

Interaktionsprogrammering

TDDC73

Anders Fröberg

anders.froberg@liu.se

Outline

- Historic perspective
 - Where are we coming from
 - Why do things look like they do
- Course outline

West of House 0/0

In the beginning there was a command line

ZORK I: The Great Underground Empire
Infocom interactive fiction - a fantasy

Entire 14 inch screen devoted to a single application

Copyright (c) 1981, 1982, 1983, 1984,
1985, 1986 Infocom, Inc.

All rights reserved.
ZORK is a registered trademark of
Infocom, Inc.

Release 52 / Serial number 871125 /

Users operated advanced system through natural language like syntax

West of House
And it was all good (especially for developers)

You are standing in an open field west
of a white house, with a boarded front
door.

And then came the gui and it was all down hill from there (for the developers)

There is a small mailbox here.

>fuck mailbox

I do not know the word "fuck."

>_

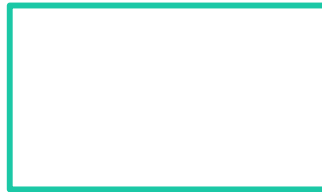
..and then came the GUI



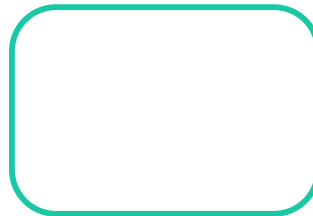
..and then came the GUI



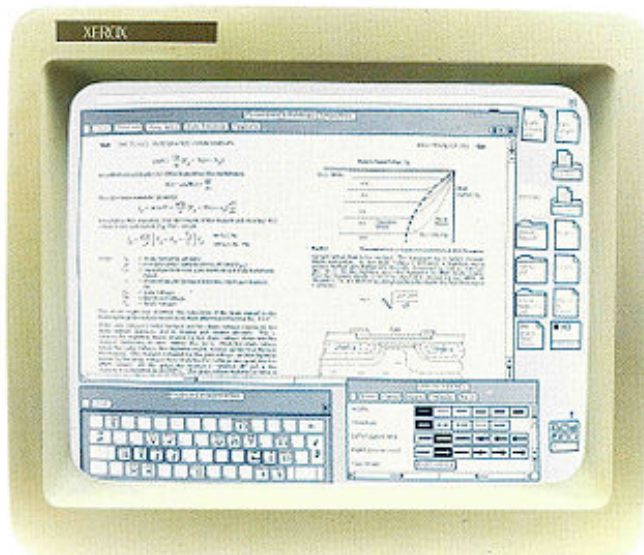
..and then came the GUI



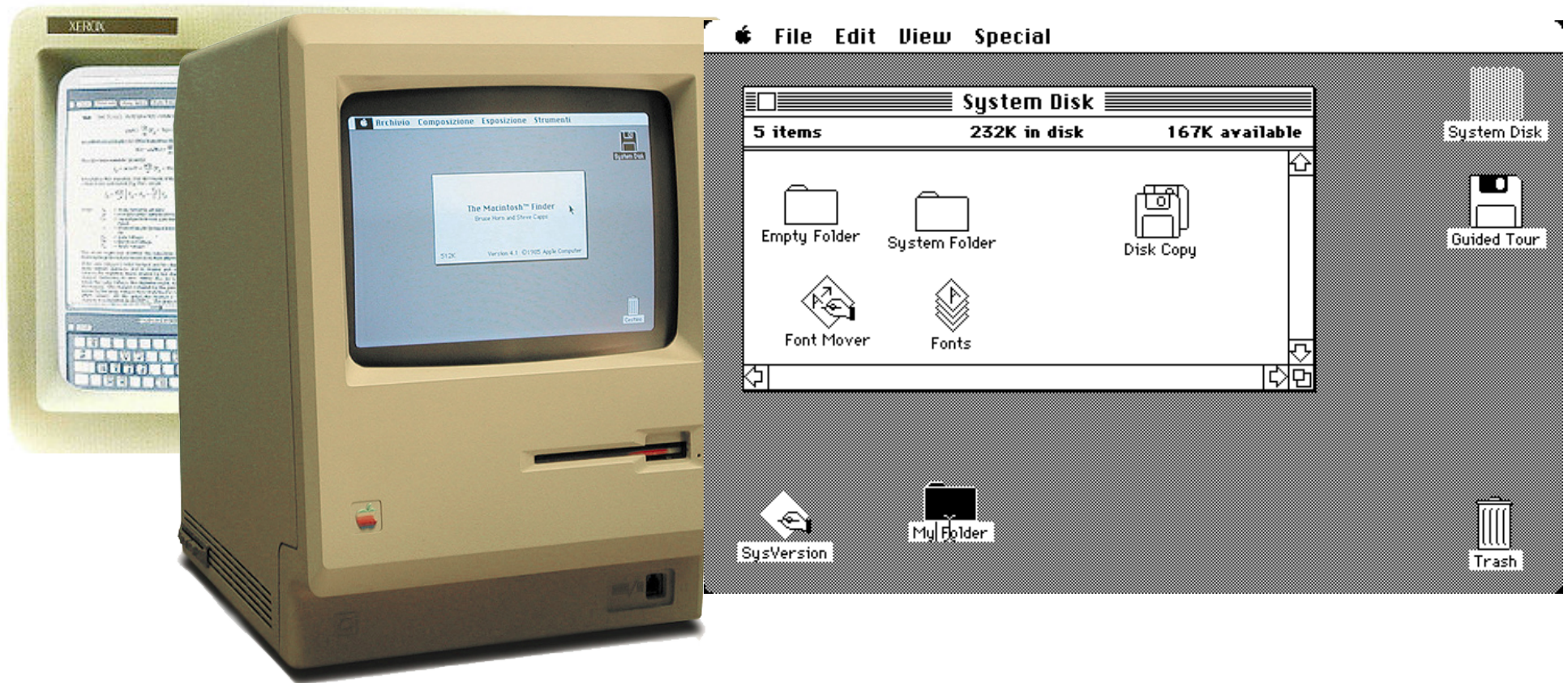
..and then came the GUI



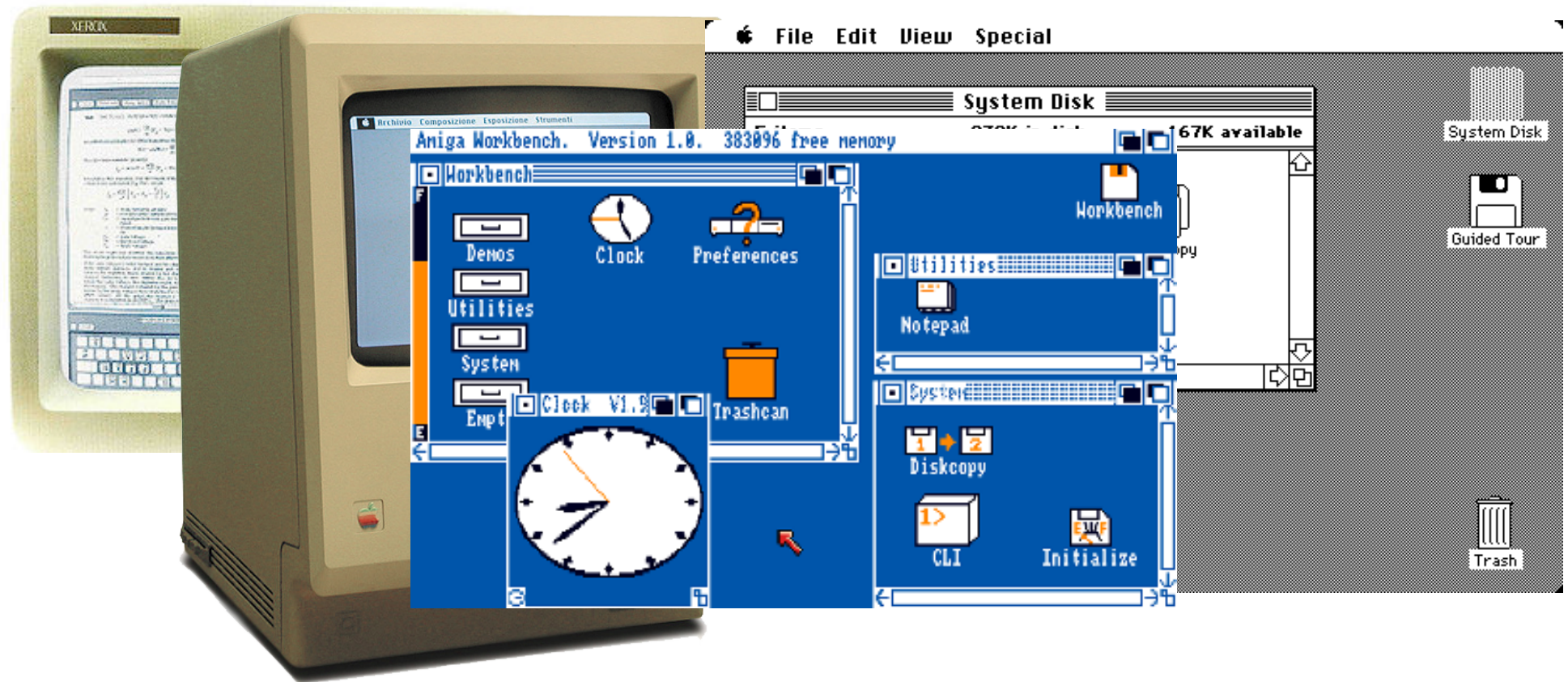
..and then came the GUI



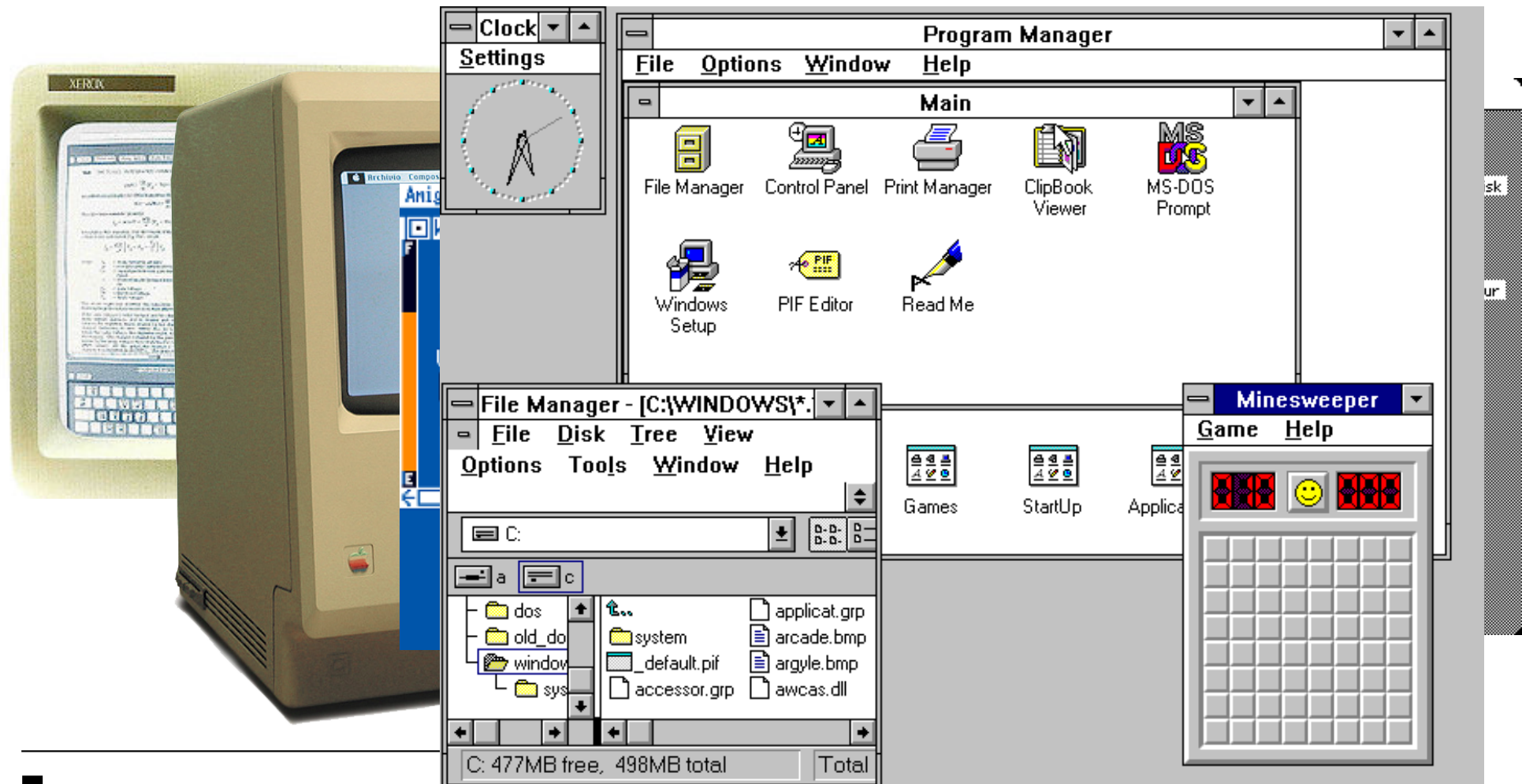
..and then came the GUI



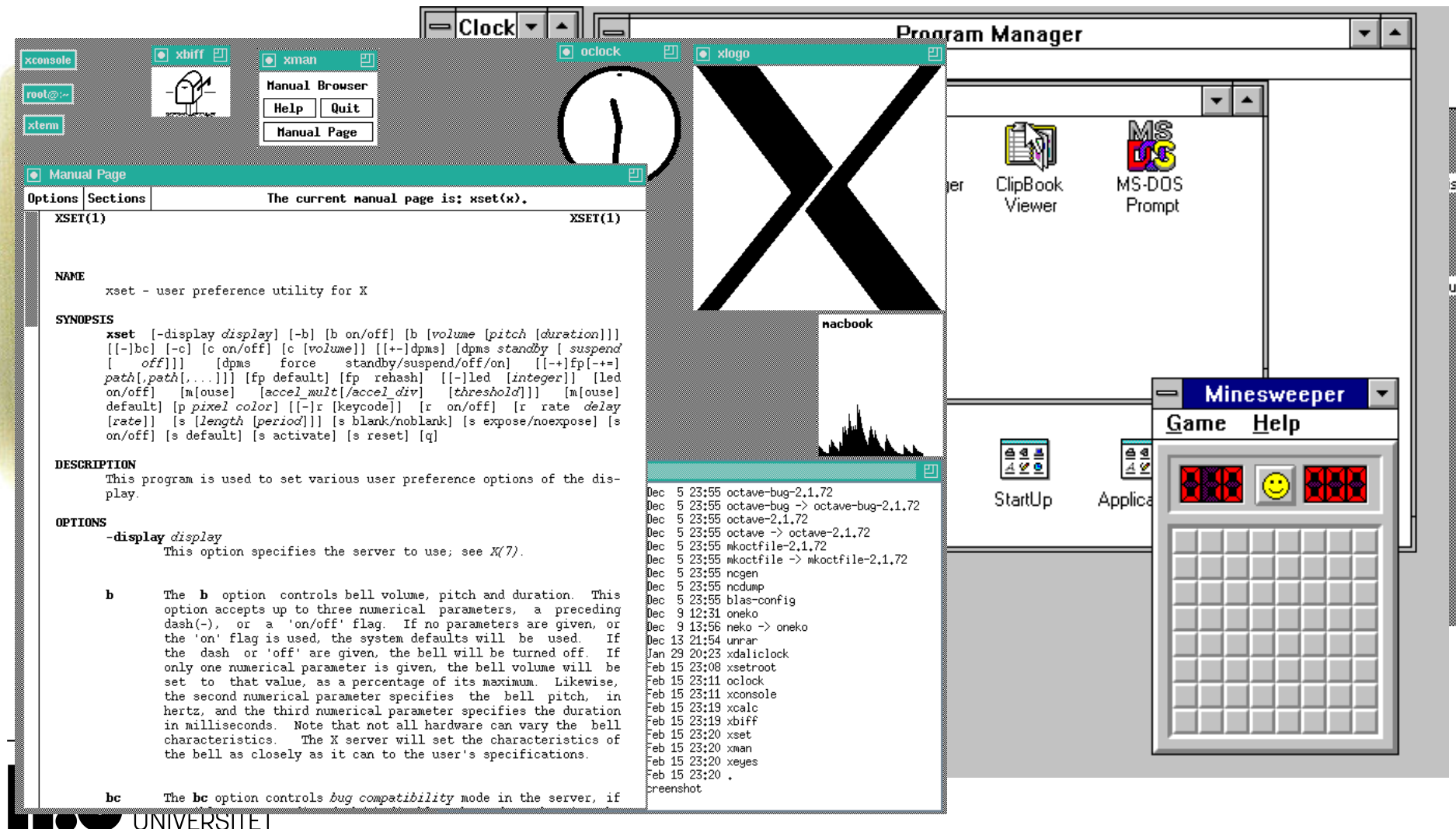
..and then came the GUI



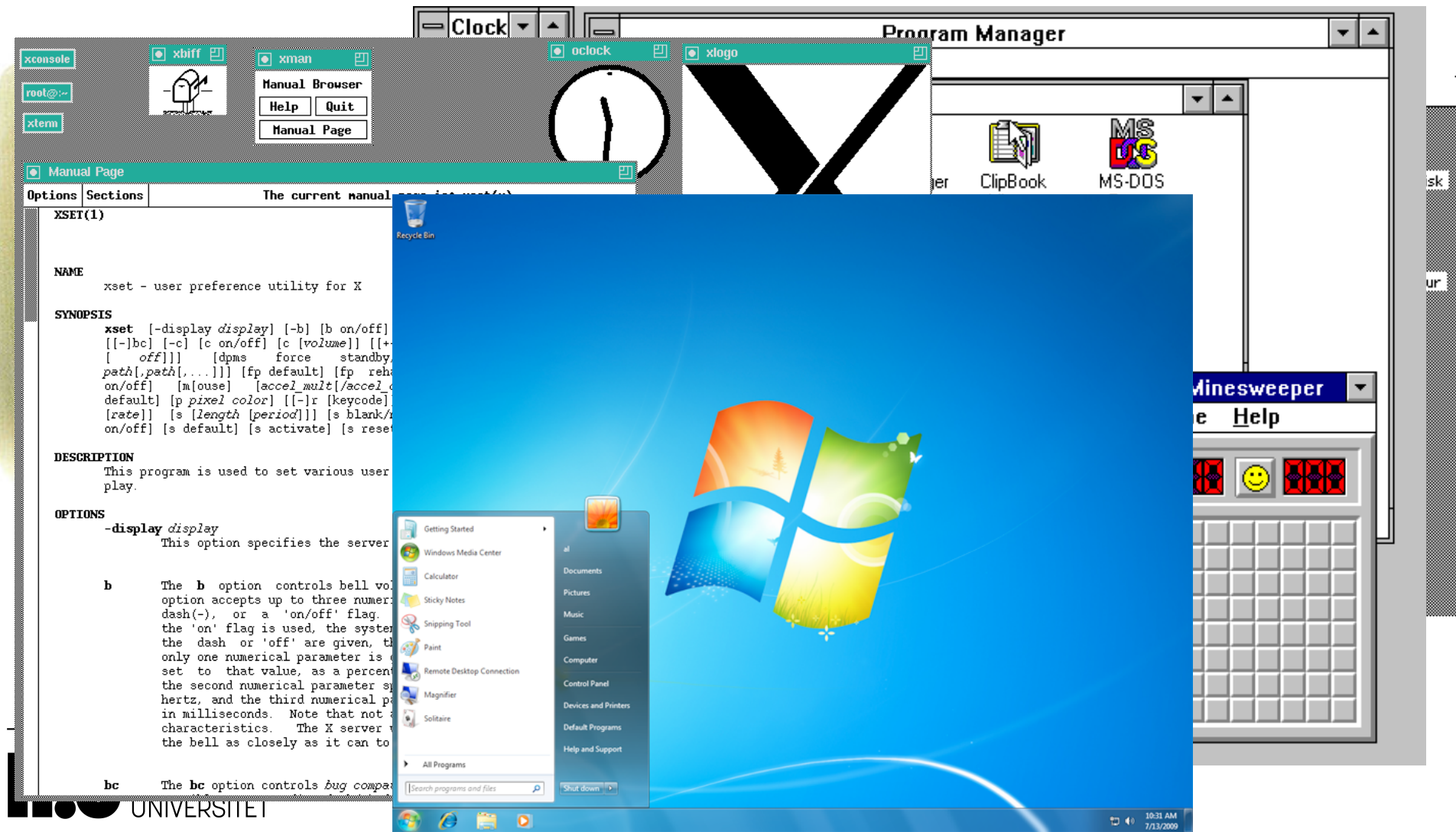
..and then came the GUI



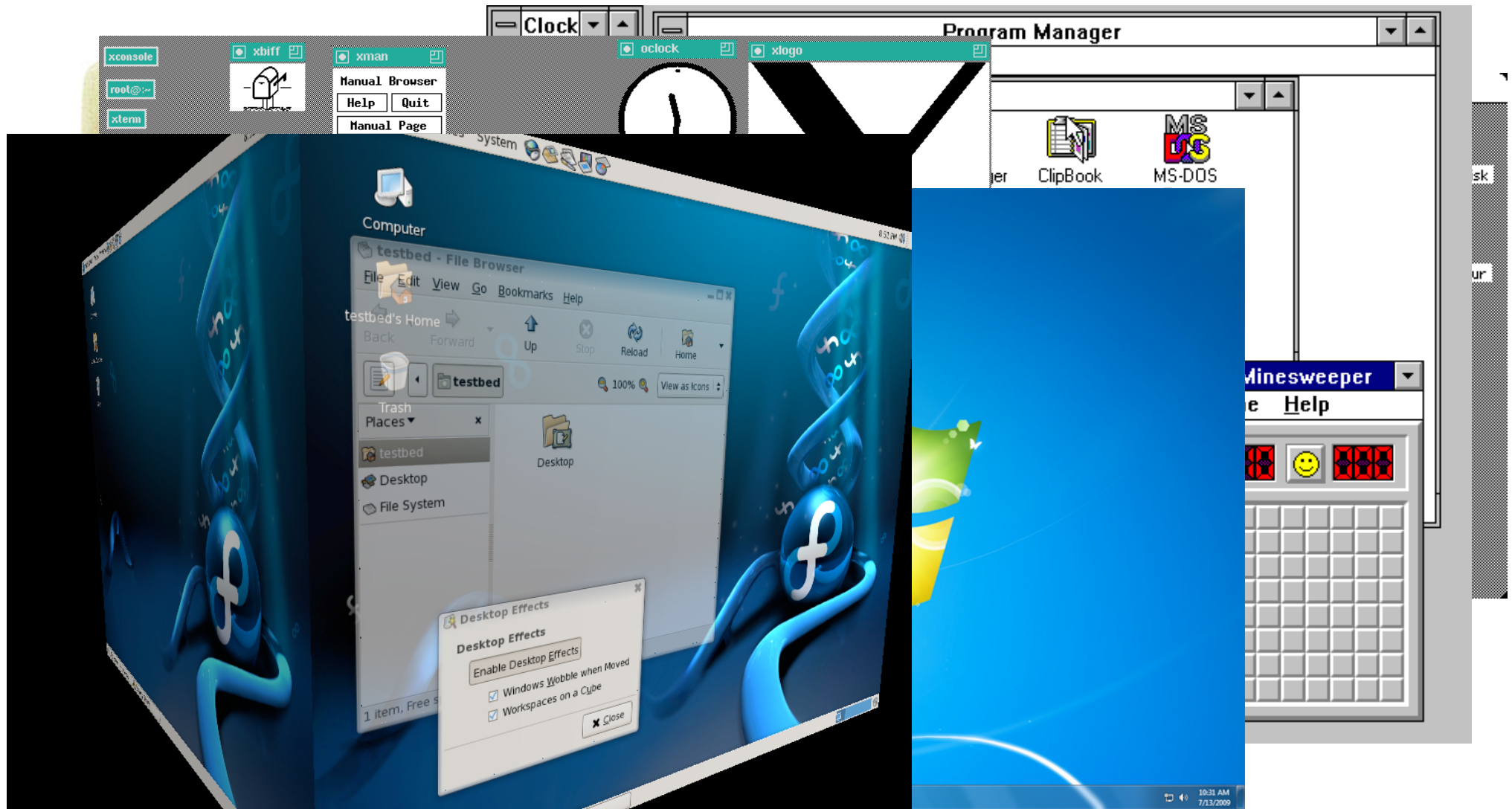
..and then came the GUI



..and then came the GUI



..and then came the GUI







Command line ->GUI

- New problem arises for developers
- Complex interaction structure
