

# Interactive Systems (ISY)

## Auditorium Exercise 02

# Lectures

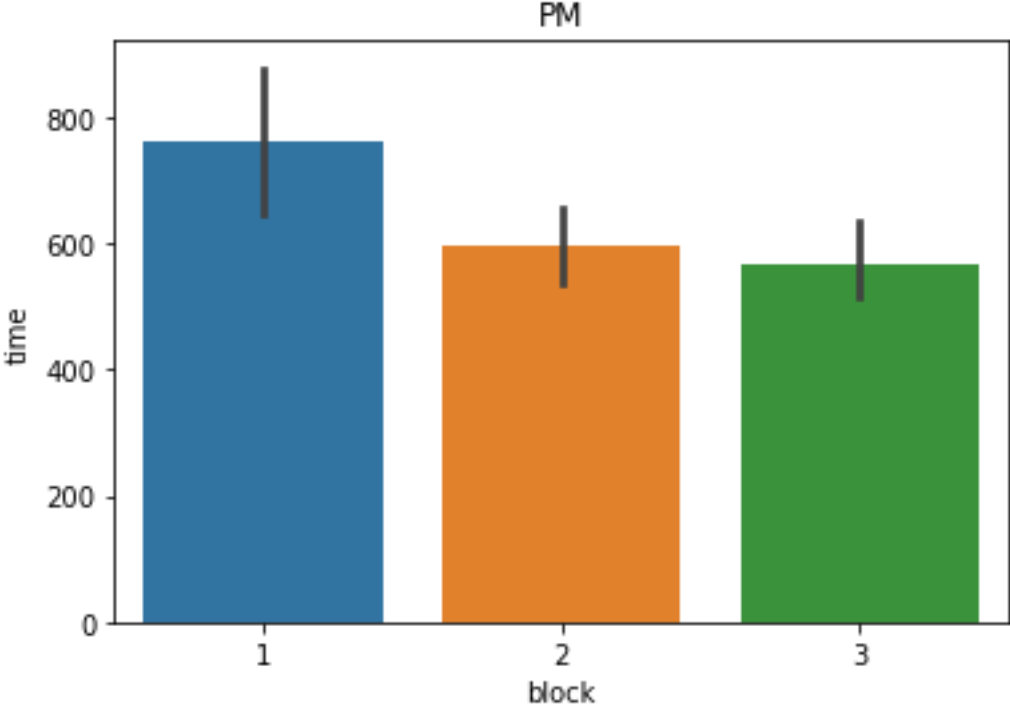
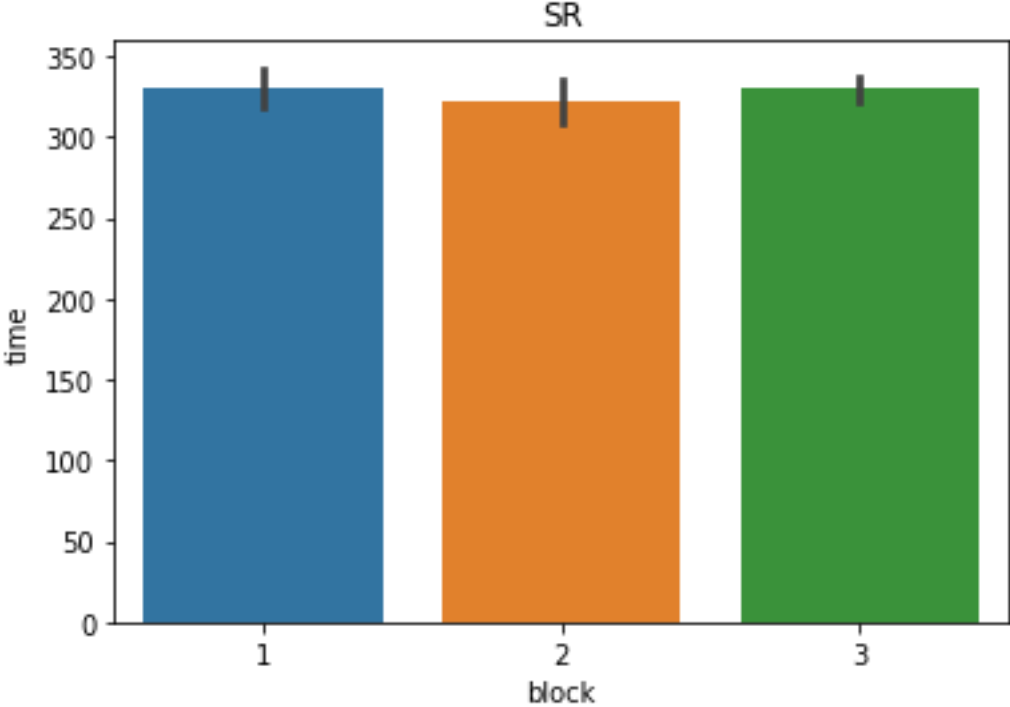
Session	Date	Topic	Details
1	2.4.	Introduction	human performance, empirical research, modeling
2	9.4.	Interaction elements	input devices, interaction elements, states, layouts
	16.4.	Event handling	events, bindings, reactive programming, scene graph
3	23.4.	Scene graphs	event delivery, coordinate systems, nodes, animation, concurrency
4	30.4.	Interaction techniques	alignment and pointing techniques
5	7.5.	Interaction techniques	
6	14.5.	Web-based user interfaces	document object model, client-server issues
	21.5.	Pfingstwoche	
7	28.5.	Web-based user interfaces	reactive Programming for the Web
8	4.6.	Experiments and data analysis	designing experiments, hypothesis testing
9	11.6.	Modeling interaction	descriptive and predictive models, keystroke-level model, regression
10	18.6.	Visualization	visual encodings, perceptual accuracy, treemaps, dynamic queries
11	25.6.	Human-Centered AI	introduction to human-centered AI, human control and automation, examples
12	2.7.	Deep learning in HCI	guidelines for human-AI interaction, neural networks
13	9.7.	Deep learning in HCI	convolutional and recurrent NNs, face recognition, gesture recognition

# ASSIGNMENT 1

# Assignment 01

- Korrekturen und Punkte können im Upload Tool eingesehen werden
  - Entweder nach Abgabe des nächsten Assignment
  - Oder durch “previous submissions” Button in Kursübersicht
- N submissions

