

# CECS 277 – Project 1

## Dungeon Master

Create a program that allows a user to explore a dungeon maze and fight monsters that they encounter along the way. Use the UML diagram on the next page, the example output, and the descriptions below to help you create your classes.

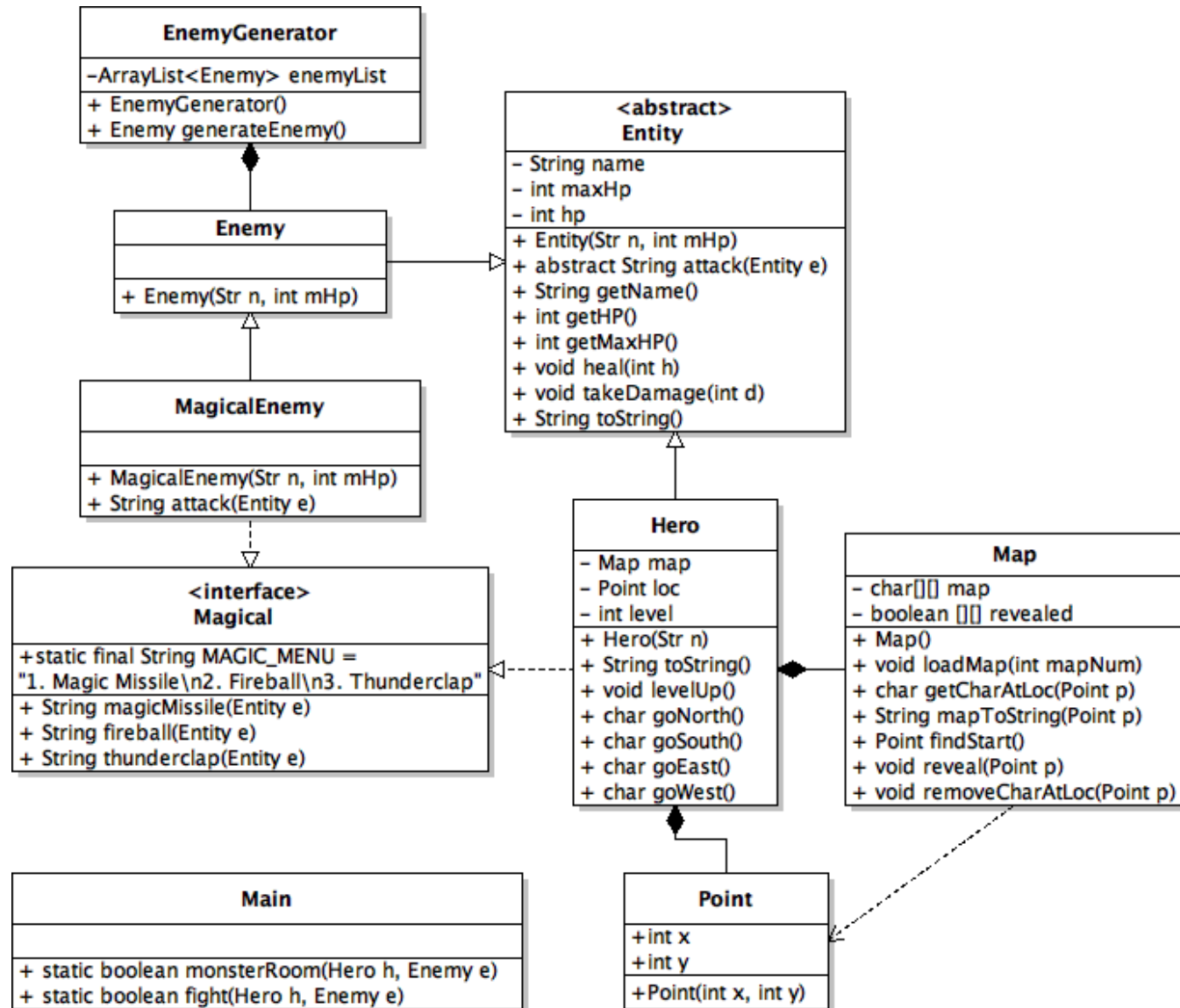
Classes:

1. Entity – describes a character in the game
  - a. an entity has a name and some hit points (maxHp is the maximum amount of hp an entity can have and hp is initialized to the maxHp).
  - b. heal method should increase the entity's hp by the amount passed in, but it should not go above the maxHp value.
  - c. takeDamage method should decrease the entity's hp by the amount passed in, but it should not go below 0.
  - d. toString method should display the name and hp over maxHp.
2. Hero – describes the character that represents the user.
  - a. the hero has a map, a location on the map, and a level.
  - b. Construct a hero with a name, some hp, a new Map, at the start of level 1.
  - c. direction methods should change the hero's location (if that location is within the bounds of the map), reveal that location, get the character at that location, remove it and then return it.
  - d. levelUp method increments the hero's level and loads the next map (note: the level should continue to increase, but the maps are numbered 1, 2, and 3, and should be repeated in that order (a finish is the next map's start)).
  - e. attack methods should each do a different amount of damage within a random range and returns a string representing that damage.
  - f. toString should display the name, hp, level, and map.
3. Map – represents the dungeon maze
  - a. a map has a 5x5 set of characters representing the types of rooms in the maze, and a 5x5 set of booleans that allow you to determine if that room has been visited yet.
  - b. loadMap reads in the map from the file and stores it in the character array.
  - c. mapToString returns a string of the map with the hero's current position, revealed rooms, and any unrevealed rooms represented by 'x's.
4. Enemy – represents an enemy the hero will encounter
  - a. an enemy has a name and some hp
  - b. attack method should do some damage within a random range and return a string representing that damage.
5. MagicalEnemy – represents a magical enemy
  - a. attack methods randomly select one of the three spells. Each of the spell methods should do a different amount of damage within a random range and return a string representing that damage.
6. Magical – defines magical abilities, implemented by Hero and MagicalEnemy.
7. EnemyGenerator – creates random enemies to encounter in the maze
  - a. constructor reads the file and adds the different enemies to the ArrayList (do not assume you know the length of the file).

- b. generateEnemy method randomly selects an enemy from the list, then randomly selects either Enemy or Magical Enemy, then copies over the name and hp to construct a new enemy of that type. If it is a magical enemy, then add a title like “Magical”, “Wizard”, or “Warlock” to the name (that way you know it’s supposed to be casting spells).
8. Main
  - a. prompt the user to enter a name, then construct a Hero with that name.
  - b. display the hero with the map and have the user choose a direction.
  - c. get the resulting character from the hero’s direction methods
    - i. x – location was out of bounds
    - ii. n – nothing here
    - iii. s – start (nothing here)
    - iv. f – finish – level up the hero to move to the next map
    - v. i – item – the hero finds a health potion. Heal the hero for 25 hp
    - vi. m – monster – fight an enemy by calling monsterRoom.
  - d. monsterRoom displays the enemy and then repeatedly prompts the user to fight or to run away. If they choose to fight, call the fight method. If they run away, then choose a random direction to move the hero. Return true if the hero is still alive after the entire encounter.
  - e. fight allows the user to choose to do a physical attack or a magical attack. If they choose physical, then call the hero’s attack method. If they choose magical, then allow them to choose from the magical menu for which spell to cast. The enemy then attacks back if it is still alive. Return true if the hero is still alive.
  - f. repeat from b until the user quits or the hero dies.

## Notes

- You can use the Point class from the java.awt library to keep the location of the hero, or make your own Point class.
- EnemyGenerator class reads in the file in the constructor to make a list of templates. The generateEnemy method randomly chooses from the template list and constructs a new enemy (ie. do not return the template’s reference). Make sure the enemies you create are of the right type (either physical or magical).
- The attack and magic attack methods should do a random amount of damage to the entity passed in. Return a string representing the attack with the amount of damage done to the entity (see example output). The MagicalEnemy class’s attack method should choose a random magic attack to do (ie. magical enemies should not have a physical attack).
- Please do not add any extra instance variables or methods to the UML.
- Ask questions about any methods you do not fully understand.
- This project will be expanded on for Project 2.



