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

Before we start

- Both at LiU and via Teams
  - Designed and planed for at LiU
  - But will stream and record on Teams.
- Feedback from last year
  - Getting started guide to easy
  - Project easier than the lab

In the beginning there was a command line

Entire 14 inch screen devoted to a single application

Users dazzled with green on black

Users operated advanced system through natural language like syntax

And it was all good (especially for developers)

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

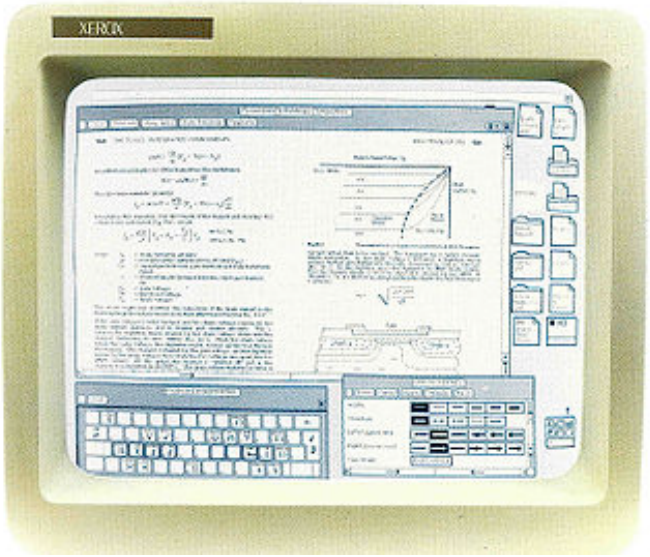
..and then came the GUI

..and then came the GUI

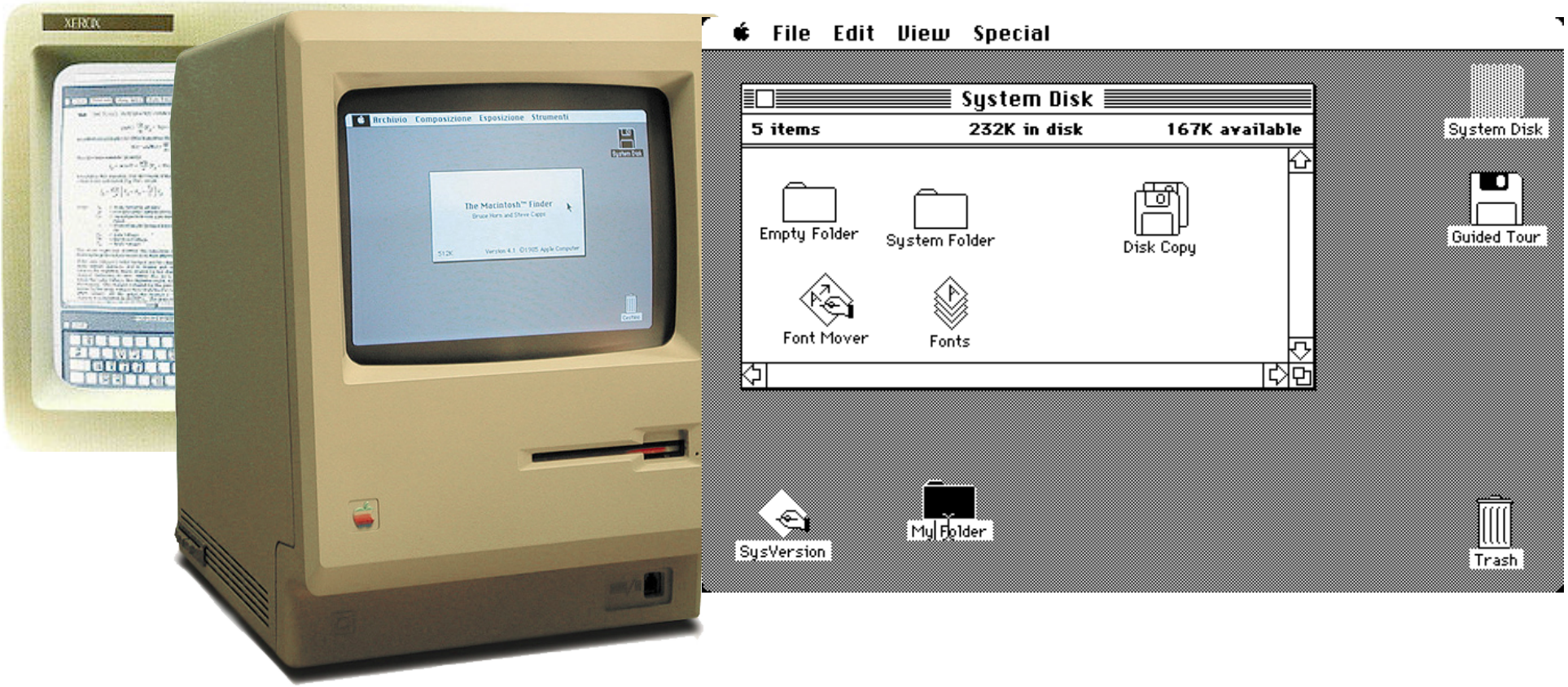
..and then came the GUI

..and then came the GUI

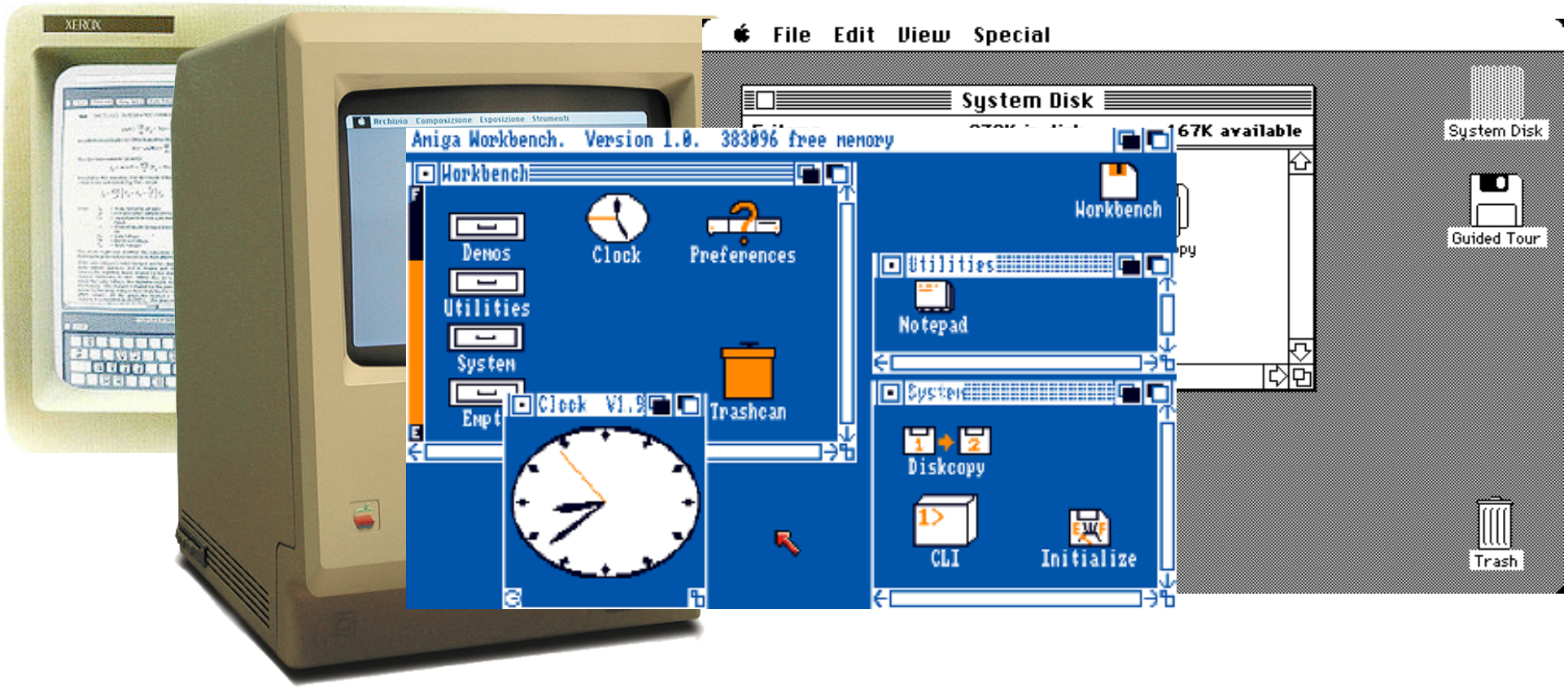
..and then came the GUI



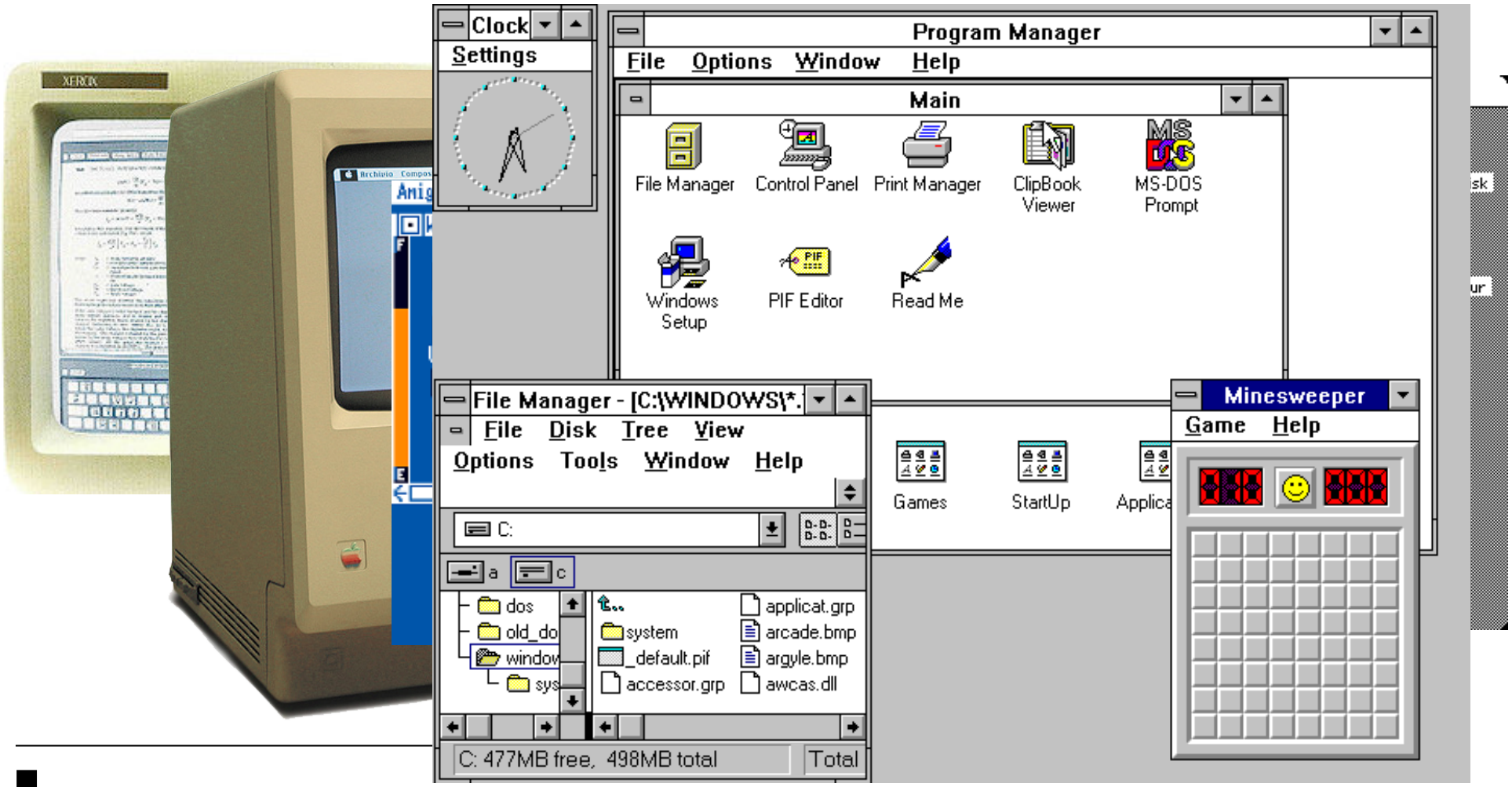
..and then came the GUI



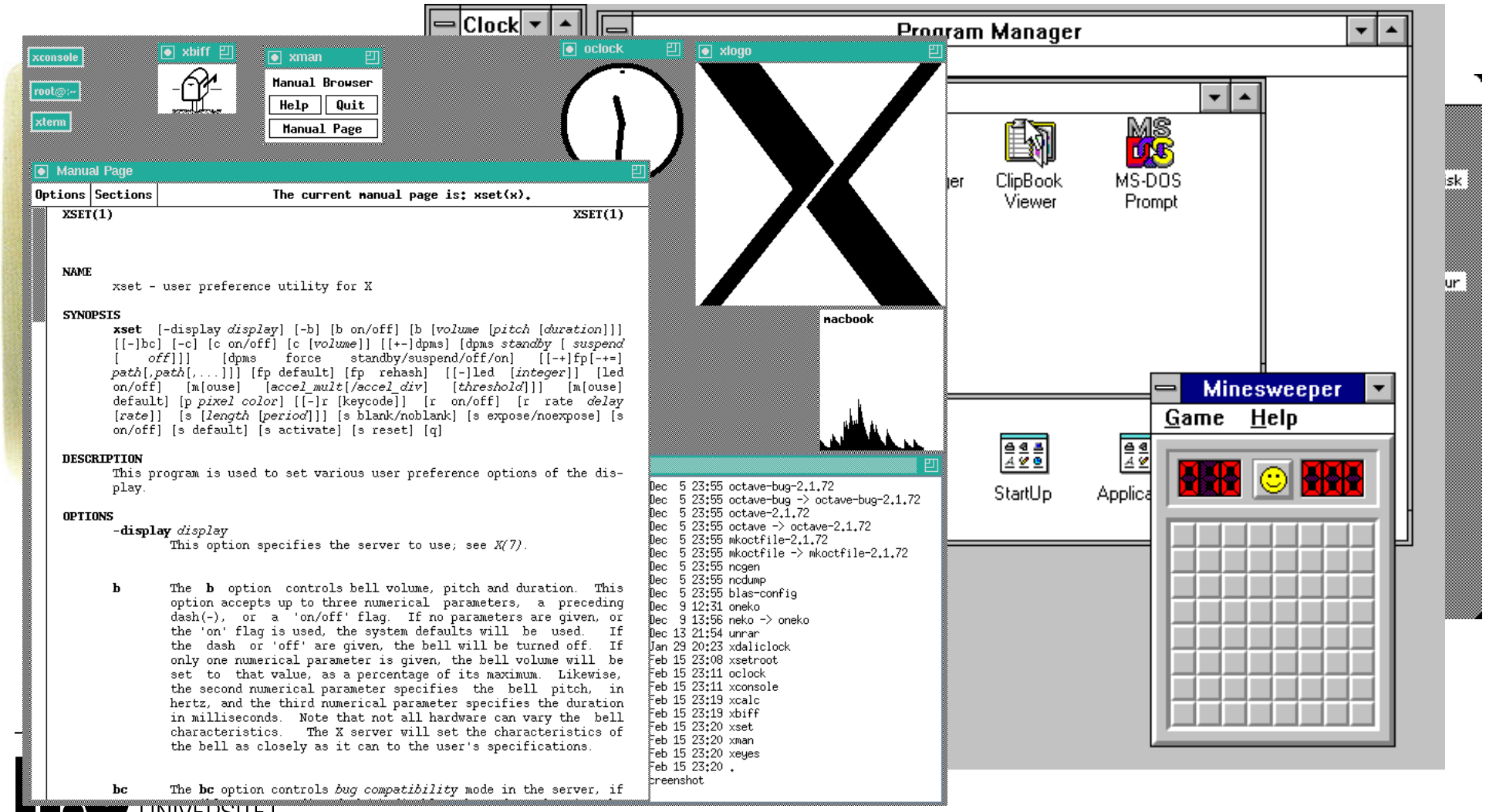
..and then came the GUI



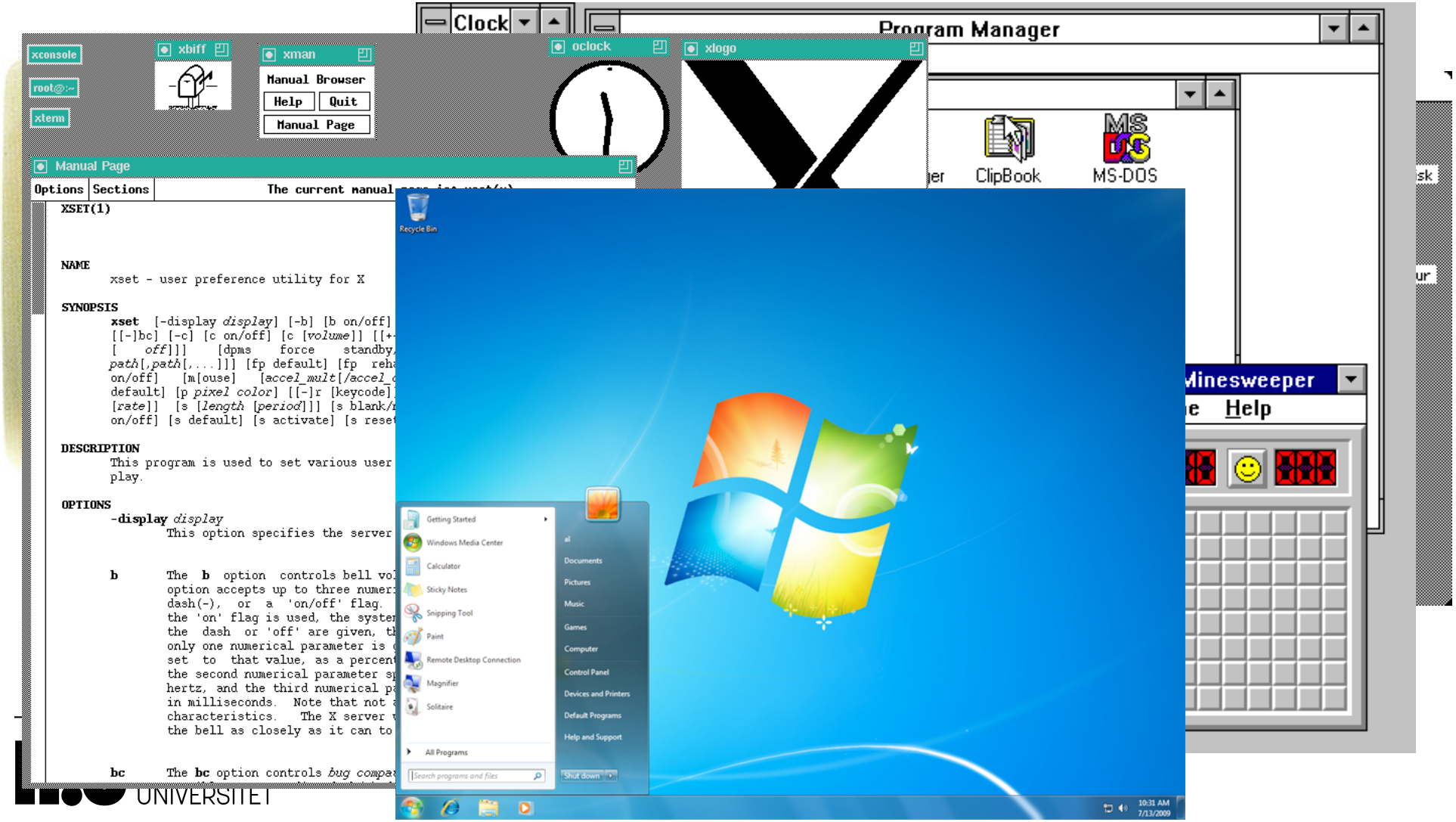
..and then came the GUI