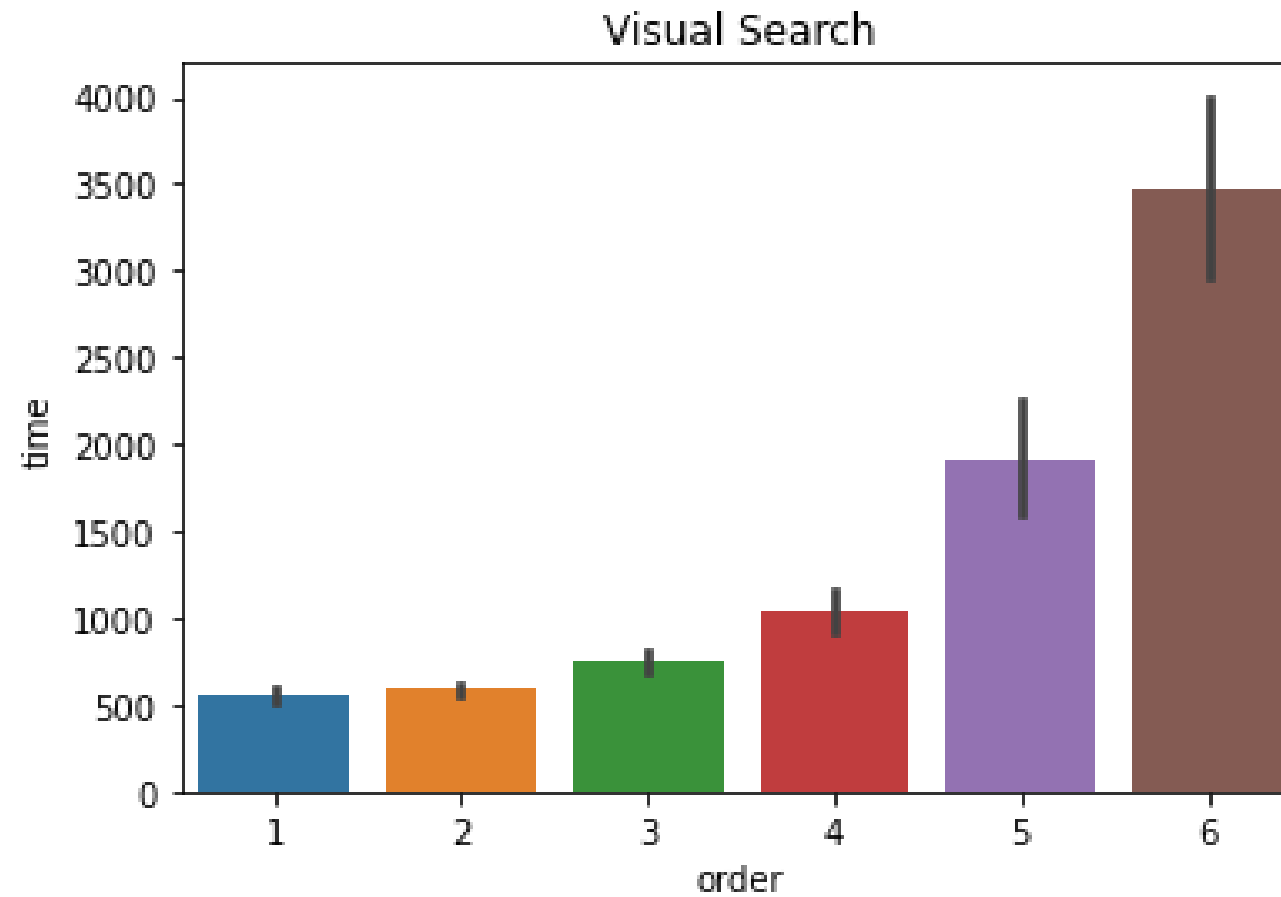
# Exercise 1



# Fragen

- Zeigt sich ein Lerneffekt?
- Wie lassen sich diese Ergebnisse erklären?

## Exercise 2



## Frage

- Gab es bei euren Ergebnissen starke Abweichungen?



Praktisches Beispiel

# AUSWERTUNG GESAMMELTER ERGEBNISSE

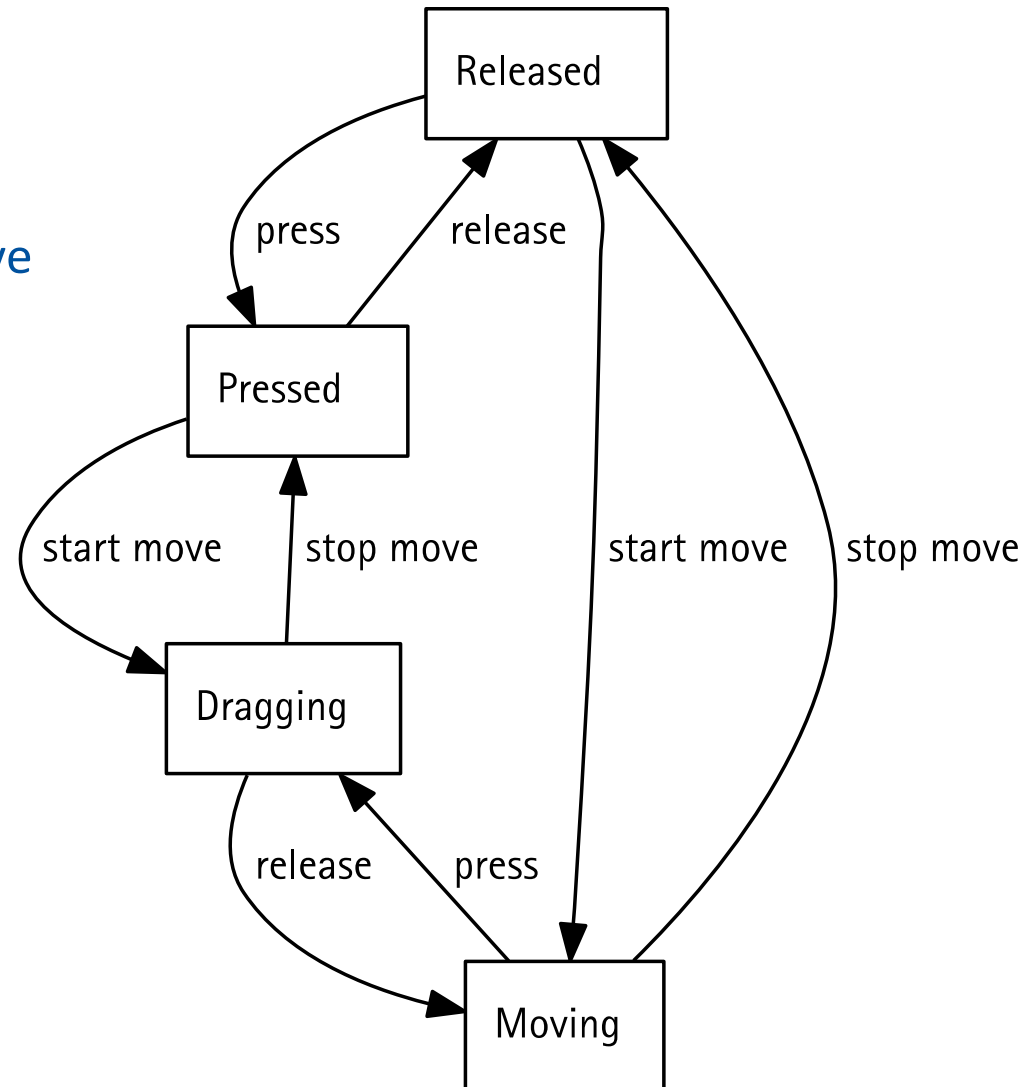
# Auswertung gesammelter Ergebnisse

- Laden Sie sich dafür eine Kopie des Spreadsheets als CSV-Datei herunter:  
[https://docs.google.com/spreadsheets/d/1WBpnhI\\_C96u3IQ5Qpulw1csWjGcQet-TiYBgXsmMhlo](https://docs.google.com/spreadsheets/d/1WBpnhI_C96u3IQ5Qpulw1csWjGcQet-TiYBgXsmMhlo)
  - Lesen Sie die CSV-Datei in Jupyter Notebook mit Pandas in ein Dataframe ein.
- Berechnen Sie über alle Teilnehmer das mittlere Alter und die Standardabweichung. Wie viele weibliche, diverse und männliche Teilnehmer gab es? Wie viele Teilnehmer sind Rechts-, Links- oder Beidhänder? Wie viele Teilnehmer haben an den jeweiligen Experimenten teilgenommen?
  - Sind die Ergebnisse repräsentativ für die Bevölkerung?
- Lesen Sie sich in der folgenden Veröffentlichung den Abschnitt „Participants“ durch:  
<https://www2.hci.uni-hannover.de/papers/pfeiffer2015CHICruise.pdf>
  - Was sagt der Abschnitt aus? Warum ist dieser Abschnitt wichtig?
- Stellen Sie die Reaktionszeiten für jeden einzelnen Teilnehmer für das Experiment SR graphisch mit Seaborn in einem Plot dar. Wählen Sie eine Plot-Variante (barplot, boxplot oder violinplot) aus und begründen Sie, warum Sie sich für diese entschieden haben.

