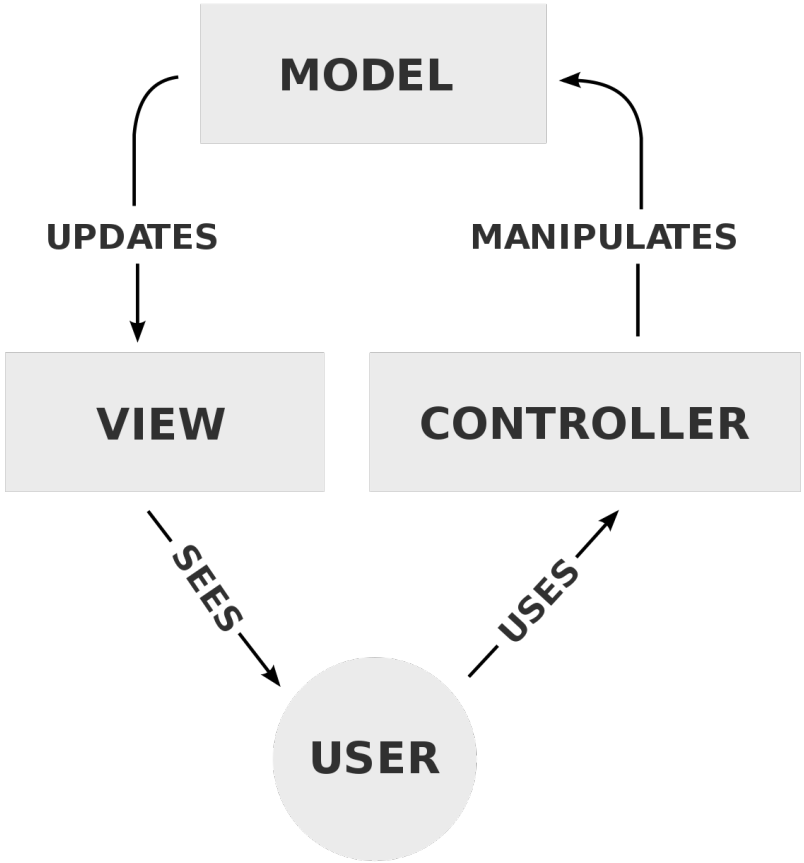
GEOG 178/258
Week 8:

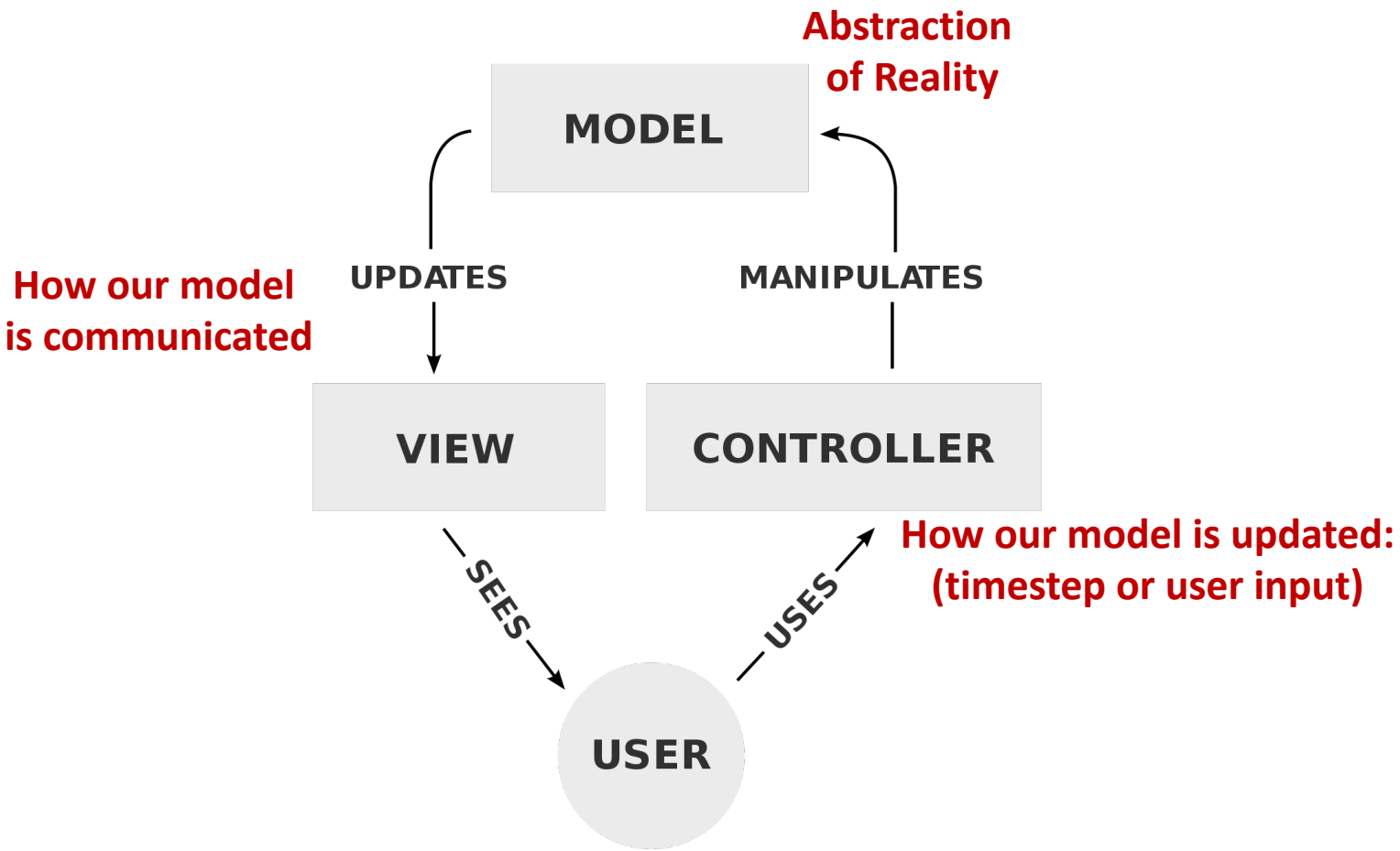
Model Viewer Controller: MVC

mike johnson

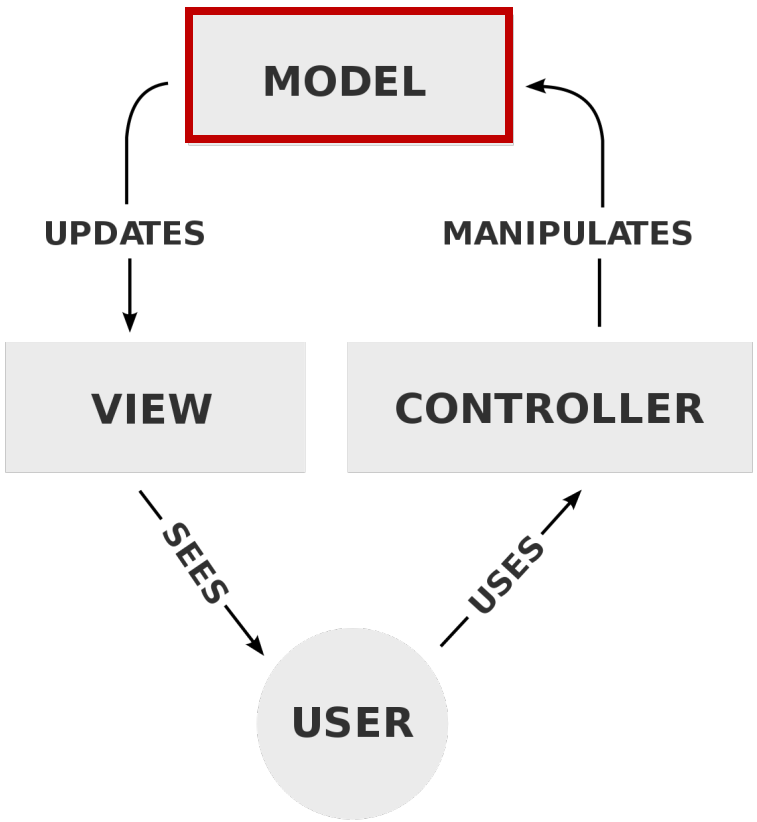
Model View Controller



Model View Controller



Model View Controller



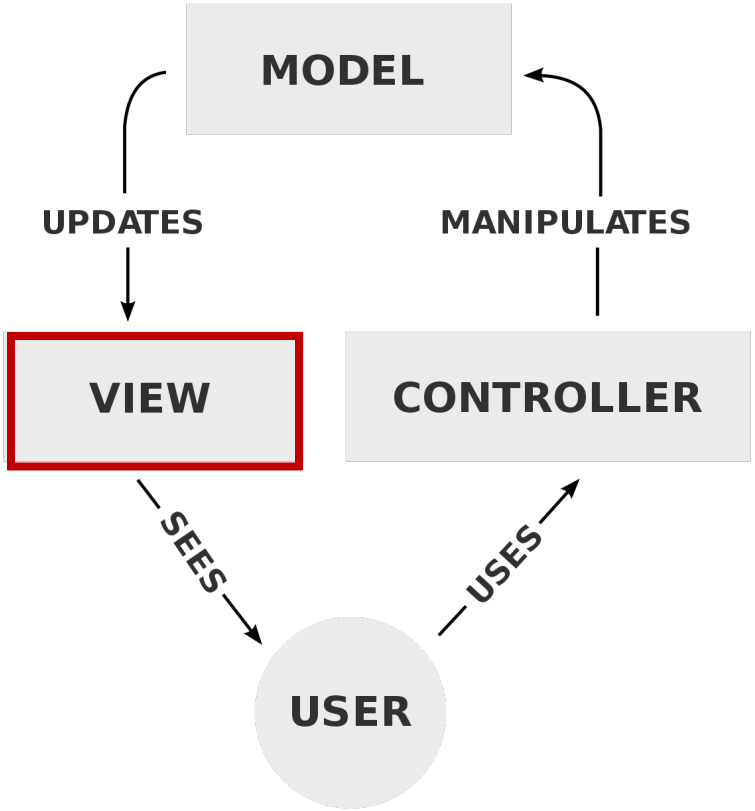
1) What are we trying to model? (Movement, physical process, behavior…?)

2) What is the abstraction we are interested in?

3) How will we represent the pieces in the most basic form (think back to the ‘essence’ of the object)

4) What classes can we inherit and which need to be built?