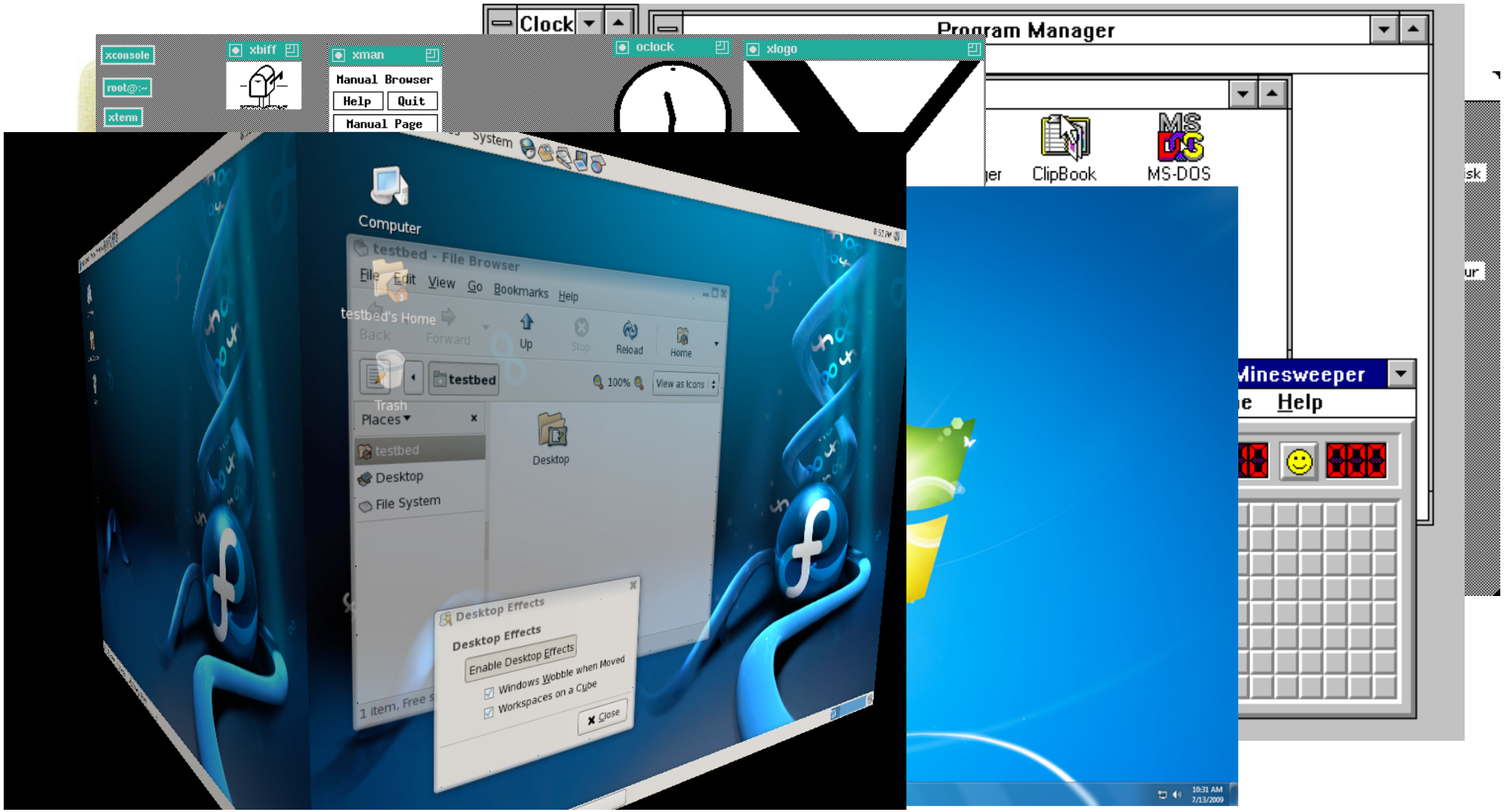
..and then came the GUI



..and then came the GUI



..and then came the GUI




1:14 PM
07/12 | WINNIPEG, 23°C
SAT
Jul 12
TOMORROW, JUL 13
Meeting
10 AM – 12 PM
MON, JUL 14
Seth dr
apt
2:30–3:30 PM
Dr Martinez
Karate
Google
Email
Travel
People
Office
Finance
Facebook
Settings
Clock
oclock
xlogo
Desktop Effects
Enable Desktop Effects
Windows Wobble when Moved
Workspaces on a Cube
Close
Trash
Places
testbed
Desktop
File System
9:00 AM
Categories
#Woman
#Man
#Baby
#Travel
#Tech
#Food&Drink