# LECTURE RECAP

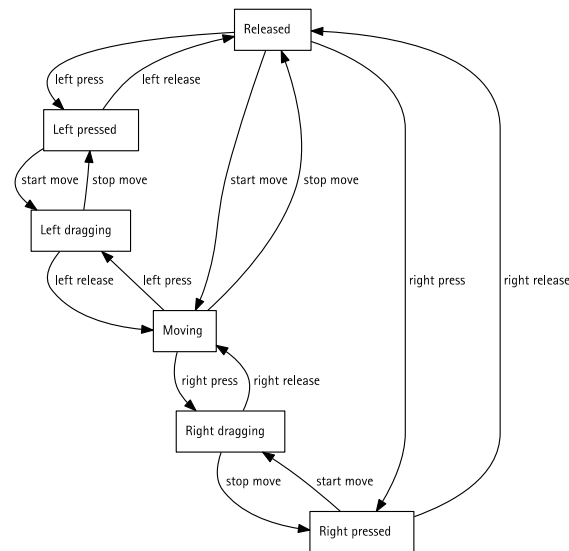
# Single-Button Mouse

- Actions
  - Press, release, start move, stop move
- States
  - Pressed, released, moving, dragging
- State transition diagram



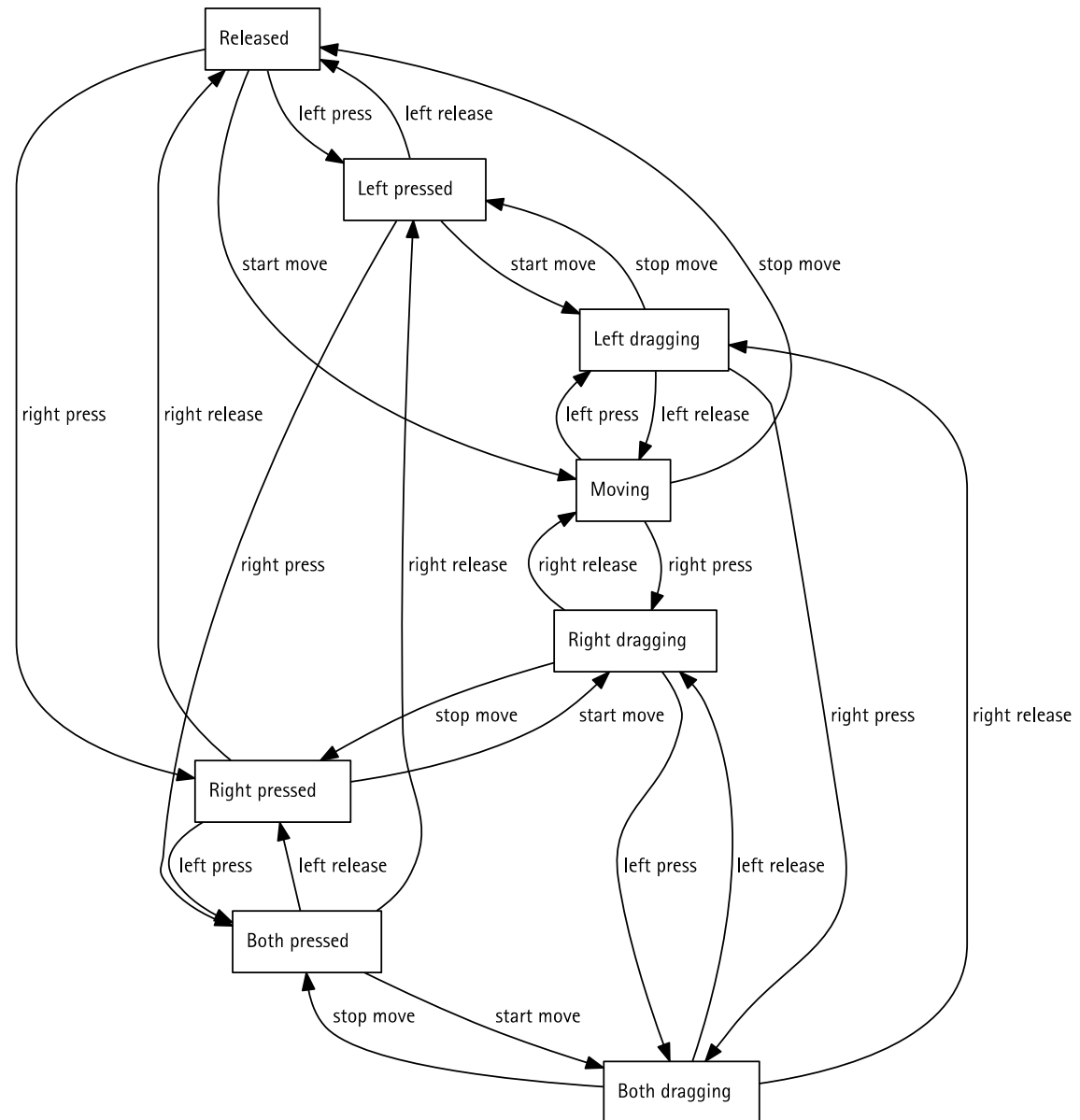
# Two-Button Mouse

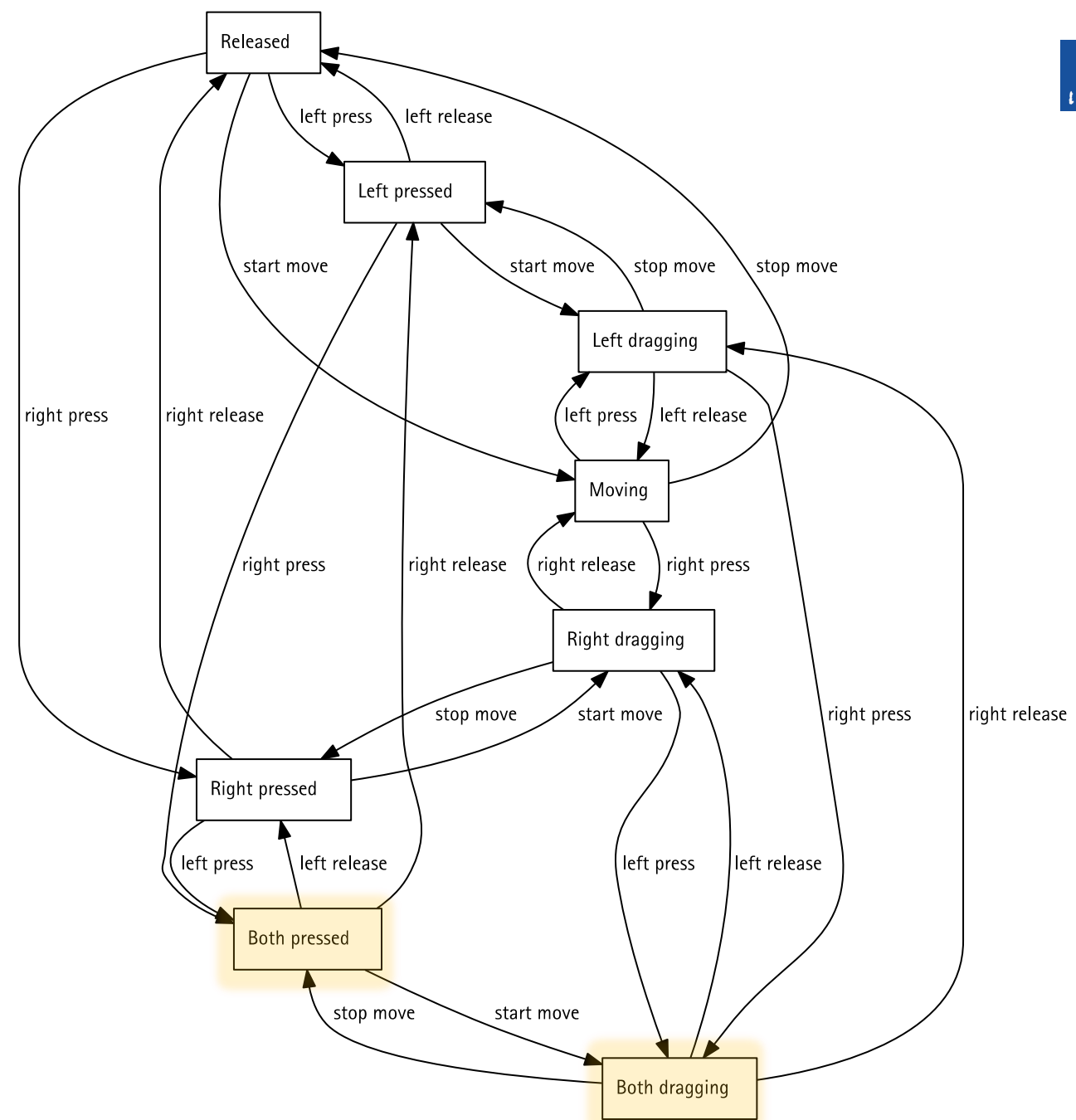
- 6 Actions
  - Left press, left release, right press, right release, start move, stop move
- 6 States
  - Left/right pressed, released, moving, left/right dragging
- State transition diagram



