CECS 277 – Project

Dungeons & Monsters

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 program.

Classes:

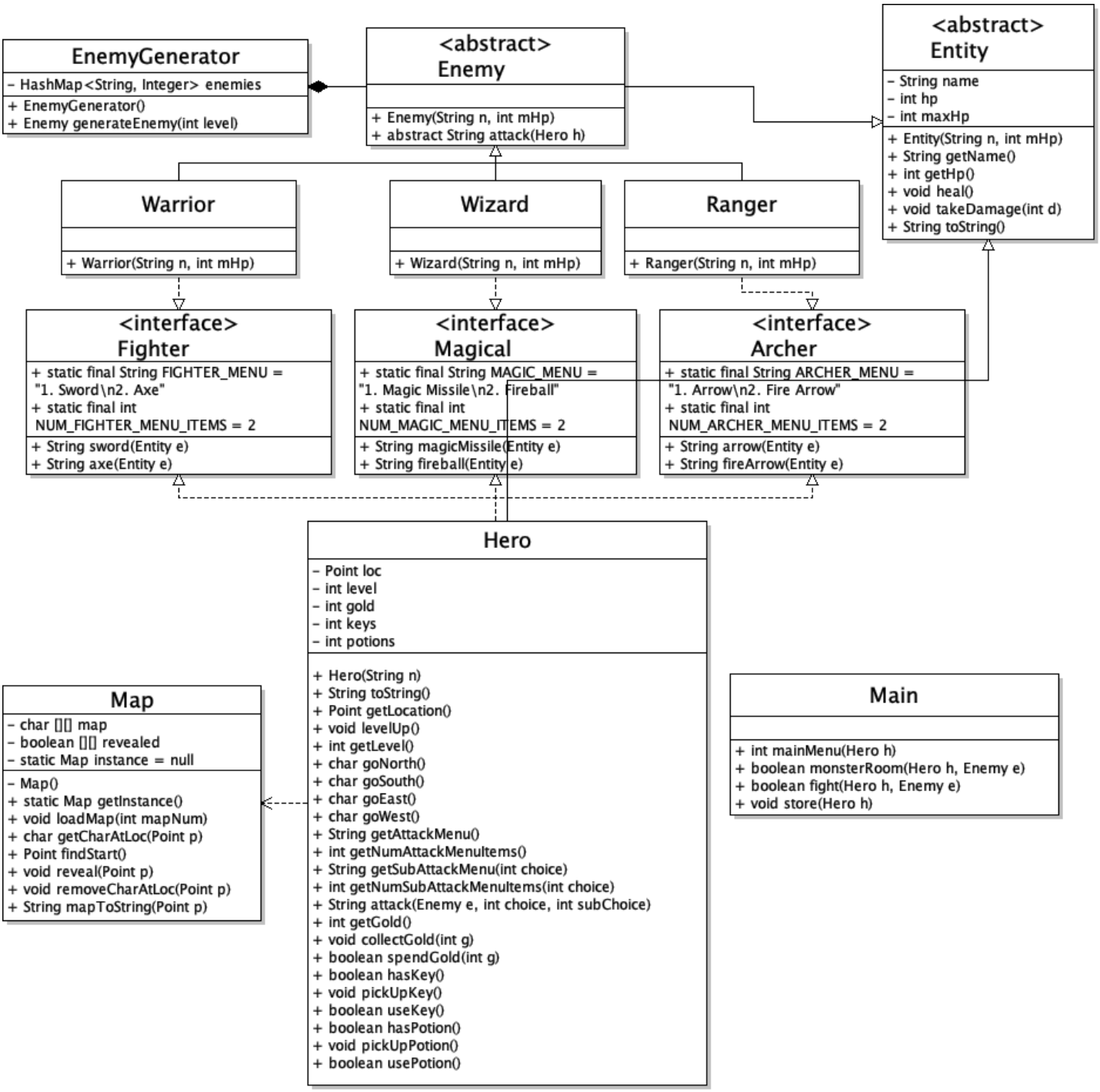
1. Entity – abstract - describes a character in the game
    a. an Entity has a name and some hit points (maxHp is the maximum amount of hp an entity can have and hp is initialized to the maxHp).
    b. heal method should reset the hp to the maxHp value (ie. does a full heal).
    c. takeDamage method should decrease the Entity's hp by the amount passed in, but it should never go below 0.
    d. toString method should display the name and hp over maxHp (ex. 15/25).
2. Hero – describes the character that represents the user.
    a. the Hero has a location on the map, a level, some gold, potions, and keys.
    b. The Hero is constructed with a name, begins at level 1 at the start position of the map and is given default values for hp, gold, potions, and keys.
    c. direction methods should update the Hero's location (if that location is within the bounds of the map), reveal that location, get the character at that location, 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 (each finish is the next map's start).
    e. attack method should call the selected ability method. Each does a different amount of damage to the enemy within a random range and returns a string representing that damage.
    f. attack menu method should return the string "1. Physical 2. Magical 3. Ranged" for the different types of attacks to choose from. Submenu method should return the selected menu from the interface.
    g. toString should display the name, hp, level, gold, potions, keys, and map.
3. Map – represents the dungeon maze. Map is a Singleton.
    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 – abstract - represents an enemy the Hero will encounter
    a. Abstract attack method will be overridden by the different enemy types.
5. Warrior/Wizard/Ranger – the different types of enemies.
    a. attack method randomly selects one of the enemy's abilities to attack the Hero with. Each of the ability methods for each enemy type should do random damage to the Hero and return a string representing that damage.
6. Fighter/Magical/Archer – interfaces that define the abilities of the Hero and the enemies.
7. EnemyGenerator – factory to create random enemies to encounter on the map.