1:14 PM
August
70°F
9:00 AM
Categories
IPHONE
Music
Screen
Mirroring
Clock
testbed
Desktop
File System
Trash
Places
Help
Stop
Reload
Home
100%
View as Icons
Desktop Effects
Enable Desktop Effects
Windows Wobble when Moved
Workspaces on a Cube
Close
Office
ettings
tv

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
  - Svelte
- 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)

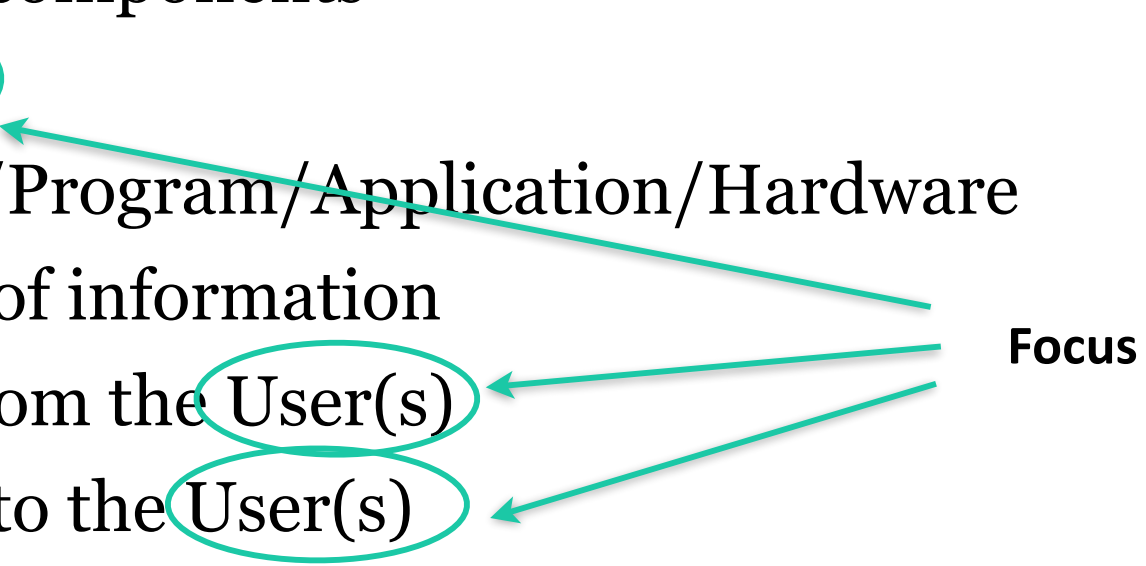
1. (countable) The part of a software application that a user sees and interacts with.