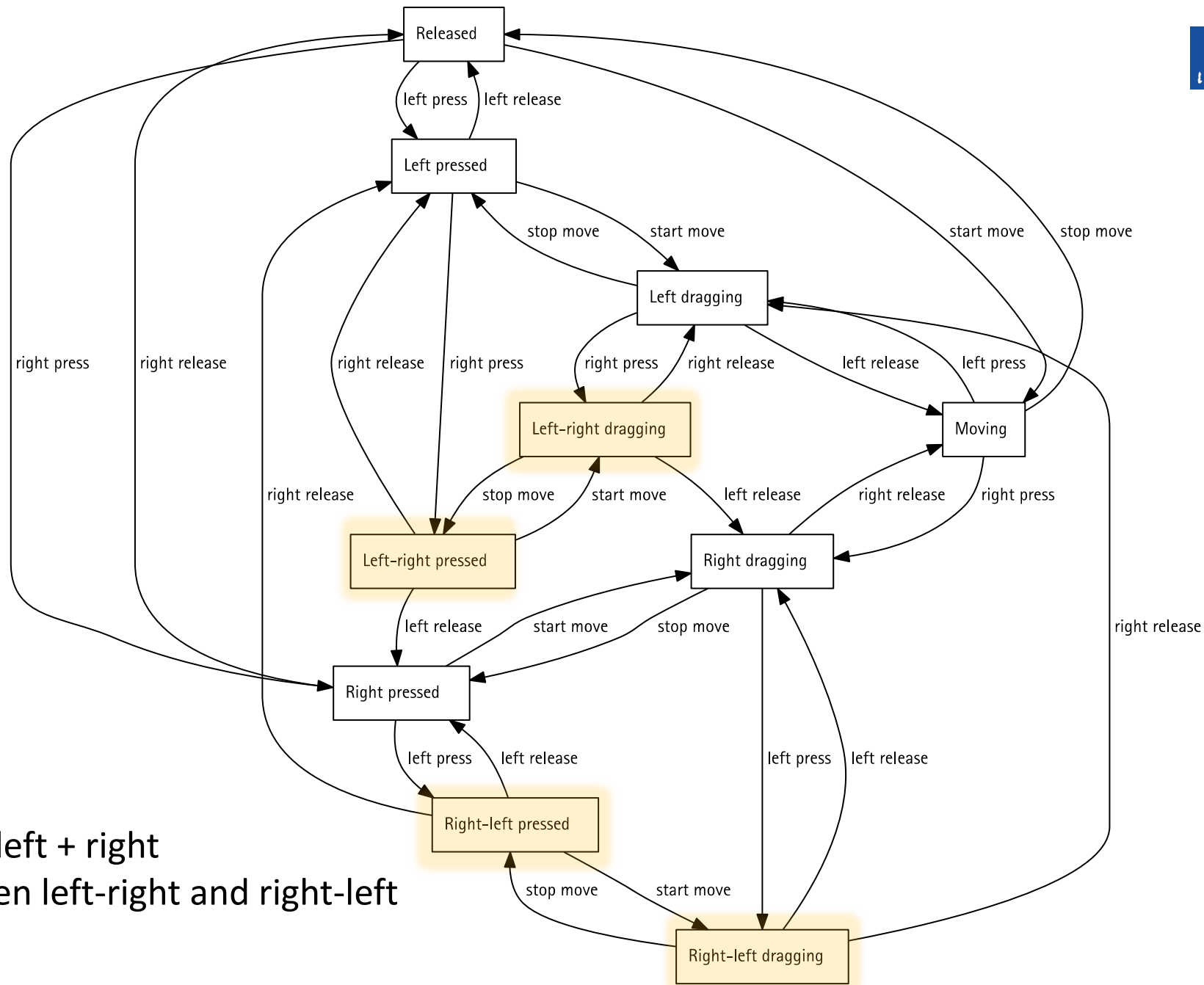
## Two Mouse Buttons

- Distinction between
  - Press left-then-right
  - Press right-then-left
- Useful?





+ dragging with both left + right



+ dragging with both left + right  
+ differentiate between left-right and right-left



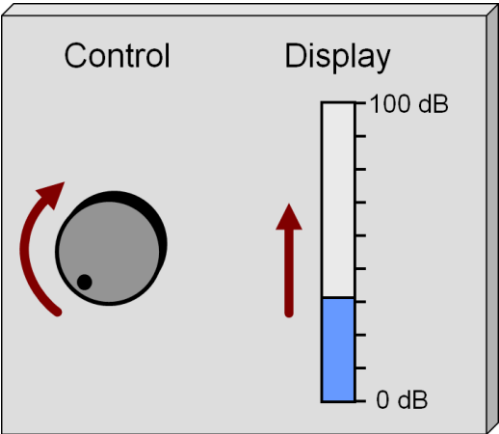
## Two Mouse Buttons

- Distinction between
  - Press left-then-right
  - Press right-then-left
- Typical Applications?



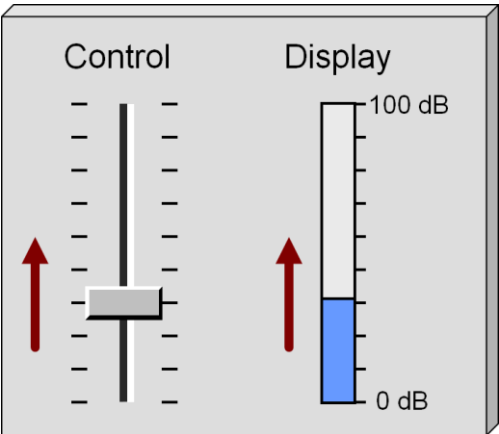
# Spatial Relationships

Learned  
relationship



DOF	Control	Display
x		
y		+
z		
$\theta_x$		
$\theta_y$		
$\theta_z$	+	

Natural  
relationship



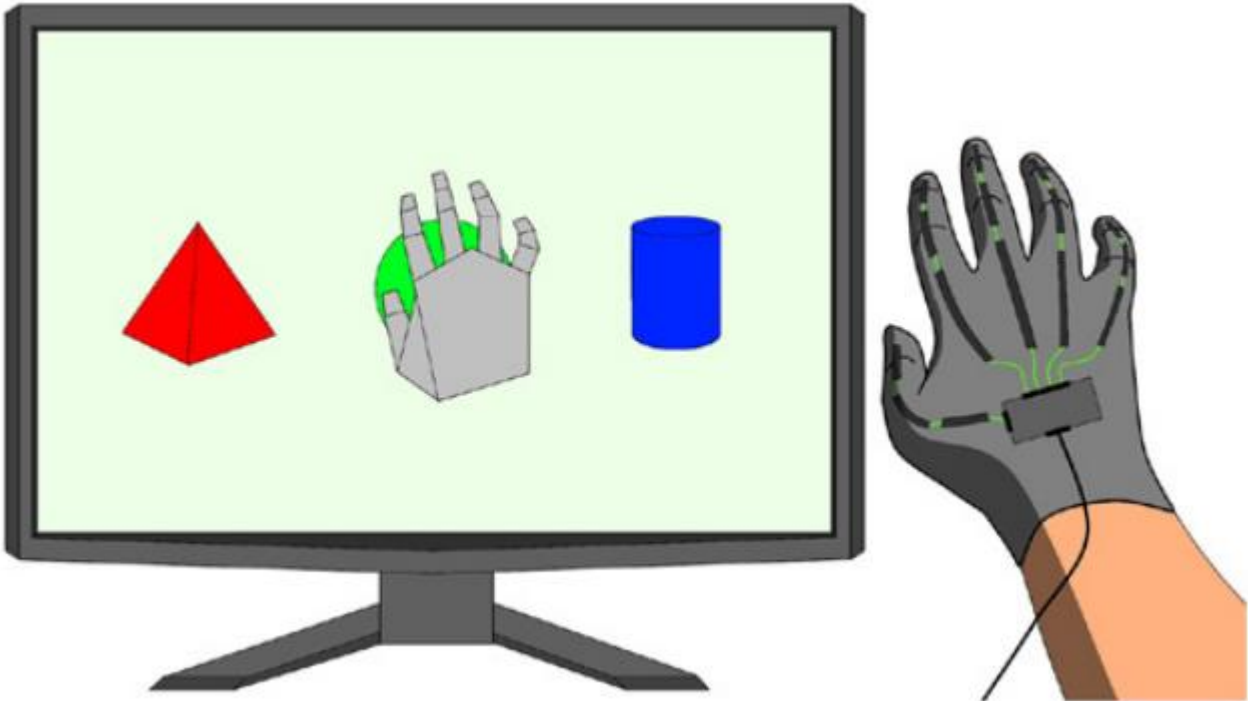
DOF	Control	Display
x		
y	+	+
z		
$\theta_x$		
$\theta_y$		
$\theta_z$		

MacKenzie: Human-Computer Interaction - An Empirical Research Perspective.