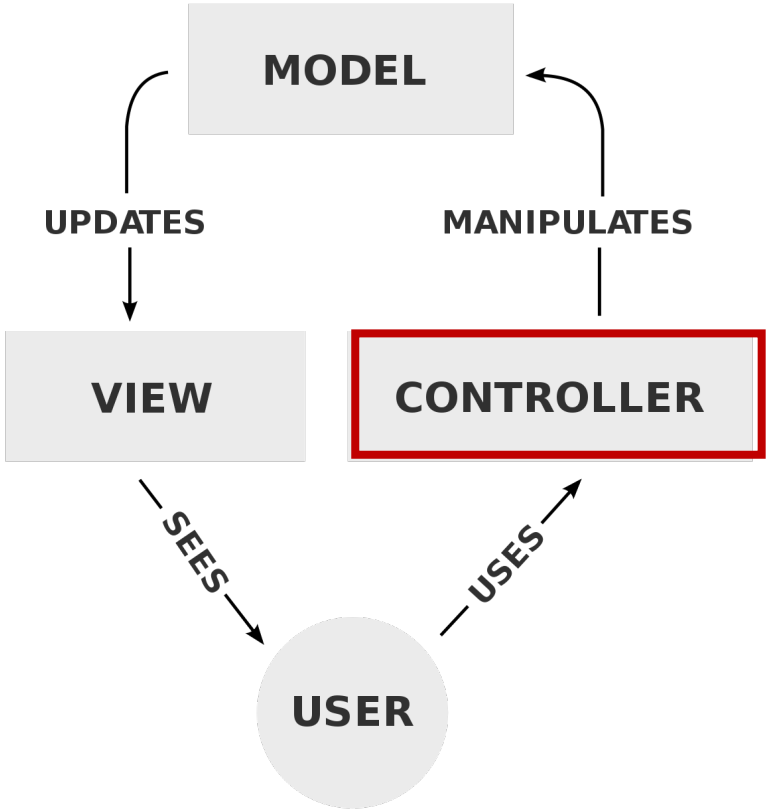
Model View Controller



1) How will results communicated? (Map, plot, text…)

2) What aspects of the model are we interested in as ‘results’

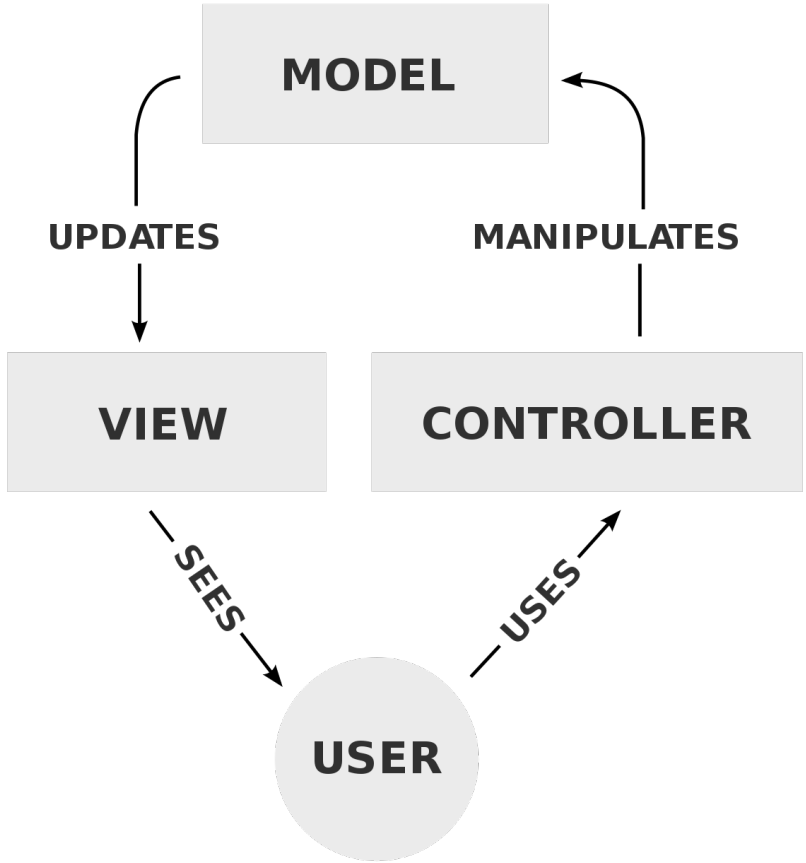
Model View Controller



1) How will the model run? (timesteps, user input, single case…)

Model View Controller

Member Variables



PaintComponent

Listeners

Model View Controller: Signature

```
public class tinyGIS extends JPanel implements MouseListener, ActionListener {
```

Model **View** **Controller**

Model View Controller: Code

```
public class tinyGIS extends JPanel implements MouseListener, ActionListener {

// MODEL

    private int x;
    private Point y;
    private boolean z;

    public tinyGIS() {
    }

//VIEW

    public void paintComponent(Graphics g) {
    }

// CONTROL

    public void mouseClicked(MouseEvent e) {

    public void mousePressed(MouseEvent e) {

    @Override
    public void mouseReleased(MouseEvent e) {
    }

    public void mouseEntered(MouseEvent e) {

    public void mouseExited(MouseEvent e) {

    @Override
    public void actionPerformed(ActionEvent e) {
    }
}
```

