Mouse events

COMP 102

Victoria University of Wellington

GUI: Mouse input

- Just like other UI, except don't have to put any UI on screen
  - Each press / release / click on the graphics pane will be an event
  - Must tell UI the listener: the object::method to call when a mouse event occurs

```
UI.setMouseListener(this::doMouse);
```

  - Must define method to say how to respond to the mouse.
    parameters: kind of mouse event and position of mouse event

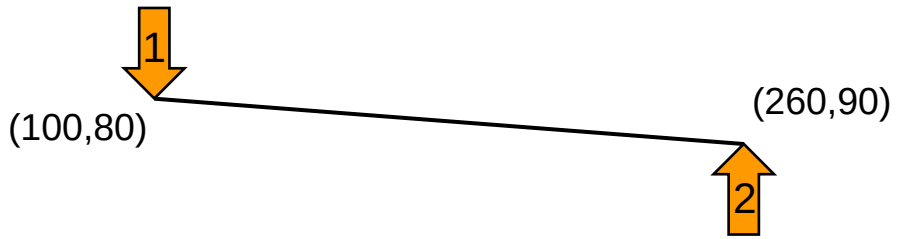
```
public void doMouse(String action, double x, double y) {
    if (action.equals("pressed") ) {
        // what to do if mouse button is pressed
    }
    else if (action.equals("released") ) {
        // what to do if mouse button is released
    }
    else if (action.equals("clicked") ) {
        // what to do if mouse button is clicked
    }
}
```

where action occurred

press-release in same place

Using the mouse.

- Want to let user specify input with the mouse,
  - eg: drawing lines



- Typical pattern:
  - On "pressed",
    - just remember the position
  - On "released",
    - do something with remembered position and new position

Mouse Input

```
public class LineDrawer {      /**Let user draw lines on graphics pane with the mouse. */
    private double startX, startY;  // fields to remember "pressed" position
    public void setupGUI(){
        UI.setLineWidth(10);
        UI.setMouseListener(this::doMouse);
        UI.setDivider(0.0);
    }
    public void doMouse(String action, double x, double y) {
        if (action.equals("pressed") ) {
            this.startX = x;
            this.startY = y;
        }
        else if (action.equals("released") ) {
            UI.drawLine(this.startX, this.startY,  x,  y);
        }
    }
```

Mouse Input

Simple mouse events: UI.setMouseListener(this::doMouse);

- pressed
- released
- clicked

Mouse movement: UI.setMouseMotionListener(this::doMouse);

- pressed, released, clicked
- dragged
- moved