a. constructor reads the file and adds the different enemies and their base hp to the HashMap (do not assume you know the length of the file).
b. generateEnemy method randomly selects an enemy from the map, then randomly selects an ability type (Fighter/Magical/Archer), then copies over the name and base hp to construct a new enemy of that type. Use the level value passed in to modify the base hp so that as the Hero progresses through the levels, the difficulty will increase.

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 – item store where the Hero can buy potions or keys.
        iv. f – finish – if the Hero has a key, then increase the Hero's level and load the next map.
        v. i – item – the Hero randomly finds a potion or a key.
        vi. m – monster – create an enemy to fight then call monsterRoom.
    d. monsterRoom displays the enemy and then repeatedly prompts the user to fight, run away, or drink a potion (only displayed if Hero has one). 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 encounter.
    e. fight does a single round of damage by allowing the user to choose to do a physical, magical, or ranged attack. Then, depending on their selection, displays the corresponding submenu. Those two selections are passed to the Hero's attack method to attack the enemy. The enemy then attacks back if it 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 you can make your own Point class.
- As the Hero moves in the map some encounters are removed once they are finished. Item rooms and defeated monsters are removed, start, finish, and monsters that were not defeated (ie. Hero ran away) are not removed.
- The ability damage methods (ex. sword or fireball) should each do a different amount of random damage to the entity that is passed in. Return a string representing the attack with the amount of damage done to the entity (see example output). The amount of damage to be done is up to you, but try to make it proportional to the Hero's mHp (ie. if the Hero only has 25 mHp, then the damage methods should probably have a random range of ~1-5 dmg). I need to test your programs, so please do not make it overly difficult for me to get to level 4.
- Please do not add any extra instance variables or methods to the UML.
- Ask questions about any methods you do not fully understand.


