to 0 before yours reaches 0.

Player Controls:

1. Normal Attack
2. Special Attack (50% chance to miss but higher damage)
3. Heal (recovers 10 HP)

Game Mechanics:

Normal Attack: Deals Random Damage upto your attack power

Special Attack: Risky attack, 50% chance to miss, but deals 2x damage if hit

Heal: Adds 10 HP. Each class has different max HP.

3. Classes

CLASS	HEALTH	ATTACK POWER	DESCRIPTION
Warrior	**120**	**8**	High durability, weaker attack
Mage	**80**	**15**	Fragile but strong attacks
Rogue	**100**	**10**	Balanced stats

4. Software Design:

The program is structured using structs and enums for clarity and maintainability.

Core Components:

1. **Player struct**: Holds name, health, attack power, and class type
2. **Class enum**: Distinguishes between Warrior, Mage, and Rogue.
3. **Game Loop**: Alternates turns between player and computer until one is defeated.

Main Functions:

1. attack() - Handles damage calculations.
2. playerTurn() - Prompts player for action and executes it.
3. computerTurn() - Randomly chooses computer's action and executes it.
4. getClassName() - Converts class enums to strings.

Program Flow:

1. Player selects a class at the start.
2. Player and computer take turns.
3. Each turn consists of choosing and performing an action.
4. Health is updated accordingly.
5. Game ends when one character's health drops to 0 or below.
6. The structure is simple and modular, making it easy to expand.

5. How To Play:

Choose Your Class:

- Upon starting, you will be asked to pick a class:
  1 for Warrior, 2 for Mage, 3 for Rogue.
- Your class affects your starting health and attack power.

Gameplay Loop:

- The game is turn-based. You will always go first.
- On your turn, you'll be given 3 options:
  - 1: Normal Attack — Safe, consistent damage.
  - 2: Special Attack — Risky, powerful if it lands.
  - 3: Heal — Restore 10 HP, up to your max health.

Enemy Turn:

- After your move, the computer will also pick a random action.
- It follows similar logic, attack, special attack, or heal.

Winning or Losing:

- Keep an eye on your health and the enemy's.
- The game ends when either the player or computer reaches 0 HP.

Tips:

- Use normal attacks when playing it safe.
- Take the gamble with special attacks when you need to turn the tide.
- Don't forget to heal when your HP gets low

References:

https://canvas.hw.ac.uk/courses/27292/assignments/170726