Operation: Dungeon

A simple Role-Play Game (RPG) Battle System

# Summary

Your party is resting in a tavern when an old man approaches you with tales of a dungeon with untold riches hidden away from the plundering hands of lesser adventurers. If you are willing to venture upon this quest, then the legendary “A of Seeing Sharpness” may be yours. Get it? Eh? Fine, in reality, you are a computer science student (someone who has faced far more fearsome foes than some run of the mill adventurer; did Conan ever have to make a compiler? Didn’t think so.) and for this semester’s project you will be creating a battling system similar to, but much more simple than, traditional turn based RPGs (*Final Fantasy 1-8*) Now to go over a bit of the specifics starting with:

## What this project doesn’t include

* Story
  + Seriously, scripting out a proper story would require an entire school year to properly implement.
* Adventuring
  + You are in the dungeon. That’s it. No towns to get resupplies. No sitting around a campfire. Just imagine a game made entirely out of random encounters (and the combat that they include).
* Epic 50+ hours of content
  + I want to be able to get a decent feel for your creation in 10 minutes or less.
* Leveling system
  + These are a pain to balance properly, so a no-go (at least until every last thing listed is finished, working perfectly, and you are bored to tears waiting to do more work)

## With those points out of the way, let’s talk about what the project will actually include:

* Heroes
  + You will have 3 heroes in your party that will take part in every battle:
    - A Warrior
      * Does physical damage.
      * High defense
    - A Mage
      * Does magic damage.
      * Low defense
    - A Cleric
      * Can heal party members
      * Low attack
      * Medium defense
  + Each hero will have the following stats:
    - Strength – relates to physical strength
    - Intelligence – relates to mental (magical) strength
    - Attack – a function of strength and intelligence.
    - Defense – How much incoming damage is reduced by
  + They will also have the following properties:
    - Hit Points – How much health they currently have (out of a max pool)
    - Skill Points – relates to using skills outside of basic hit/block (magic attacks or special physical attacks)
    - Speed – How quickly will they react in combat.
* Enemies
  + You will have 3 different types of enemies:
    - Bandit – Just a simple bandit
      * Low defense
      * Low hit points
      * Not very strong
    - Ogre – A slow moving fellow that packs powerful physical attacks.
      * High Defense
      * Medium hit points
    - Dragon
      * Strong attacks
        + Breath Fire – hits all heroes
        + Swipe Attack – hits just one hero
      * Slow
      * High Hit points
  + Enemies will have the same types of stats at heroes.
* Game will take place in a Windows Form
  + Sprites will be displayed for each character (and enemy)
  + You will control the game through the use of Controls (both default and custom)
  + It will be turn based, the order in which characters and enemies act will be based on their speed property.
  + Properties like hit points and skill points will be visible on screen and update as needed.
  + Do not worry about animations!
    - Have a label (or custom control) that tells you each action taken.
  + Heroes will fight between 1-3 (inclusive) enemies per encounter.
    - Heroes need a way of targeting which enemy they want to attack.
  + Game saves stats like:
    - Highest number of levels beaten in one game.
    - High score.

# Project Turn-ins:

## 1st Turn-in (Pre-Production) - 200 points:

* UML Diagram of classes necessary to create game.
  + Created in UMLet
* Mock-up of game form
  + Created in Visual Studio.
* **Due: 4/12/17**

## 2nd Turn-in (Progress Check) – 200 points:

* At least one hero character and one enemy working for one battle.
* (you can have 3 of one type of hero / enemy visible if you still want proper formatting).
* **Due: 4/24/17**

## 3rd Turn-in (Completed Project): - 200 points:

* Full implementation of all features listed.
* Game plays well with no exceptions / errors being thrown in the course of up to 10 minutes of play.
* **Due: 5/3/17**

# Code Requirements:

* All code must be written with an emphasis placed on function and maintainability with special attention paid to OOP principles. This means:
  + Descriptive comments
    - Methods have /// comments
    - Each class has a comment at the top detailing its purpose and who wrote it.
  + No commented out code left in the completed project
    - It either works or it doesn’t and if the final project doesn’t use it, I don’t care about it.
  + Meaningful variable names
  + Different classes written in different .cs files
* **This will be worth 25% each for the Progress Check and Completed Project.**