# Spatial Relationships

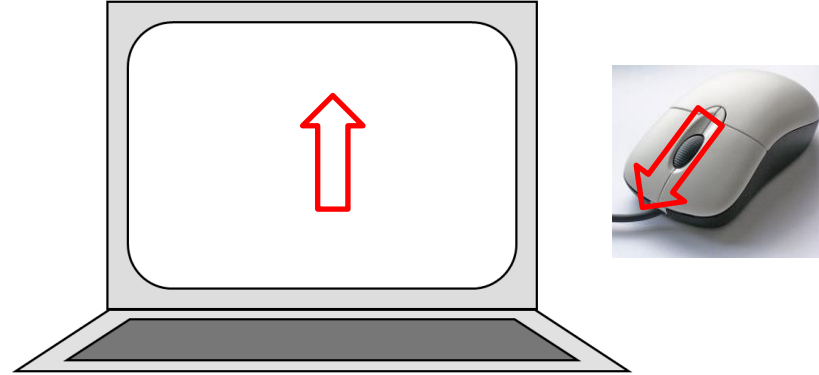
- Is this possible?

DOF	Control	Display
x	+ ●————● +	●————● +
y	+ ●————● +	●————● +
z	+ ●————● +	●————● +
$\theta_x$	+ ●————● +	●————● +
$\theta_y$	+ ●————● +	●————● +
$\theta_z$	+ ●————● +	●————● +

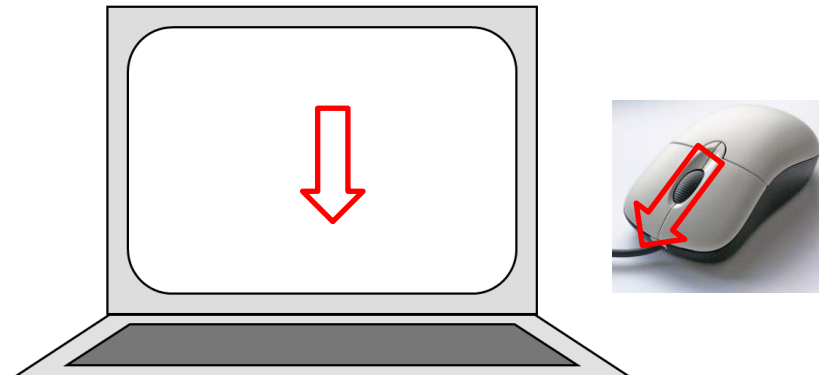


## Which spatial relationships is more natural? Why?

- A: Wheel forward: content moves up

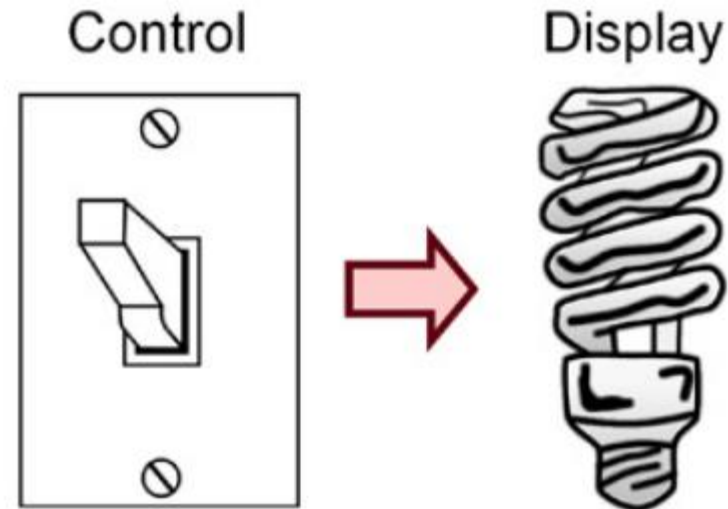


- B: Wheel forward: content moves down



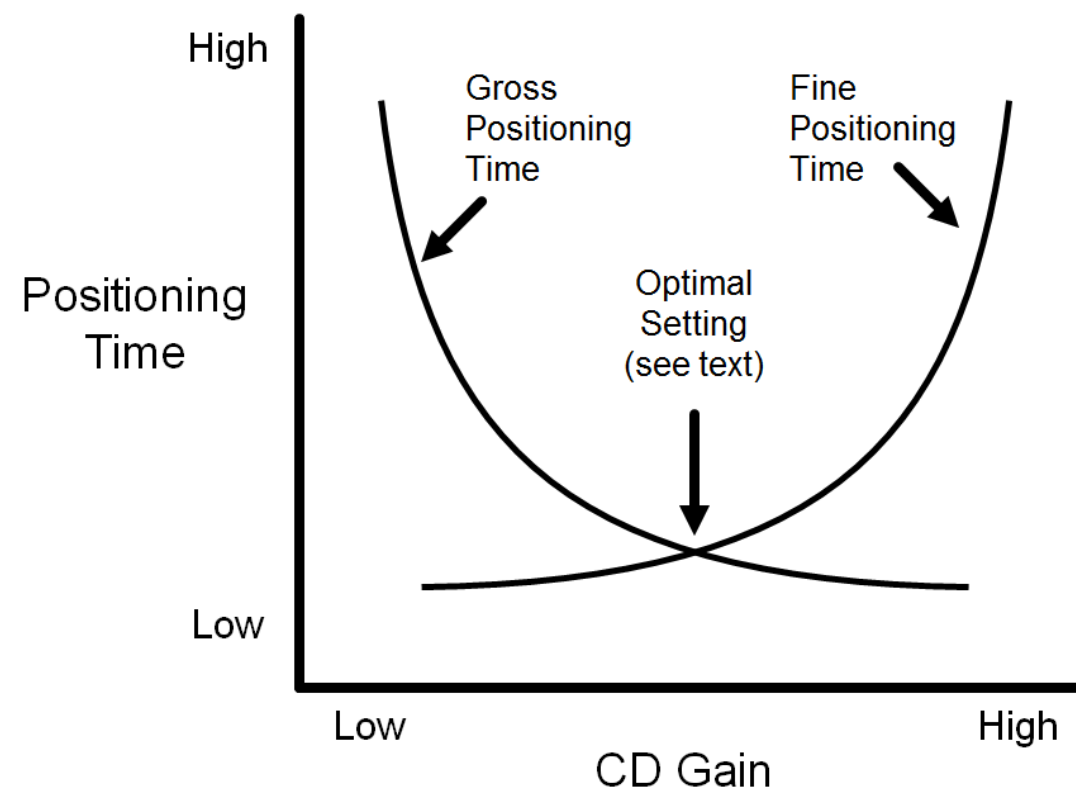
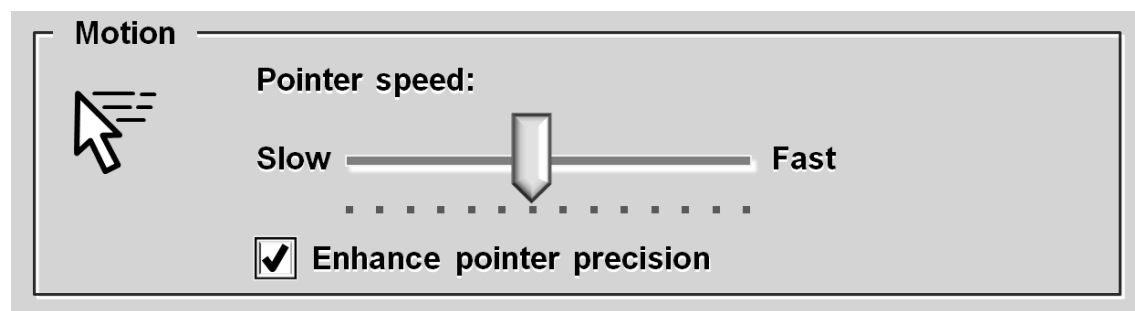
# Spatial Relationships

- Is the light on or off?
  
