

Problem Decomposition Revisited (Again and Again...): Object Oriented Design

There's more...?



Outline

- **Object Oriented Design: Now that we know the classes**
 - Design the methods



Software Development Life Cycle

1. Understand the Problem = Requirements Analysis
2. Work out the Logic = Design
3. Convert it to Code = Implementation
4. Test/Debug
5. Maintenance

Today we will start to analyze and design the solution to a program.

The Problem – For Reference

- **The Game.** The Wumpus World game takes place in a cave with different rooms in it. You can think of the cave as an $N \times M$ rectangular grid. The player always starts in position 0,0, which is guaranteed to be safe (but it may still be smelly or breezy or glittery).

The objective of the game is to find the gold. The player will know when he/she is in a room with the gold because there will be a "glitter" in that room. If the player detects a glitter, he/she can pick up the gold and the game is won.

Bottomless pits are present in some of the rooms. There is a 20% chance that any given room will have a pit. All rooms adjacent to a pit are breezy, that is, a player entering a room adjacent to a pit will detect a breeze. If the player moves into the room with a pit, he/she falls in and dies a horrible death.

There is only one wumpus in the cave, and he is also placed at random. Rooms adjacent to the wumpus are smelly, that is, a player will detect a stench in a room adjacent to a wumpus. The wumpus cannot move. If the player enters a room with the wumpus, he/she will be eaten, and, once again, die a horrible death.

There is also only one room in the cave that contains the gold. Unlike the other objects, the player has to be in the same room as the gold in order to detect a glitter. Like the wumpus, the gold is placed at random.

The player can move up, down, left, or right. The player also has one arrow. Once it's used up, it's gone. It can be used to shoot a wumpus, and can be shot in any direction the player can move in. If the player is successful in shooting the wumpus, the wumpus will emit a blood-curdling scream, and will no longer be a threat. The only other action the player can perform is to "grab gold".

When the player first starts the game, he/she does not know (and cannot see) where the location of pits, gold and the wumpus are. The only clues are whether the current room is breezy, smelly, or glittery.

Cave
rooms
width
height

20% of rooms
are pits

Wumpus location

Gold location

draw

initialize itself ←

toString

Player

x, y position

arrow

move

draw

init

shoot

grabGold

getX

getY

getArrow

Room

visited or not
breeze
stench
gold
wumpus
pit

draw
toString
setConditions
getConditions
init

Class Cave

draw(self)
--init-- (self, width, height)
string toString(self)

Class API's

Class:	Returns	Method	Parameters	Description
Room		__init__		Construct a room
		setCondition	condition, value	Set the condition of a room (e.g. breezy, has a Wumpus, etc.)
	boolean value	getCondition	condition	Returns the value of a room condition
		draw	image size, x, y	Draws the room image with contents at the size and x, y location
Cave		__init__	width, height	Constructs an NxM cave and adds all objects (pits, breezes, etc.)
		draw		Draws the cave with all its rooms
Player		__init__		Constructs a player
	int x	getX		Returns the x location of the player
	int y	getY		Returns the y location of the player
	boolean	getArrow		Returns whether the player has an arrow left or not
	boolean	move	height, width, direction	Changes players location and returns True if successful, False otherwise
		shootArrow		Removes the arrow from inventory
		draw		Draws the player at his/her current location

Recap: Important Methods to Have

- Constructor
- Accessors (Getters)
- Mutators (Setters)
- Equals
- toString

Summary

- **Object Oriented Design**

- Identify the classes
- Identify what information each class needs
- Identify what each class needs to do
- Identify use cases
- Define the API
- Define the instance variables
- Finally – write some code!

