

Programming of Interactive Systems

Anastasia.Bezerianos@lri.fr

(Nolwenn.Maudet@lri.fr)

Week 7 : b. Exam preparation

Anastasia.Bezerianos@lri.fr

(part of this class is based on previous classes from Anastasia,
and of T. Tsandilas, S. Huot, M. Beaudouin-Lafon, N. Roussel, O. Chapuis)

General info

Duration: 2 hours

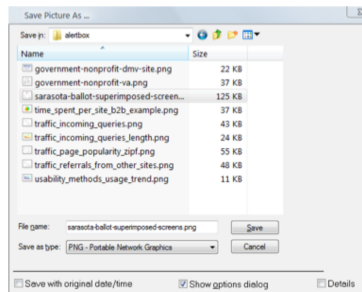
Authorized material: any document in paper form

Time: (most likely) Tue 15/11 @ 9:30

Widgets,
event-based programming,
SM,
and MVC

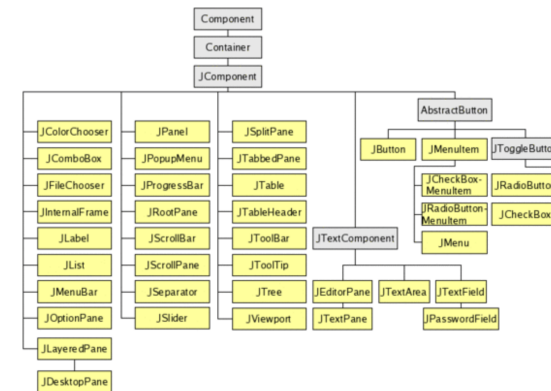
Example (widgets + events)

Identify the components (widgets), their events and their listeners

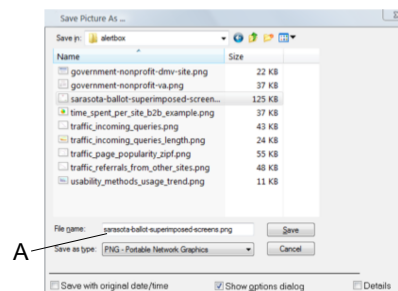


You don't necessarily have to follow the naming and precise structure of Swing

Hierarchy of Swing components



Can you create the hierarchy of the containers and components of this UI?



What layout manager(s) would you use and how?

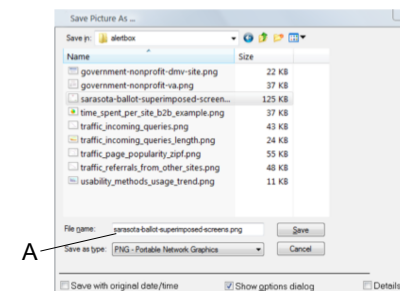
What about the events and their listeners?

Example:

Think about how to handle selection events within the list (tabular form) of the files

Describe:

Widget
Event
Listener
Related widgets and relation



Example (MVC)

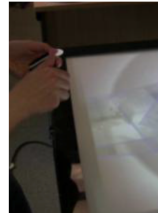


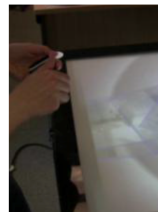
Figure — In this example a user is approaching a digital table with their mobile. A halo appears at the edge of the mobile. The user can flick the image towards the halo. When the image slides off the edge of the mobile it disappears, and reappears on the table.

Example (MVC)



Where is the application? In the mobile device or in the tabletop?

Example (MVC)



Where is the application? In the mobile device or in the tabletop?

