Preparing for the final exam

Exam material

UI Programming: Widgets, Events, Listeners, Model-View-Controller (MVC) architecture

Modeling user interaction, state machines

UI Design principles: C.R.A.P., gestalt laws, perfomance & memory

User-centered design & evaluation: conceptual models, feedback, scenarios, sketches, and storyboards, evaluation & usability criteria

Phrasing and chunking, modes & gestures

Input: Direct, indirect, relative, absolute pointing, CD gains, multitouch

Additional information

Duration: 2 hours

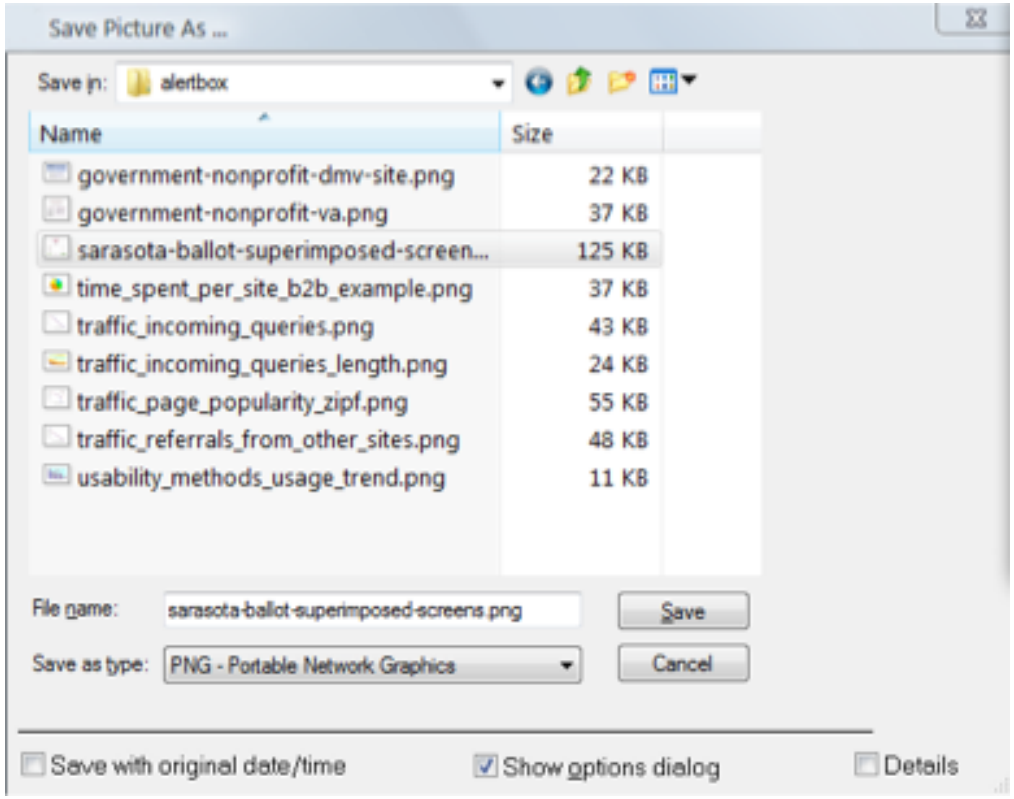
Authorized material: any document in paper form

Language: either English or French

Widgets, event-based programming, and MVC

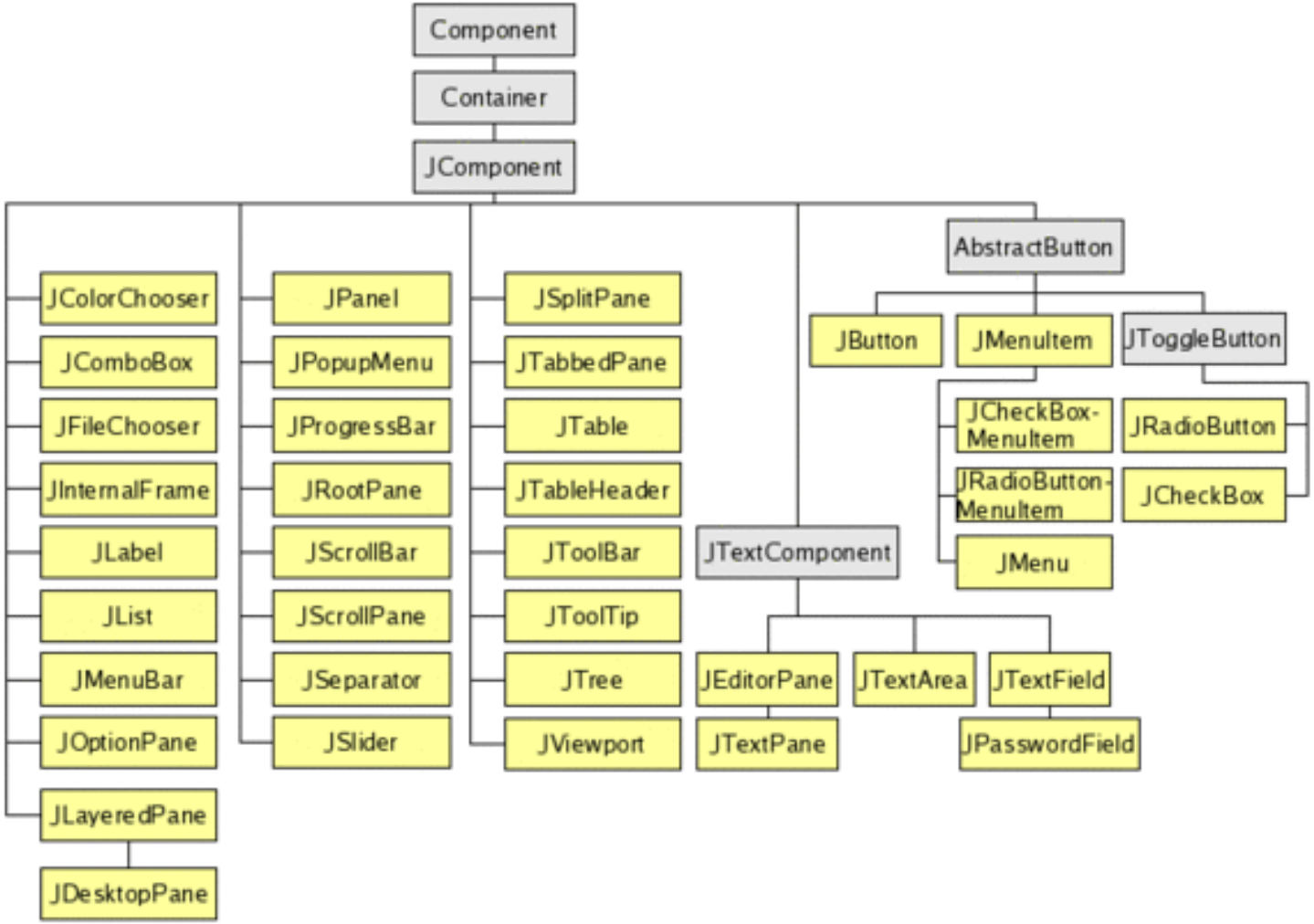
Exam 2013 – A.

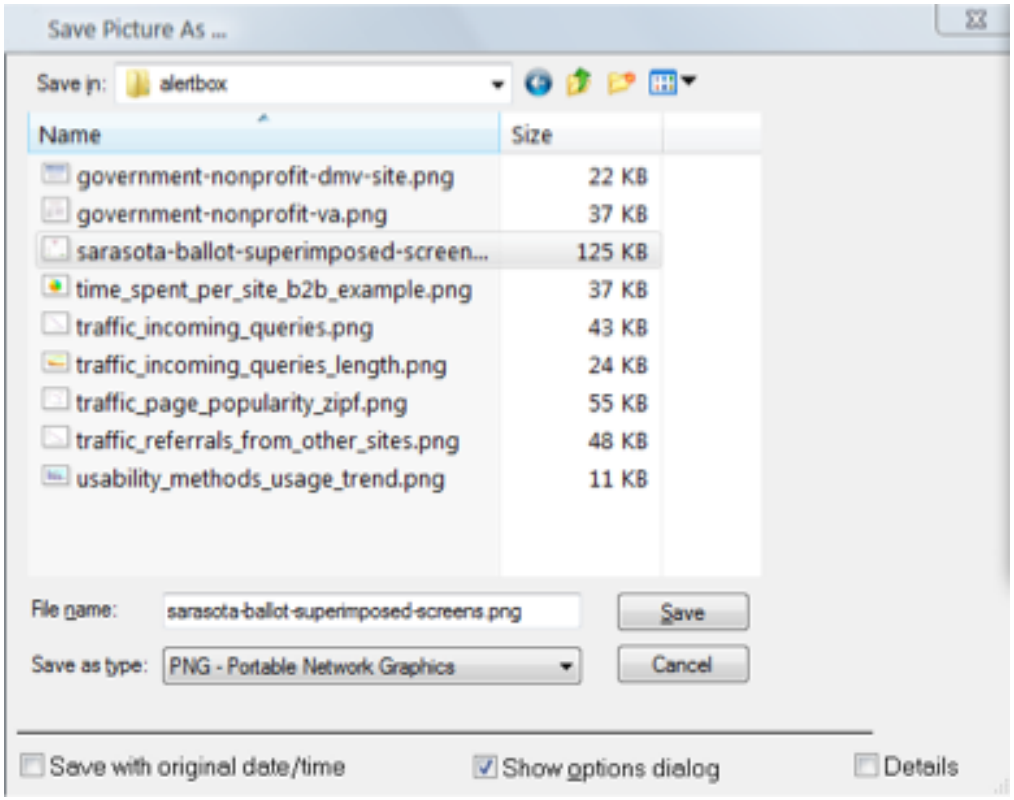
Identify the components (widgets), their events, and their listeners