- Cultural influence
  - **US, Canada, Europe:**
    - UP = ON
  - **UK, India, Australia:**
    - UP = OFF
  - **Germany:**
    - „OHR-Regel“
    - Oben-Hinten-Rechts ist an



# Control-Display (CD) Gain

How should the transfer function be designed so that the mouse can be positioned optimally?



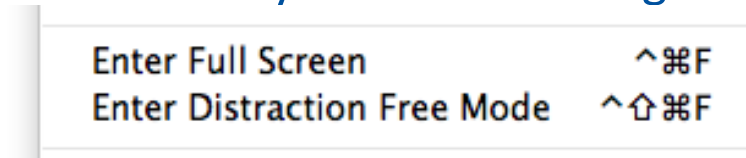
# Widgets (aka Controls)

- Widgets are tools to manipulate the content
  - Each widget solves a specific interaction problem
- Verb-object relationship
  - Widgets (tools) are the verb
  - Content (data) are the objects
  - Example: Pick a color (tool, verb) from an image pixel (data, object)



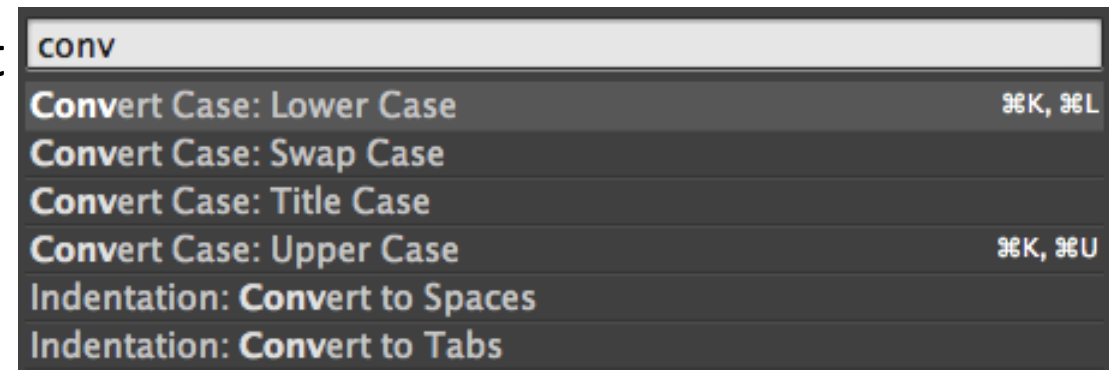
# “No Chrome” Example: Sublime Text Editor

- “Distraction free” mode for full concentration on text editing
  - “You and your text. Nothing else.”



- Each function has a keyboard shortcut
- Function access through search interface
  - Supports discoverability
- Transient views
  - Widgets only shown while invoking a function

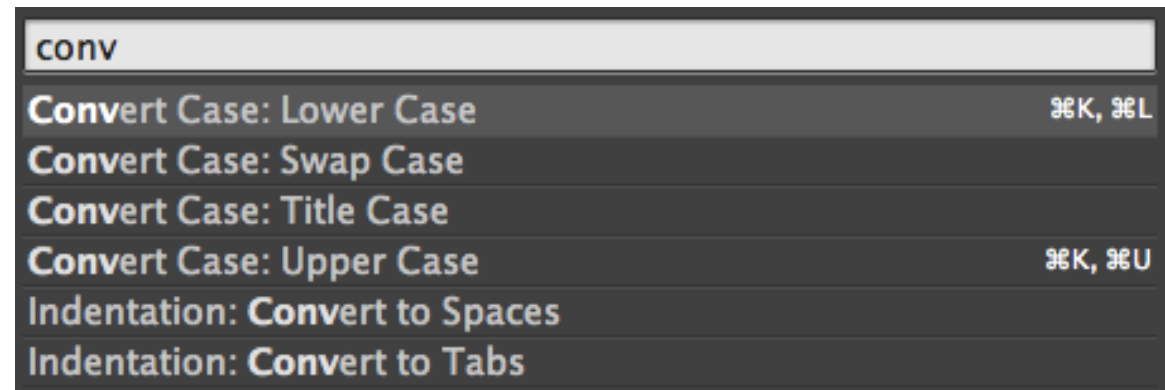
↑+⌘+P





## “No Chrome” Example: Sublime Text Editor

- Guter Ansatz?
- Vorteile? Nachteile?

