

**Documentation of the project**

Discipline: **Design pattern**  
Teacher: Ukibassov Baubek  
Student: Miramuly Zharaskhan  
Specialize: Digital Engineering 2<sup>nd</sup> Course  
Project title: Solid Adventure Game

**Introduction**

This project is a refactored version of a simple RPG game. The main goal was to break down a monolithic game structure into smaller, manageable parts using SOLID principles. It helps to better understand OOP and how to write clean, scalable code.

**SOLID Principles in My Code****Single Responsibility Principle (SRP)**

Each class has one job. Player, CombatManager, LevelManager, and ItemManager handle their own responsibilities.

**Open/Closed Principle (OCP)**

New enemies or items can be added without changing existing code.

**Liskov Substitution Principle (LSP)**

Any enemy can be replaced with another without breaking the game.

**Interface Segregation Principle (ISP)**

IAttackable interface ensures flexibility in combat.

**Dependency Inversion Principle (DIP)**

High-level modules do not depend on low-level modules; they rely on abstractions.

**What I Learned**

I learned how to structure a game properly using OOP, making the code modular and easy to manage. SOLID principles helped me separate concerns, reduce dependencies, and ensure the game could scale without breaking existing features. Now, adding new enemies, items, or mechanics is smooth and doesn't turn into a headache. Simple, clean, and structured—exactly how code should be.

Diagramm:

