CS-1180 Project 4: Catacomb Crawler

Inspiration: Brandon Walker - CS 1181 Game Jam - Spring 2022

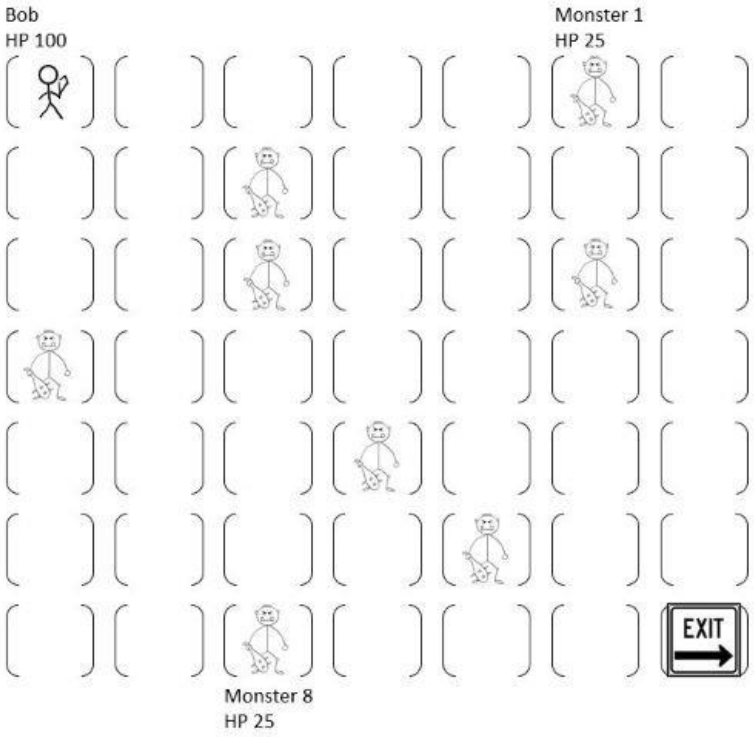
PURPOSE: Demonstrate competency with basic object-oriented design principles

PROBLEM: Your task is to implement a game called Catacomb Crawler.

The user chooses the name of their hero and the size of the catacomb N, where N is between 5 and 10 inclusive. The catacomb is an N-by-N square of rooms (for example, if the user chooses 7, then the catacomb has 7 rows with 7 rooms in each row, for a total of 49 rooms). The hero character starts out with 100 health in the room in the northwest corner of the catacomb. The goal of the game is for the hero to make it to the catacomb's exit in the southeastern room before they run out of health.

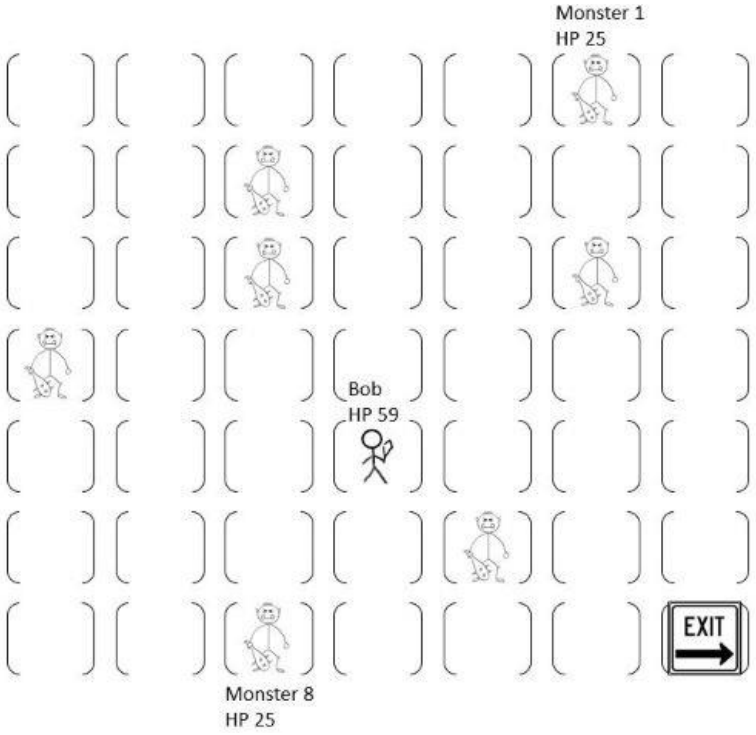
There is one monster in the catacomb for every six rooms (e.g., if there are 49 rooms, there are 8 monsters). The monsters are named Monster 1, Monster 2, etc. and start out with 25 health. Each monster's location in the catacomb should be determined randomly. No monster can start in the following locations:

- North-west corner, where the hero enters the catacomb.
- South-east corner, where the hero exits the catacomb.



