COMP2396B (OOP & Java)
Tutorial # 3
(Java GUI)

Java GUI exercise

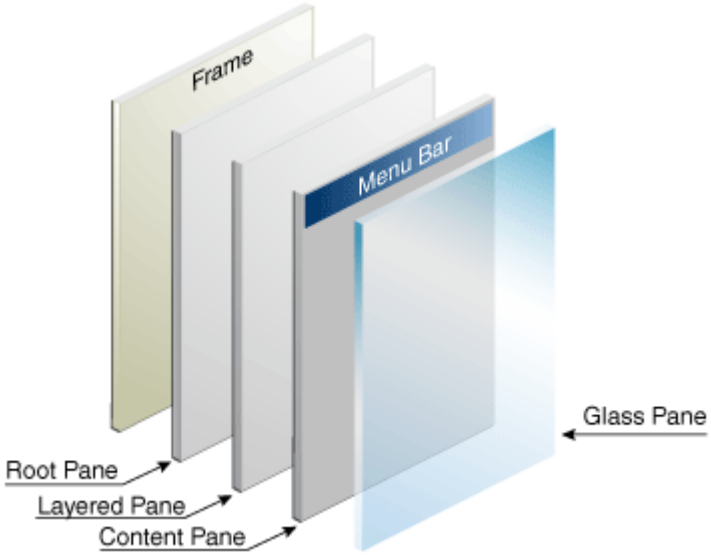
In this Lab exercise, you are going to:

1. Create a Java GUI
2. Create JLabel, JTextField, JButton
3. Implement Action Listener
4. Implement a GUI Infix to Postfix calculator

Java GUI

To appear onscreen, every GUI component must be part of a *containment hierarchy*. A containment hierarchy is a tree of components that has a top-level container as its root.

Each GUI component can be contained only once. If a component is already in a container and you try to add it to another container, the component will be removed from the first container and then added to the second.



Create JFrame

1. Create a Java Project called “JavaGUI”
2. Create a Java Class “main” (main.java)
3. Add the following code:

```
JFrame frame = new JFrame("Calculator");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Create JLabel, JTextField and JButton

1. Add the following code:

```
JLabel Label = new JLabel("Infix to Postfix calculator");

Label.setPreferredSize(new Dimension(200, 100));


JTextField TextBox = new JTextField();

JButton Button = new JButton("Calculate");
```

Using Layout

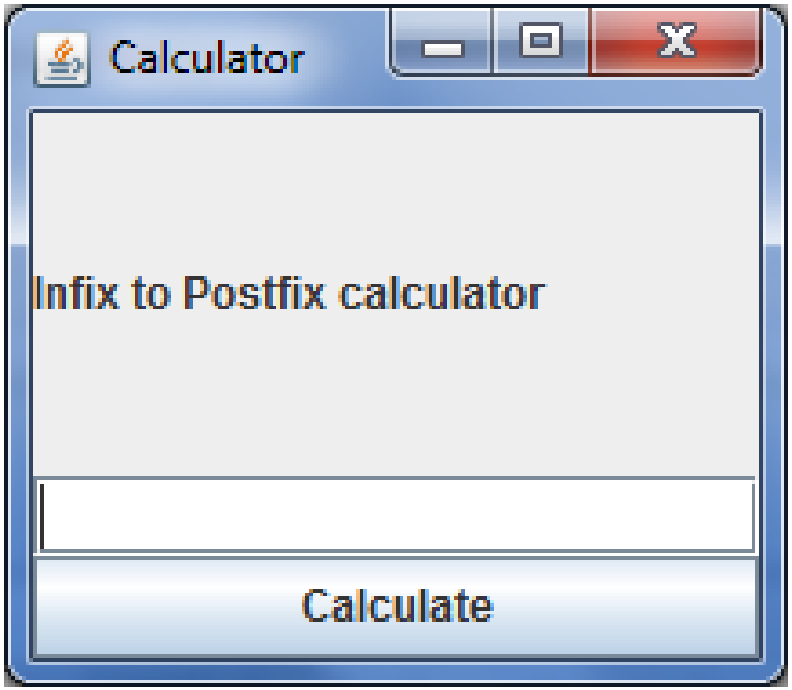
1. Map the Label, Textbox and Button in to the frame:

```
frame.getContentPane().add(Label, BorderLayout.NORTH);

frame.getContentPane().add(TextBox,
BorderLayout.CENTER);

frame.getContentPane().add(Button, BorderLayout.SOUTH);
```

Add Action Listener

```
Button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("Hello World");
    }
});
```

Final output



To Do

When the button “Calculate” is clicked:

1. Read the input from JTextField
2. Calculate the infix to postfix
3. Display the result in the JTextField

Assignment 3

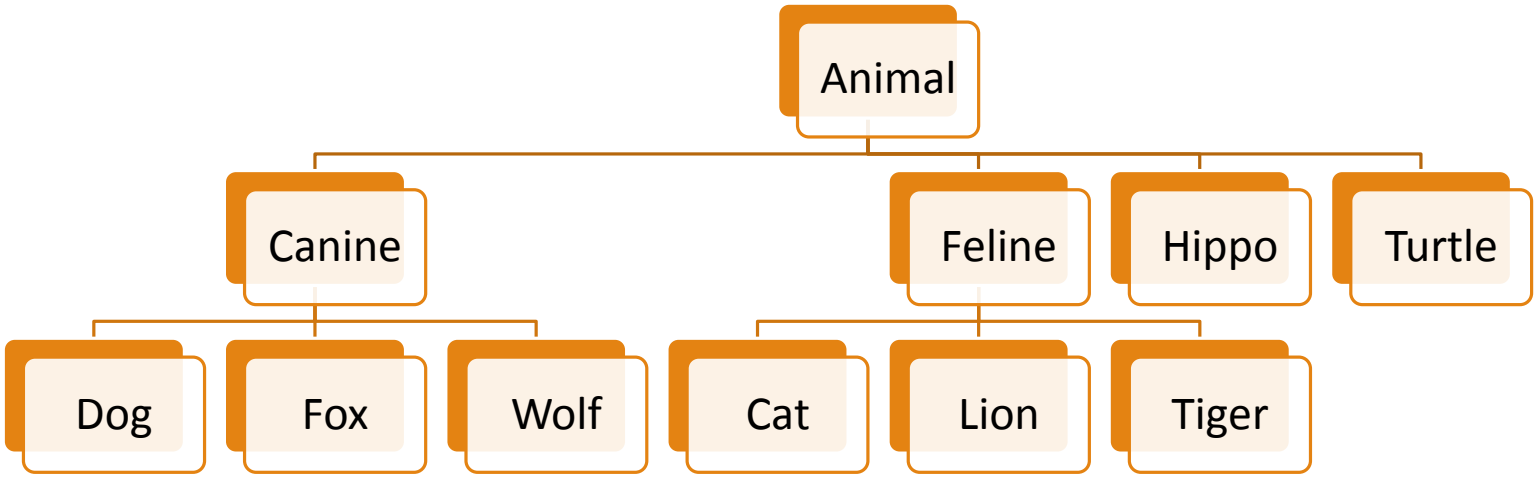
DUE: 27 MARCH, 2017 23:30

Assignment 3

- An extension of Assignment 1
- Simulated forest
- Display it in the GUI

Similarities & Differences

- Same animal hierarchy



- Same rules of animal moving and attacking

Similarities & Differences

Differences:

1. Size is different: 15×15
2. Two modes: auto / manual
3. Can start a new round when only one is living

Tasks

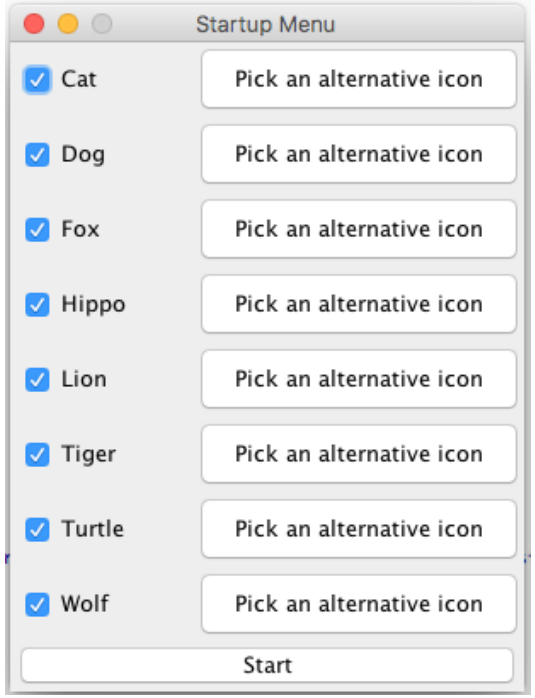
- Part I: Setup menu
- Part II: Initialization
- Part III: Auto mode / Manual mode
- Part IV: Animal moving
- Part V: Animal attacking
- Part VI: New round

Part I: Setup menu

1. select the animals to be included
2. select the images for animals