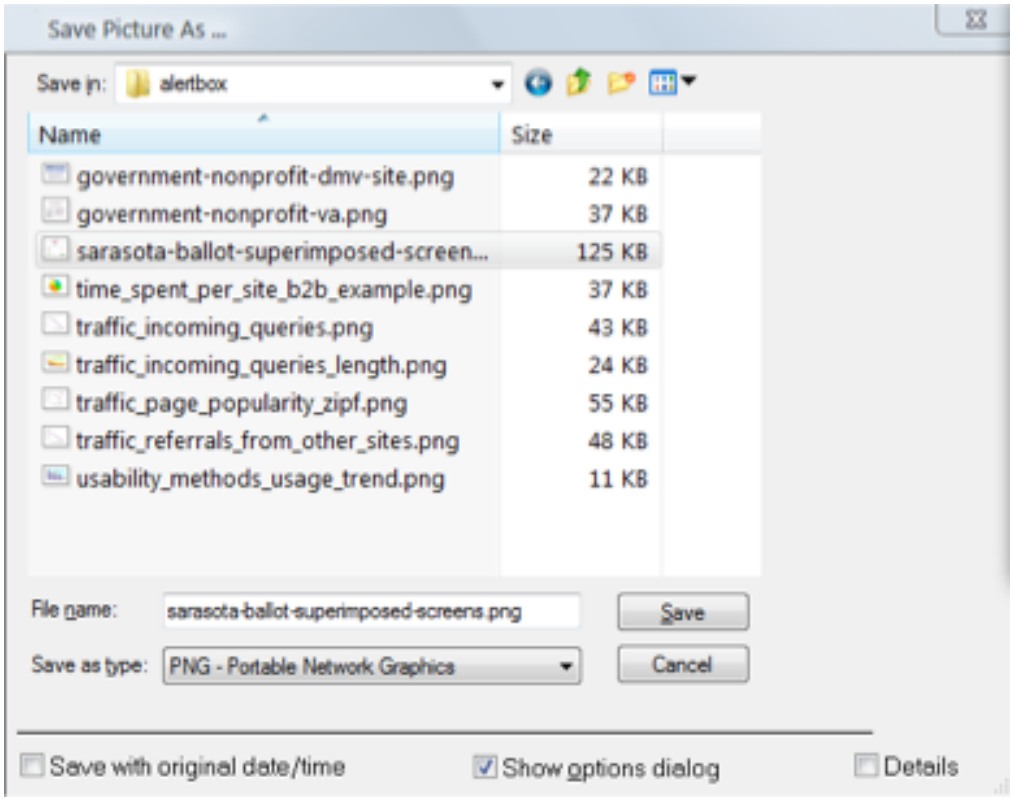
You don’t necessarily have to follow the naming and precise structure of Swing

Hierarchy of Swing components





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



What about the events and their listeners?

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

Exam 2013 – B (MVC)

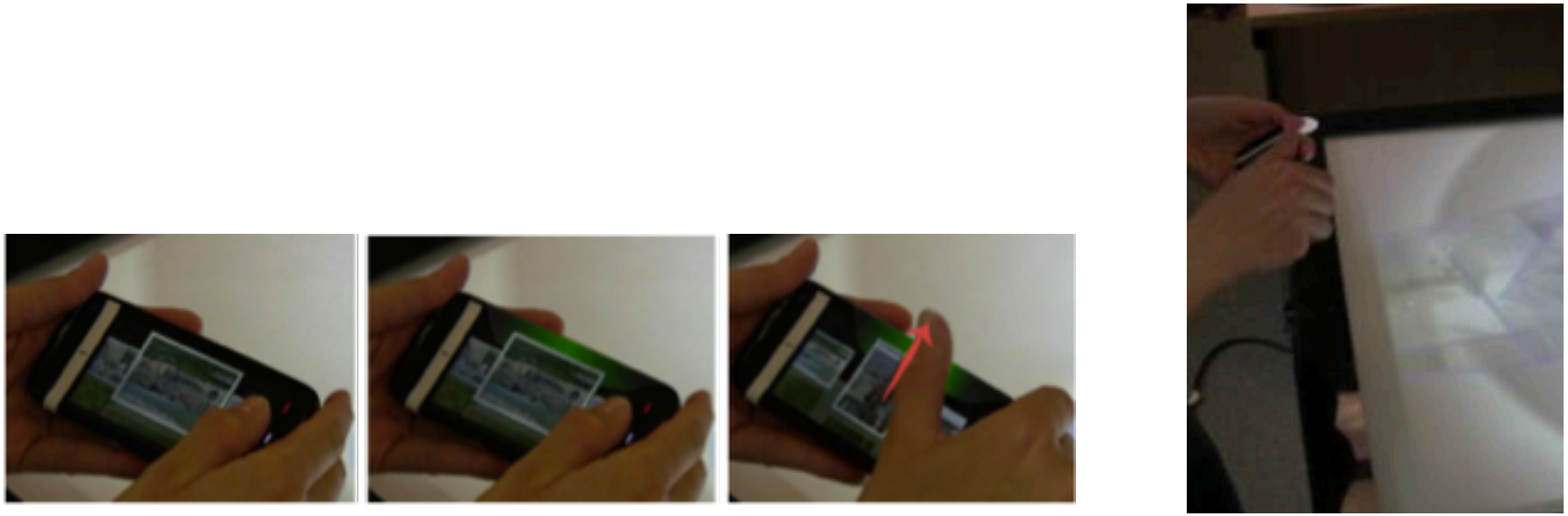
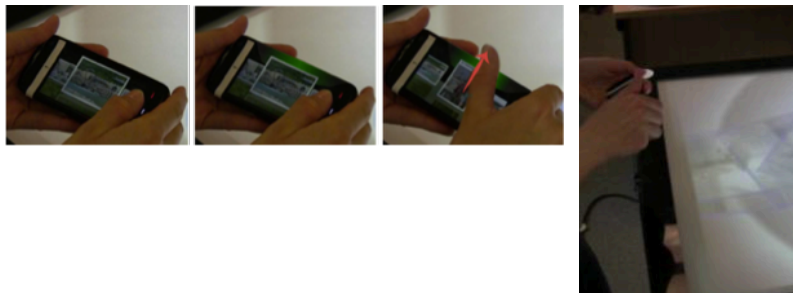


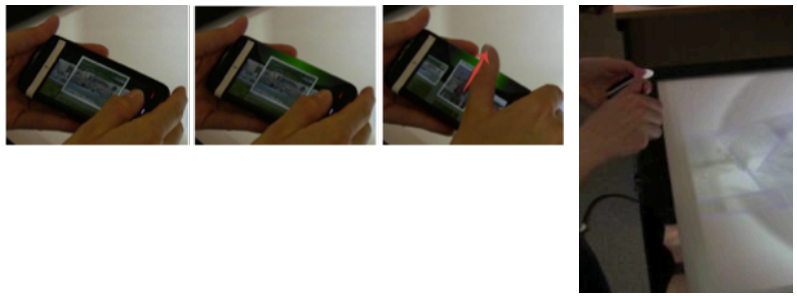
Figure 2— In this example a user is approaching a digital table with their mobile. A halo appears at the edge of the mobile. The user can flick the image towards the halo. When the image slides off the edge of the mobile it is disappears and reappears on the table.

Exam 2013 – B (MVC)



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

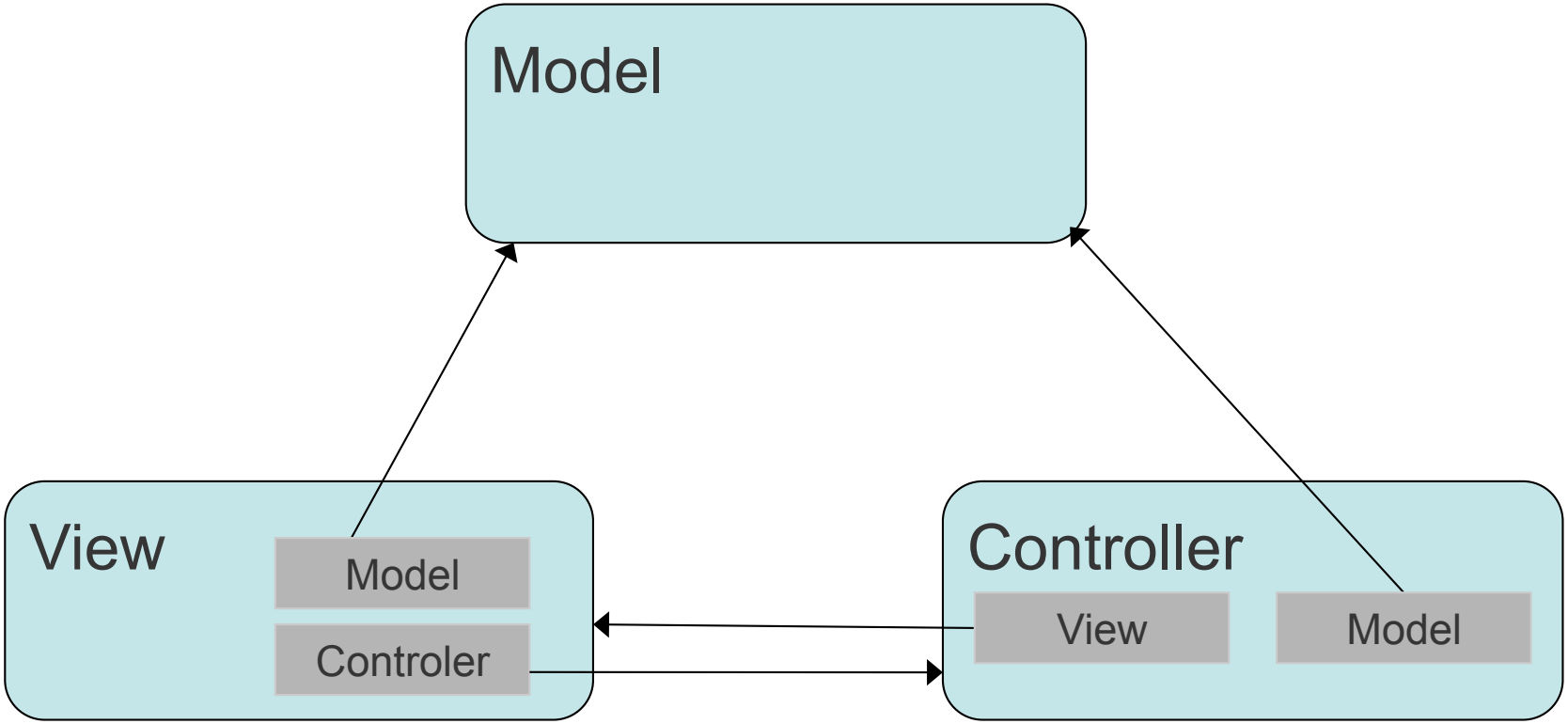
Exam 2013 – B (MVC)