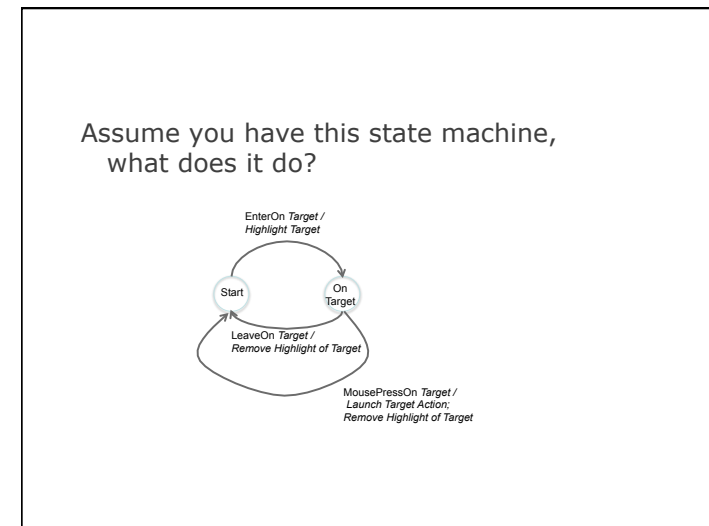
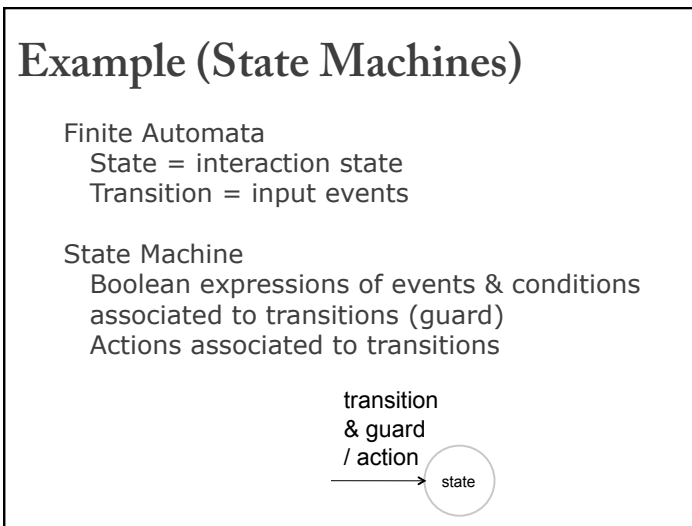
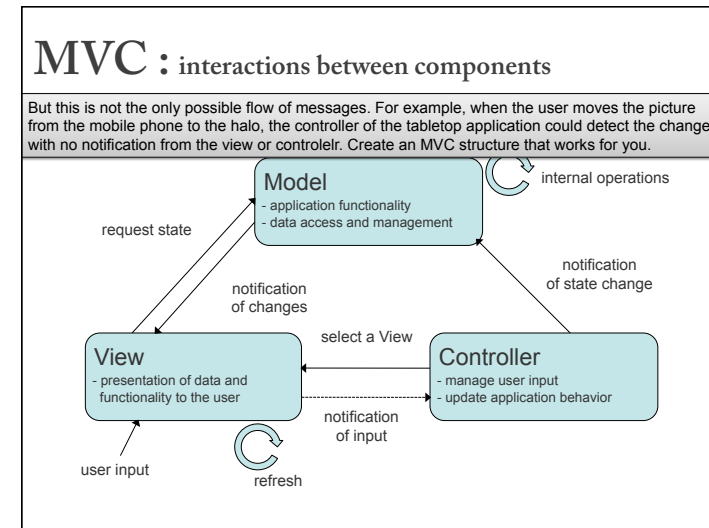
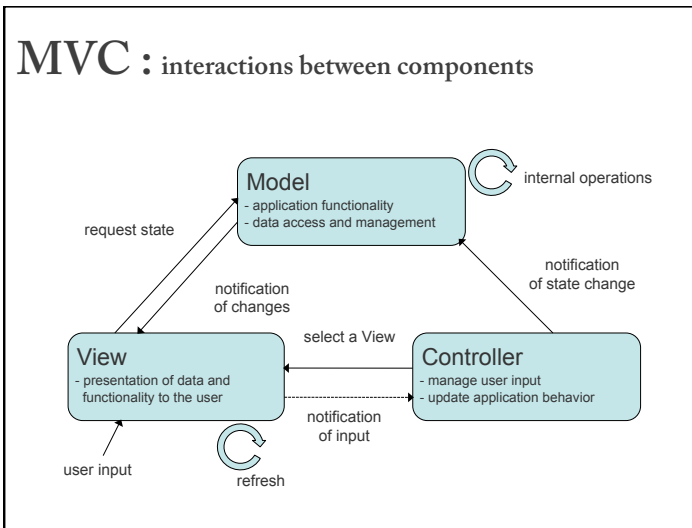
How could this application be implemented?