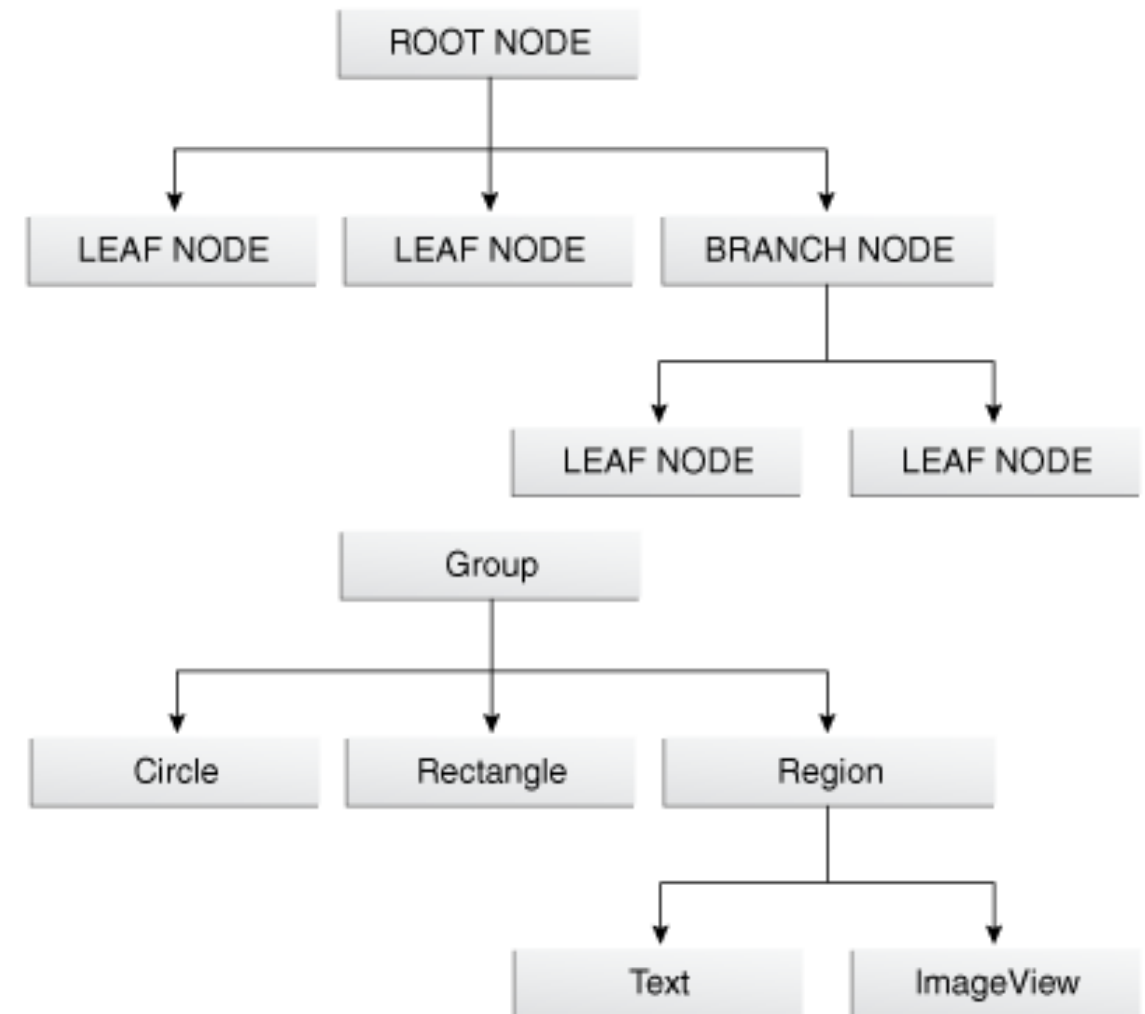


# Scene Graphs

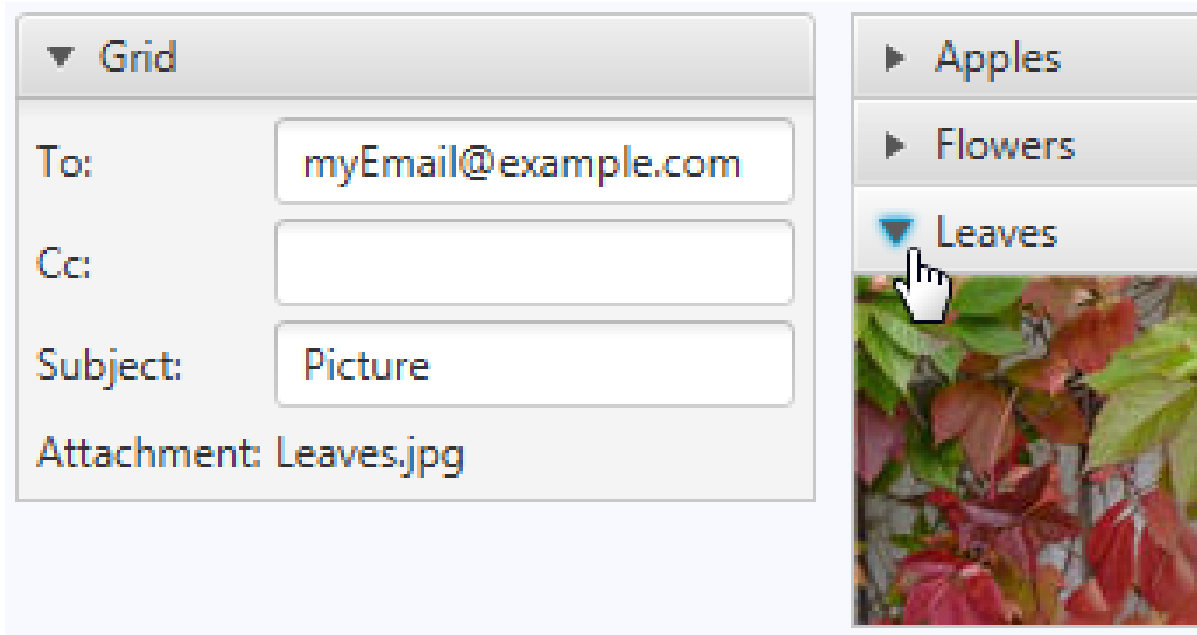
- Scene graph: Set of trees of graphical objects (nodes)
  - Retained mode API: System maintains model of all graphical objects
- Commonly used in video games and 3D graphics
- Allows post-hoc transformations of the structure
  - Size changes
  - Structure changes
  - Animation
  - Effects, etc.
- Creating graphics by modifying the scene graph
- System manages details of graphics rendering
  - Efficiency
  - Less application code

# Scene Graphs

- Represents hierarchy of GUI elements
  - Also used in video games and 3D graphics
- Allows post-hoc transformations of the structure
  - Size changes
  - Structure changes
  - Animation
  - Effects, etc.
- System manages details of graphics rendering
  - Efficiency
  - Less application code



# JavaFX Hierarchy of Scene Graph Nodes in a Layout



<https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/accordion-titledpane.htm>

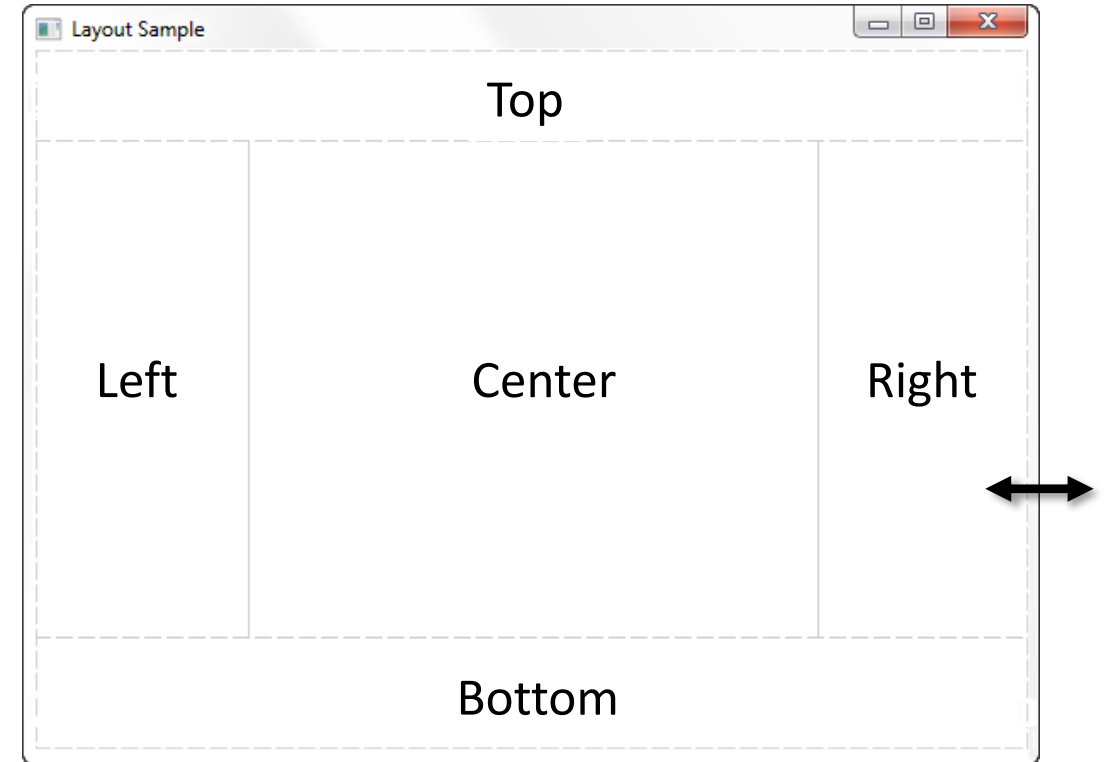
- Which elements would be the topmost nodes in this view?
- Which user actions may affect the visible image?

# Layout Panes

- UI layout: Position and size of UI elements
- Manually laying out widgets is tedious
- Layout panes implement the arrangement of UI elements
- When a window is resized, layout panes reposition and scale the UI elements they contain according to
  - **their layout strategy**
    - row, column, grid, flow, anchors, etc.
  - **the size constraints of the widgets**
    - preferred size, min/max size, alignment
    - not all nodes are resizable (Text, Group, shapes)

