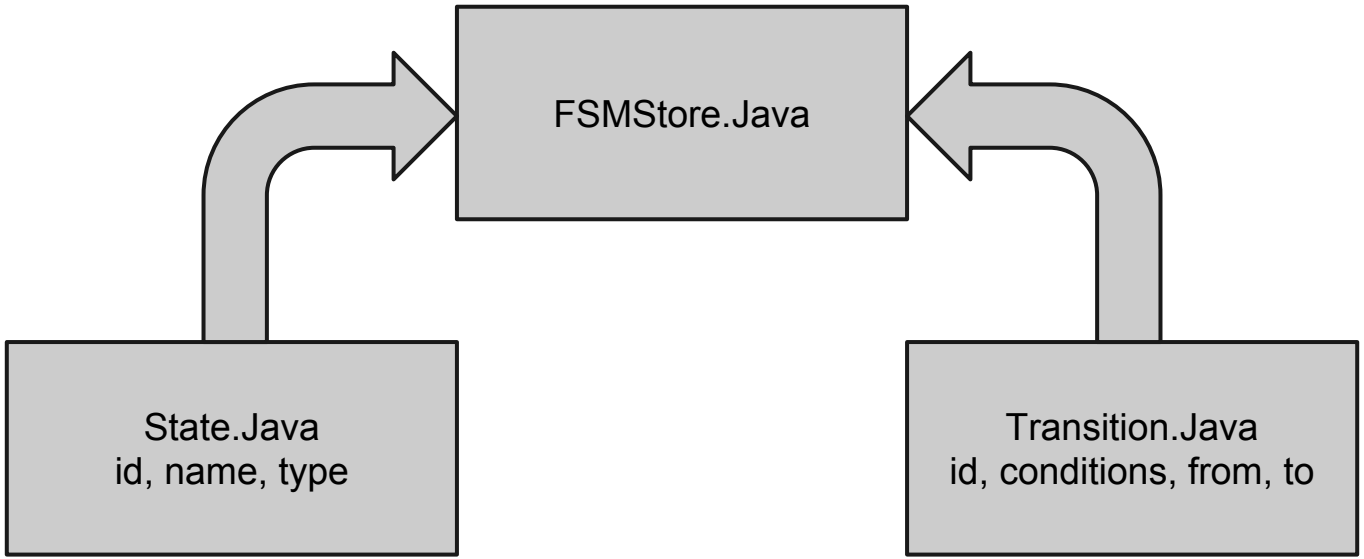
Finite State Machine

Frank, Rudy & Nate

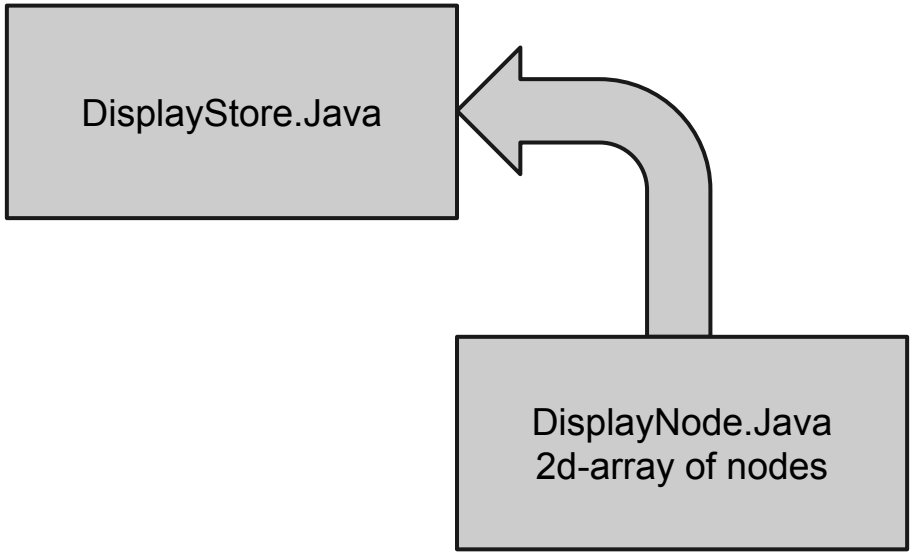
Storing the FSM



FSMStore API

FSMStore	State	Transition
addState(name,type) id addTransition(name,fromID,toID) id removeState(id) removeTransition(id) containsState(id) boolean containsTransition(id) boolean getState(id) State getTransition(id) Transition numStates() & numTransitions()	getID() id getName() name rename(name) setType(type)	getID() id getFromID() fromID getToID() toID addCondition(name) removeCondition(name) numConditions() getLabel()

