Barbie vs Zombies

CRC Cards :

Class: Game	
Responsibilities:	Collaborators:
Print locations of every object. Print health of Barbie. Check if Barbie is surrounded by Zombies. Check if Barbie stepped on a pear. Print a welcome message. Take user inputs to play the game. Check if the user won or lost the game. Start the game.	Barbie Zombie **Pear**

Class: Pear	
Responsibilities:	Collaborators:
Obtain current row and column. Print current location.	Location

Class: Barbie	
Responsibilities:	Collaborators:
Move. Check Health. Edit Health. Set Health. Print current location. obtain current row or column. Check if Barbie's dead.	Location

Class: Zombie	
Responsibilities:	Collaborators:
Move randomly. Print current location. obtain current row and column.	Location

Class: Location	
Responsibilities:	Collaborators:
Set the position of an object. Get the position of an object. Change the position of an object.	

Design Decisions:

First off, I created my four classes:

- Location
- Zombie
- Barbie
- Pear

The Location class was created later, when I realized, instead of using two instance variables in each class for holding rowPosition and columnPosition, I can create a separate class for Location of each object. My main idea was giving location to each object (Barbie, Zombies, Pears) in a fixed size map, in this case I decided on 5x5, and used these locations to compare objects with each other to see if Barbie satisfies the game winning conditions or not. Zombie, Barbie, and Pear classes share similar methods, and they all use Location class for determine, and edit their positioning on the map. Starting of the Game class, we create List of zombies and List of pears to put on our map, and one Barbie for user to play. I did not want to randomize the zombie and pear locations, to find the best strategy to beat the zombies in this version.

I did not add extra internal walls inside the 5x5 grid. Only walls are the ones blocking the user from leaving the game zone.

Barbie starts the game from the middle of the map (2,2), with 5 health. The four zombies are located on each corner of the grid. Pear's locations are fixed. Every time we run the game; the pears will be on the same spot as well as zombies. Zombie's move randomly, with the help of Random class. Pears don't move.

Starting of the game there will be:

- Welcome message.
- information of the Game.
- Barbie's location
- All four Zombie's location.
- Location of pears.
- Command words user can enter.
- Warning message about wrong command words or trying to leave the map.

After the first move and rest of the moves, there will be:

- Information message If a pear is picked up.
- Barbie's location
- Zombie's location
- Barbie's health
- Remaining pears' location.

Rules:

- If one or more zombie's move a square next to Barbie, meaning right left up or down of you, not diagonal, Barbie loses one health for each zombie.
- If a zombie and Barbie both move to the same square, zombie eats barbie, game over.
- If Barbie steps on a pear, she gains one health. Pear disappears from the map.
- If Barbie tries to leave the map (by hitting a wall), or enters the wrong command, her location does not change, but the zombies will move.
- Goal is collecting all the pears without dying.

I added additional comments on each of the classes and methods I used on this project.

Lastly, here is a visualization of the map for the beginning of my game.

Z(1,2,3,4) = Zombies

Z(1)				Z(2) Pear
	Pear			
		Barbie	Pear	
Pear				
Z(3)		Pear		Z(4)