Example (MVC)



Where is the application? In the mobile device or in the tabletop?

How could this application be implemented?
Create a diagram that shows the interaction between the View, the Controller, and the Model for the mobile and the tabletop apps



You have access to:

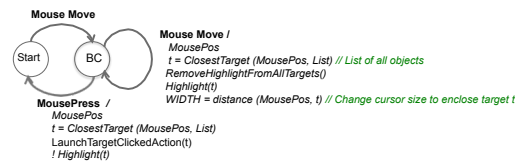
List = IntersectTargets (mousePos, WIDTH)

Target = ClosestTarget (mousePos, List)

TimeOut (programmable) event to start after n sec with Arm(n), a fuction Disarm() cancels the Arm call if TimeOut has not started

Create the interaction for Bubble cursor:
closest target always selected on click

Create the interaction for Bubble cursor
closest target always selected on click

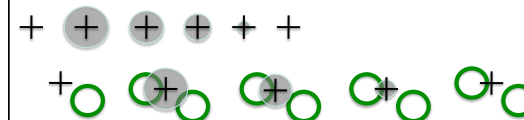


Dynaspot:

area cursor of MAXWIDTH active when speed > MAXSPEED

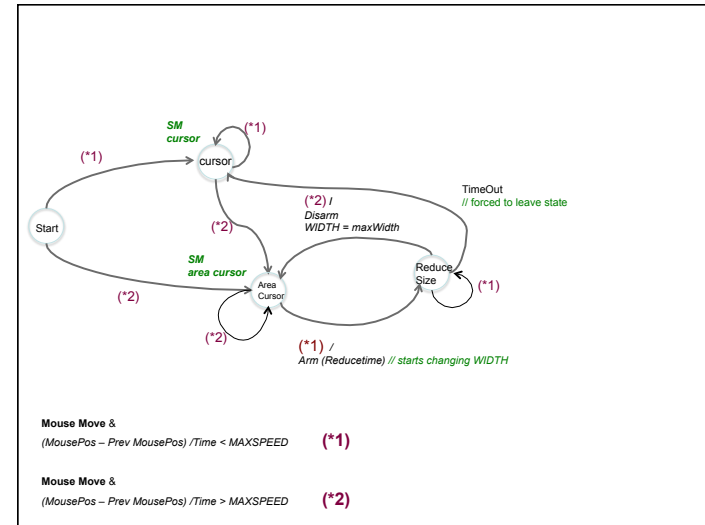
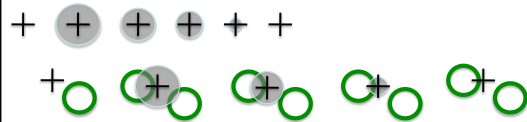
when speed drops an animation starts that takes REDUCETIME during which the area of the cursor becomes smaller until 0 and the cursor becomes a regular cursor

during REDUCETIME we can accelerate, and go to area cursor again



Dynaspot:

Hint, use the previous SM



Example (Interaction)

Imagine you are asked to program the following behavior:

When I drag a circle over another, the second circle starts moving to the opposite direction so that they do not overlap (it is pushed)

How would you program this (what events would you listen for, what tests would you do, how would you move/change the position of the second object, etc)?

Assume you know your mouse position, you are dragging the first circle from its center, you know the radius of your circles (r_1, r_2), and you can calculate the intersections of objects.

Example (Interaction)

- We will need a single JFrame with a panel, and we will overwrite its paintComponent function to draw ellipses.
- We'll need a mouse listener attached on the frame.
- At mouse click we will check if the first ellipse is selected (under cursor position) and set a variable "dragging" to true: if (underCursor(posx, posy, ellipse1)) dragging = true;
- For every event I store mouse position in prevx, prevy (so I can check for distance differences in last step)
- At mouse drag, if "dragging" is true, I will check at every point if ellipse1 touches ellipse2, ellipse1.intersects(ellipse2).
- If it does not intersect, then I move ellipse1 to the new mouse position ellipse1.setPosition(posx, posy)
- If it does intersect, I still move ellipse1 as before, but I also move ellipse2. I will first get its previous position ellipse2.getPosition(), and then the new position for ellipse2 will be the previous position + dx, dy (where dx = x-prevx, and dy = y-prevy)