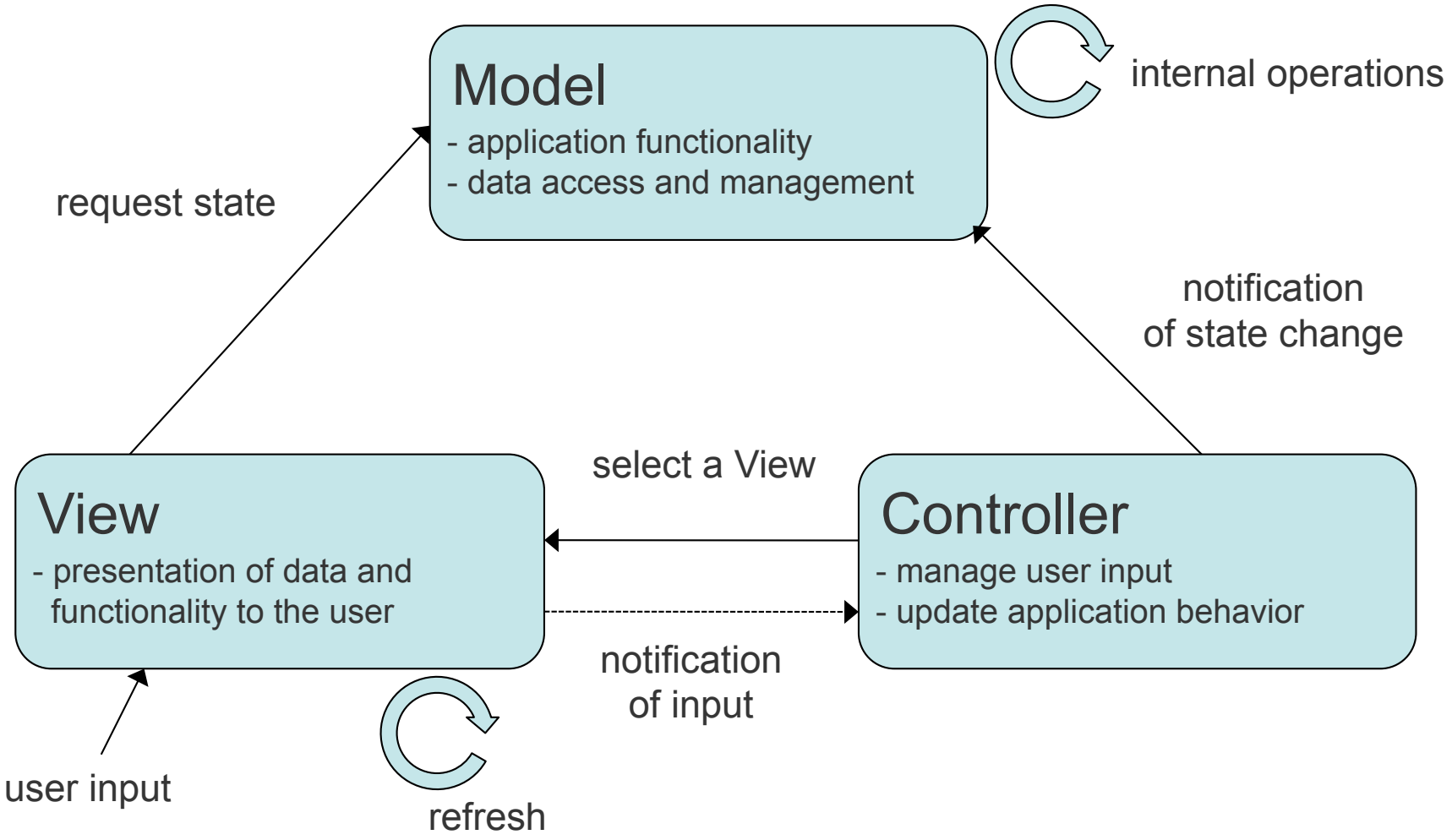
Where is the application? In the mobile device or in the tabletop?

Create a diagram that shows the interaction between the View, the Controller, and the Model for the mobile device and the tabletop application.

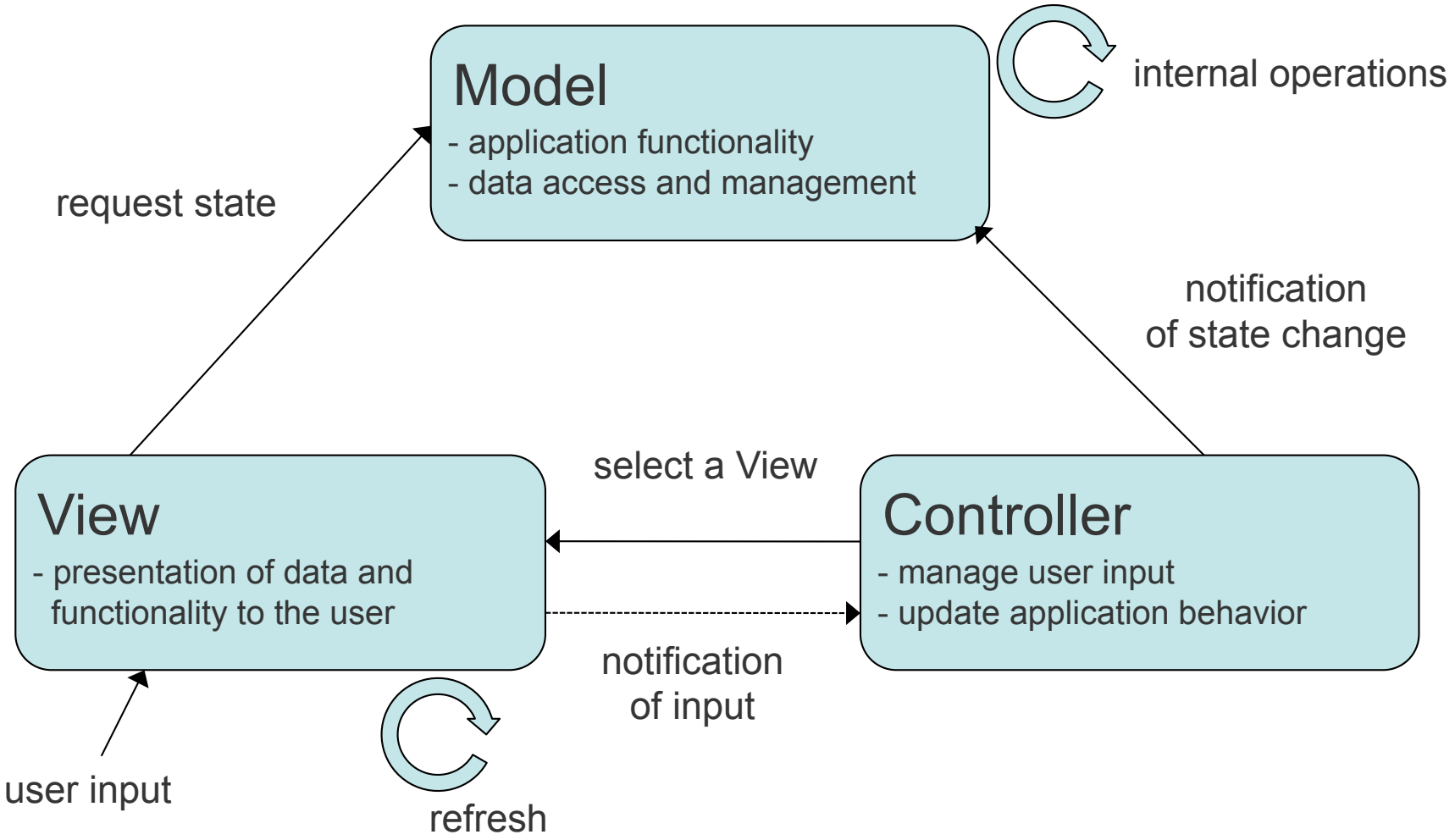
remember...



remember...


Model
- application functionality
- data access and management
internal operations
request state
notification
of state change
select a View
View
- presentation of data and
functionality to the user
Controller
- manage user input
- update application behavior
notification
of input
user input
refresh

remember...



But this is not the only possible flow of messages. For example, when the user moves the picture from the mobile phone to the halo, it is the controller of the tabletop application that detects the change. There is no notification from the view.