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)

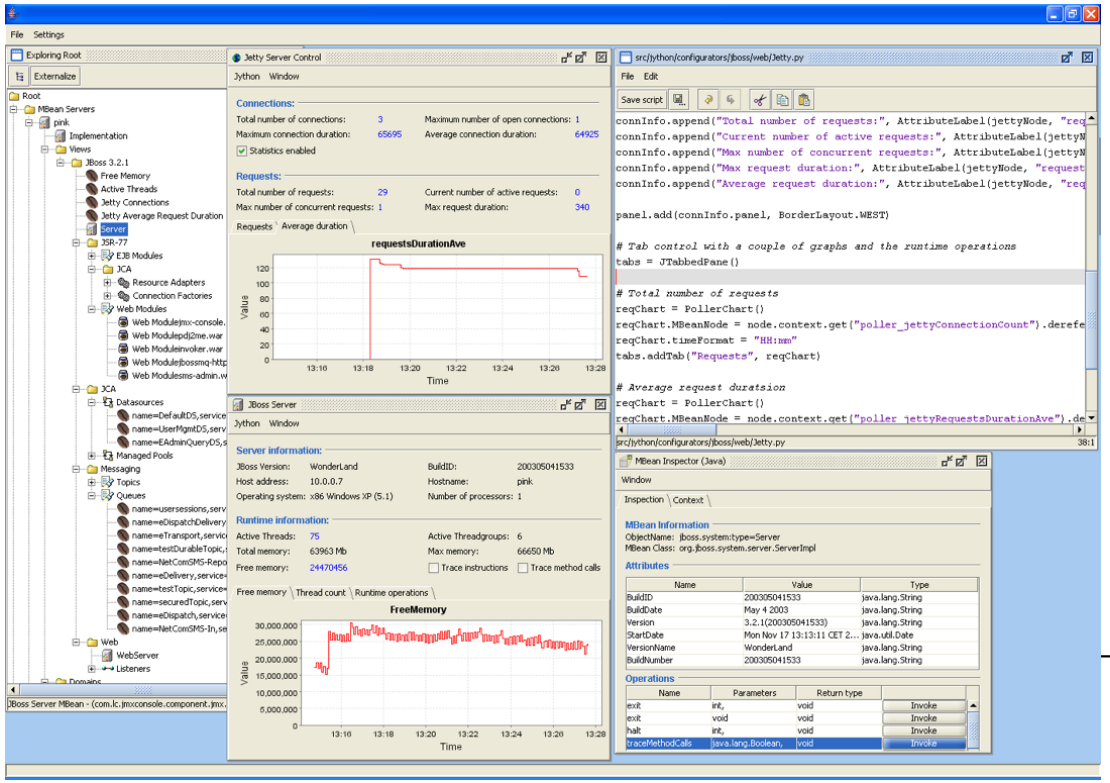
GUI for programmers

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



What constitutes a (G)UI

- The worlds smallest GUI



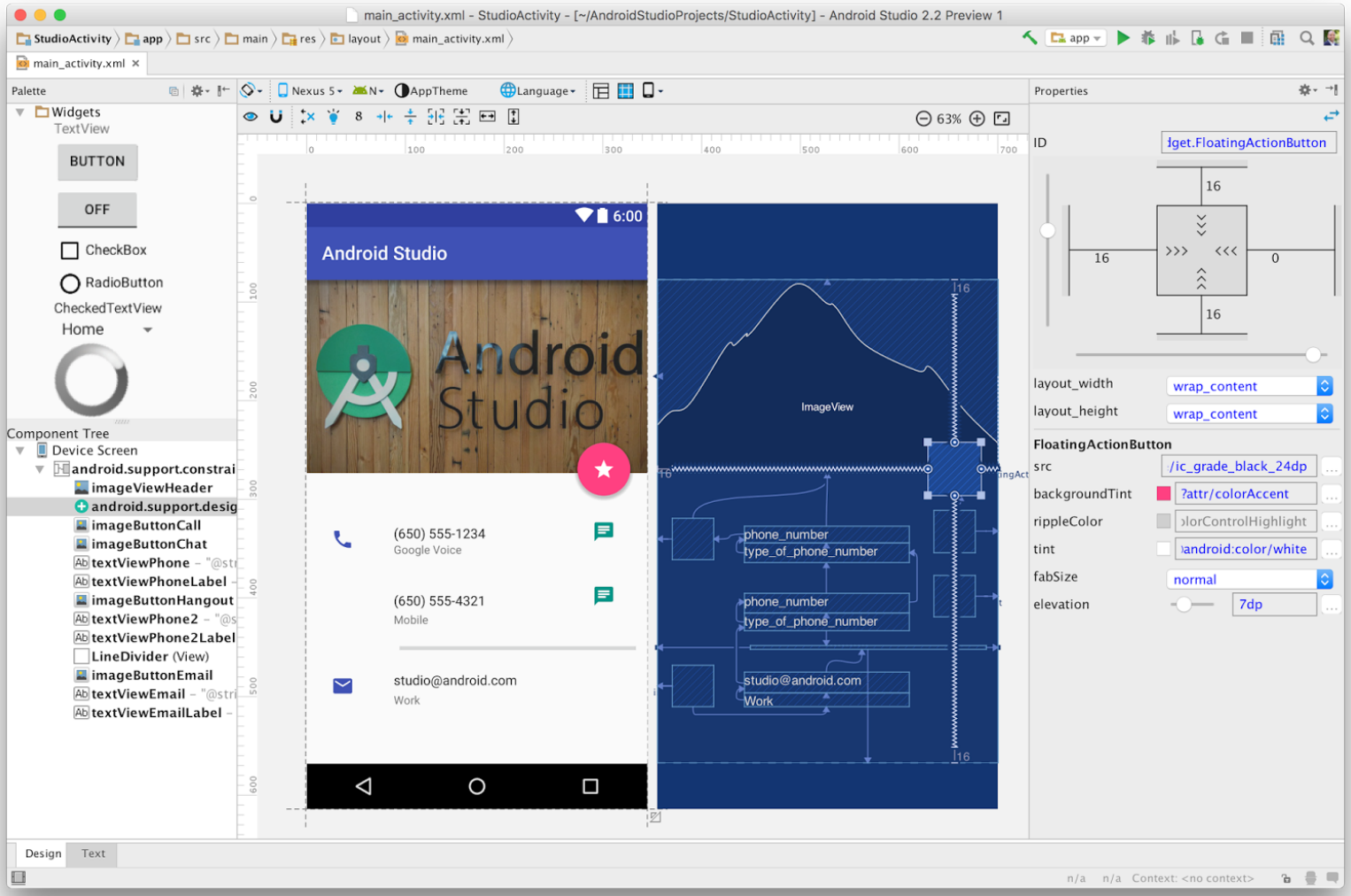
A GUI for a programmer

```
public void
paint(graphics g){
g.moveTo(100,100);
g.setColor(“ffffff”);
g.drawlineTo(200,300);
}
```

