

**VGP133 Intro to C# Programming**

**Final Project**

*Estimated hours:​ 16-24 hours*

**

Assignment Goals:

* Apply and practice the foundational, object-oriented programming, file structures, and .NET libraries in C#
* Create a medium-scale project to apply all the topics learned in C#

Assignment Hand-in

* Submit the project on Omnivox
* **Naming Format: VGP133\_Final\_LastName\_FirstName**
* Please hand in the **BUILD ZIP file** **AND the ZIP file of the C# Visual Studio Project**. **Failure to have both will result in deductions in the final mark.**

Assignment Overview

You will create a small-scale text-based RPG game in C# using Visual Studio. Please read all the instructions first.

* This is an individual project. You must write all the code here yourself.
* For the Final Milestone (Week 11), I’ll play the game in front of you and ask you some questions about your game and code. I may deduct marks for failing to answer questions related to your code or algorithm.
* I can regularly check-in with everyone at the start of every week to see progress. Please use the time in class to work on this project as well.

You are free to design how you’d like to play, but it must have the following features:

**Milestone 1 (Week 9)**

* Character Creation
  + Once the player starts the game, allow them to create their character. Ask the player for the following player information, and have them input accordingly:
    - Name
    - Hair Color
    - Gender
    - Age
  + The player will also have battle-specific stats
    - Max HP
    - Current HP
    - ATK
    - DEF
* Basic Exploration
  + The player will have a main overworld, where they can choose where to go, or what to do. Here are the choices they can select:
    - Go to Town
      * Refill Player HP back to Max (Inn)
      * Buy Consumable Items such as Potions. (Item Shop)
      * Buy Equippable Weapons and Armor. These items strengthen the character during battle. (Weapon Shop)
      * Review the player’s character appearance (name, hair color, gender, and age), and their current battle stats.
    - Go to the Forest
      * When the player enters the forest, the following can occur:
        + 50% of the time, they encounter a monster. Make sure to have a pool of monster the player can challenge. You can tweak the odds of the chances each monster can spawn.
        + 50% of the time, they will receive a random set of item reward and money. You can determine the loot table they receive.
    - Go to the Mountains
      * Same as going to the forest, but there will be harder monsters if they encounter them. You are free to pick from the pool of monsters you create for Milestone 2.
    - Go to the Boss Castle
      * Once the player enters this, they will fight two battles with a random monster, followed by the final boss.
      * If they defeat the boss, they win the whole game.
      * If they lose, they will be sent back to the main overworld and the penalty for losing applies. They also must go through the gauntlet of monsters again.
    - Check your Inventory
      * Create an Item class for the items the player can purchase
      * Create a List of Items that will contain all the items the player has
      * Shows all the items that you have in your inventory
      * Allows you to sort through them by:
        + Quantity
        + Name
        + Type (Consumable or Equipment)
    - Change Equipment (Milestone 2)
    - Save the Game (Milestone 3)
    - Load the Game (Milestone 3)

**Milestone 2 (Week 10)**

* Monsters
  + The monsters will inherit from a Monster class that contains the following:
    - Properties (you may have more if needed)
      * Name
      * ATK
      * DEF
      * HP
      * Gold Dropped
    - Methods. A monster can choose one of two attacks
      * **Basic Attack**: does the damage to the player. The same damage formula will occur (ATK-DEF)
      * **Special Attack**:
        + It MUST be different for every monster you create (use class inheritance/polymorphism)
        + It must NOT be just a basic multiplier, but at least something unique it can bring to the battle (e.g. A powerful attack but skips the next turn, A very strong attack with big chance to miss, Lifestealing from opponent, etc.)
    - Create **AT LEAST 5 different monsters** in the game. Each of them inheriting from the Monster class. They can have different abilities that they use during battle.
* Combat System
  + The combat system will be basic where the player can choose to attack, use an item, or flee.
    - **ATTACK:** Player will deal damage equal to their ATK power MINUS the opponent’s DEF. The damage cannot be less than 1.
    - **USE ITEM:** Player can choose to use an item in their inventory that heals their HP.
    - **FLEE:** The player runs from battle and goes back to the main overworld.
  + After the player’s turn, it will be the monster’s turn.
    - **ATTACK** (use 70% of the time): The monster will do a basic attack on the player.
    - **SKILL** (use 30% of the time): The monster will do their special attack to the player.
  + Battle Result
    - **WIN:** When the player beats the monster, they will receive some gold, which they can use to buy other items in town.
    - **LOSE:** When the player reaches 0 HP during battle, they will be returned to the main overworld, and lose half of their money.
* Change Equipment
  + The player can equip a Weapon and an Armor to their character to strengthen their stats in battle
  + When entering the Change Equipment Screen, the player can select which equipment to replace (either Weapon or Armor)
  + If they replace the Weapon, they must see the weapon they currently equipped, and all the weapons they have in their inventory that they can equip.
    - Vice versa for Armor
  + Make sure exchanging your equipment works, and doesn’t end up in some bugs (e.g. losing the equipment, duplicating the equipment, or permanently increasing player’s stats)

**Milestone 3 (Week 11)**

* Implement the Save and Load functionality
  + Have three different slots the player can save the data into
  + Make sure to display the Name of the player on the Save data as well
  + When saving, make sure to save all the important information of the **player data**, equipped weapon/armor, and **current inventory**, and any other relevant information that needs to persist to future play sessions.
  + You may use either Text, JSON, XML, or Binary to store the file. (XML/JSON is recommended)
* Polishing, Error Tracking, and Bug Fixing
  + Make sure everything is well polished and the game can easily be played without the need of external presence.
  + The UI and instructions should be self-explanatory.
  + Code commenting is optional but encouraged.
  + Implement proper names to the items, monsters, skills and characters in the game.

Marking Procedure

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Criteria** | **Mark** | **Total** |
| 1 | The text prompts of the whole text adventure is understandable and easy to navigate |  | 10 |
| 2 | Navigating across different screens and areas are functional and does not result in soft locks. |  | 15 |
| 3 | Inventory Management Screen is complete and fully functional without errors. |  | 10 |
| 4 | Changing Equipment is functional without major bugs. |  | 10 |
| 5 | The Combat System is functional for the regular monsters and the boss. |  | 25 |
| 6 | Saving and Loading features are all implemented and bug-free. |  | 20 |
| 7 | Creativity of the Game. Names of monsters and skills are creatively named and not just placeholders |  | 10 |