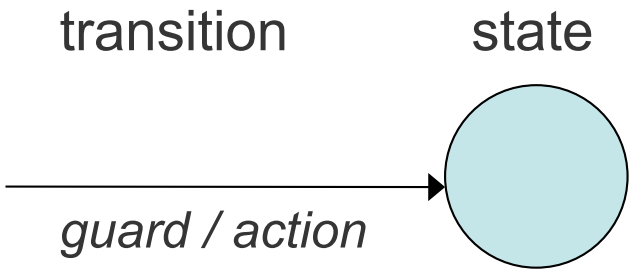
Modeling interation: state machines

State machines

Finite Automata

State = interaction state

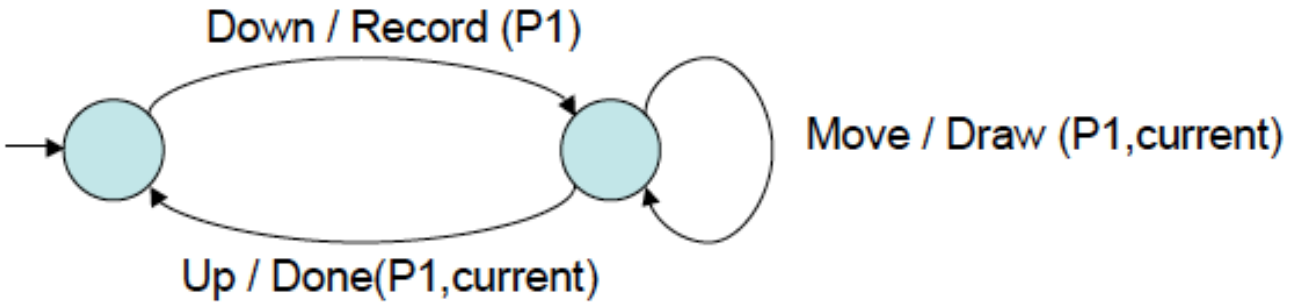
Transition = input events



State Machine

- boolean expressions of events & conditions associated to transitions (guard)
- actions associated to transitions (not always present)

Example:



The touch surface of a smartphone detects the contact point of a single finger.

The surface sends events when the finger touches the surface (FingerDownEvent), slides on the surface (FingerDragEvent) or leaves the surface (FingerUpEvent). Each event contains the coordinates of the contact point and a value that represents the time (in milliseconds) from the beginning of the session.

To facilitate the programming of interaction with graphical objects (widgets) of the UI of the smartphone, we decided to create a layer of events at the level of the widgets. More precisely, events at this level are produced when the finger touches a widget (OnWidgetEvent), taps on a widget (TapWidgetEvent), moves on a widget (DragWidgetEvent), stays on the widget for more than T = 300 ms (PauseWidgetEvent) or is raised from the widget (ReleaseWidgetEvent).

The touch surface of a smartphone detects the contact point of a single finger.

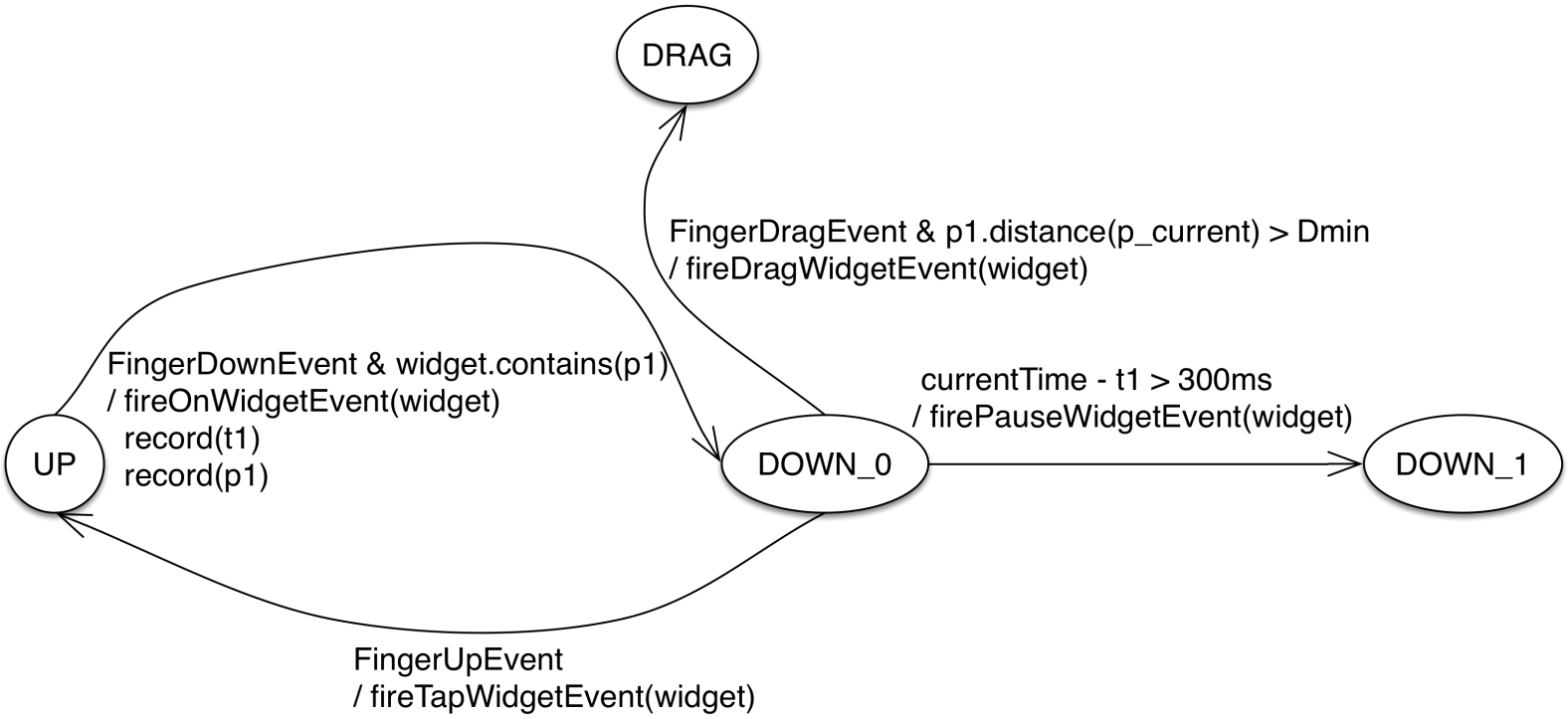
The surface sends events when the finger touches the surface (FingerDownEvent), slides on the surface (FingerDragEvent) or leaves the surface (FingerUpEvent). Each event contains the coordinates of the contact point and a value that represents the time (in milliseconds) from the beginning of the session.

To facilitate the programming of interaction with graphical objects (widgets) of the UI of the smartphone, we decided to create a layer of events at the level of the widgets. More precisely, events at this level are produced when the finger touches a widget (OnWidgetEvent), taps on a widget (TapWidgetEvent), moves on a widget (DragWidgetEvent), stays on the widget for more than T = 300 ms (PauseWidgetEvent) or is raised from the widget (ReleaseWidgetEvent).

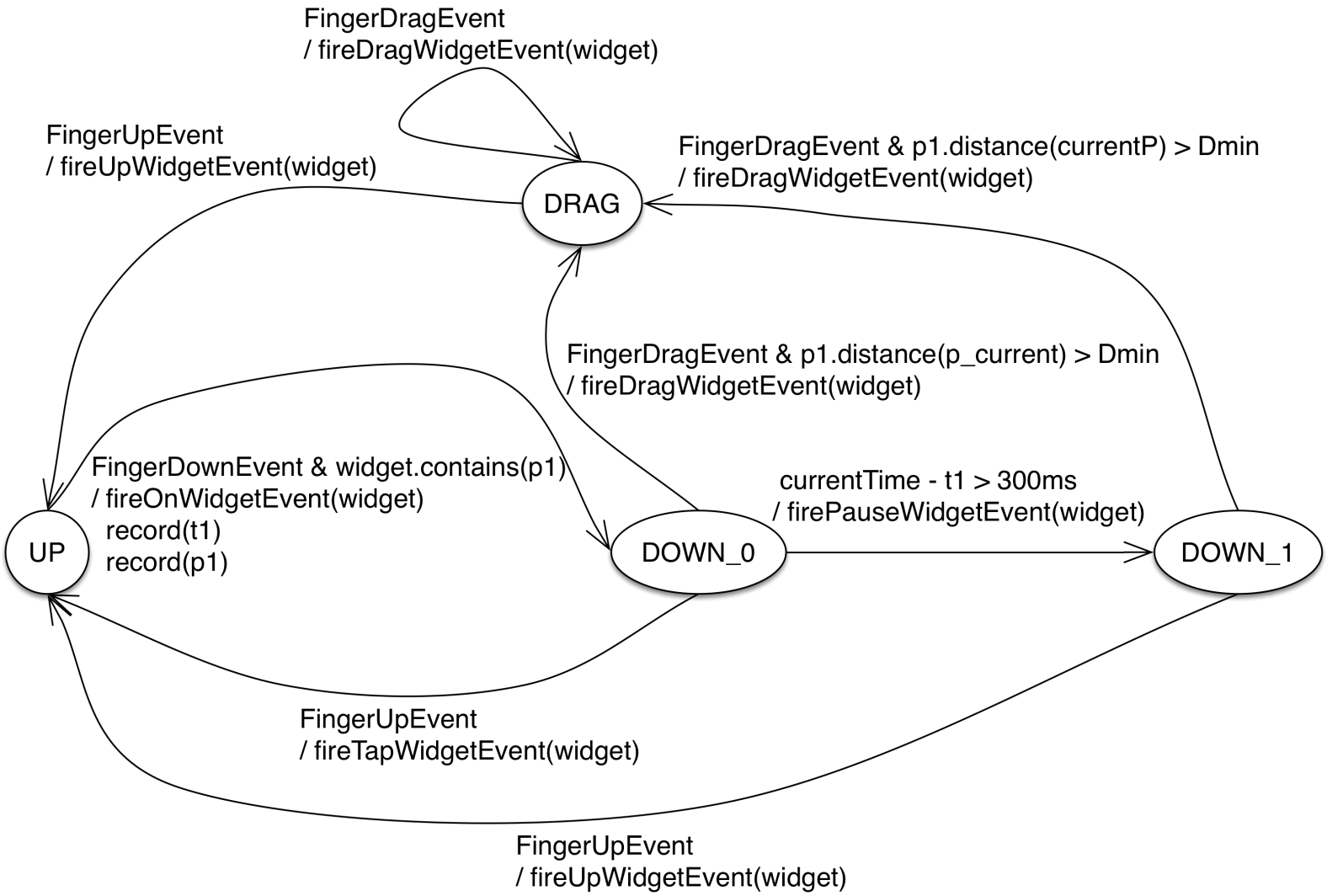
Define a state machine that manages the creation of the events OnWidgetEvent, TapWidgetEvent, DragWidgetEvent, PauseWidgetEvent, and ReleaseWidgetEvent.