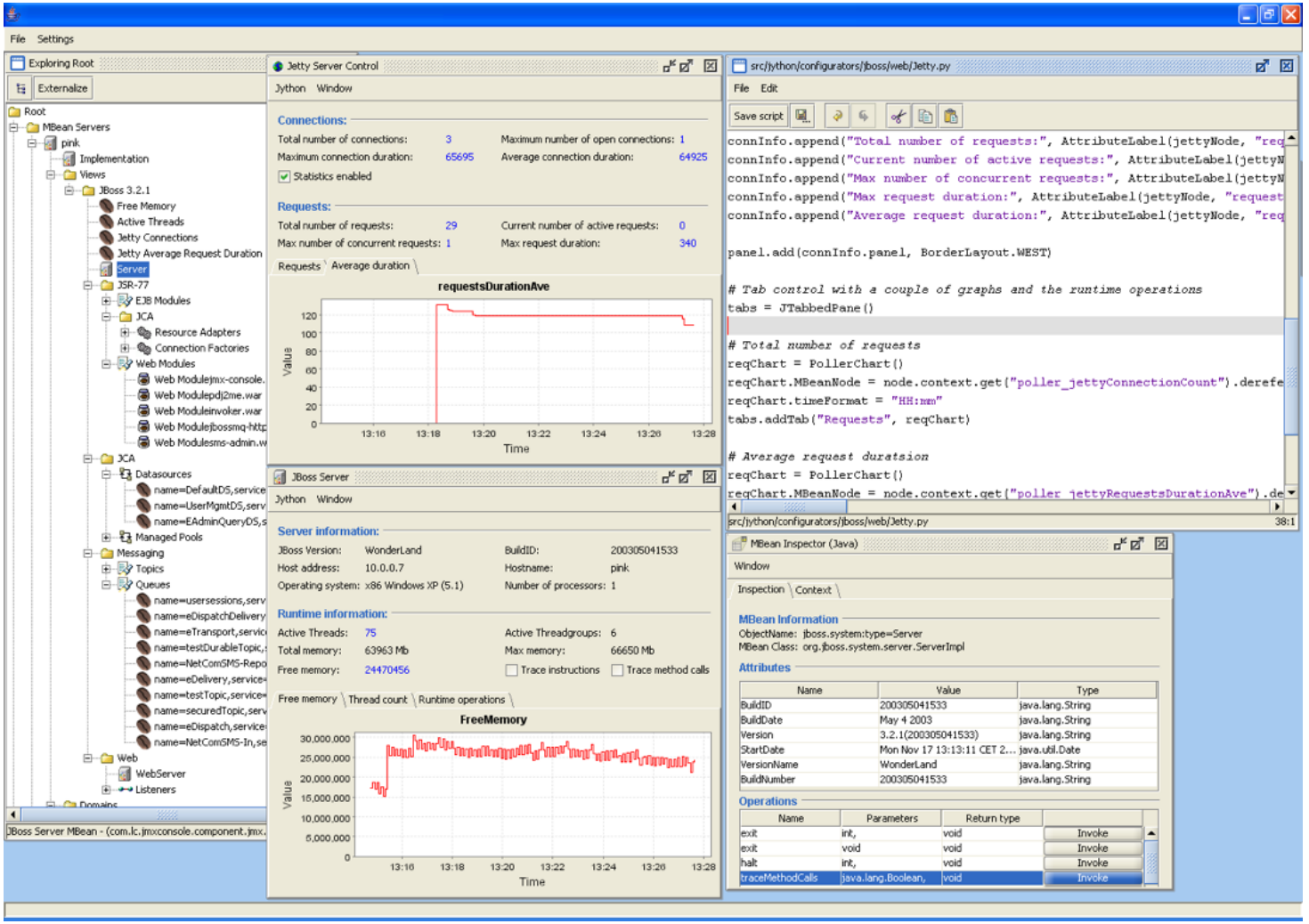
Spans from low level to very high level

```
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

Android/Java

```
protected void onDraw(Canvas canvas) {
  super.onDraw(canvas);
  // Draw the shadow
  canvas.drawOval( mShadowBounds, mShadowPaint );

