

Lab Assignment #2: The Game of Life Refactored

Due Date: 12 April, 2016 @ 6pm

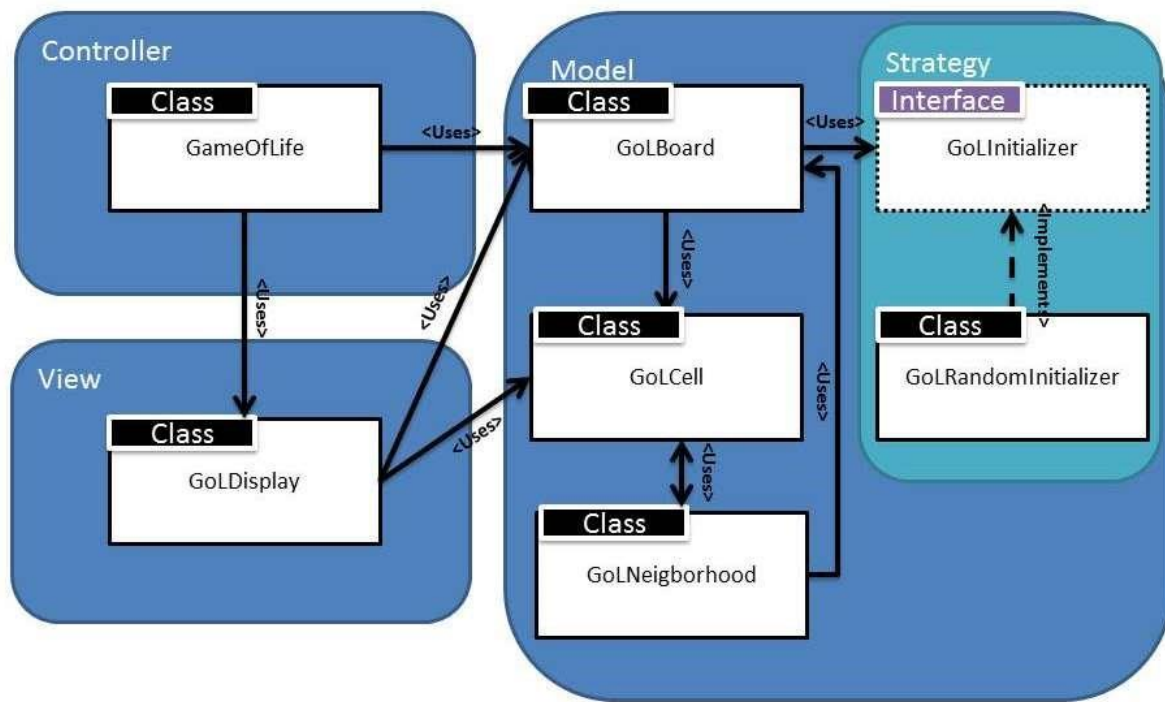
Instructions for Submitting the Lab

For this lab assignment, the completed lab will consist of an electronic copy of the Java code file you develop. The completed code file should be submitted to the instructor via Blackboard **before the end of class** on the due date.

This lab will be graded based on the following factors:

- Commenting: Are all classes, methods and variable declarations (both properties and private variables) documented with inline comments
- Functionality: does the submitted code compile and run? Does it meet the requirements given below?

In this lab assignment, you will be refactoring your previous GameOfLife application to implement the Model-View-Controller pattern and a Strategy pattern according to the design below:



Required Classes and Interfaces (see diagram above)

Class GameOfLife

This is the executable controller class for the Game of Life simulation. It should initialize other objects and control stepping through the rounds, calling the appropriate methods in both the GoLBoard class and the GoLDisplay class. It should not contain any code related to updating cells, etc.

Method	Returns	Description
main	void	Entry point for the program

Class GoLBoard

This class implements a model for the game board.

Method	Returns	Description
constructor ()	n/a	Calls the overloaded constructor below with the default GoLRandomInitializer
constructor (GoLInitializer myInitializer)	n/a	Created a new GoLBoard initialized by the provided initializer
nextRound()	GoLBoard	Updates the board for the next round by calling the updateCell method of the individual cells, passing a GoLNeighborhood to them. Returns the updated board.
getCell(int x, int y)	GoLCell	Returns the cell at the specified coordinates
copyBoard()	GoLBoard	Returns a copy of the current board
reset()	GoLBoard	Re-initializes the board
reset(GoLInitializer myInitializer)	GoLBoard	Re-initializes the board with the provided GoLInitializer
getCurrentRound()	int	Returns number of the current round
getBirths()	int	Gets number of births from last round
getDeaths()	int	Gets number of deaths from last round

Class GoLCell

This class implements a model for an individual cell.

Method	Returns	Description
constructor (boolean)	n/a	Creates a new cell that is alive if the parameter is true or false otherwise
updateCell(GoLNeighborhood myNeighborhood)	boolean	Updates the cell using the given neighborhood and returns true if alive and false if dead
isAlive()	boolean	returns true if alive and false if dead

Class GoLNeighborhood

This class implements a model for the neighborhood of an individual cell.

Method	Returns	Description
constructor (GoLCell[][] cellNeighbors)	n/a	Creates a new neighborhood using the 3x3 array provided
getCell(int x, int y)	GoLCell	Gets the cell at coordinates x,y in the neighborhood

Interface GoLInitializer

This class implements an interface for initializing a board.

Method	Returns	Description
getCellArray()	GoLCell[][]	Returns an array of cells

Class GoLRandomInitializer

This class implements an interface for initializing a board.

Method	Returns	Description
constructor ()	n/a	Creates new GoLRandomInitializer with a random seed
constructor (long seed)	n/a	Creates new GoLRandomInitializer with a given seed
getCellArray()	GoLCell[][]	Returns an array of cells

Class GoLDisplay

This class implements a model for the neighborhood of an individual cell.

Method	Returns	Description
constructor ()	n/a	Creates a new display object
displayBoard(GoLBoard gameBoard)	void	Outputs the current board along with round number, births and deaths