Faculty of Arts & Sciences
Department of Computer Science
CMPS 200—Introduction to Programming
Assignment 8 – Due Friday Nov 16, 2012

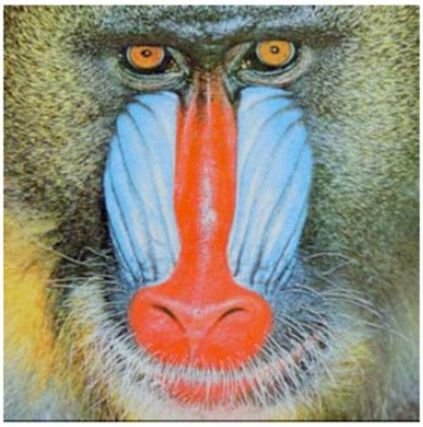
Notes and Announcements

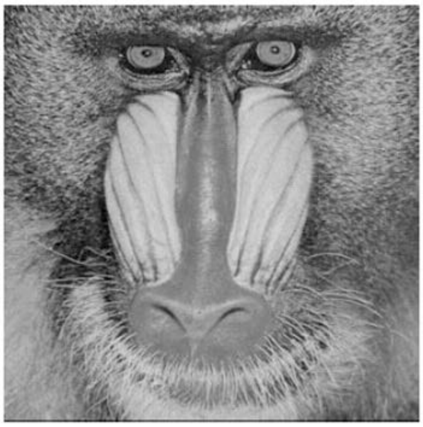
- Reading material for this week: Color and Picture classes; More on the String class.
- Reminder: Exam 2 will take place on Saturday Nov 17, 9:00-11:00am. Exam will be closed book.
- You may use your own laptop during the exam, but you have to make sure it has enough battery power to last you for the whole exam period.
- We will have a "Programming Clinic" open to all on Friday 2:00-5:00pm in Bliss 209.

Exercises

1. GrayImage

Write a program `GrayImage.java` that takes the name of an image as a command line argument and displays a grayscale version of the image. You may use the *monkey.jpg* image (posted on moodle) to check your program.





2. Smooth

Write a program `Smooth.java` that takes the name of an image as a command line and displays a smoothed version of the image where every pixel color is replaced by the average of the colors of its 4 neighbors.

3. PulseHeart

Write a method:

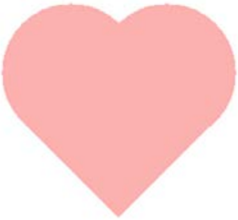
```
public static void heart(double size, Color c)
```

that draws a heart-shaped figure similar to the one below (you can draw the shape as a diamond with two circles to the upper left and upper right sides). `size` refers to the diameter of the circular part of the shape, and c is the color used for the drawing.

Write a method:

```
public static void pulseHeart(double size)
```

that draws a heart with its color changing every 100 milliseconds. You may want to change the color continuously (say the color goes from blue to red in 10 steps, so that the animation cycles every second).



4. Similar

Write a program Similar.java that take two file names as command line arguments and prints true or false depending on whether or not the files have the same contents. Your program should check the files character by character and ignore case differences (upper/lower case) in the characters that are letters. Write appropriate test files to check your program.

5. RemoveDuplicates.

Write a program `RemoveDuplicates.java` that takes three file names as command line arguments. The first two files contain lists of integers (the first line of each file contain the number of integers in that file). Your program should write to the third file the union of the numbers in the first two files (without any duplicates). Hint: Store the contents of the files in arrays first.

6. TicTacToe

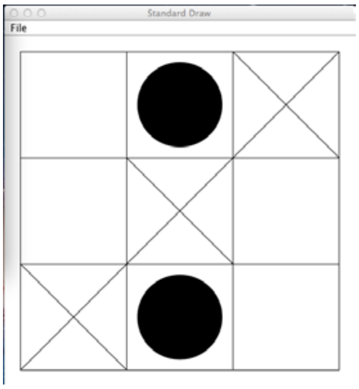
Tic-Tac-Toe is a two-player game played on a 3 by 3 grid. Players (represented by X and O) take alternating turns trying to place three of their letters in a line horizontally, vertically, or diagonally. The game is won by the player who completes three-in-line. The game may end in a tie if the entire board fills up with no player completing such a line. You are to write a TicTacToe game simulation where a user plays against the computer.

- We will assume the cells of the grid are labeled from 0 to 8 and stored in an array. The entries of the array may be 0 (empty cell), 1 (selected by user); or 2 (selected by computer).
- The program chooses randomly whether the user or the computer starts.
- If it is the user's turn to play, he/she will enter an integer from 0 to 8 to indicate the cell selected. (the program should make sure the selected cell is empty and ask the user to re-enter a value in case the chosen selection is not possible).
- If it is the computer turn, a cell among the available should be chose. Your program may use any strategy to choose the cell. For example, a random integer between 0 and 8 may be generated until an empty cell is chosen. You may also use a more elaborate strategy.
- Below are the declarations of some methods you may want to write to decompose the program into manageable pieces.

```
// Reads an integer from the user and marks the corresponding entry in the cells array.
// Returns true is user wins after cell is selected
// Method keeps getting user input until a valid cell is selected
public static boolean userTurn(int[] cells)

// Pick a cell from the available ones and marks it in the array cells.
// Returns true is user wins after cell is selected
public static boolean computerTurn(int[] cells)

// Checks if the array cells has a winning line for user (id=0) or computer (id=1)
public static boolean checkWin(int[] cells, int id)

// Draw the current state of the game on a canvas
public static void drawGrid(int[] cells)
```

An example invocation of the game might produce:

```
C:> java TicTacToe
Enter a cell: 0
The computer selects cell 1
Enter a cell: 1
Already selected
Enter a cell: 4
The computer selects cell 7
Enter a cell: 8
Congratulations. You win!
```


Standard Draw
File