Assume that a TapWidgetEvent is created when the delay between an event OnWidgetEvent and an event ReleaseWidgetEvent is less than T = 300 ms and there is no significant finger movement (Dmin < 5 mm).

Solution


DRAG
FingerDragEvent & p1.distance(p_current) > Dmin
/ fireDragWidgetEvent(widget)
FingerDownEvent & widget.contains(p1)
/ fireOnWidgetEvent(widget)
record(t1)
record(p1)
currentTime - t1 > 300ms
/ firePauseWidgetEvent(widget)
UP
DOWN_0
DOWN_1
FingerUpEvent
/ fireTapWidgetEvent(widget)

Solution


FingerDragEvent
/ fireDragWidgetEvent(widget)
FingerUpEvent
/ fireUpWidgetEvent(widget)
FingerDragEvent & p1.distance(currentP) > Dmin
/ fireDragWidgetEvent(widget)
DRAG
FingerDragEvent & p1.distance(p_current) > Dmin
/ fireDragWidgetEvent(widget)
FingerDownEvent & widget.contains(p1)
/ fireOnWidgetEvent(widget)
record(t1)
record(p1)
currentTime - t1 > 300ms
/ firePauseWidgetEvent(widget)
UP
DOWN_0
DOWN_1
FingerUpEvent
/ fireTapWidgetEvent(widget)
FingerUpEvent
/ fireUpWidgetEvent(widget)

Exam 2012-A

Create a state machine for the three techniques:

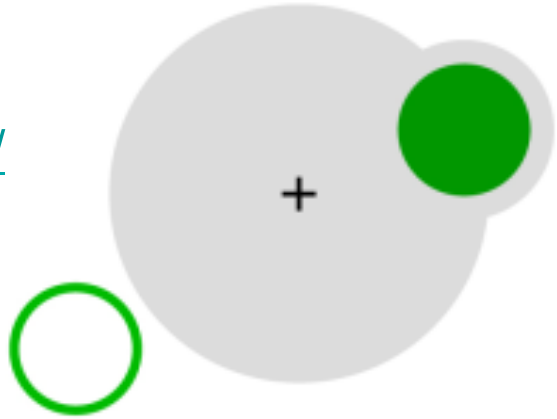
- Area cursor: area around cursor, can click on targets when inside
- Bubble cursor: closest target always selected on click
- Dynaspot:
  - area cursor of MAXWIDTH active when speed > MAXSPEED
  - when speed drops an animation starts that takes REDUCETIME during which the area of the cursor becomes smaller until 0 and the cursor becomes a regular cursor
  - during REDUCETIME we can accelerate, and go to area cursor

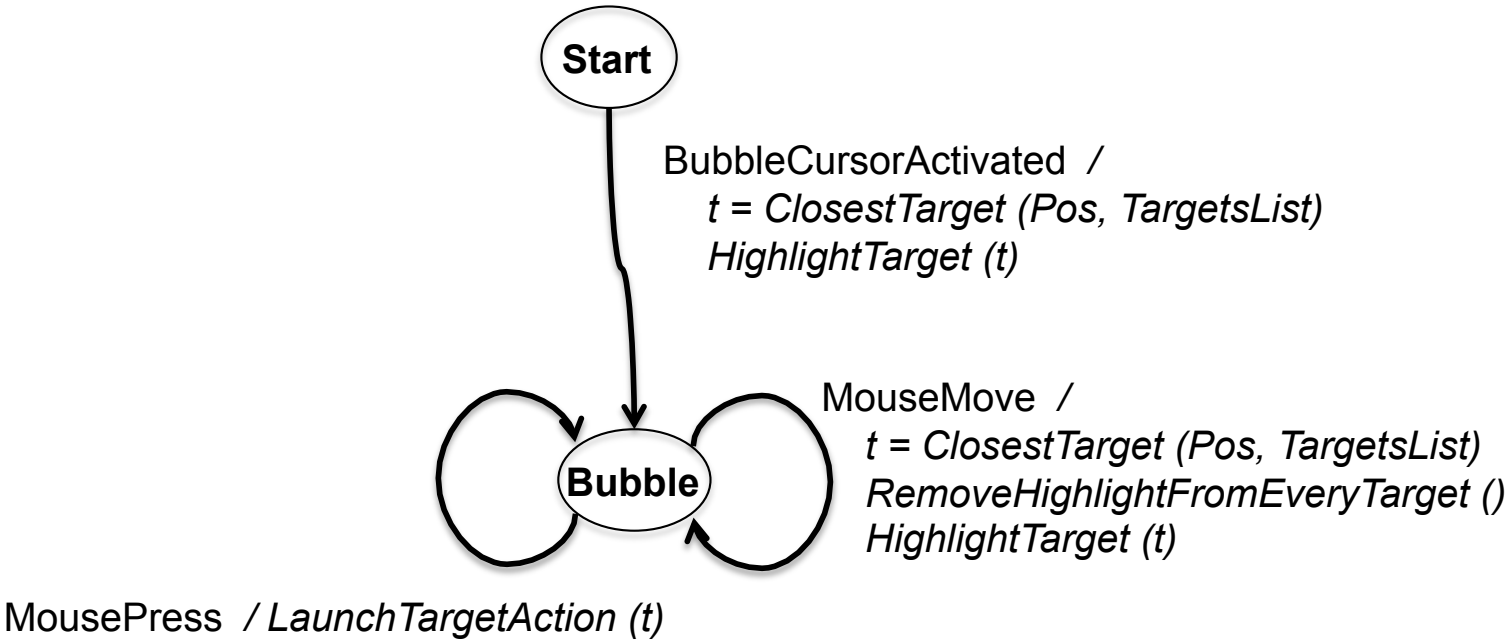
Can use:

- List = IntersectTargets (mousePos, WIDTH)
- Target = ClosestTarget (mousePos, List)
- ResetTimer(), GetTime (): start a timer and see time passed
- TimeOut (programmable) event to start after n sec with Arm(n), a fuction Disarm() cancels the Arm call if TimeOut has not started

Example

Bubble Cursor http://www.tovigrossman.com/BubbleCursor/



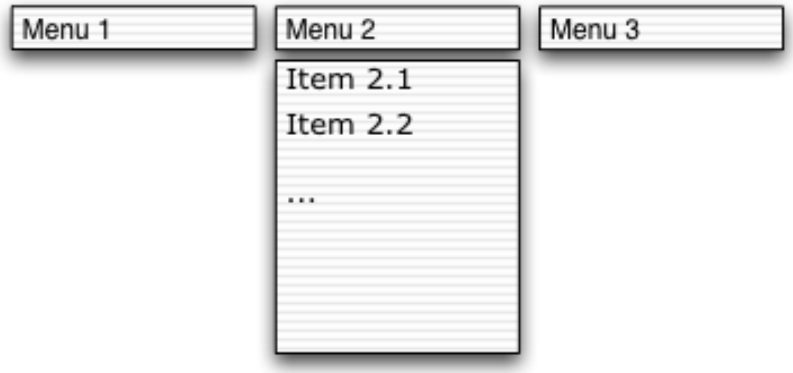


Other examples:

You may be asked to create a state machine that desribes a small set of multitouch gestures or a technique for mid-air interaction.