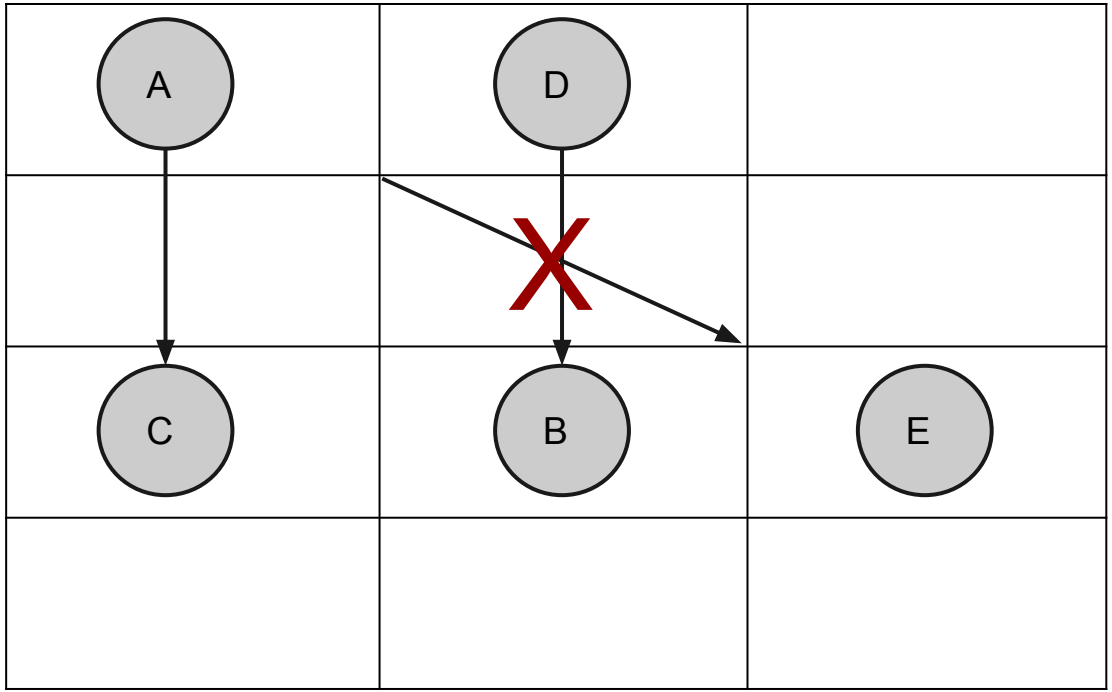
Storing the Grid



DisplayStore API

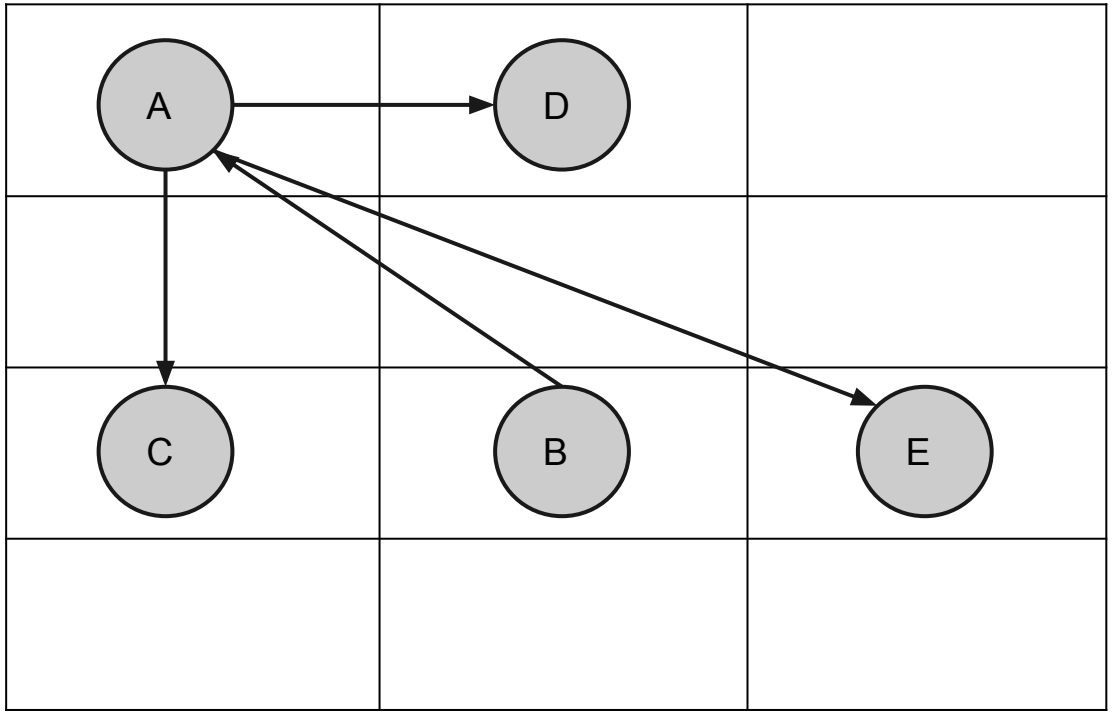
DisplayStore	DisplayNode
getRow(id) → row getColumn(id) → column addState(x,y,id) → id removeState(x,y) moveState(x1,y1,x2,y2) → bool getState(x,y) → id	getID() → id containsState() → bool setNode(id) → id removeNode() → id

Change in Design



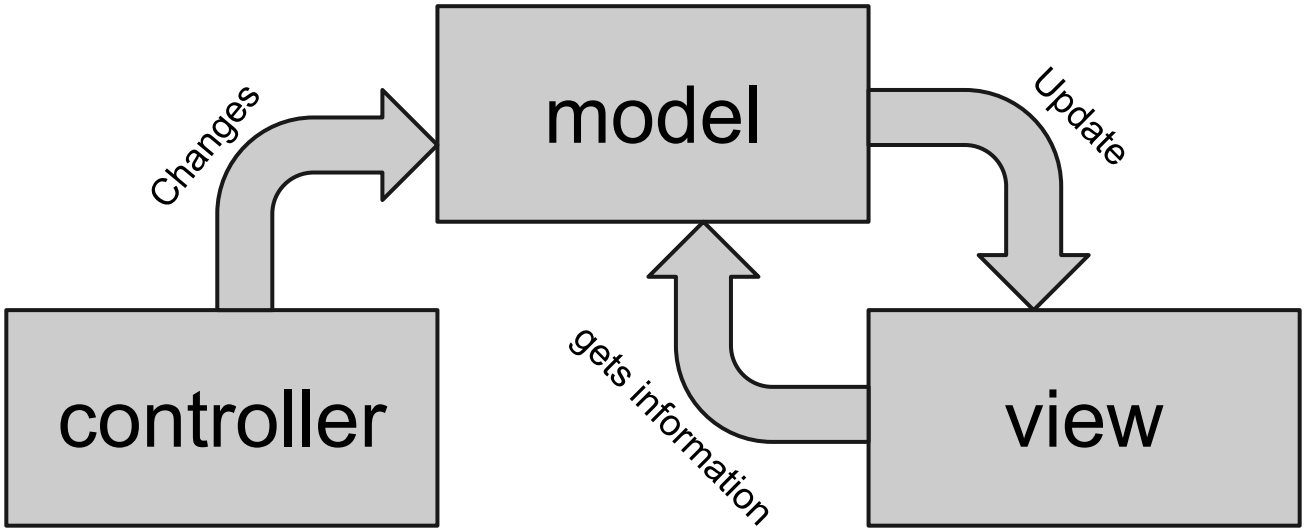
Original Implementation

Change in Design (cont)