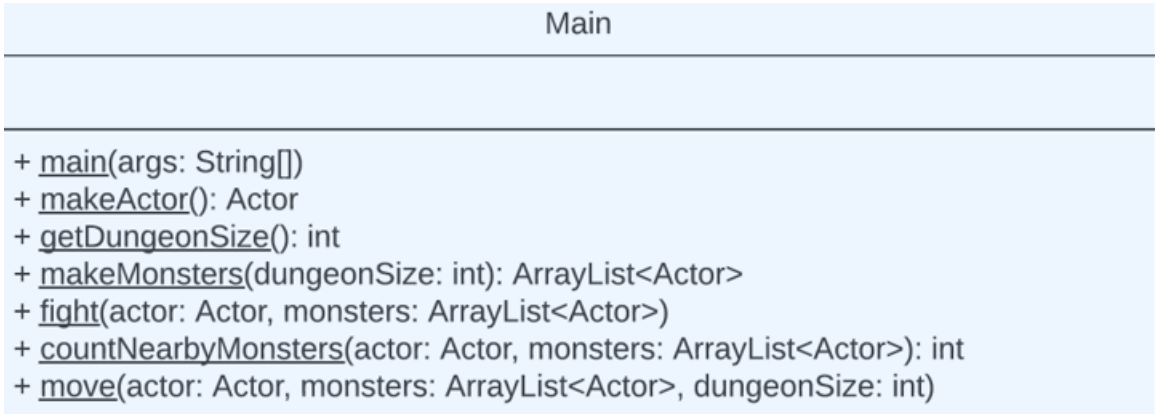
The user should be repeatedly asked which way they want to move (north, south, east, or west). User input should be case-insensitive (i.e. nOrtH ->north, etc.) The hero has an excellent sense of smell and can determine the number of monsters in adjacent rooms (i.e., rooms directly to the north, south, east or west of the hero's current position). The number of nearby monsters is given to the user before prompting them for the direction they want to move.

If the hero moves into a room that has one or more monsters in it, the hero fights each monster to the death, one at a time. The hero does between 1 and 10 hit points of damage (randomly determined) with each attack, and monsters do between 1 and 5 damage (randomly determined). The hero uses 2 health points just by moving between rooms. If the hero dies during a fight, the game is over. If a monster dies, they should be removed from the collection of monsters so that they cannot attack the hero again and the hero can no longer smell them.



REQUIREMENTS: Your program must implement good use of object-oriented program techniques and logical methods. Solutions that do not have more than one class or only use the main method in the driver class will receive a 0. Below are UML diagrams of an Actor class and Main driver class that you may refer to for building your solution.

Actor
- name:String - health:int - maxDamage:int = 5 - row:int - col:int
+ Actor() + Actor(char_name: String, health: int) + isAlive: boolean + hasEscaped(dungeonSize: int): boolean + inSameRoom(other: Actor): boolean + inAdjacentRoom(other: Actor): boolean + hit(other: Actor) + move(direction: String, dungeonSize: int): boolean + toString(): String



REQUIRED DOCUMENTATION: Refer to the JavaDoc Guidelines document in your Lab section in Pilot for the javadoc requirements for documenting classes & methods. Projects that do not contain javadoc will receive a 50% reduction.