Models of user performance

Menus

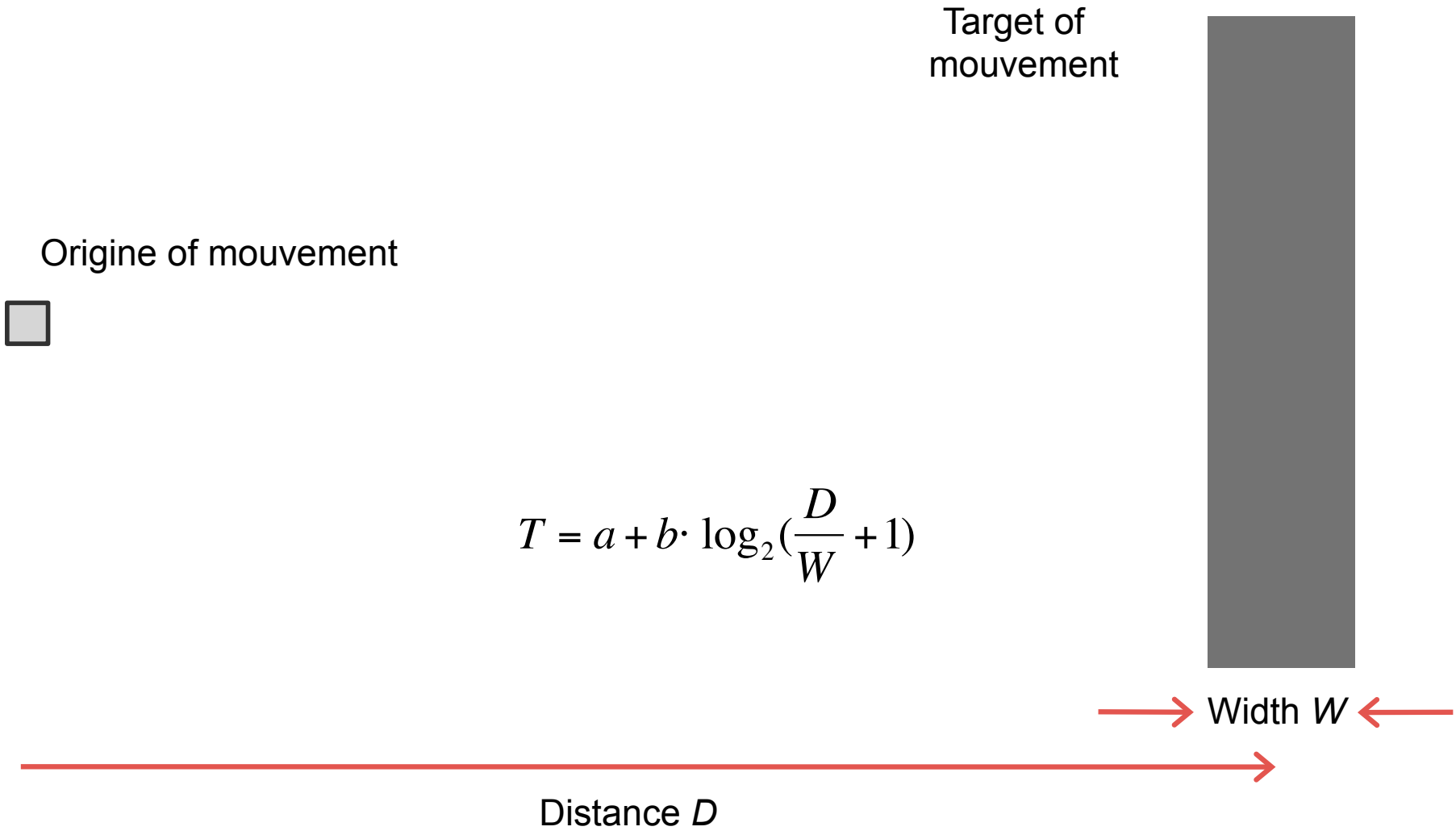


Discuss the advantages & disadvantages of different ways of improving a menu design:

- [1] Double the height of the menu items
- [2] Order the menu items in alphabetic order
- [3] Order the menu items by frequency of use
- [4] Group the items based on their semantics

Use criteria from human performance and models or laws to justify your analysis

Fitts' law



$$T = a + b \cdot \log_2\left(\frac{D}{W} + 1\right)$$

Fitts' law

Top better because all buttons are bigger

or

Bottom better because "back" is more frequent

Hick-Hyman's law

Describes the time to take a simple decision as a function of number of choices

$T = a + b \cdot log_2(n+1)$

n : number of choices
a, b : constants

We divide the total number of choices and re - search : binary search

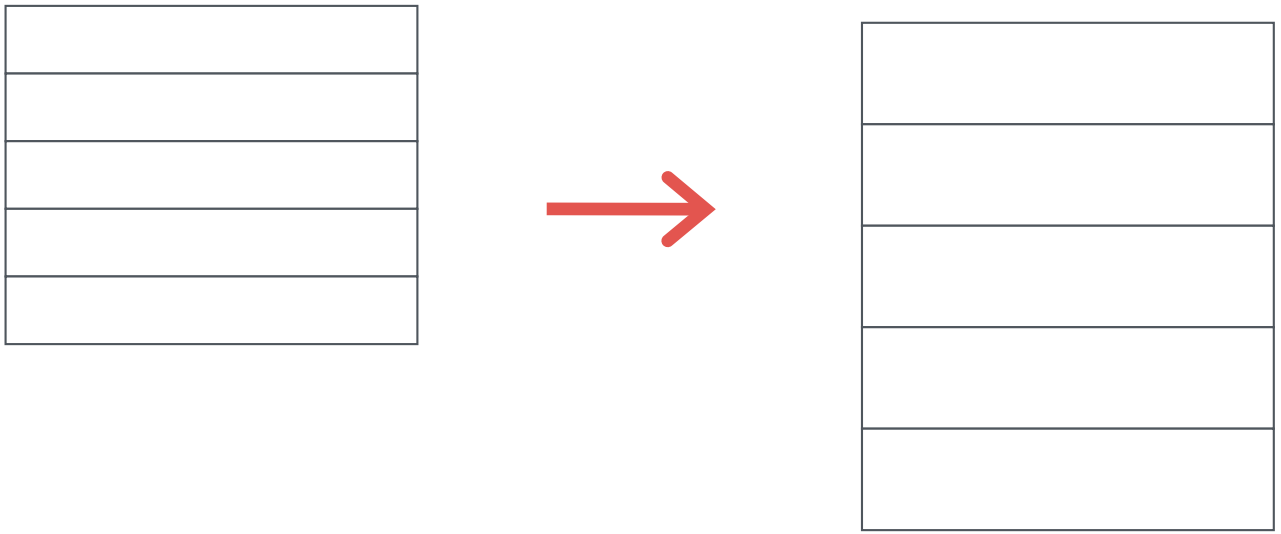
Other factors?