 - Elaborate graphics
 - Multiple interaction mechanisms
 - Non-linear command structure

Solutions

- Iterative methods
- Software Development Kits (SDKs)
 - Swing
 - Motif
 - .Net Forms
 - React React-Native
 - Flutter
 - Vue
- GUI builders
- Specialized GUI languages

Some stats

- Already in '94 almost all Unix contained a GUI
- Half to code is GUI code
- Time spend equal all other parts together
- Benefit Tools
 - Reduces code by 83%
 - Time spend reduced by a factor 4 (Building GUI)

So what is this UI thing

Wikipedia

The user interface (also known as human computer interface or man-machine interface (MMI)) is the aggregate of means by which people—the [users](#)—[interact](#) with the [system](#)—a particular [machine](#), device, [computer program](#) or other complex [tool](#). The user interface provides means of:

- [Input](#), allowing the users to manipulate a system
- [Output](#), allowing the system to indicate the effects of the users' manipulation.

Wiktionary

Noun

[user interface](#) (plural [user interfaces](#))

1. ([countable](#)) The part of a [software application](#) that a [user](#) sees and interacts with.

GUI for programmers

- Two main components
 - User(s)
 - System/Program/Application/Hardware
- Two pipes of information
 - Input from the User(s)
 - Output to the User(s)

GUI for programmers

- Two main components
 - User(s)
 - System/Program/Application/Hardware
- Two pipes of information
 - Input from the User(s)
 - Output to the User(s)

Focus

Three teal arrows originate from the word 'Focus' and point to the three instances of 'User(s)' in the list: the first in the main components list, and the two in the information pipes list.