3. a VERY SIMPLE example:
4. Hints:

   Check box

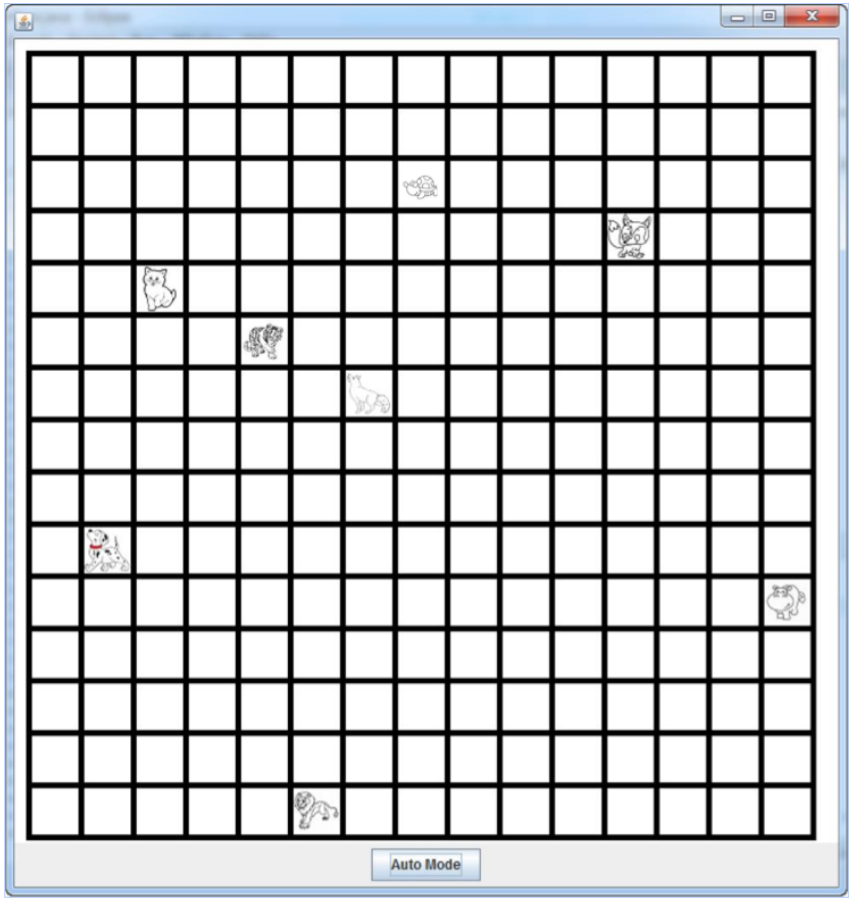
   List / Combo box



Part II: Initialization

- Start simulating the forest after click the start button
- Initialization (Graphics should be used)
  - A table: **15*15**
  - Animals included with their icons(one object for each animals)
  - One button controlling the mode
- Print Information, for example:
  - -------- START --------
  - xxxx initialized at x,y
  - ……

Part II: Initialization



Part III: Auto mode/Manual mode

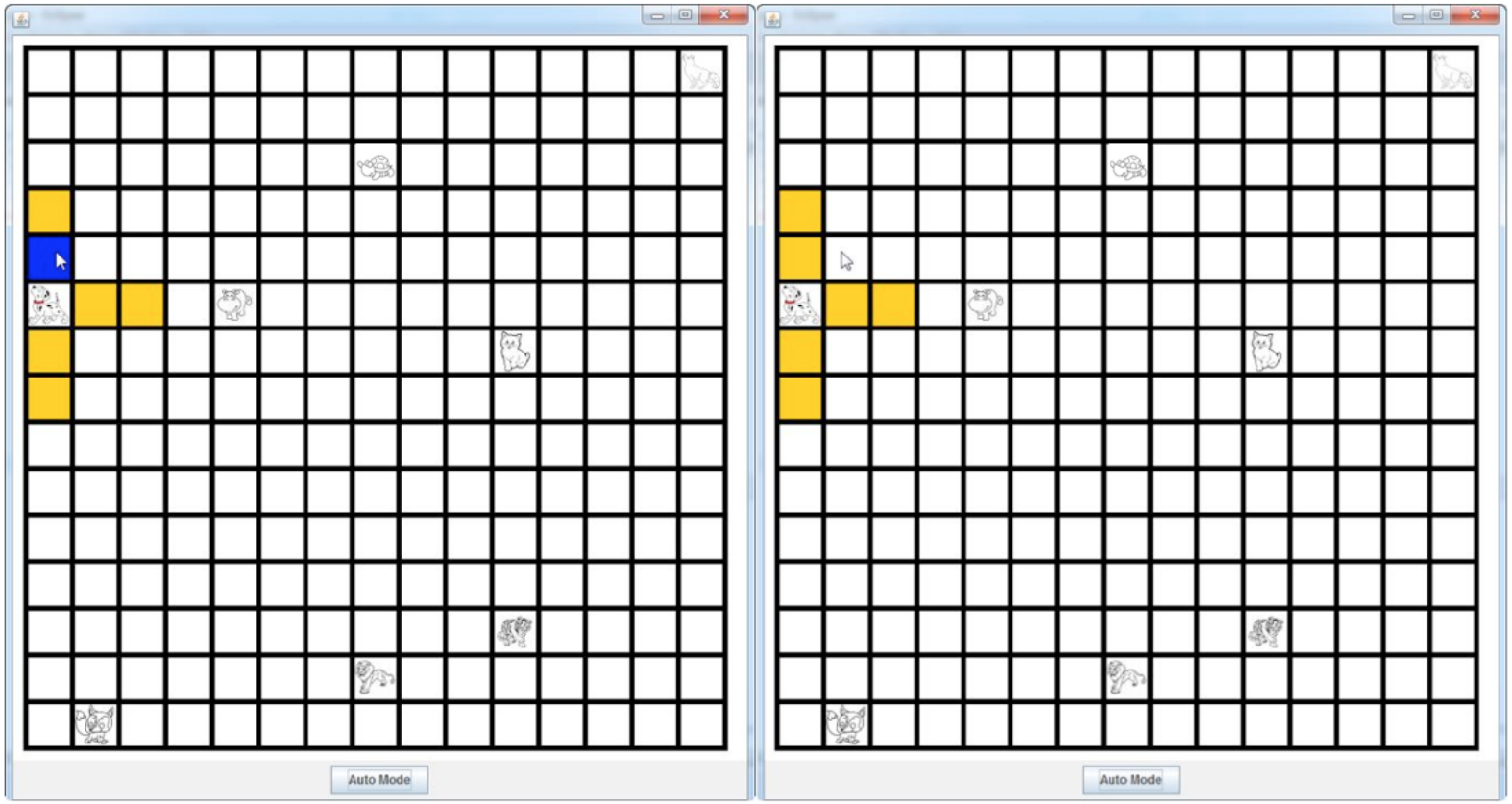
- Auto mode:
  - Animals move (or attack) automatically in the same manner as described in Assignment 1
- Manual mode:
  - Control the animals by dragging the mouse and update the layout & icons

Part IV: Animal moving

- On pressing:
  - Show the move(attack) range of the specified animal in Orange
  - Show other animals in its attack range in Red
- On dragging:
  - The cell that the cursor goes into becomes Blue as a real-time process

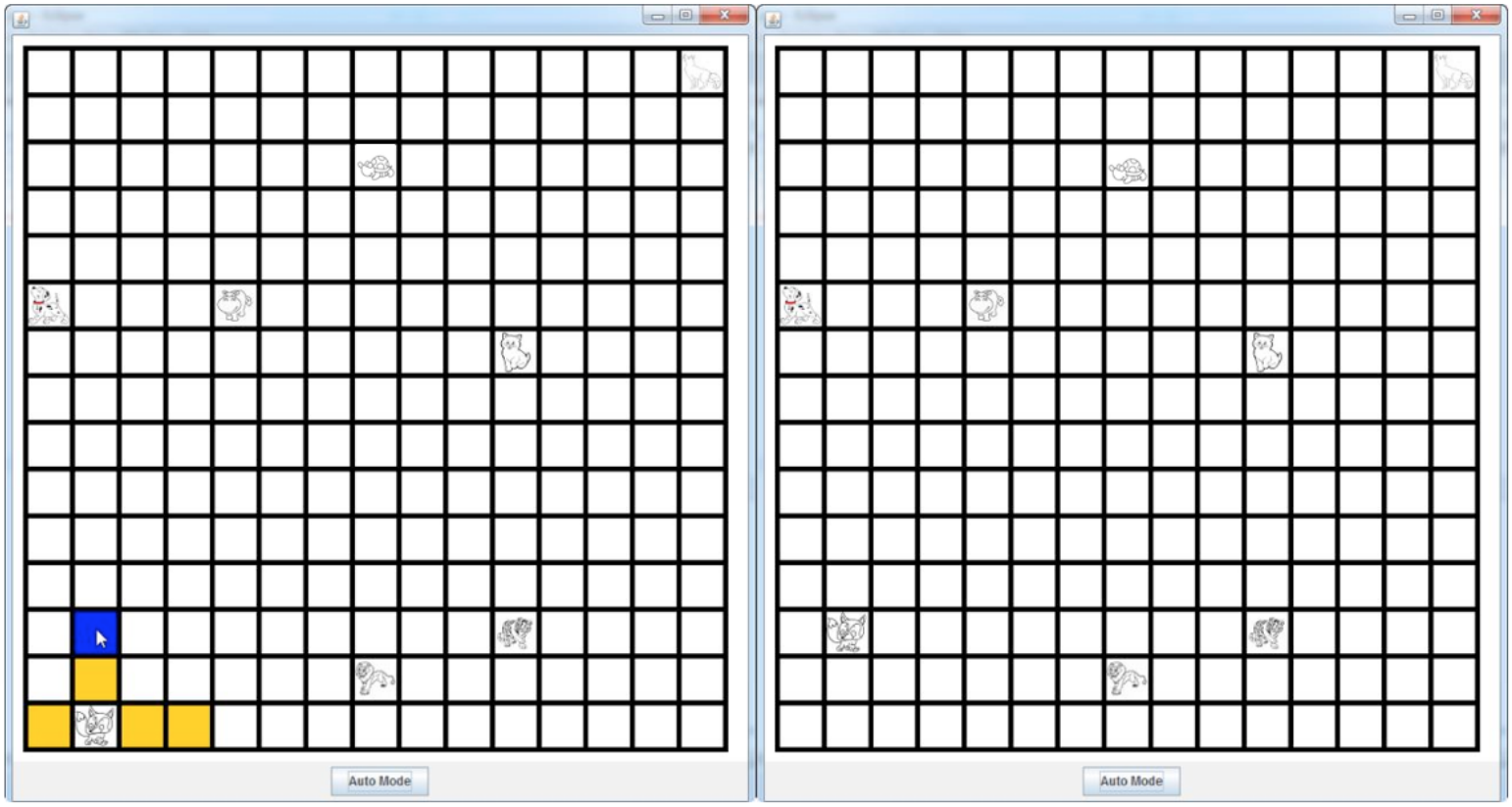
Part IV: Animal moving

Drag and move mouse



Part IV: Animal moving

Drag and release on a reachable place