What is your name, traveler? Link  
 Link  
 HP: 25/25  
 Level: 1  
 x x x x x  
 x x x x x  
 \* x x x x  
 x x x x x  
 x x x x x  
 1. Go North  
 2. Go South  
 3. Go East  
 4. Go West  
 5. Quit  
 2  
 You've encountered a Kobold Warlock  
 Kobold Warlock  
 HP: 4/4  
 1. Fight  
 2. Run Away  
 1  
 1. Physical Attack  
 2. Magic Attack  
 1  
 Link attacks Kobold Warlock for 1 damage.  
 Kobold zaps Link with Thunderclap for 7 damage.  
 Kobold Warlock  
 HP: 3/4  
 1. Fight  
 2. Run Away  
 1  
 1. Physical Attack  
 2. Magic Attack  
 2  
 1. Magic Missile  
 2. Fireball  
 3. Thunderclap  
 2  
 Link hits Kobold Warlock with a Fireball for 6 damage.  
 You defeated the Kobold Warlock!  
 Link  
 HP: 18/25  
 Level: 1  
 x x x x x  
 x x x x x  
 s x x x x  
 \* x x x x  
 x x x x x  
 1. Go North  
 2. Go South  
 3. Go East  
 4. Go West  
 5. Quit  
 3  
 You found a Health Potion! You drink it to restore your health.  
 Link  
 HP: 25/25

Level: 1  
 x x x x x  
 x x x x x  
 s x x x x  
 n \* x x x  
 x x x x x  
 1. Go North  
 2. Go South  
 3. Go East  
 4. Go West  
 5. Quit  
 3  
 There was nothing here.  
 Link  
 HP: 25/25  
 Level: 1  
 x x x x x  
 x x x x x  
 s x x x x  
 n n \* x x  
 x x x x x  
 1. Go North  
 2. Go South  
 3. Go East  
 4. Go West  
 5. Quit  
 3  
 You've encountered a Giant Rat  
 Giant Rat  
 HP: 1/1  
 1. Fight  
 2. Run Away  
 1  
 1. Physical Attack  
 2. Magic Attack  
 1  
 Link attacks Giant Rat for 4 damage.  
 You defeated the Giant Rat!  
 Link  
 HP: 25/25  
 Level: 1  
 x x x x x  
 x x x x x  
 s x x x x  
 n n n \* x  
 x x x x x  
 1. Go North  
 2. Go South  
 3. Go East  
 4. Go West  
 5. Quit  
 2  
 You've encountered a Orc  
 Orc  
 HP: 4/4  
 1. Fight  
 2. Run Away  
 2  
 Link  
 HP: 25/25

Level: 1

x x x x x  
x x x x x  
s x x x x  
n n n n x  
x x \* n x

1. Go North
  2. Go South
  3. Go East
  4. Go West
  5. Quit
- 3

There was nothing here.

Link

HP: 25/25

Level: 1

x x x x x  
x x x x x  
s x x x x  
n n n n x  
x x n \* x

1. Go North
  2. Go South
  3. Go East
  4. Go West
  5. Quit
- 3

You found the exit. Proceeding to the next level.

Link

HP: 25/25

Level: 2

x x x x x  
x x x x x  
x x x x x  
x x x x x  
x x x x \*

1. Go North
  2. Go South
  3. Go East
  4. Go West
  5. Quit
- 1

There was nothing here.

Link

HP: 25/25

Level: 2

x x x x x  
x x x x x  
x x x x x  
x x x x \*  
x x x x s

1. Go North
  2. Go South
  3. Go East
  4. Go West
  5. Quit
- 2

You're back at the start.

Link

HP: 25/25

Level: 2

x x x x x  
x x x x x  
x x x x x  
x x x x n  
x x x x \*

1. Go North
  2. Go South
  3. Go East
  4. Go West
  5. Quit
- 5

Game Over