EnemyGenerator
- HashMap<String, Integer> enemies
+ EnemyGenerator()
+ Enemy generateEnemy(int level)
<abstract>
Enemy
+ Enemy(String n, int mHp)
+ abstract String attack(Hero h)
<abstract>
Entity
- String name
- int hp
- int maxHp
+ Entity(String n, int mHp)
+ String getName()
+ int getHp()
+ void heal()
+ void takeDamage(int d)
+ String toString()
Warrior
+ Warrior(String n, int mHp)
Wizard
+ Wizard(String n, int mHp)
Ranger
+ Ranger(String n, int mHp)
<interface>
Fighter
+ static final String FIGHTER_MENU =
"1. Sword\n2. Axe"
+ static final int
NUM_FIGHTER_MENU_ITEMS = 2
+ String sword(Entity e)
+ String axe(Entity e)
<interface>
Magical
+ static final String MAGIC_MENU =
"1. Magic Missile\n2. Fireball"
+ static final int
NUM_MAGIC_MENU_ITEMS = 2
+ String magicMissile(Entity e)
+ String fireball(Entity e)
<interface>
Archer
+ static final String ARCHER_MENU =
"1. Arrow\n2. Fire Arrow"
+ static final int
NUM_ARCHER_MENU_ITEMS = 2
+ String arrow(Entity e)
+ String fireArrow(Entity e)
Hero
- Point loc
- int level
- int gold
- int keys
- int potions
+ Hero(String n)
+ String toString()
+ Point getLocation()
+ void levelUp()
+ int getLevel()
+ char goNorth()
+ char goSouth()
+ char goEast()
+ char goWest()
+ String getAttackMenu()
+ int getNumAttackMenuItems()
+ String getSubAttackMenu(int choice)
+ int getNumSubAttackMenuItems(int choice)
+ String attack(Enemy e, int choice, int subChoice)
+ int getGold()
+ void collectGold(int g)
+ boolean spendGold(int g)
+ boolean hasKey()
+ void pickUpKey()
+ boolean useKey()
+ boolean hasPotion()
+ void pickUpPotion()
+ boolean usePotion()
Map
- char [][] map
- boolean [][] revealed
- static Map instance = null
- Map()
+ static Map getInstance()
+ void loadMap(int mapNum)
+ char getCharAtLoc(Point p)
+ Point findStart()
+ void reveal(Point p)
+ void removeCharAtLoc(Point p)
+ String mapToString(Point p)
Main
+ int mainMenu(Hero h)
+ boolean monsterRoom(Hero h, Enemy e)
+ boolean fight(Hero h, Enemy e)
+ void store(Hero h)

```
What is your name, traveler? Link
Link
HP: 25/25
Level: 1
Gold: 25
P: 1 K: 0
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 Wizard
HP: 3/3
1. Fight
2. Run Away
3. Drink Potion
1
1. Physical Attack
2. Magical Attack
3. Ranged Attack
1
1. Sword
2. Axe
1
Link slashes Kobold Wizard for 1 damage.
Kobold Wizard zaps Link with Magic
Missile for 5 damage.
Kobold Wizard
HP: 2/3
1. Fight
2. Run Away
3. Drink Potion
1
1. Physical Attack
2. Magical Attack
3. Ranged Attack
2
1. Magic Missile
2. Fireball
2
Link hits Kobold Wizard with a Fireball
for 6 damage.
You defeated the Kobold Wizard!
You find 3 gold on the corpse.
Link
HP: 20/25
Level: 1
Gold: 28
P: 1 K: 0
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 Key!
Link
HP: 20/25
Level: 1
Gold: 28
P: 1 K: 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: 20/25
Level: 1
Gold: 28
P: 1 K: 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 Froglok Warrior
HP: 1/1
1. Fight
2. Run Away
3. Drink Potion
1
1. Physical Attack
2. Magical Attack
3. Ranged Attack
1
1. Sword
2. Axe
1
Link slashes a Froglok Warrior for 4 damage.
You defeated the Froglok Warrior!
You find 5 gold on the corpse.
Link
HP: 20/25
Level: 1
Gold: 33
P: 1 K: 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 Warrior
HP: 4/4
1. Fight
2. Run Away
3. Drink Potion
2
Link
HP: 20/25
Level: 1
Gold: 33
P: 1 K: 1
x x x x x
x x x x x
s x x x x
n n n n x
x x * m x
1. Go North
2. Go South
3. Go East
4. Go West
5. Quit
3
You've encountered a Goblin Ranger
HP: 1/1
1. Fight
2. Run Away
3. Drink Potion
1
1. Physical Attack
2. Magical Attack
3. Ranged Attack
1
1. Sword
2. Axe
1
Link slashes a Goblin Ranger for 3 damage.
You defeated the Goblin Ranger!
You find 2 gold on the corpse.
Link
HP: 20/25
Level: 1
Gold: 35
P: 1 K: 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 find a locked gate.  Luckily you have
a key!  You proceed to the next area.
Link
HP: 20/25
Level: 2
Gold: 35
P: 1 K: 0
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: 20/25
Level: 2
Gold: 35
P: 1 K: 0
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
Welcome to the store.  What would you
like to buy?
1. Health Potion - 25g
2. Key - 50g
3. Nothing, just browsing...
3
Link
HP: 20/25
Level: 2
Gold: 35
P: 1 K: 0
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
```