CSCI1530 Computer Principles and Java Programming 2014-2015 Second Term
Department of Computer Science and Engineering
The Chinese University of Hong Kong

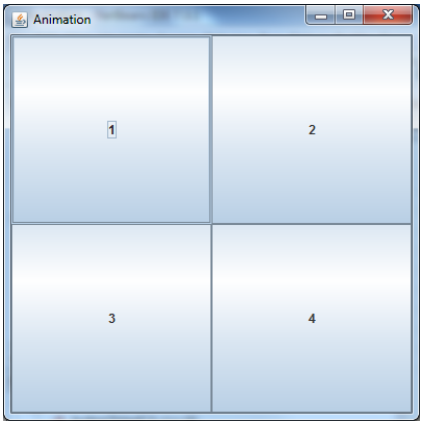
Due date: 20 March 2015 (Fri) **Assignment 4** Full mark: 100
Expected normal time spent: 5 hours

Animation Game (扑傻瓜)

Aims: 1. Practising flow control.
2. Processing String using the API methods.
3. Basic Graphical User Interface (GUI) programming.

Requirements:

1. We are going to implement an interesting animation game using Java. The program asks the user for an animation sequence such as 12343241223 and blinks four graphical buttons (labelled 1, 2, 3 and 4) accordingly. When a button is highlighted, the user shall attack (click) the button.



2. The program firstly creates and initializes the game animation window as shown above.

In a new class **`Animation`**, define the method **`public static void main(...)`**:

```
// window creation:
JFrame window;
window = new JFrame("...");
window.setSize( 400, 400 );
window.setLayout( new GridLayout(2, 2) );
window.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );

// create a new object for tracking button clicks
ButtonTracker act = new ButtonTracker();

// create and place the buttons properly
JButton label1 = new JButton("1");
label1.addActionListener(act);
window.add(label1);
// create, prepare and add other buttons to the window
// ...

window.setVisible(true);

// read user animation sequence input (for many rounds until user dismisses)
// ...
```

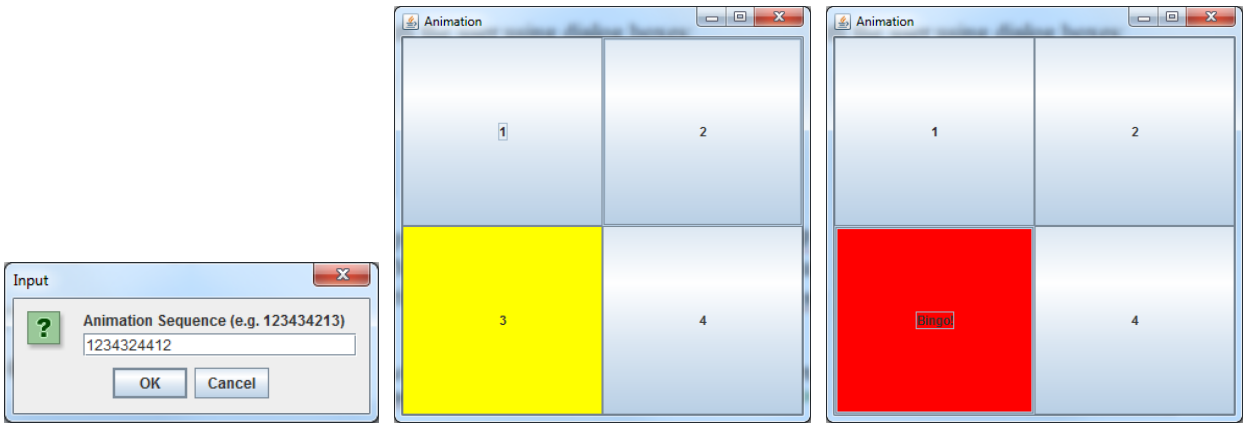
```
// for each digit in the input sequence
//   blink the corresponding button at 1 second intervals, e.g.

label1.setBackground(Color.yellow);  // set button color to yellow

// wait for ~1000 milliseconds by checking System.currentTimeMillis() repeatedly

label1.setBackground(null);  // restore the original color of the button
label1.setText("1");         // restore the original text label of the button
```

After setting up the window in method **`main()`**, we have to prepare another method for handling user clicks on the buttons. The **`ButtonTracker`** object will deal with this method internally.