What constitutes a (G)UI

- The worlds smallest GUI 

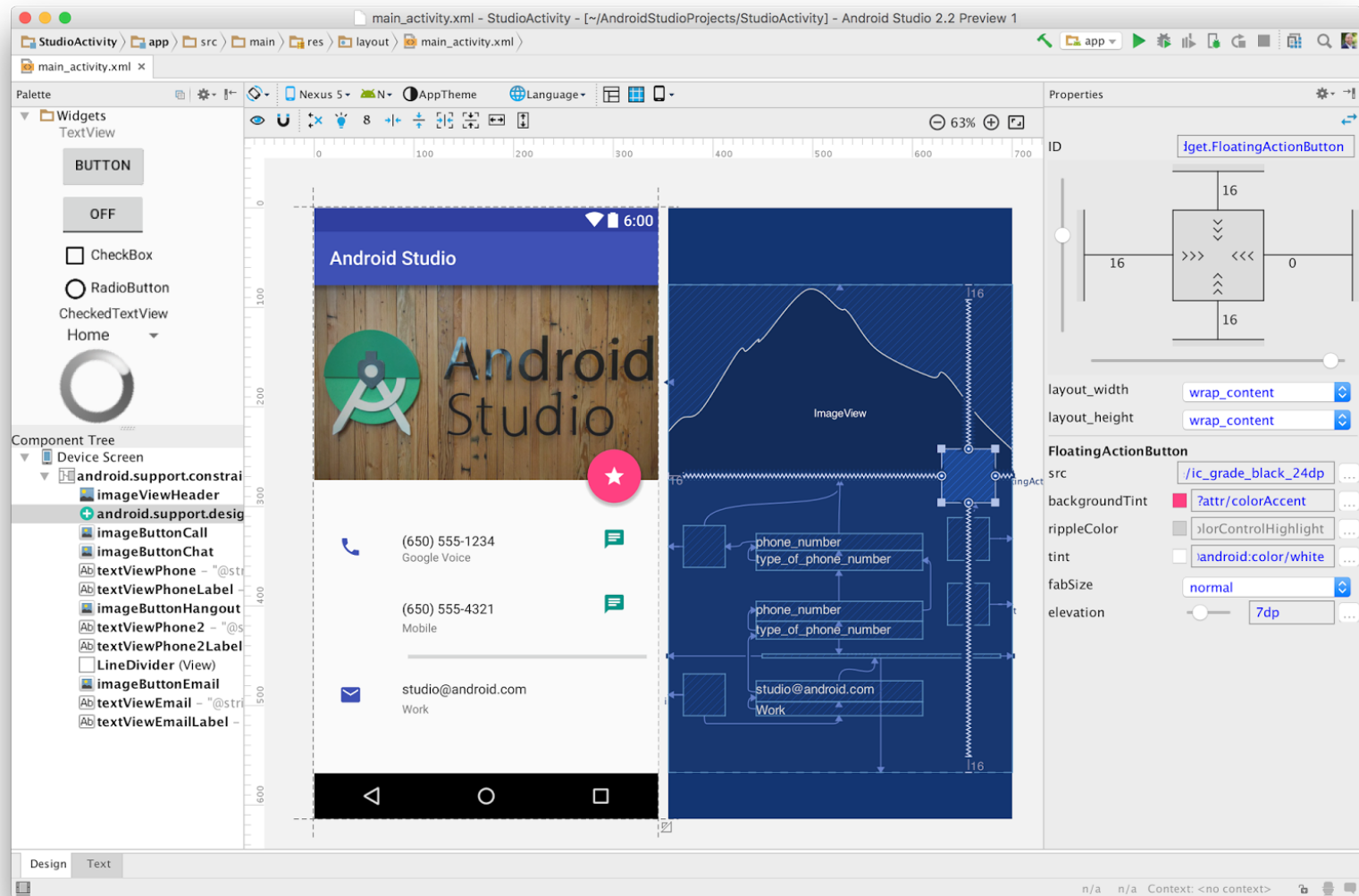
A GUI for a programmer

```
public void  
  paint(Graphics g){  
    g.moveTo(100,100);  
    g.setColor("ffffff");  
    g.drawLineTo(200,300);  
  }
```

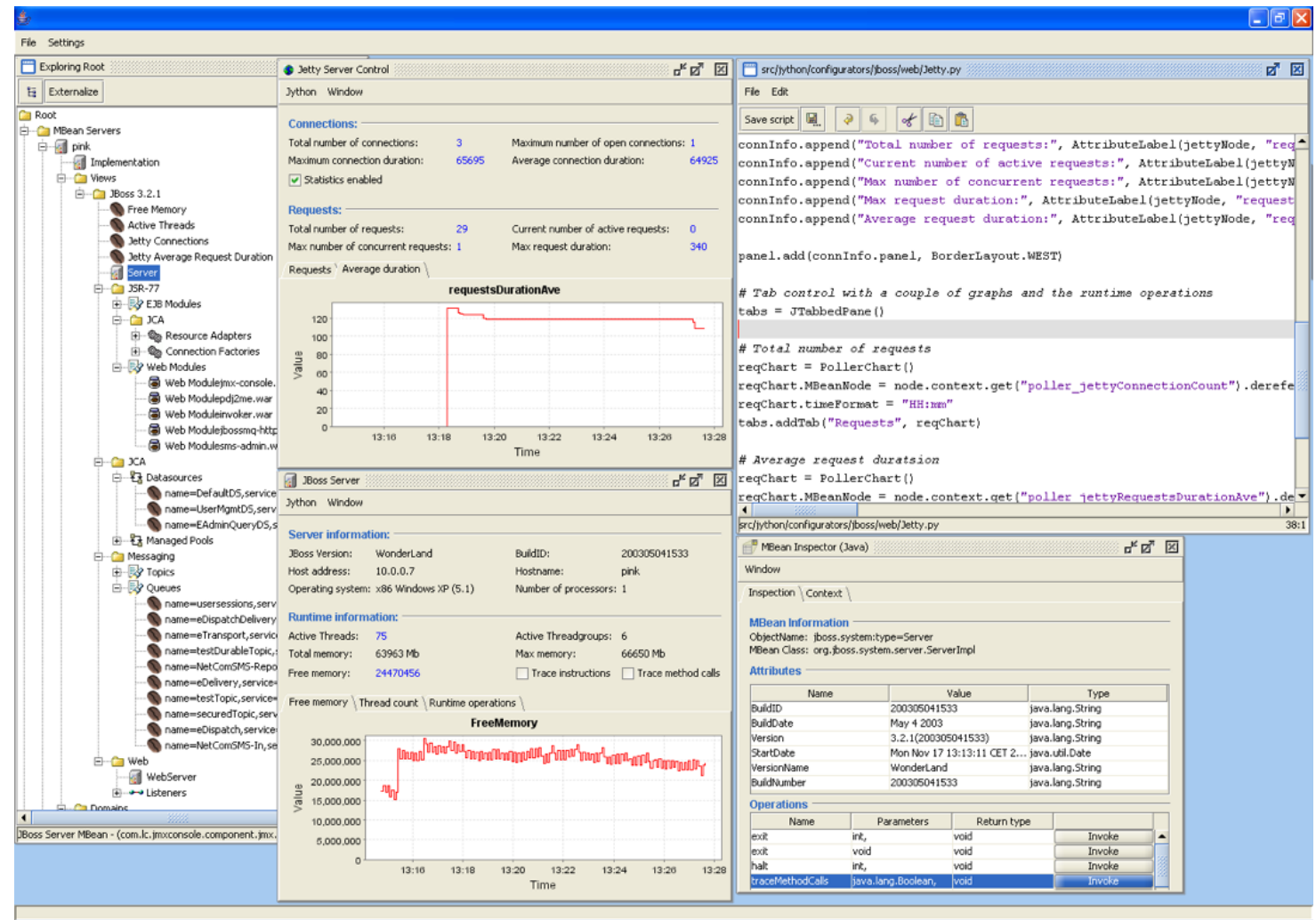
Spans from low level to very
high level

```
val intent = Intent()  
intent.type = "image/*"  
intent.action = Intent.ACTION_GET_CONTENT  
startActivityForResult(  
    Intent.createChooser(intent, "Select Picture"), PICK_IMAGE);
```


