Teamwork Project Assignment for the OOP Course @ SoftUni

Create a 2D / 3D object-oriented **role-playing game** of your own choice. Your project must meet all the requirements listed below.

Requirements

- **Use C# or Java** – the game should be written in either C# or Java
  - The game can be console-based or written using some kind of a Graphical User Interface (e.g. WPF, Windows Forms, Swing, AWT, Java FX, etc.)
  - You are not allowed to use any Game engine technology (e.g. Unity, etc.)
- **Work in team** – all team members should contribute
  - **Use GitHub** or another source control system as a project collaboration platform
  - **Each team member should have commits in 5 different days**
- **Quality OOP code**
  - The project should be properly structured and should **follow the good practices of OOP**. Use data encapsulation, use exception handling properly, use inheritance, abstraction and polymorphism properly, follow the principles of strong cohesion and loose coupling.
- **The game should implement the following object-oriented assets:**
  - At least **5 interfaces** (with one or more implementations)
  - At least **15 classes** (implementing the application logic)
  - At least **3 abstract classes** (with inheritors)
  - At least **1 exception class** (with usage in your code)
  - At least **3 levels of depth in inheritance**
  - At least **1 polymorphism** usage
  - At least **1 structure** (only in C#)
  - At least **1 enumeration**
  - At least **1 event** (aside from key listeners)
  - At least **1 use of functional programming**
- **Gameplay and UX**
  - The game should have intuitive and smooth gameplay
  - The UI should be user-friendly

Forbidden Techniques and Tools

You are **NOT allowed** to copy / paste an existing project from Internet.

Game Description

A Role Playing Game (RPG) is a game in which players assume the roles of characters in a fictional world.

- There can be several **players** and **computer-controlled characters** (allies, enemies) and of different type (warriors, mages, creatures, etc.)
- The game characters can wear **items** with different **effects** and uses (swords, shields, guns, etc.)
- There can be several **skills** and **abilities** that the player or the enemies may use (jumping, casting spells, teleporting, etc.)
- There can be several **character interactions** (attacking, healing, buying)

Deliverables

Put the following files in a **ZIP archive** and submit it (each team member submits the same file):

- The complete **source code** of your project (C#/Java files, images, sounds and other files).
- A **presentation** of your project (e.g. PowerPoint slides) of your project. It should provide the following information:
  - Project name and purpose – what you have created?
  - Team name, list of team members.
  - Contributions of each team member.
  - Technical description.
- Any other information (optionally).

Public Project Defense

Each team will have to deliver a **public defense** of its work in front of the other students, trainers and assistants. Teams will have **only 10 minutes** for the following:

- **Demonstrate** the game (very shortly).
- Show the **source code** and explain how it works.
- Explain how each team member has **contributed**: display the commit logs in the Source Control System you are using.
- Optionally you might prepare a **presentation** (3-4 slides).

Please be **strict in timing**! On the 10th minute mark you **will be interrupted**! It is good idea to leave **the last 2 minutes for questions** from the other students, trainers and assistants.

Be **well prepared** for presenting maximum of your work for minimum time. Bring your own laptop. Test it preliminary with the multimedia projector. Open the project assets beforehand to save time.

Assessment Criteria

- **Gameplay** (game experience from a user point of view) – **0…6**
- **Code structure** (Class structure, Interfaces, proper Inheritance) – **0…6**
- **Code quality** (correct naming, code formatting, separation of concerns, correct application of OOP principles) – **0…10**
- **Teamwork*** (GitHub used; regular contribution; distribution of tasks) – **0…3**
- **Bonus** (bonus point are given for implementing optional functionalities according to the type of project you choose) – **0…5**

* If not all team members have contributed to the project, this **does not affect** the teamwork score.

Give Feedback about Your Teammates

You will be invited to **provide feedback** about all your teammates, their attitude to this project, their technical skills, their team working skills, their contribution to the project, etc. The feedback is important part of the project evaluation so **take it seriously** and be honest.