In method **`public static void buttonClicked(JButton b)`**:

```
System.out.println( b.getText() );   // you shall see the clicked text label

Color c = b.getBackground();

// compare c with Color.yellow, see if the clicked button is highlighted or not

b.setBackground(Color.red);   // show red on Bingo!

b.setText("...");             // show Bingo! (text will be restored in main())
```

3. The program interacts with the user using an input dialog box as well as a window:



4. Animation sequence is entered as a String through a JOptionPane input dialog box. We assume the user input is always valid.

```
String input;
// pops up an input dialog
input = JOptionPane.showInputDialog( "..." );

if (input == null)    // it means the user dismissed the dialog box
      System.exit(0);
```

5. Using String methods, we can extract and inspect each of the characters in the input, thus running the animation in the window accordingly.

6. The program generates delay by recording and examining the System current time (in ms). One second is equivalent to 1000 ms. Note that timing and comparisons need not be exact.

7. The user input dialog and the animation all happen in the method **`main()`** in the class **`Animation`**. Responses to the user clicks (i.e., Bingo! on red) shall happen in the method **`buttonClicked()`** in the class **`Animation`**.

8. In addition to your own work in the above class **`Animation`**, you also need this provided class **`ButtonTracker`**:

```
// package declaration and some more import statements to be fixed by NetBeans...
package game;
import static game.Animation.buttonClicked;

// this is a given class, you shall just adopt it as is without any modification:
class ButtonTracker implements ActionListener {
    @Override
    public void actionPerformed( ActionEvent eventObject ) {
        buttonClicked( (JButton)( eventObject.getSource() ) );
    }
}
```

9. Demonstration and sample program (in the form of a web-based applet):
   http://www.cse.cuhk.edu.hk/csci1530/asgdemo/AnimationDemo.html
   Your work should be a stand-alone Java application, not an applet.

Your Task:

1. Before you code, draw your own flow chart and think twice.

2. **Create** a new NetBeans project named **`Animation`**, with a package named **`game`** and main class **`Animation`**.

3. The main class should include a proper *header comment block*, similar to the one appeared in assignment 1. It should include course code and course name, title of the assignment, brief description of your work, your name, your SID, date of the work, as well as your statement of originality and declaration of understanding the guideline on academic honesty.

4. Zip and Submit your *whole NetBeans project folder* in an archive file **`Animation.zip`** via our Online Assignment Collection Box on Blackboard <**`https://elearn.cuhk.edu.hk`**>

5. Enjoy your game!

Marking Scheme and Notes:

0. We will test playing your program vigorously for proper game features. So do you!

1. The submitted program should be free of any typing mistakes, compilation errors and warnings.

2. Comment/remark, indentation, style are under assessment in every programming assignments unless specified otherwise. Variable naming, proper indentation for code blocks and adequate comments are important.

3. Remember to do your submission before 6:00 p.m. of the due date. No late submission would be accepted.

4. If you submit multiple times, **ONLY** the content and time-stamp of the **latest** one would be counted. You may delete (i.e. take back) your attached file and re-submit. We ONLY take into account the last submission.

University Guideline for Plagiarism

Attention is drawn to University policy and regulations on honesty in academic work, and to the disciplinary guidelines and procedures applicable to breaches of such policy and regulations. Details may be found at http://www.cuhk.edu.hk/policy/academichonesty/. With each assignment, students are required to submit a statement that they are aware of these policies, regulations, guidelines and procedures.

Faculty of Engineering Guidelines to Academic Honesty

MUST read: http://www.erg.cuhk.edu.hk/erg-intra/upload/documents/ENGG_Discipline.pdf (VPN required if you are not in the CUHK campus network)