# BorderPane

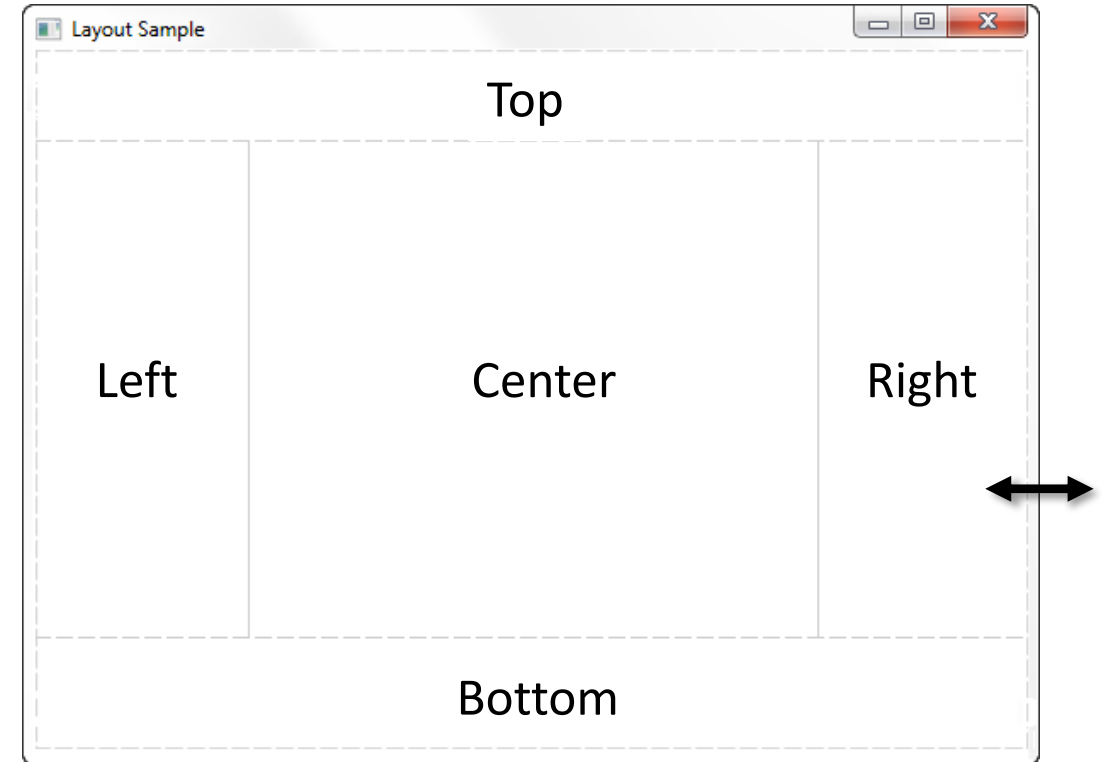
- Why exactly was this layout chosen for the BorderPane?



[http://docs.oracle.com/javase/8/javafx/layout-tutorial/builtin\\_layouts.htm](http://docs.oracle.com/javase/8/javafx/layout-tutorial/builtin_layouts.htm)

# BorderPane

- What happens if the window width is changed?
- A: left, right constant;  
top, center, bottom larger
- B: center constant;  
top, bottom, left, right larger
- C: all larger;  
Ratio left : center : right remains constant



Praktisches Beispiel

# LAYOUTS IN JAVAFX



## Example: FlowPane

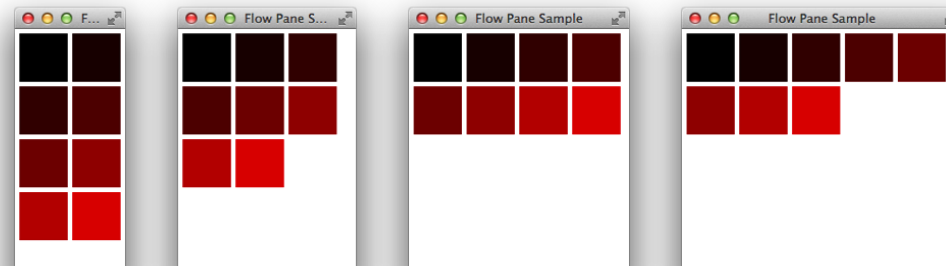
- Nodes flow horizontally (rows) or vertically (columns) and wrap at the pane boundary
  - Similar to line wrap in a text editor

- Example

```

FlowPane flow = new FlowPane();
flow.setPadding(new Insets(5, 5, 5, 5));
flow.setVgap(5); flow.setHgap(5);
Rectangle[] rs = new Rectangle[8];
for (int i = 0; i < rs.length; i++) {
    Color c = new Color((double) i / rs.length, 0, 0, 1);
    rs[i] = new Rectangle(50, 50, c);
    flow.getChildren().add(rs[i]);
}

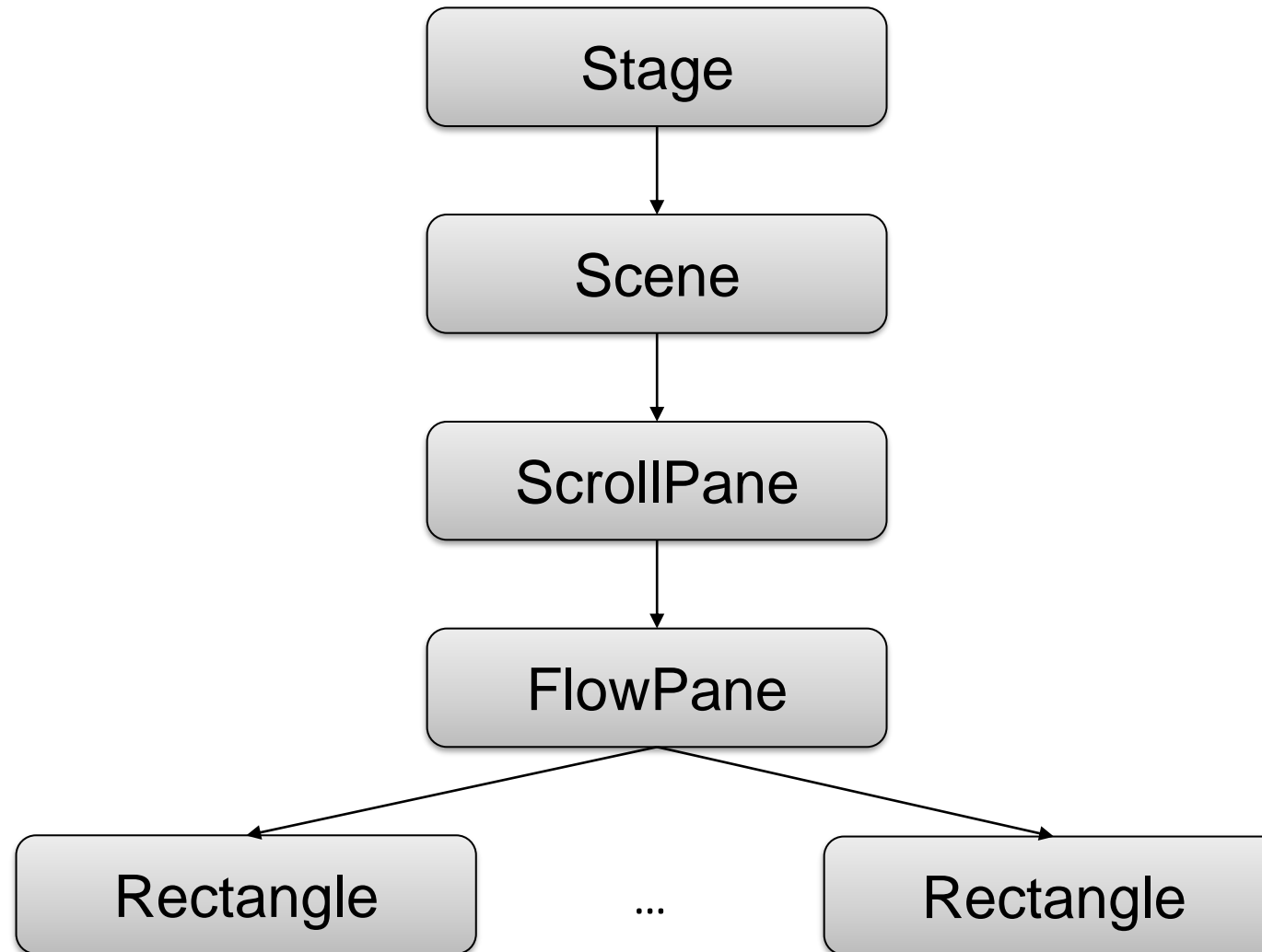
```



see also: TextFlow, TilePane

# Praktisches Beispiel: FlowPane

## Scene Graph: Flow Pane mit Scrollbar



# ASSIGNMENT 2

## Assignment 02

- Abgabe bis Montag (15.04. 23:59)
- PDFs für Aufgaben 1, 2 & 3
- Exportiert das Java Projekt als Zip Datei
  - Wenn die .zip Datei, die ihr an das Assignment System schickt, eine weitere .zip Datei beinhaltet, dann habt ihr alles richtig gemacht