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

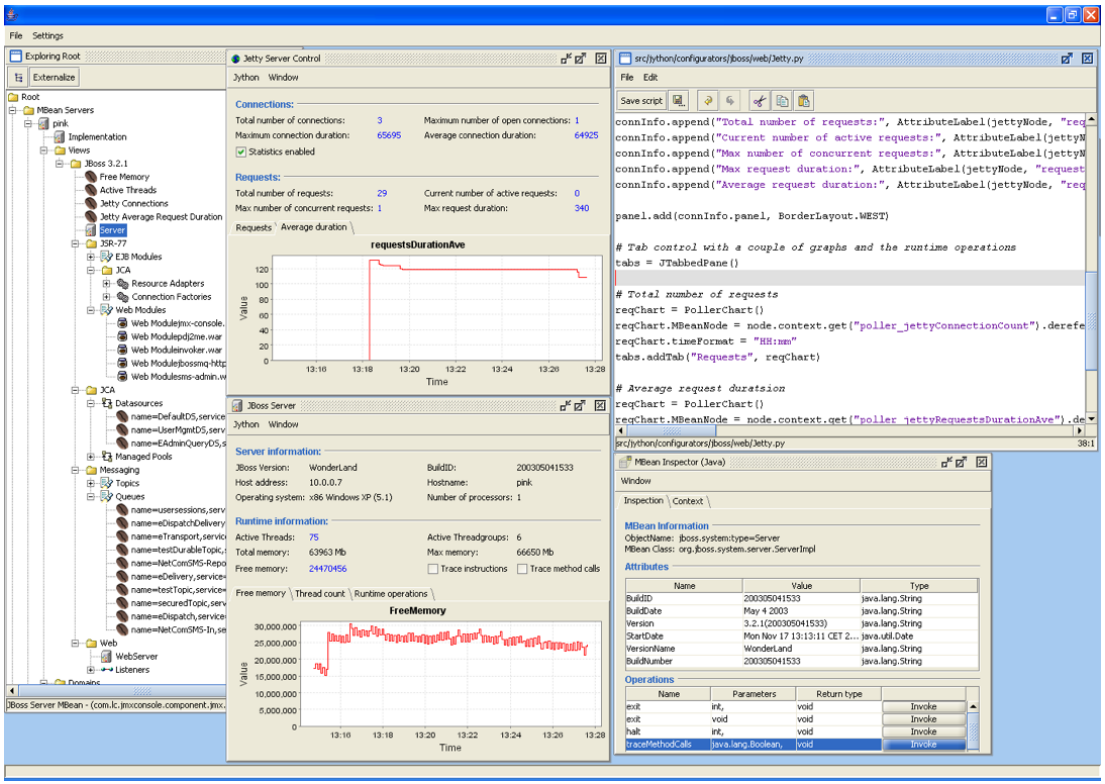
Spans from low level to very high level

Android/Kotlin

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

How do we get from

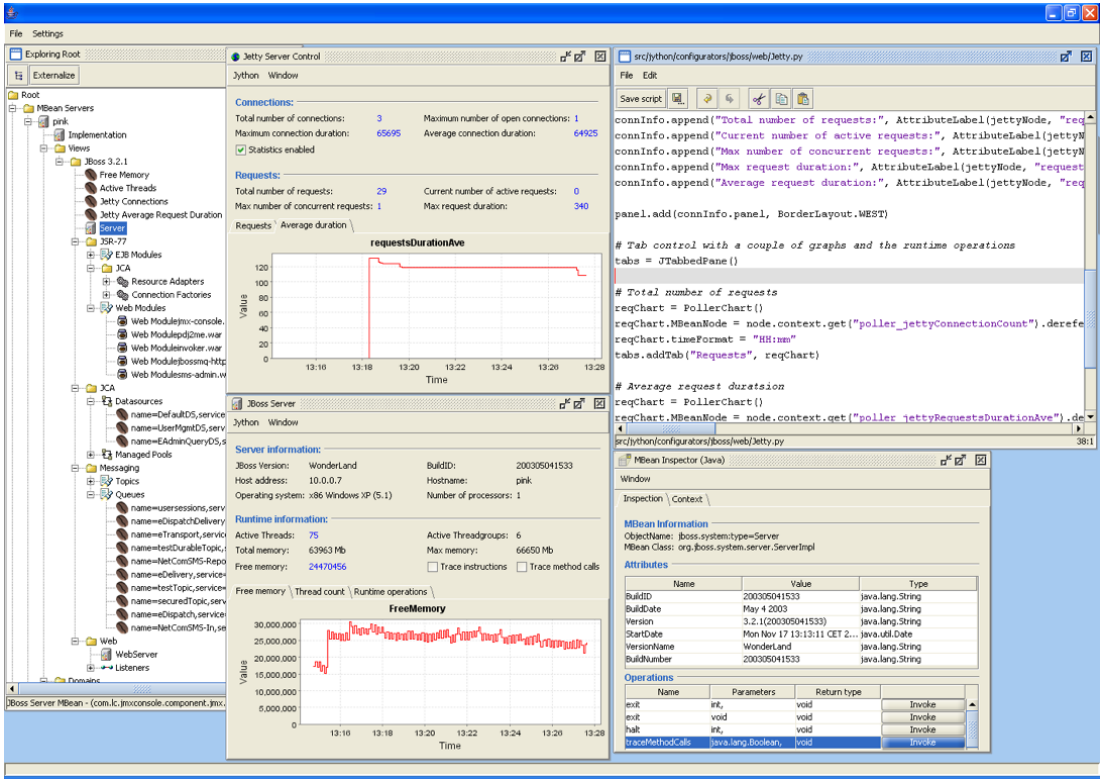
What user wants and needs (user/system requirements)