...and even visual



GUI for the User



And GUI for You in this course

```
protected void onDraw(Canvas canvas) {  
    super.onDraw(canvas);  
    // Draw the shadow  
    canvas.drawOval( mShadowBounds, mShadowPaint );  
  
    canvas.drawText(mData.get(mCurrentItem).mLabel, mTextX, mTextY, mTextPaint);  
}
```

Android/Java

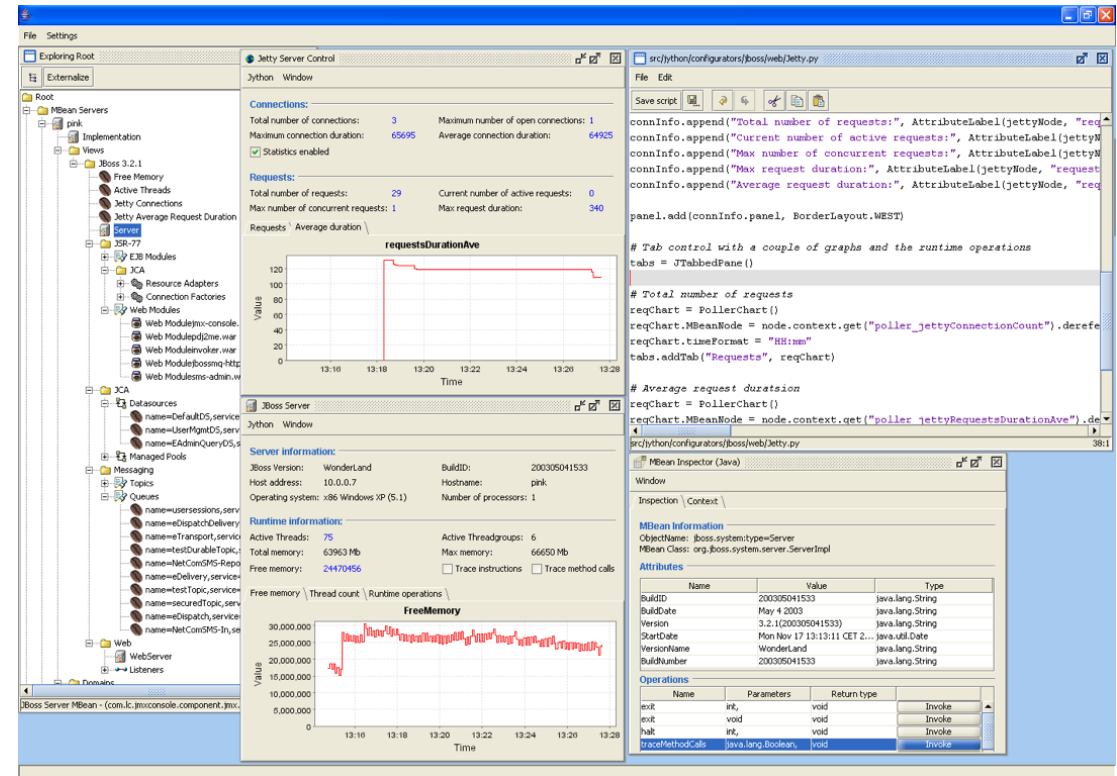
Spans from low
level to very high
level

```
val intent = Intent()  
intent.type = "image/*"  
intent.action = Intent.ACTION_GET_CONTENT  
startActivityForResult(  
    Intent.createChooser(intent, "Select Picture"), PICK_IMAGE);
```

Android/Kotlin

How do we get from

What user wants and needs (user/system requirements)



How do we get from

29

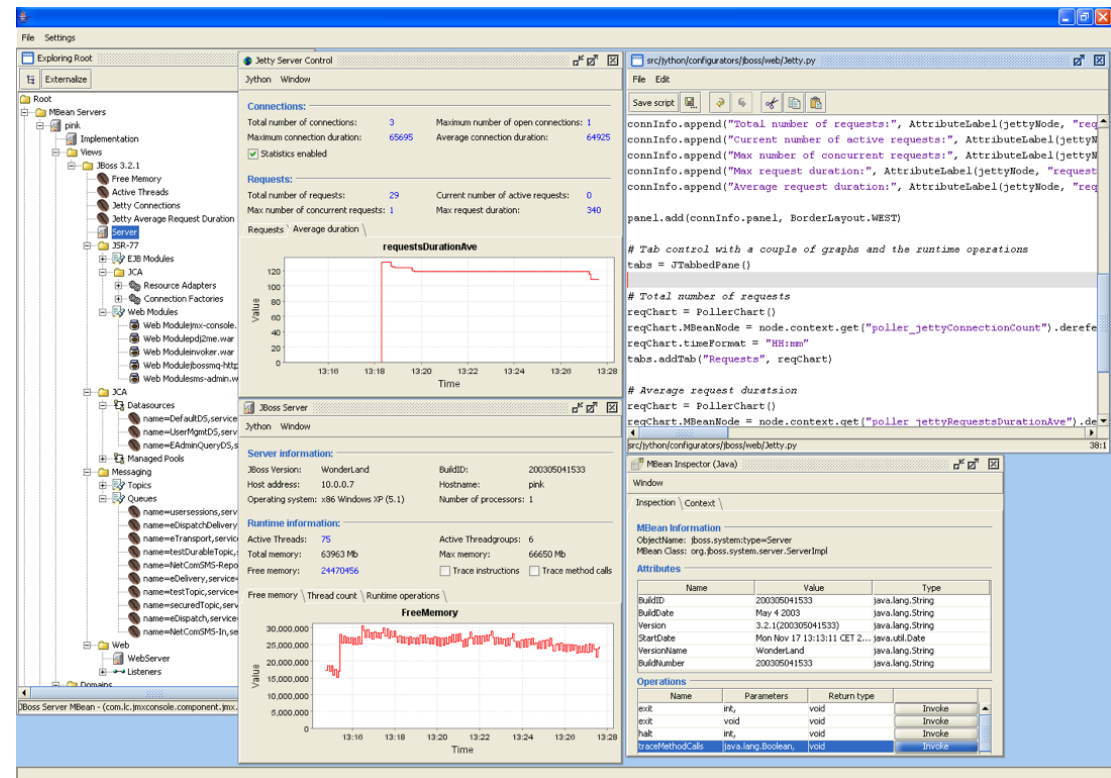
```
protected void onDraw(Canvas canvas) {
```

```
    super.onDraw(canvas);
```

```
    // Draw the shadow
```