In addition, you will use markdown syntax to create a companion document for your project. Note that when you create a Java Project in VSCode there is a file named README.md in the project folder. The .md means you can use markdown in this file. A link to a guide to using markdown is included in Project 4. Use Ctrl + Shift + V to preview what your markdown would look like using VSCode. Your README.md should contain a minimum of the following outline - fill in the outline with your own details about your project:

```
# Project Title

## Project Description
Describe what the purpose of the project is.  Include what it does in summary
and what programming concepts it reinforces

## Project Guide

### Dependencies
To run your code, what does a user need to have installed?

### How to run the project
If they have things installed, how can they run your project?

### How to play the game
Tell a user how to play your game.  Be sure to add special details if you
implemented something extra!

## Lessons Learned
Tell us what you learned along the way. Examples include:
- what concepts you practiced / better understood
- bugs you detangled
```

EXAMPLE: Below is an example of a run through the game.

```
What is your name, heroic adventurer? Bob
How wide of a catacomb do you want to face (5-10)? 500
That is not a valid catacomb size!
How wide of a catacomb do you want to face (5-10)? 5
Bob at 0, 0 with 100 health
You smell 0 monsters nearby.
Which way do you want to go (north, south, east, west)? potato
You can't move that way!
Which way do you want to go (north, south, east, west)? north
You can't move that way!
Which way do you want to go (north, south, east, west)? east
Bob at 0, 1 with 98 health
You smell 0 monsters nearby.
Which way do you want to go (north, south, east, west)? south
Bob at 1, 1 with 96 health
You smell 1 monsters nearby.
Which way do you want to go (north, south, east, west)? south
Bob at 2, 1 with 94 health
You smell 0 monsters nearby.
Which way do you want to go (north, south, east, west)? east
Bob at 2, 2 with 92 health
You smell 1 monsters nearby.
Which way do you want to go (north, south, east, west)? east
Bob at 2, 3 with 90 health
You smell 3 monsters nearby.
Which way do you want to go (north, south, east, west)? south
Bob at 3, 3 with 88 health versus Monster 2 at 3, 3 with 25 health
You hit for 1
You get hit for 2
You hit for 8
You get hit for 5
You hit for 7
You get hit for 4
You hit for 4
You get hit for 1
You hit for 3
You get hit for 4
You hit for 6
Monster 2 has been defeated!
Bob at 3, 3 with 70 health
You smell 0 monsters nearby.
Which way do you want to go (north, south, east, west)? north
Bob at 2, 3 with 68 health
You smell 2 monsters nearby.
Which way do you want to go (north, south, east, west)? south
Bob at 3, 3 with 66 health
You smell 0 monsters nearby.
Which way do you want to go (north, south, east, west)? east
Bob at 3, 4 with 64 health
You smell 0 monsters nearby.
Which way do you want to go (north, south, east, west)? south
You have escaped the catacomb
```

EXTRA CREDIT (ADDITIONAL GAME FEATURE): You can earn two points of extra credit on your overall grade in this course by developing an additional feature to the game. If you do this, explain the additional feature you have made as a comment at the top of your driver class. Please know that help room assistance will be limited for extra credit features, and primary features will be prioritized. If you are not sure that the extension you have in mind is "meaningful," check with a lab assistant before implementing it. Some ideas are:

- The hero can run away in the middle of a fight
- A monster has a random chance of running away in the middle of a fight if it is losing
- Some rooms in the catacomb have gold in them, and the hero's goal is not only to survive but to accumulate as much gold as possible
- Some rooms have potions in them that can restore health, but if the hero drinks the potion, they lose a turn
- The hero can find monster traps or weapons in rooms that can be used during fights to increase the amount of damage the hero does
- Display a map of the catacomb to show where the hero and monsters are in the catacomb

RUBRIC: Projects that don't compile will receive a 0. Use block comments for any non-compiling code. Projects that only have a main method or only use one class will receive a 0. Projects that do not meet javadoc guidelines for classes & methods will receive a 50% deduction.

(70 pts) **Base Functionality**

[15] The game is initialized as described, with the hero, a valid catacomb size, and the correct number of monsters randomly placed in the catacomb per the catacomb size

[15] The hero can move between rooms and invalid moves are rejected

[10] The hero is correctly warned of the number of monsters in adjacent rooms

[10] Heroes & monsters lose health per requirements when fighting or moving

[10] Dead monsters are removed from the catacomb

[10] The program recognizes when the hero has died or has succeeded in escaping the catacomb

(30 pts) **Style & Documentation**

[10] Program is cleanly organized using methods and classes

[10] README.md contains project documentation per outline in Required Documentation section

[10] The program is well-commented, using javadoc standards and meaningful comments

(20 pts) **Extra Credit** (20 points == 2% additional credit in the course)

[5] Create a comment at the top of you driver class that describes the feature

[5] Update user manual in README.md to include feature since it changes how the game is played

[10] An additional game feature is implemented in your game