



Object Oriented Programming

Project

Team

- 1)Sumaiya Akter (2233081495)**
- 2)Sadia Islam (2233081485)**
- 3)Maria Mahjabeen Mim (2233081491)**
- 4)Md.Antoreep Bin Bayzid (2233081490)**
- 5)Fahad Hossain Lam (2233081509)**



Class Design

Player Class

- **Purpose:** Base class for all characters.
- **Properties:** Name, Health, Energy, XP
- **Methods:**
 - **Attack:** Virtual method for character-specific attacks.
`//public virtual void Attack(Player target)//`
 - **TakeDamage:** Reduces health when attacked.
`//public void TakeDamage(int damage)`
 - **Dodge:** Placeholder for dodge action.
`//public virtual void Dodge ()//`
 - **IsAlive:** Checks if the character is alive.
`//public bool IsAlive()//`

Warrior Class

- **Purpose:** Represents a melee fighter with high health
- **Inheritance:** Derived from Player.
- **Special Features:**
 - Specialized melee attacks using energy.
 - High health and moderate energy.

Warrior Class Methods

- **Attack:** Performs a strong melee attack using energy.
`//public override void Attack(Player target)//`
- **Dodge:** Provides a dodge mechanism for combat.
`//public override void Dodge ()//`

Mage Class

- **Purpose:** Represents a spellcaster with high damage.
- **Inheritance:** Derived from Player.
- **Special Features:**
 - Casts spells with high damage but requires more energy.
 - Balanced health and energy.

Mage Class Methods

- **Attack:** Casts a spell to inflict damage.
`//public override void Attack(Player target)//`
- **Dodge:** Provides a dodge mechanism for combat.
`//public override void Dodge ()//`

Boss Class

- **Purpose:** Represents a strong and durable enemy.
- **Inheritance:** Derived from Player.
- **Special Features:**
 - Unique attack style.

Boss Class Methods

- **Attack:** Unleashes a powerful attack on the player.
`//public override void Attack(Player target)//`

Static NPC Class

- **Purpose:** Handles quest system and storyline guidance.

NPC Class Method

- **Quest:** Gives a quest to the player
`//public static void Quest ()//`

Code Flow

Character Selection:

- Player chooses either a Warrior or Mage.

Quest Interaction:

- NPC offers a quest to retrieve a relic.
- Accepting the quest rewards XP.

Boss Battle:

- Turn-based system: Player chooses to attack or dodge.
- Boss counterattacks if the player doesn't act.

```
//while (player.IsAlive() && boss.IsAlive())//
```

Game Outcome:

- Player wins if the boss's health reaches zero.
- Game over if the player's health depletes.

Sample Output

- **Character Selection:** Choose your character: 1. Warrior 2. Mage
- **Quest Interaction:** A strange villager offers you to retrieve an ancient relic!
- **Boss Battle:**
 - Hero swings their sword!
 - Dragon unleashes a powerful attack!
 - You defeated the boss! Congratulations!

Conclusion

This code demonstrates foundational RPG mechanics in C#, including character selection, turn-based combat, and a simple quest system. It provides a great starting point for building more complex RPG games.

thank
you!