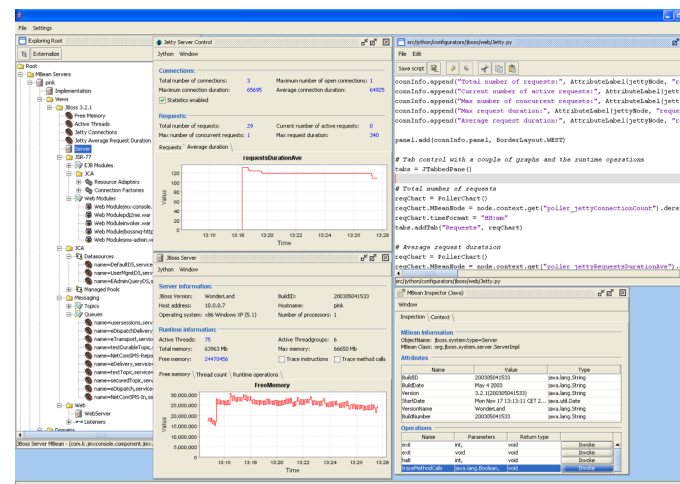
```
    canvas.drawOval( mShadowBounds, mShadowPaint );
```

```
    canvas.drawText(mData.get(mCurrentItem).mLabel, mTextX, mTextY, mTextPaint);
```



Solution

SDKs
for (gui-)code:
How to build it



Interaction Patterns
for user requirements:
What to build

Now how do you get there

- Welcome to this course
- More interaction programming
- This is a programming course (G2 level)
 - Focus on writing code
 - Focus on writing for others
- You will learn from working not listening

**People generally
remember...
(learning activities)**

10% of what they read

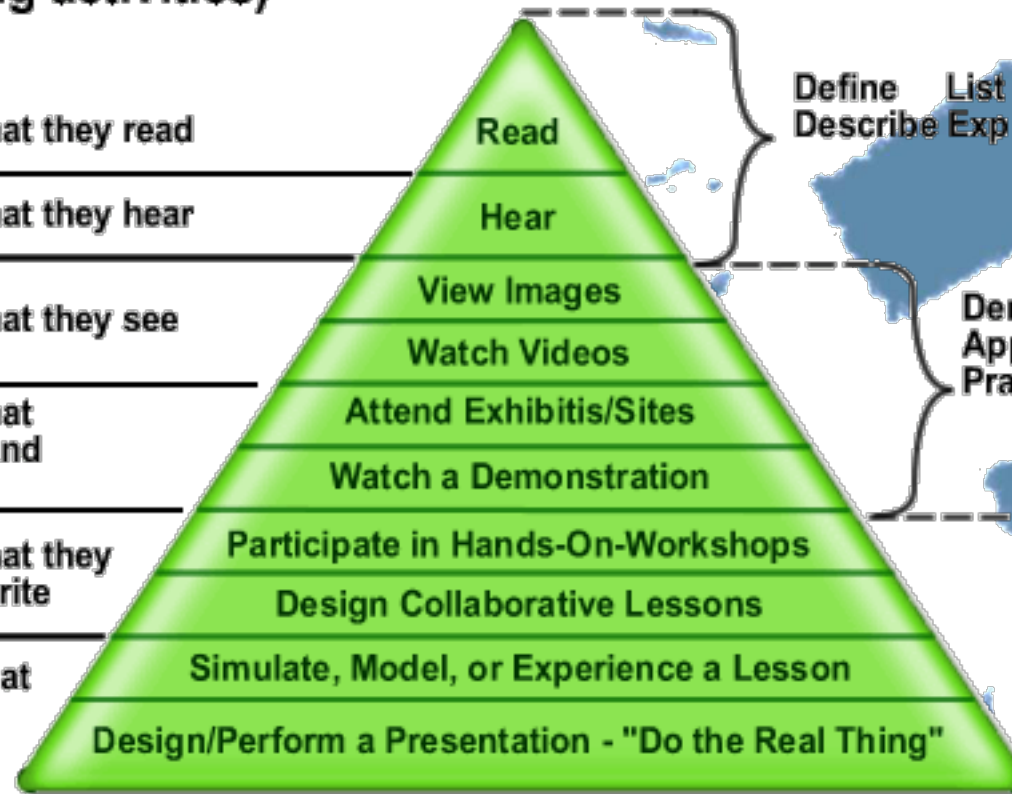
20% of what they hear

30% of what they see

50% of what
they see and
hear

70% of what they
say and write

90% of what
they do.



**People are able to...
(learning outcomes)**

Define
Describe

List
Explain

Demonstrate
Apply
Practice

Analyze
Define
Create
Evaluate

LINKÖPINGS UN

Course outline - Your work

- 3 Labs
 - Component placement (Three frameworks)
 - User interaction and component coupling
 - Communication with the rest of the world
- Mini project
 - Your own mini SDK
 - Implement 2 (at least) high-level interaction patterns in a SDK.
 - You may need to develop supporting components

Course outline - our work

- Introduction to Course (right now)
- Two ui lectures
 - Closely related to the labs
- One lecture on adjacent topics
 - Testing
 - Framework/Library design

Labs

- Cover basic GUI-programming
 - Component placement
 - Component Interaction
 - Component-System/network interaction

Mini project

- A SDK with of interaction patterns (two of them)
- A example app using your SDK

Grading for Course

- 3 Labs and mini project
- 4
 - grade 3 + Testing or Getting started tutorial
- 5
 - grade 3 + Testing and Getting started tutorial

Examination

- To pass the course you need to
 - Pass the labs (assistants)
 - Pass the mini project (assistants)
 - Pass the oral examination (Anders)