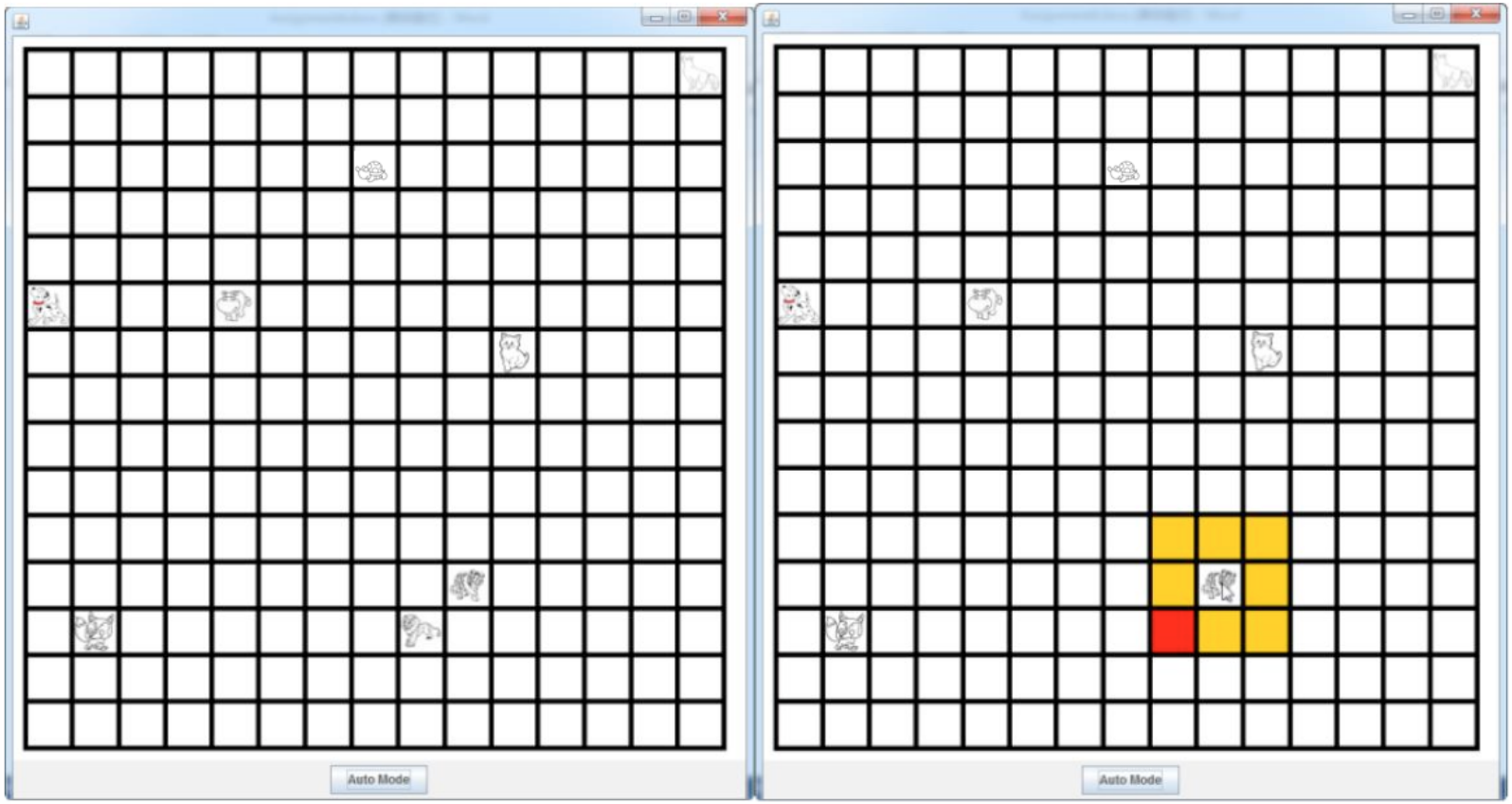


Part V: Animal attacking

- when it attacks: the same as Assignment 1:
  - when the animal moves to another animal's cell
  - when an animal on its way
- same attacking rules
- same print information
- same way to place dead bodies

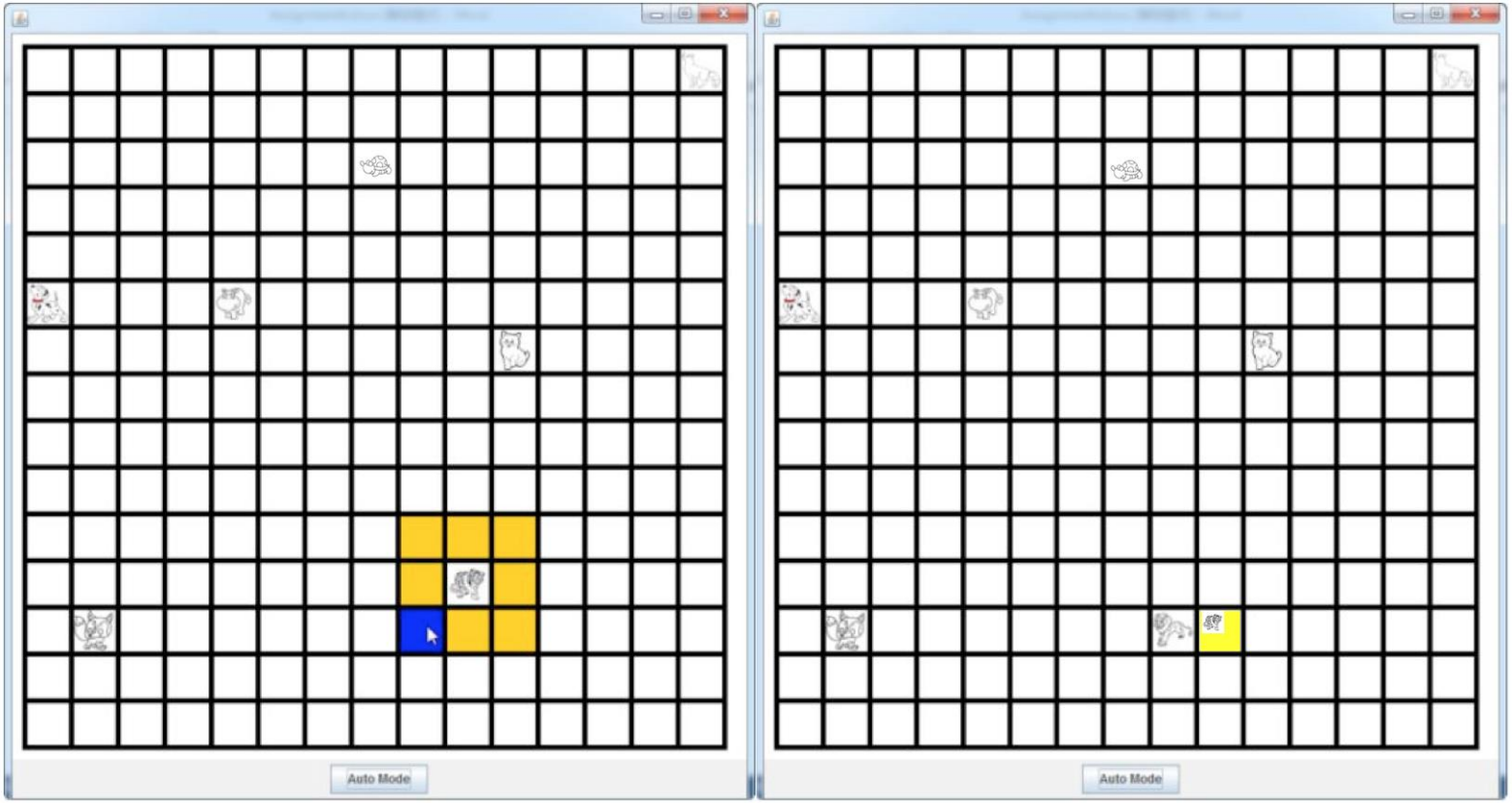
Part V: Animal attacking

- GUI: use red to represent defender



Part V: Animal attacking

- GUI: use yellow to represent dead body position, with icon on it



Part VI: New round

When there is only one animal in the forest:

- Print information(for example):
  - -------- END --------
  - The survivor is xxx at x,y
- Start a new round(immediately / with one click)

Marking

80% functionality:

- You will get part of the full marks if you implement some of the features
- Higher marks: no exception during runtime

20% program design

- OOP, GUI, graphics, frames, panels, buttons, action listeners
- Precise interface & performance

No marks if no Javadoc

No marks if there's compilation error

Useful suggestions

Perhaps most challenging so far

- Make your program well organized
- Classes, methods: modularize your program

Put functionality at first!

Start it early!