memorisation

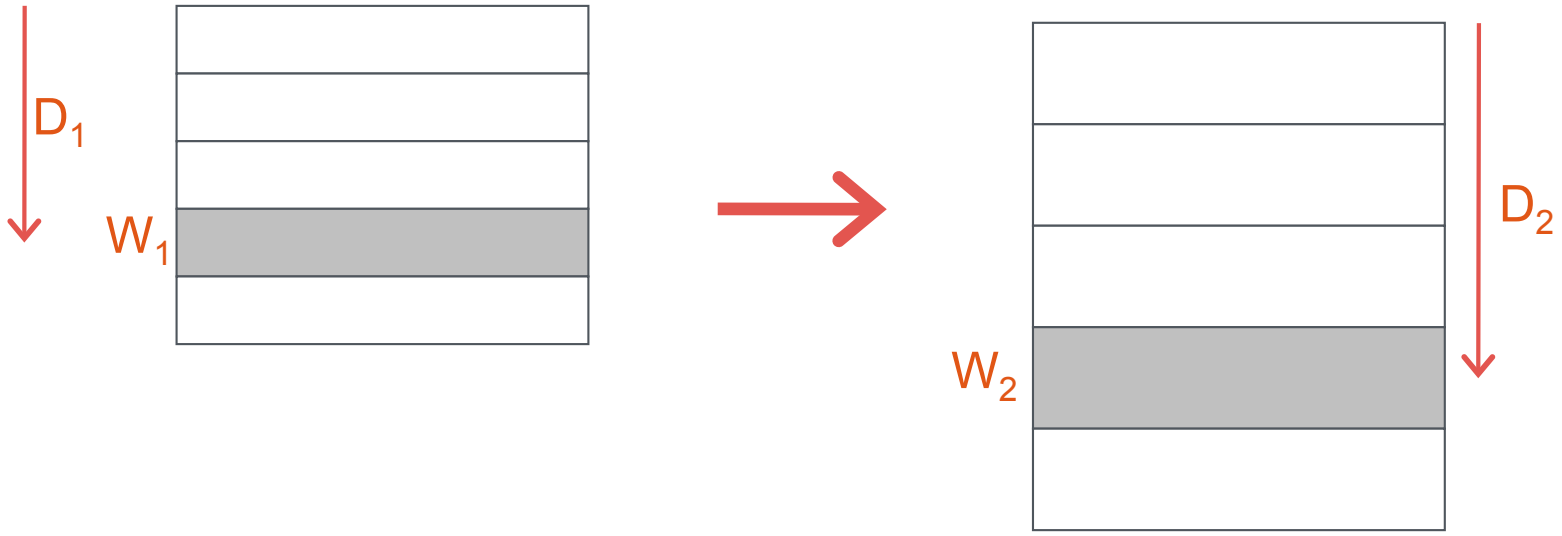
visual search

…

Enlarging the height of the items



Enlarging the height of the items

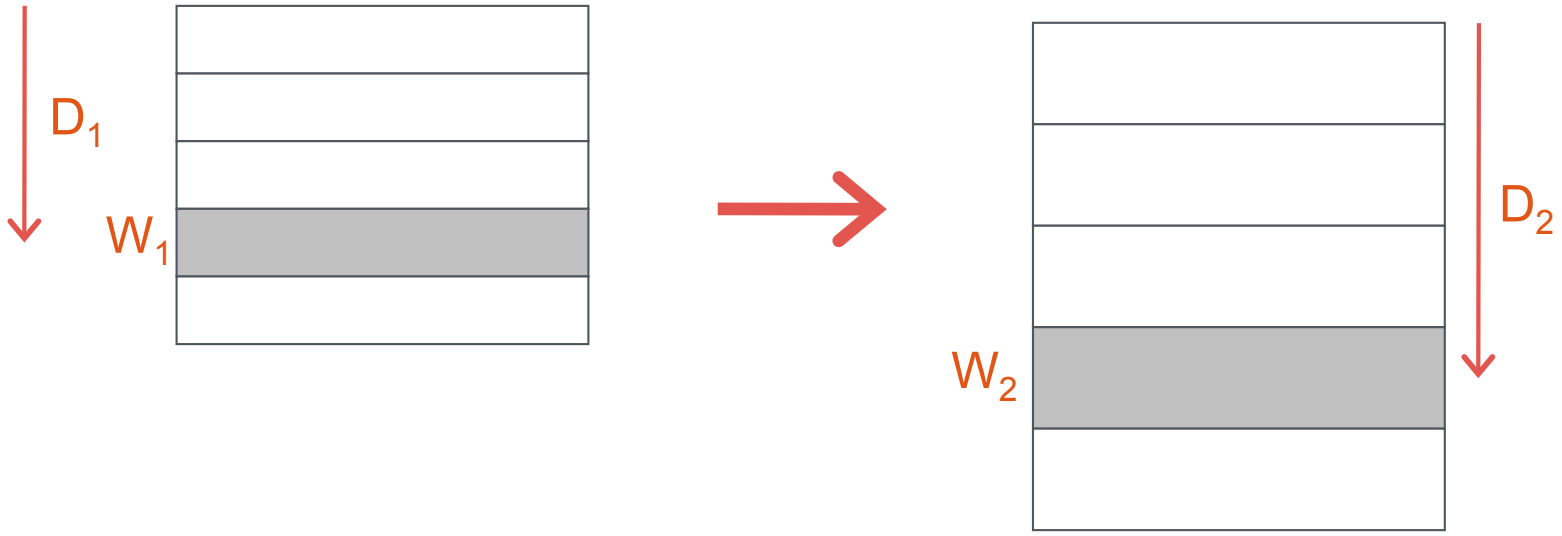


$$T_1 = a + b \cdot \log_2(\frac{D_1}{W_1} + 1)$$

$$T_2 = a + b \cdot \log_2(\frac{D_2}{W_2} + 1)$$

$$T_2 \overset{?}{<} T_1$$

Enlarging the height of the items



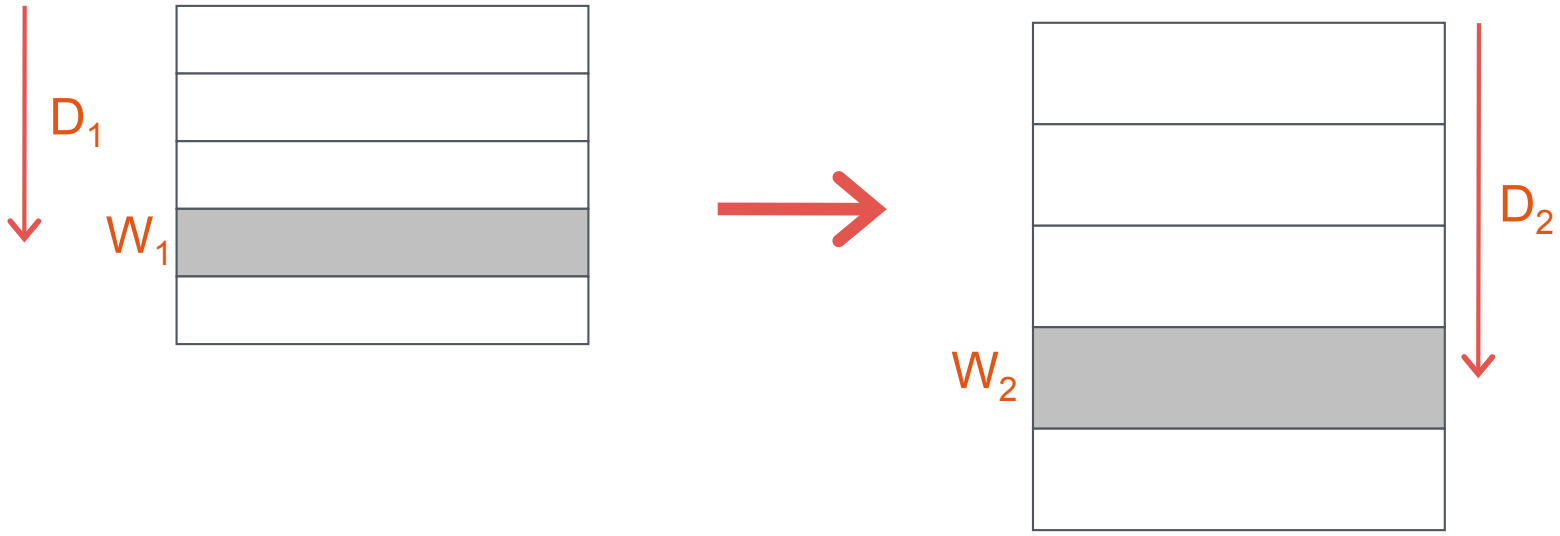
$$T_1 = a + b \cdot \log_2(\frac{D_1}{W_1} + 1)$$

$$T_2 = a + b \cdot \log_2(\frac{D_2}{W_2} + 1)$$

$$W_2 = \alpha \cdot W_1 \Rightarrow D_2 = \alpha \cdot D_1$$

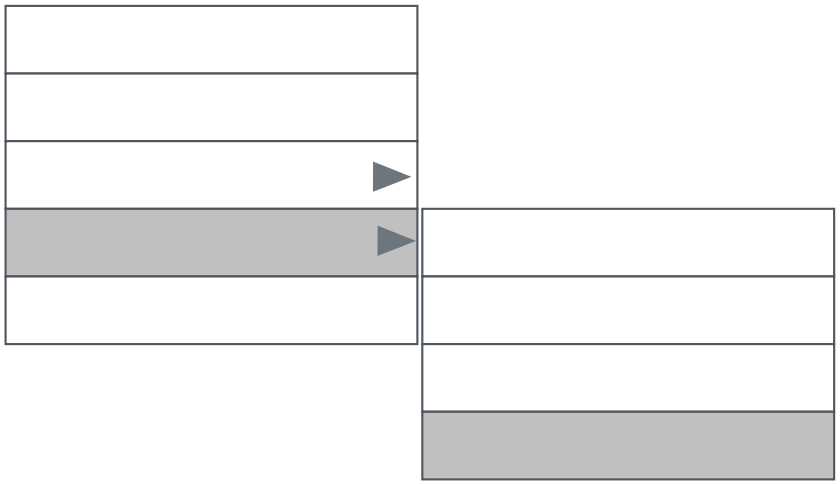
$$\Rightarrow T_2 = T_1$$

Enlarging the height of the items



Other factors that may affect performance ?

Enlarging the height of the items



How do things change if we also consider submenus?

Alphabetical order

Find France !

Allemagne
Belarus
Bulgarie
Espagne
France
Italie
Irlande
Islande
Lettonie
Norvège
Pologne

logarithmic

Norvège
Irlande
Islande
Lettonie
Allemagne
Bulgarie
Italie
Pologne
Espagne
France
Belarus

linear

Alphabetical order

Allemagne
Belarus
Bulgarie
Espagne
France
Italie
Irlande
Islande
Lettonie
Norvège
Pologne

Disadvantages?

Selection frequency

Benefits acording to Fitts' law ?

$$\overline{T} = \sum_{i=1}^{N} (a + b \cdot \log_2(\frac{D_i}{W} + 1)) / N$$

Is the mean movement time different?

Selection frequency

And the time for visual search ?

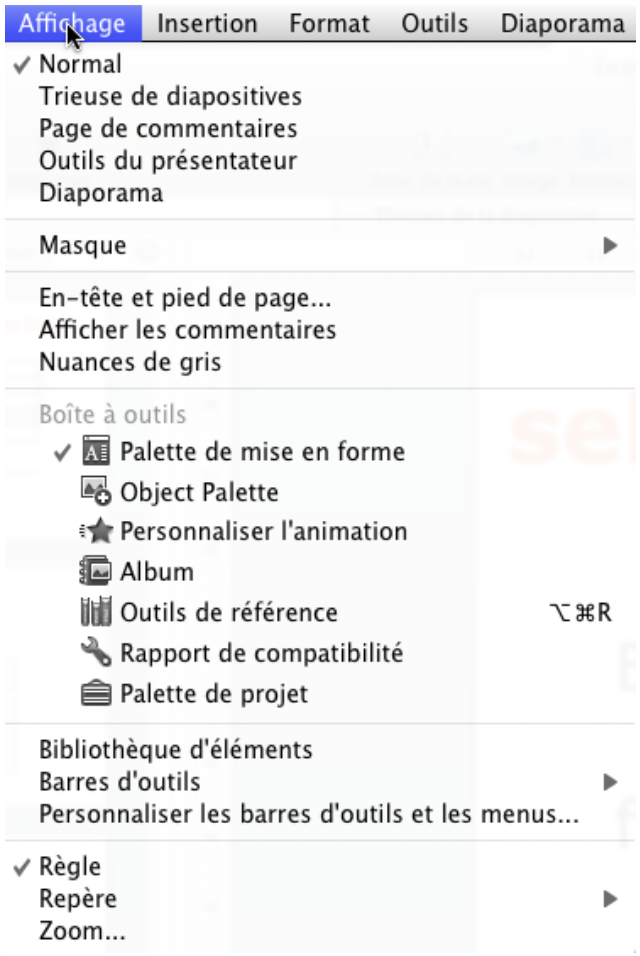
factors

... before and after the « stabilization » of frequencies
... distribution of selections

and memorization of the item position ?

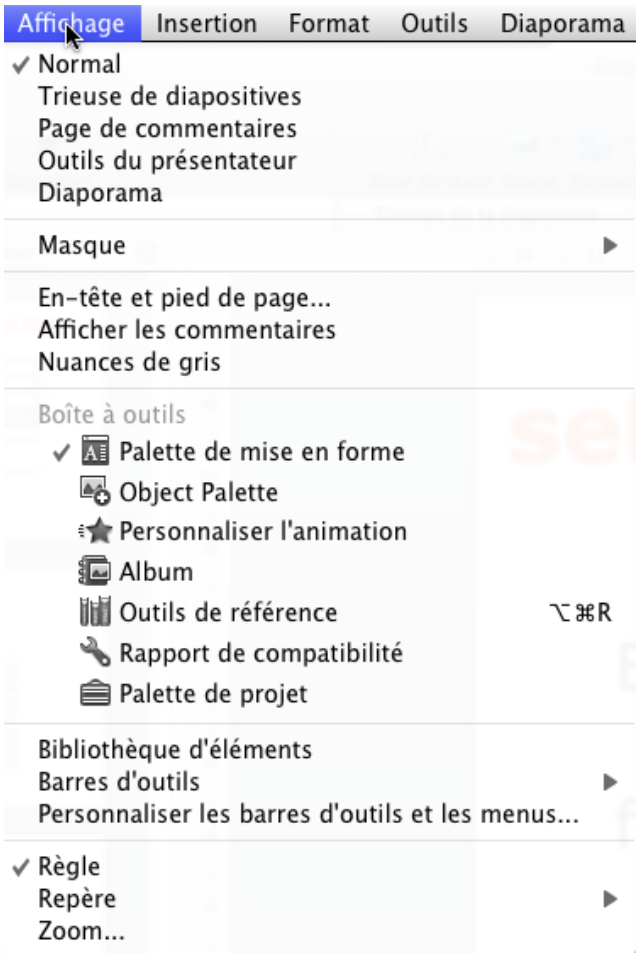
Semantic grouping

Advantages and disadvantages ?