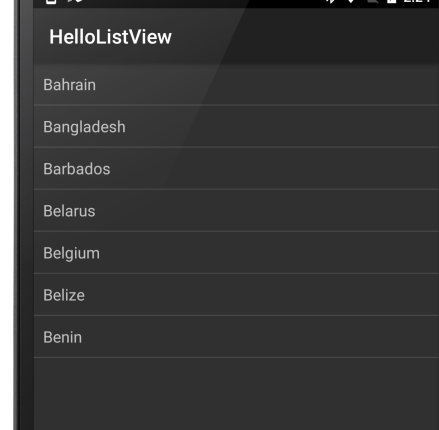
Feedback from last year

- Fun
 - But time consuming
- Lab 3 was a bit to open
 - What is “trending” on GitHub
- Project was/is easier then the labs

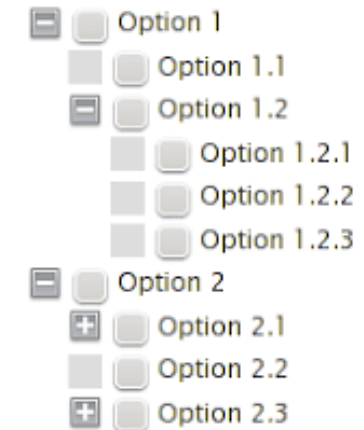
GUI SDKs

- GUI SDK contains
 - Runtime -a mapping between Abstract UI and native OS/SDK components/primitives
 - Widgets - Buttons, Menus , Text Field etc
 - Widget Messages - interoperability among widgets
 - Messaging structures for in-/output - Messages
 - Rule sets/widget constraints - for controlling layout

UI -Widgets

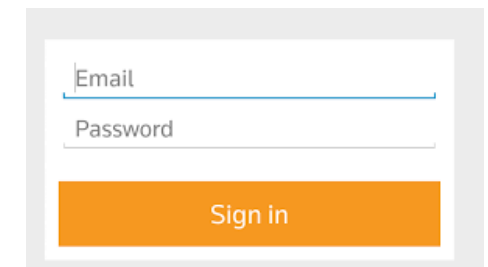


Tree Widget Test



Select up to 3 options.

- Components
 - A set of visual reusable objects
 - Buttons, Tree, Table, Text-field
 - Android : View, Button , List, EditText, TextView
- Containers
 - Placeholders for Components and Containers
 - Usually allows for nested Containers
 - Windows, Panels, Menus, Toolbars
 - Android: ViewGroup, Layouts



Widget Messages

- Messages from widgets to your code
 - Information about a change in state
 - Messages of user generated actions
 - Mouse movement, keyboard action
 - Touch , Sensors
- Widgets has a set of Messages (zero or more) you choose which you care about

Widget Messages

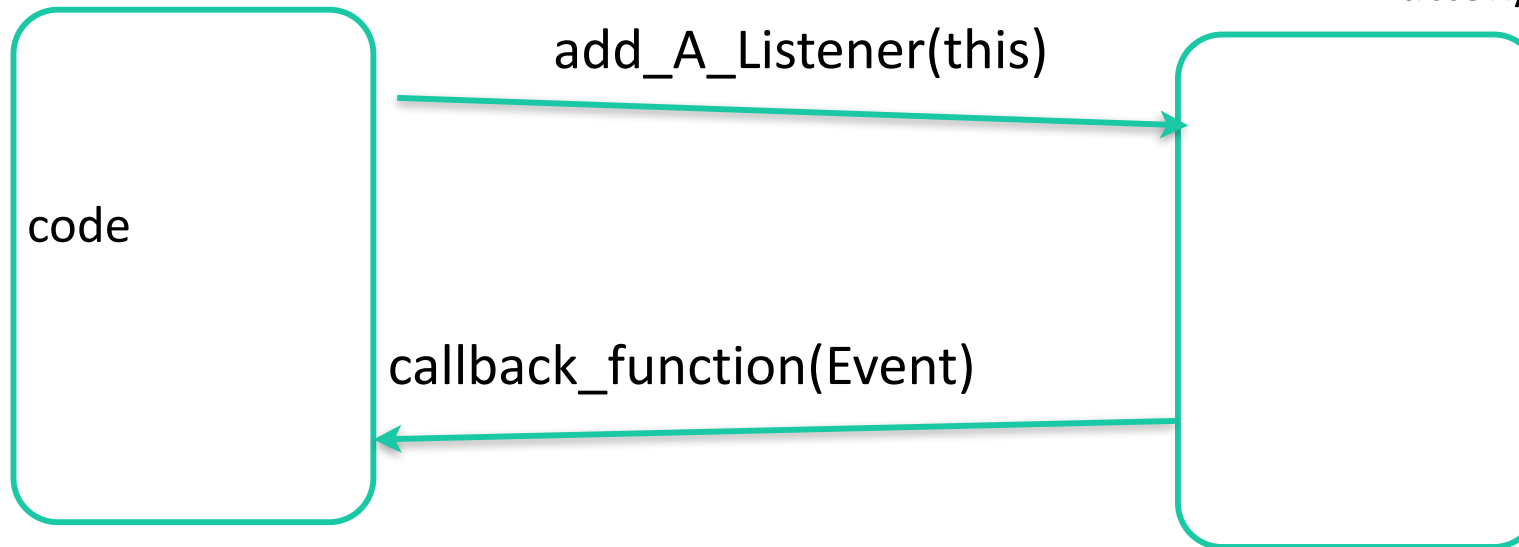
- Messages passed through callback functions
- You registers yourself with component using listeners
 - android : `setOnClickListener`
- Receive notification
 - `onClick(View v)`

Listeners



Your code

Button/Widget



Messages for in- and output

- GUI runs in a separate thread
 - Keep this thread clean only use for GUI related issues
- SDK provides mechanism for delivering info that can be received and processed in that thread

Threads

- Problem: UI not Thread safe
 - GUI code only in the GUI Event thread
- Solution: Place code on cue
 - Causes UI thread to run a given runnable when it can on the UI Event thread
- Use post any time you want to modify a UI object outside of a listener method

Android : {View}.post

```
mImageView.post(new Runnable() {  
    public void run() {  
        mImageView.setImageBitmap(bitmap);  
    }  
});
```

// OR AsyncTask

```
SwingUtilities.invokeLater(  
    new Runnable(){  
        public void run(){  
            outputArea.append(messageToDisplay);  
        }  
    }  
);
```

Rule sets and widget constraints

- Controlling widgets
 - placement, size
 - relation to other widgets
 - reaction to external changes
 - Window size change
 - Device orientation,size,resolution

Placing components

- Layouts in Android
 - Constraint, Motion Linear- Vertical/Horizontal (Grid , Table)
- Layouts in React Native
 - Flexbox
- Layouts in Flutter
 - Layout widgets
 - Single-Child widget(s)
 - Multi-Child widget(s)

Conclusion

- What I hope you take with you
 - This is a programming course you learn by doing
 - Tools aids developers - so learn them
 - Some brief hints on how to develop todays GUI application.

- Questions