Model

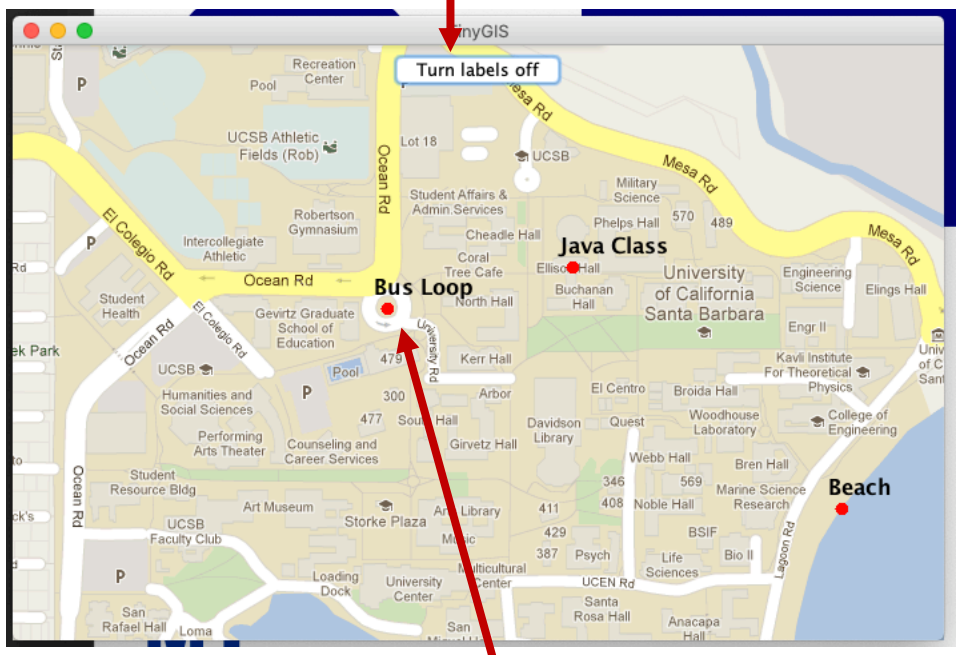
View

Controller

Model View Controller: Visual

Controller

View



Model

Model View Controller: Homework

- Implement a POI App that allows you to digitize a path as well as POIs via mouse clicks (and store both types of geometries).
- Show the names of the POIs when they are visited (via the digitized path)
- Allow the user to name POIs and check their names by clicking on them [GEOG 258 only] (http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html)
- Implement the path in a way that it can be reset which also causes the display to hide the POI names. [GEOG 258 only]

Model View Controller

HW MVC

Model:

POIs: Have **names**, and can be **visited** (Extend PointBuffer?)
Paths: Can visit POIs and be reset (PolyLine)

Viewer:

Map (campus.png; g.drawImage())
POI (w and w/o **names**) (g.fillOval(x,y))
Path (d.drawLine(x1,y1,x2,y2)

Controller:

Clicks let you **draw** new points, and **name** existing points (MouseListener)
Path can be **reset** (ActionListener?)

In Class Examples

Homework

Assignment

- Implement a POI App that allows you to digitize a path as well as POIs via mouse clicks (and store both types of geometries).
- Show the names of the POIs when they are visited (via the digitized path)
- Allow the user to name POIs and check their names by clicking on them **[GEOG 258 only]** (http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html)
- Implement the **path** in a way that it can be **reset** which also causes the display to hide the POI names. **[GEOG 258 only]**
- (We will learn about Exceptions next time (but you may have read about them before)
- Upload a zip file [LN1W5.zip] with the *.java files to GauchoSpace.
- Assignments and **executable** programs are due the day before lecture at **5pm PST** of each week.