Semantic grouping

Which gestalt principles have been applied to this menu design?



Gestalt laws of perception

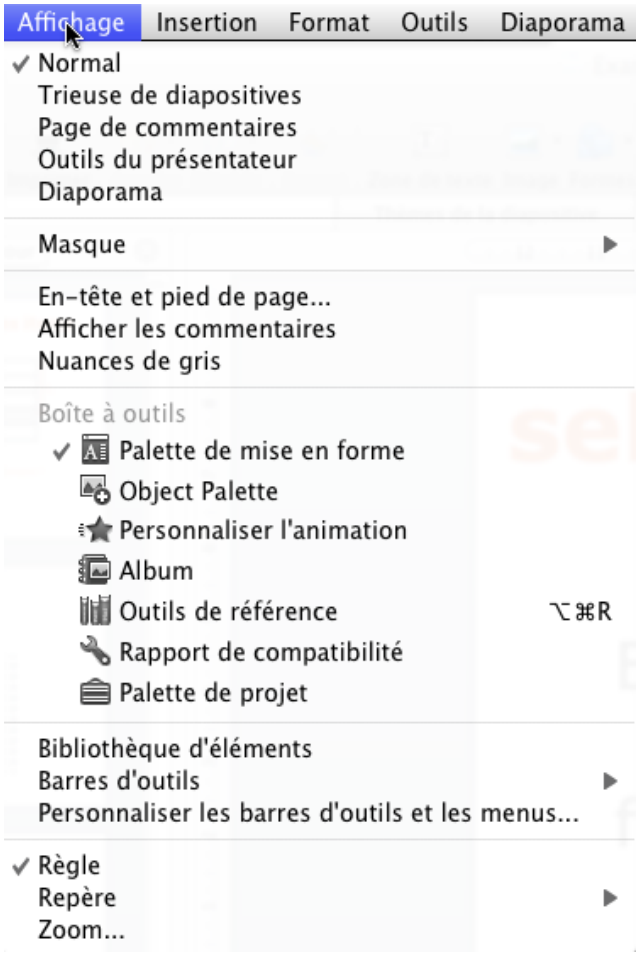
Continuity
Proximity
Similarity
Symmetry
Closure
Common Fate

etc...

Gestalt laws of perception

Continuity
Proximity
Similarity
Symmetry
Closure
Common Fate

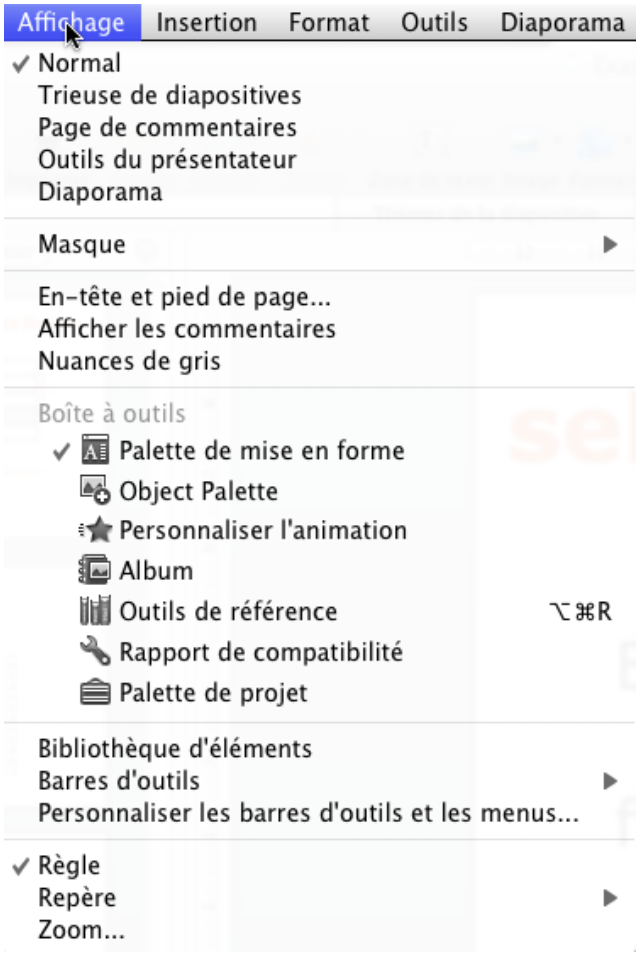
etc...



Gestalt laws of perception

Continuity ✔
Proximity ✔
Similarity ✔
Symmetry
Closure ✔
Common Fate

etc...



Design, sketching, prototyping

We may be asked to design/prototype a user interface or a specific interaction technique, e.g., one that is based on touch or gestures.

You won't have to write code but you can sketch it!

You may be explicitely asked to respect or follow some design guidelines or you may be asked to design it for certain users, e.g., novices or experts.

You may be asked to design it to address a specific problem, e.g., increase accuracy or reduce occlusion.

Example

Design an editor that helps UI designers to easily create state machines.

Specify the main functionality of the application and sketch its main screen.

You may be asked to create a storyboard to describe an interaction sequence.

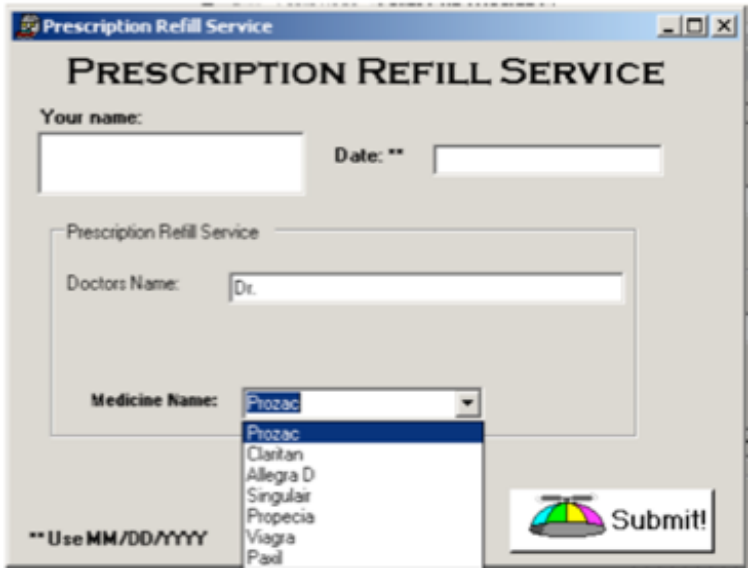
You may be asked to create a storyboard to describe an interaction sequence.

Example: Draw a storyboard that describes the main steps of creating a state machine with your proposed interface.

You may be asked to redesign a user interface to reduce problems that you identified or extend it to satisfy some new user needs.

Exam 2013

Critique the interface by using three usability criteria (next lecture) and redesign it to fix the problems



Other examples

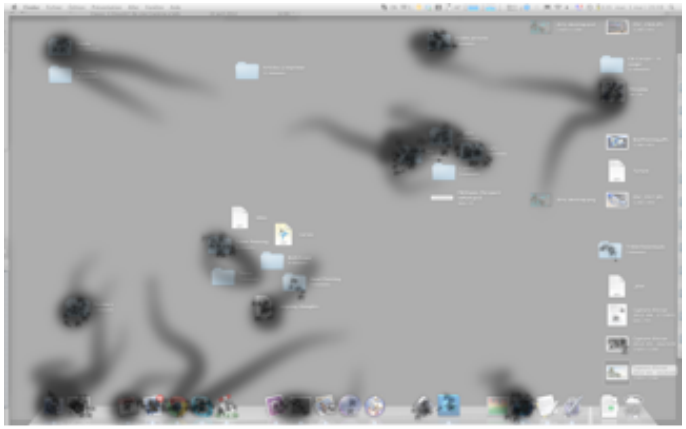
exercise 2012

An adaptive interaction technique adjusts CD gain of different screen areas depending on how often and for how long users visit them (at start all CD = 1)

Assuming that very visited areas where the cursor moves slowly are dense, we reduce CD gain to make interaction more detailed

We increase the CD of unvisited areas or areas the cursor moves fast as they likely do not contain interesting targets

We show visual feedback of “dense” areas and the algorithm continues to adapt



What are drawbacks of the technique and solutions?

exercise 2012-B : Feedback/Feedfwd

Both gray-out buttons for impossible actions (“Stop”) and list of history

Where users can input text is clear (text box)

Clicking options clearer in 2 (buttons look like buttons and not images)

2 has a better grouping of actions to guide users in what to use (e.g. back very close to fwd)

In 2 attention is attracted by larger button

exercise 2012-B : Gestalt laws

Similarity: the first 2 icons similar Shape and Color and different from the rest because of colour
Proximity: not really followed
Bad Closure: targets not grouped and boundaries not visible
Symmetry: The backward and forward buttons

Similarity: the first 2 icons similar in Shape and Color BUT their actual button shape is different, making grouping weaker
Similarity: all buttons look like buttons (including the "g" which in top does not look like a widget)
Proximity: the first 2 buttons are closer together than all the rest, thus a group
Closure: the first 2 buttons are inside an enclosure (making them grouped together as related). All other buttons also enclosed to show individual functionality and clear boundaries.