How do we get from

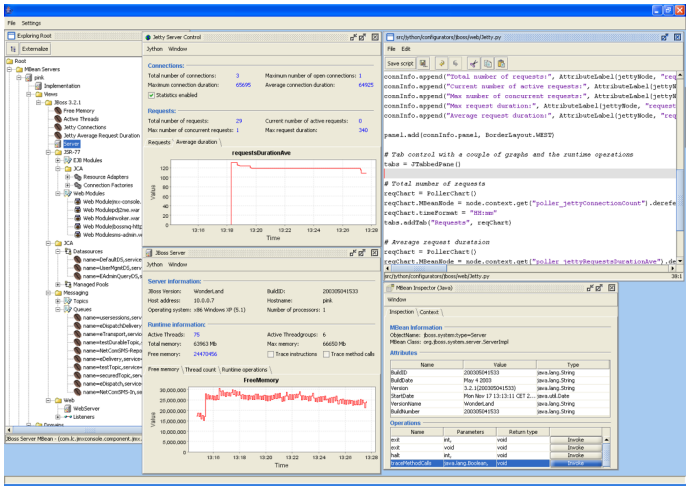
```
protected void onDraw(Canvas canvas) {
  super.onDraw(canvas);
  // Draw the shadow
  canvas.drawOval( mShadowBounds, mShadowPaint );

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



Solution

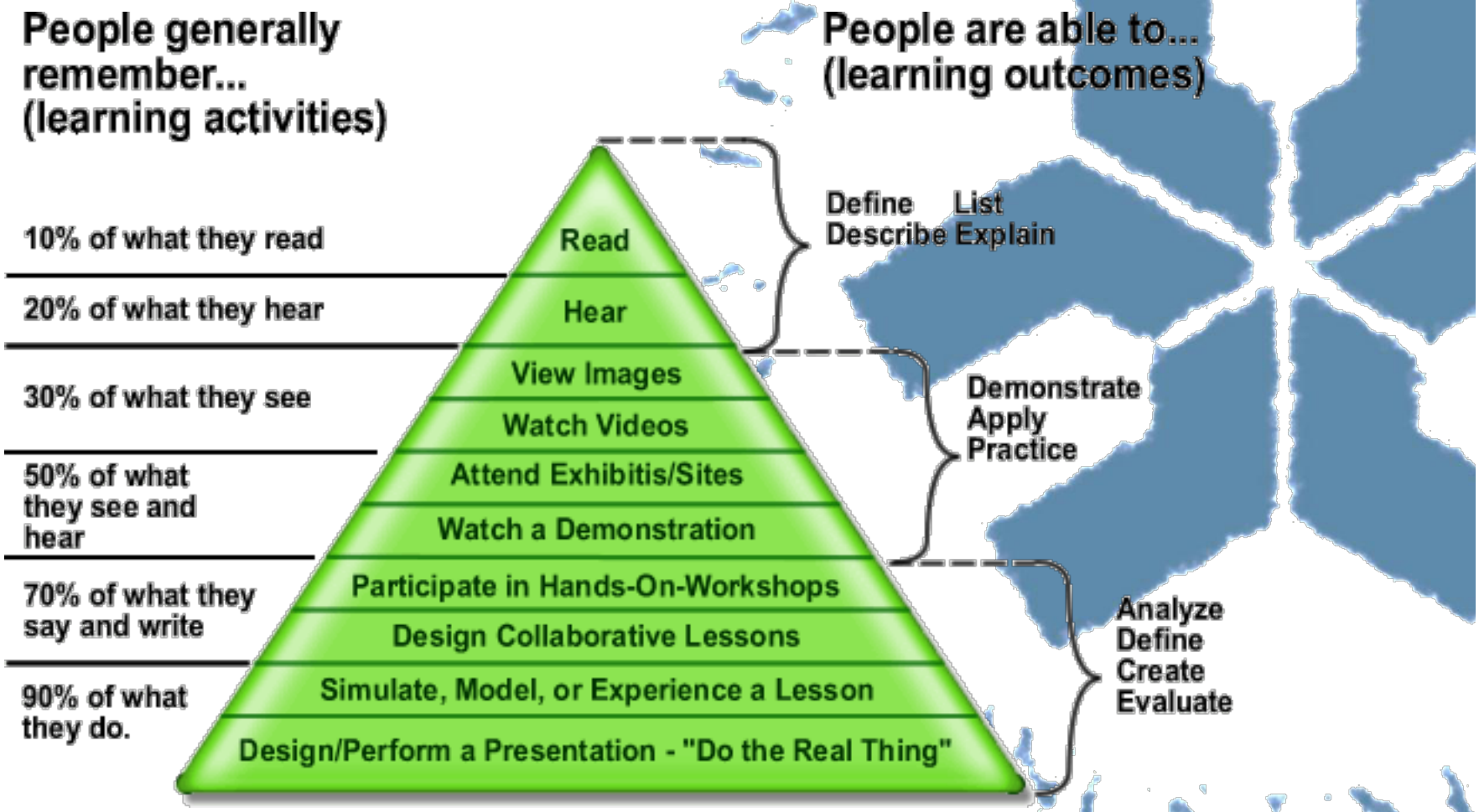
SDKs
for (gui-)code:
How to build it



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)
People are able to... (learning outcomes)
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.
Read
Hear
View Images
Watch Videos
Attend Exhibitis/Sites
Watch a Demonstration
Participate in Hands-On-Workshops
Design Collaborative Lessons
Simulate, Model, or Experience a Lesson
Design/Perform a Presentation - "Do the Real Thing"
Define List
Describe Explain
Demonstrate
Apply
Practice
Analyze
Define
Create
Evaluate

Course outline - Your work

- 3 Labs
  - Component placement (Four frameworks)
  - User interaction and component coupling
  - Communication with the rest of the world
- Mini project
  - Your own mini Library
    - Implement 2 (at lest) high-level interaction patterns in a Library.
    - 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 library

- A Library with of interaction patters (two of them)
- A example app using your Library

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)

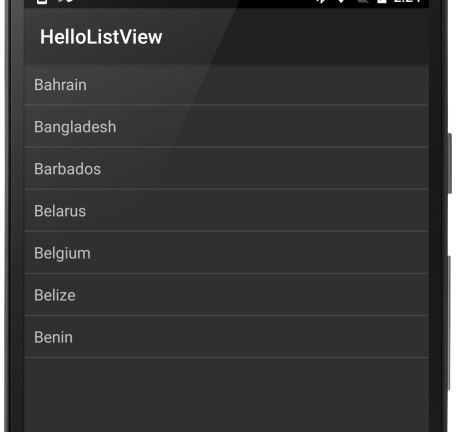
Feedback from last year

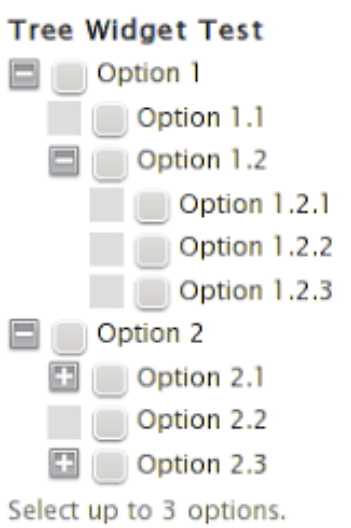
- Fun
  - But time consuming
- Lab 3 was a bit to open
  - What is “trending” on GitHub
- Project was/is easier then the labs

GUI SDKs

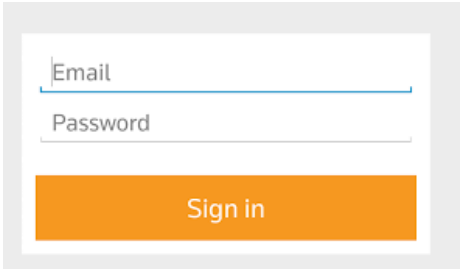
- GUI SDK contains
  - Runtime -a mapping between Abstract UI and native OS/SDK components/primitives
  - Widgets - Buttons, Menus , Text Field etc
  - Widget Messages - interoperability among widgets
  - Messaging structures for in-/output - Messages
  - Rule sets/widget constraints - for controlling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

UI -Widgets





- 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



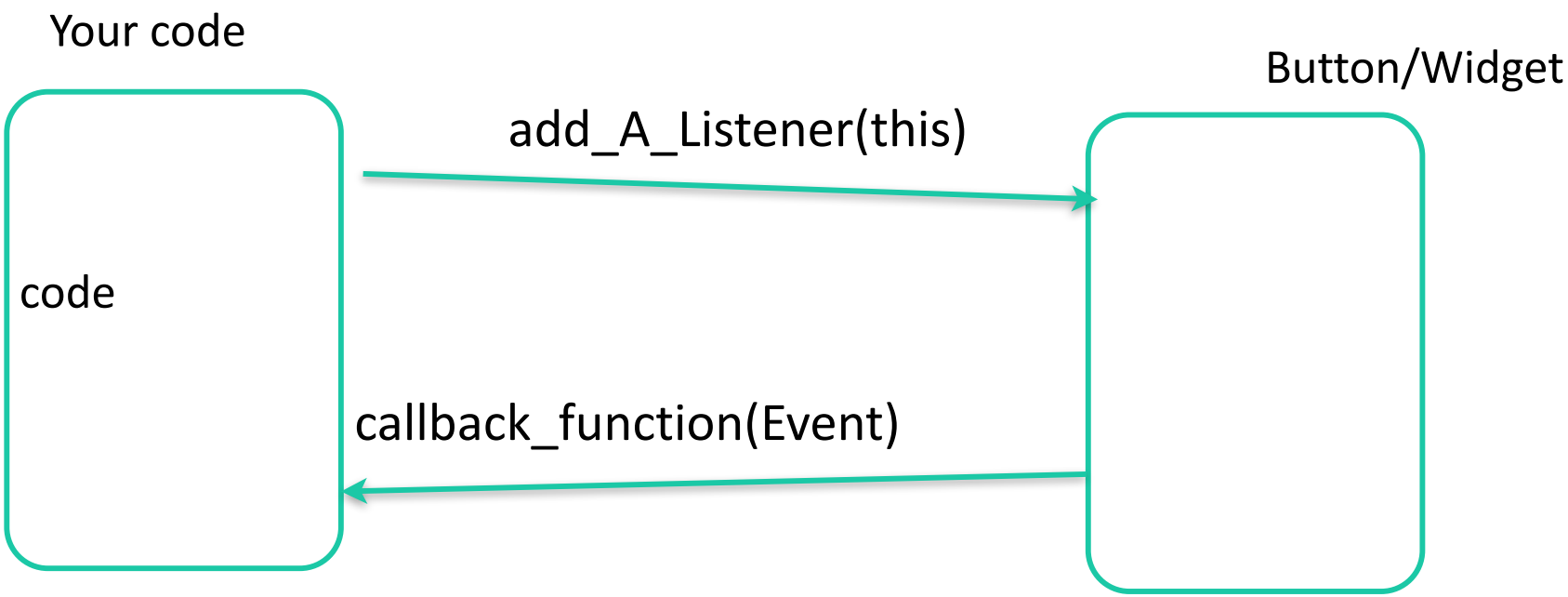
Widget Messages

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

Listeners


Your code
Button/Widget
code
add_A_Listener(this)
callback_function(Event)

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