Assignment 4: Hunt the Wumpus

Grading: EVERY assignment in this course is graded by demoing your work for 10 minutes with a TA. You are required to meet with a TA within two weeks from the due date to demo, and you receive an automatic 50 point deduction for failure to do so. If you miss a scheduled appointment, you will be penalized 10 points for rescheduling within 1 day (24 hours), 25 points within 7 days (1 week), and 50 points for anything outside of a week. Your job is to convince the TA that your program works correctly, i.e. show your TA how to use/break your program ☺

1 Hunt the Wumpus — Game description

The object of HUNT THE WUMPUS is to guide the adventurer, Jeremy Cole, to kill the Wumpus, find its hidden gold, and escape alive. The Wumpus lives in a large cave of rooms arranged in a grid, where each room at has most four tunnels leading to the rooms to the north, east, south, and west.

The adventurer starts the game in a random empty room in the Wumpus' cave. Their starting position is also the location of the escape rope that they will use to escape after they've completed their task.

There are four **events** that may be inside each room. Two of these 'events' are dangerous hazards, one is the terrifying Wumpus, and the other is the gold treasure. When the adventurer is in a room that's adjacent to a room containing an event, the player may receive a percept (a message) to inform them about the events they're close to.

The Wumpus' cave contains two of each hazard type in random locations. There can only be at most one event or hazard in each room. The location of the gold will also be random, and it cannot be in the same room as a hazard.

If the adventurer perishes while searching for the Wumpus, the player should be presented with the option to start the game over with the same cave configuration, start the game over with a new random cave configuration, or quit the game entirely.

The player wins the game if they can safely guide the adventurer through the cave to kill the Wumpus, pick up its gold, and make it back to the escape rope unharmed!

1.1 Adventurer:

Each turn you may take one of two actions to guide the adventurer Jeremy Cole:

Move: You can move through a tunnel to an adjacent room.

Fire an Arrow: Jeremy Cole has three arrows. You can aim each arrow by firing it into an adjacent room. The arrow continues flying in the same direction until it hits a wall or travels through three rooms. If the arrow enters the Wumpus' room, it pierces the Wumpus' heart and kills the monster. If Jeremy Cole runs out of arrows without having killed the Wumpus, you lose.

1.2 Wumpus:

Usually, the Wumpus is peacefully asleep in his cave. Two things wake him up, however– the adventurer entering his room, or shooting an arrow and missing the Wumpus. If the Wumpus wakes up while in the same room as Jeremy Cole, the Wumpus will angrily eat him, ending the game. If the Wumpus just wakes

up from hearing an arrow being fired, though, there's a 75% chance the Wumpus will move to a random empty room in the cave to avoid being found.

1.3 Hazards:

Bottomless pits: Two of the rooms have bottomless pits in them, if Jeremy Cole goes there, he falls into the pit, dies, and you lose. Rough day for Jeremy Cole.

Super bats: Two rooms have super bats. If Jeremy Cole enters a room that contains super bats, an angry bat grabs him and takes him to some other room at random (which could be dangerous!).

1.4 Gold:

The gold is a treasure sitting in a room that does not contain a hazard. If Jeremy Cole is in a room containing gold, he automatically picks it up and takes it with him.

1.5 Percepts:

When Jeremy Cole is in a room **directly adjacent to** to a room containing an event, the player receives the following messages:

Nearby Event	Message
Wumpus	"You smell a terrible stench."
Super Bats	"You hear wings flapping."
Bottomless Pit	"You feel a breeze."
Gold	"You see a glimmer nearby."

Notice that there's no percept for the escape rope! That means the player will have to remember where they started and find their way back to that tile after they've completed their task.

For example, if the adventurer was standing in an empty room with the Wumpus one room to the North, the Gold one room to the East, and Bats *two* rooms South, they would recieve the following messages at the beginning of their turn:

```
You smell a terrible stench.  You see a glimmer nearby.
```

Remember, the percepts don't tell you *where* the hazard or gold is– just that it's somewhere close!

2 (90 pts) Program Implementation

Requirements:

1. Your program should allow the user to play Hunt the Wumpus as described above.
2. The cave the game takes place in should be a square grid. The size of the grid (i.e. number of rooms on one side of the square) should be specified as a **command line parameter** to your program. Caves smaller than four rooms on a side aren't allowed.
3. Your code must have at least the following classes: **Room, Event, Wumpus, Bats, Pit, Gold**. You can create more classes if they would be helpful.

4. Wumpus, Bats, Pit, and Gold should all be child classes of the abstract (pure virtual) **polymorphic** Event class. Any event does something when the adventurer enters the same room as the event, and may display a message when the adventurer is nearby. Your design of the Event class should reflect this.

5. Each Room **has** at most one associated Event. Again, the design of your Room class should reflect that fact.

6. Your program should have no memory leaks.

7. **Extra credit (10 points)**: Instead of just allowing the game to be played by a human, design in such that you can create an **AI class** that plays the game for you. This AI class should use the same interface to the game that the player does– that is, they use percepts to learn about the world and make decisions.

3 Hints:

1. Polymorphism only works when you are working with references or pointers. If you use the *value* of an object directly, it may not select the correct member function.

2. Hunt the Wumpus is a game all about hiding information from the player– which might make it hard to debug! To get around that problem, you might want to create a debug-only map display so that you can tell exactly what's in the cave while you're testing your program. Just make sure you remove it or disable it before submitting the assignment!

3. Hunt the Wumpus is a very common introductory programming project. This assignment, however, is designed to specifically include the programming concepts you've seen in class so far. It's very likely that other versions of Hunt the Wumpus you could find do not correctly implement what's described above.