# Lab 8 - Concentration


---

## Introduction

Concentration is a card game in which all of the cards are laid face down on a surface and two cards are flipped face-up each turn. The object of the game is to turn over every pair of matching cards.

In this lab you will enhance a text-based Concentration program by creating a graphical user interface (GUI) for the game so that a user can play a single person version of the game by clicking on buttons.

---

## Preparation

Study the lectures on JavaFX regarding layouts, controls, event handling and model-view-controller.

---

## Problem Solving Session

You will work in teams for this portion of the lab.

When creating a program with an interactive GUI, it is good practice to make use of the Model-View-Controller (MVC) design pattern. Recall that the model is the object that contains the data and methods central for solving the problem, in this case playing the game. Now you will focus on the game model.

1. What state must the Concentration game model have in order to play the game? Note: User interface state should not appear in the model.

2. What behaviors must the Concentration game model have in order to play the game? Note: User interface methods should not appear in the model.
3. Write pseudo-code for the Concentration game model class that you have outlined above. Do you find that you need more state than you expected?
4. Outline how methods could be written for the following features.
    - Undoing a recent card flip (i.e., flipping a card that is face-up back down).
    - Resetting the entire game.
    - Cheating by showing where all the cards are.

---

# Implementation

You will complete this lab on your own.

## Java Documentation/Code

You are provided with a complete implementation of the model and a text-based view-controller. The provided classes should not be modified.

- `CardFace` class - `CardFace.java` java code
- `Card` class - `Card.java` java code
- `CardBack` class - `CardBack.java` java code
- `ConcentrationModel` class - `ConcentrationModel.java` java code
- `TextViewControl` class - `TextViewControl.java` java code

The text-based game runs as follows from a terminal window. Note the package structure of the provided code.

```
$ java -cp . term.TextViewControl
```

You will implement the following:

- `Concentration` class
- other classes for the GUI that you may define, if any.

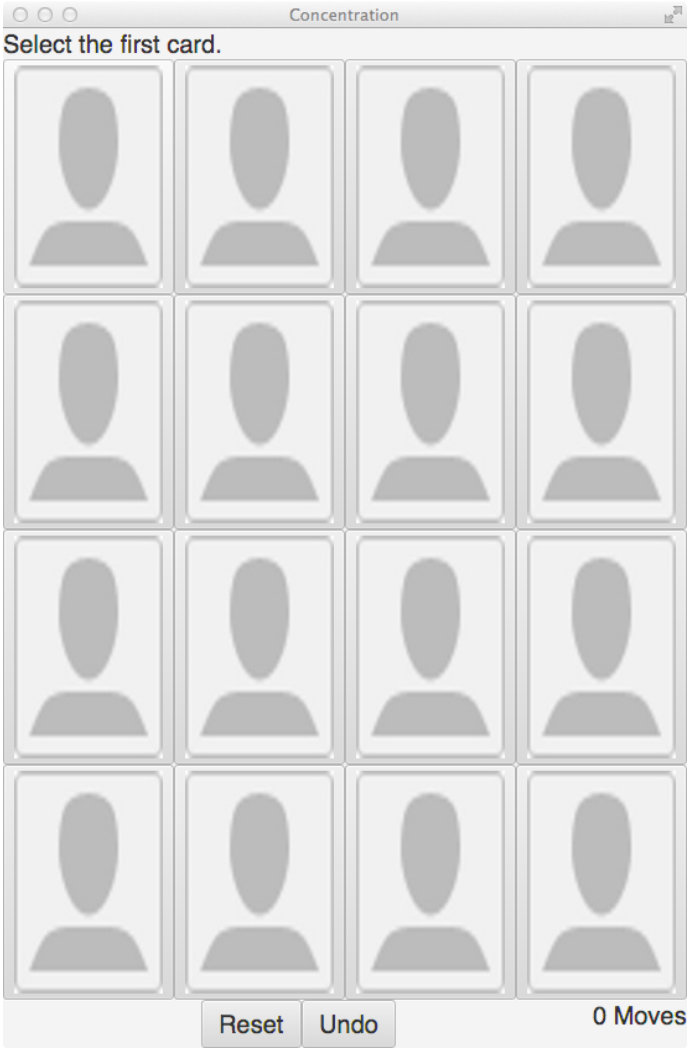
The GUI-based game should be run as follows from a terminal window.

```
$ java -cp . gui.Concentration
```