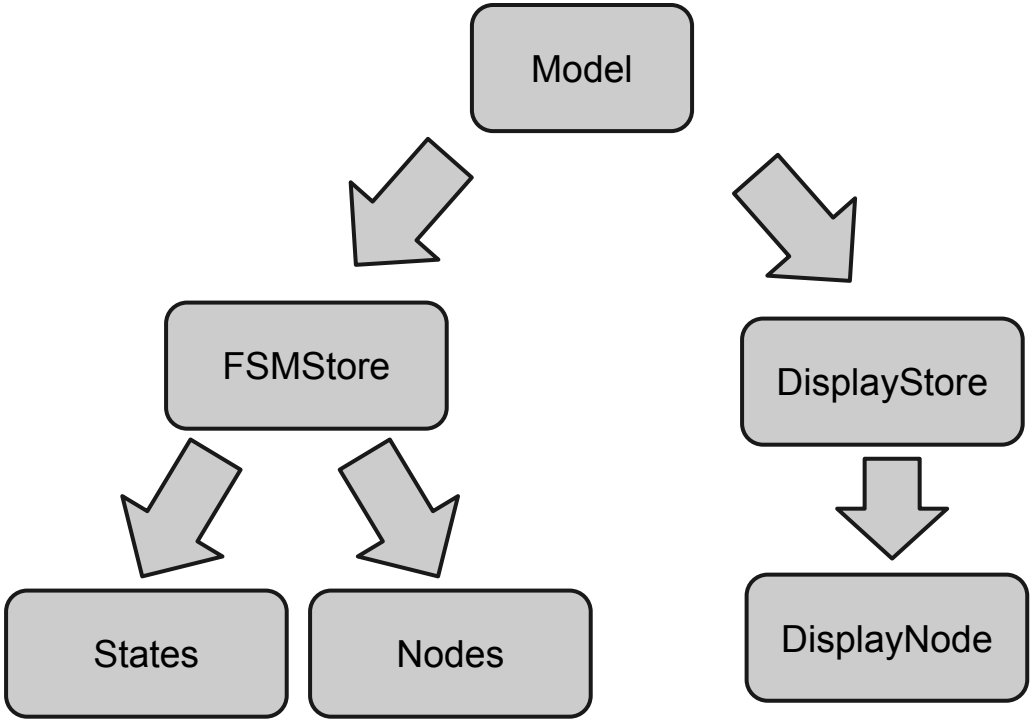


New Implementation

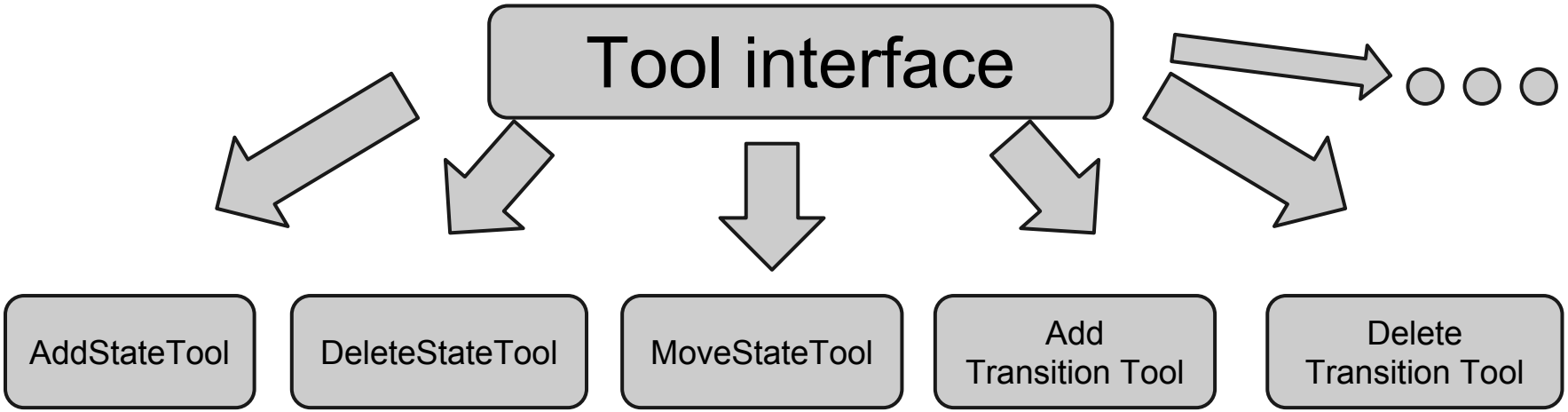
Model-View-Control Layout



Model



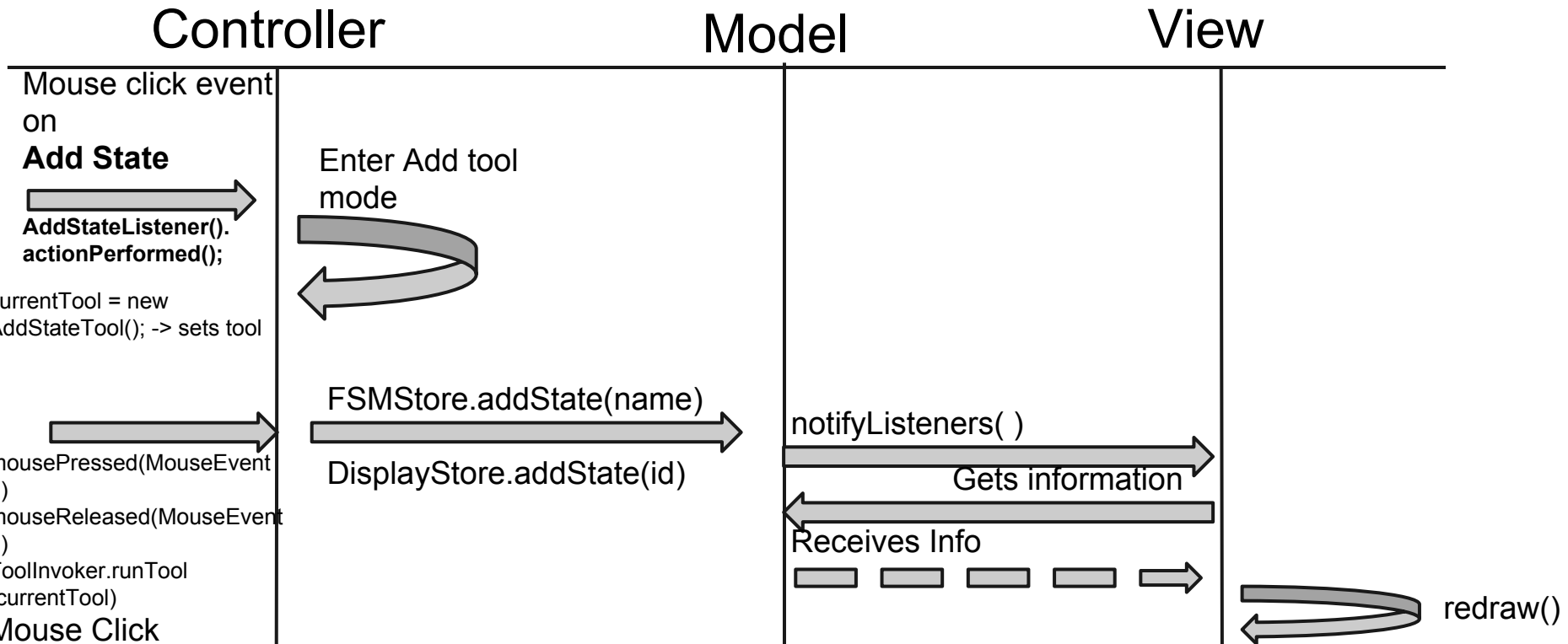
ToolBar - Command Pattern



Tool Implementation

- ToolInvoker
  - ToolInvoker.runTool(currentTool);
- Controller is a MouseListener
  - mousePressed(MouseEvent e)
- Tools are Private Classes within the controller
  - allows for access to mouse info inside of the controller without having to pass around alot of info

Click on Empty Location with create State Tool - Sequence Diagram



Save

Saving the State Machines to files would be implemented by going through the State Machine, taking its data and concatenating it to a string. This string will then be saved to a different file using the fileWriter.

Load

Load will take a string, which contains all of the elements pertinent to the class. We then call StringTokenizer on the string and on the element that we will be splicing the string around. We then throw in the “nextElement” into the load of the class below.

“;$false,-1;true,23;true,2;false,-1;$”

In this example the code first would set $ as the delimiter, and would pass the “false…-1;” into our DisplayStore. From there it would throw the new string and “;” into StringTokenizer. So the “nextElement” would be “false,-1”, which would be thrown into DisplayNode.

User Interface Diagram

Demo

YAY!

Any Questions?