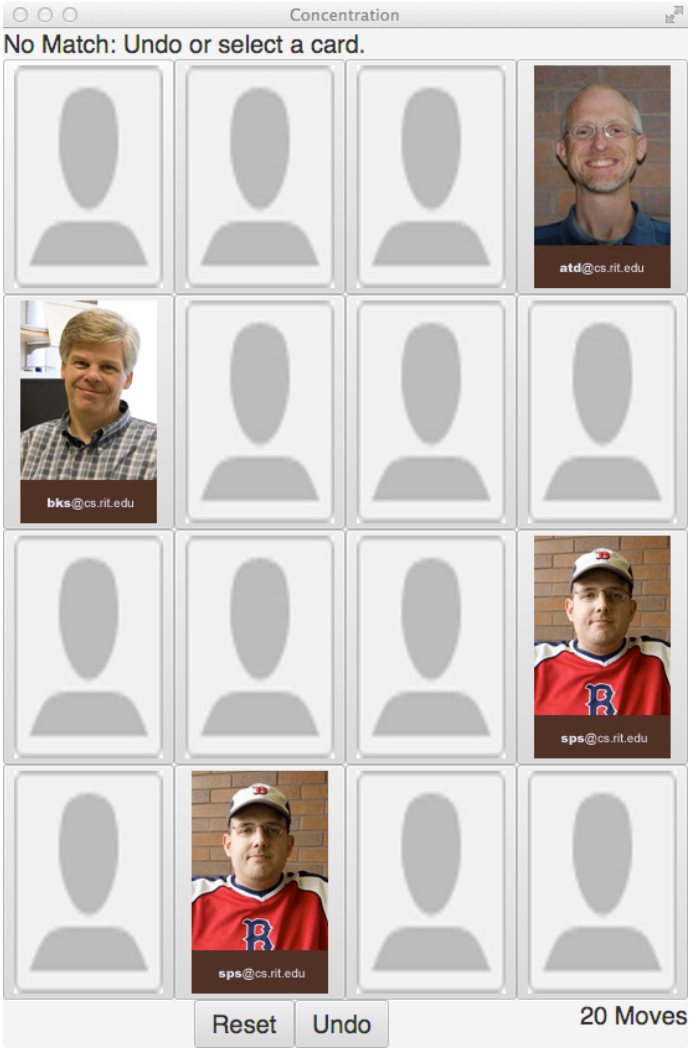
## Description

In any GUI application, there are two main concerns: how to layout the components of the GUI to make it appear as you want, and what functionality to attach to the components to make it behave as you want.

In this lab, we will specify both the look and behavior, but not the details of how these should be achieved. (Of course much of the behavior will be determined by the model.) The GUI itself should look approximately like this. The exact look will depend on your operating system.




Initial puzzle




Puzzle in the middle of play

The first thing you will need to do is decide how you will lay out the GUI to achieve this look. What layouts will you use?

Your UI must contain the following parts:

- A title bar with "Concentration" as the title;
- A message area that tells the user what move it is and what they can do next;
- The grid with all the cards (the model has the images);
  - Each card has an image and a number and matching cards have the same image and number.
  - Card displays that are initially hidden, which in the example means having an 'empty head' image.
- A reset button that turns all the cards face-down and shuffles them.
- An undo button that turns the most recent card selected back down unless it was already matched with its pair.

The user will use the mouse to make selections of cards; each card selected should be exposed to the user.

## Implementation Hints

Before you begin coding, make sure you understand how the class ConcentrationModel works. The main

things to consider are the UI layout, how you will display the cards, and how the UI will interact with the model.

Try to make the grid look as similar as possible to the examples.

---

# Submission

You should zip all your java files together, both the ones that you wrote as well as the ones that we provided. Submit the zip to the dropbox on MyCourses. (See MyCourses for deadlines for on-time and late submissions. The syllabus describes the penalty for submitting to a late drop box. No submissions are accepted after the late deadline passes.)

---

# Grading

The grade breakdown for this lab is as follows:

- Problem Solving Session: 15%
- Lab attendance and participation: 5%
- Implementation: 80%
  - *Note!*: Zero credit for submissions that do not compile!
  - Design and Functionality (MVC and game play): 70%
    - Operation of card picking: 5%
    - Operation of reset: 10%
    - Operation of undo: 10%
    - Operation of message updating: 5%
    - Concentration GUI class layout: 20%
    - Event-handling implementation: 20%
  - Code Style: 5% (see style guide and example)
  - Documentation: 5%

---

Created April 9, 2015