UNIVERSITY OF TWENTE.

APPLICATION DEVELOPMENT

LECTURE 6: INHERITANCE, USERINTERFACES

```
class AppDev {

}
```

Part of SmartProducts

INTRODUCTION

APPLICATION DEVELOPMENT

Fjodor van Slooten
W241 *(Horst-wing West)*
f.vanslooten@utwente.nl

- Inheritance
- (Prototyping) Userinterfaces
- Assignment

No lecture next week,
next lecture Tuesday June 4th

slides @ vanslooten.com/appdev

ASSIGNMENT 5

- Adding methods
- 5a: determine value of money from cents to euro’s €XX.XX
- 5b: determine return coins
- Modulo operator %: remainder* of division

```
int balance = 130; // 130 cents
int euros = balance / 100;
int cents = balance % 100;
```

euros=
1

cents=
30

```
void Userinterface::printBalance() {
   ...
   lcd->print(euros); lcd->print("."); lcd->print(cents);
}
```

prints:
1.30

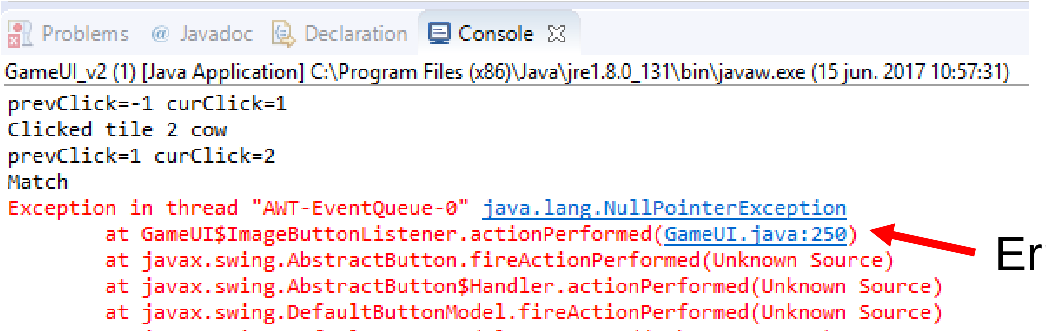
```
balance = balance - p.getPrice();
int coin50 = balance / 50;
balance = balance % 50;
int coin20 = ...
```

Assignment 5b

* The % operator returns the remainder of two numbers. For instance 10 % 3 is 1 because 10 divided by 3 leaves a remainder of 1.

CRASH…? APPLICATION NOT WORKING?

1. Scroll up in Console
2. Click on error (in own code) to go there



```
GameUI_v2 (1) [Java Application] C:\Program Files (x86)\Java\jre1.8.0_131\bin\javaw.exe (15 jun. 2017 10:57:31)
prevClick=-1 curClick=1
Clicked tile 2 cow
prevClick=1 curClick=2
Match
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
	at GameUI$ImageButtonListener.actionPerformed(GameUI.java:250)
	at javax.swing.AbstractButton.fireActionPerformed(Unknown Source)
	at javax.swing.AbstractButton$Handler.actionPerformed(Unknown Source)
	at javax.swing.DefaultButtonModel.fireActionPerformed(Unknown Source)
```

Error at line 250

Finding a problem: debug or use ***System.out.println()***

Print values of variables!

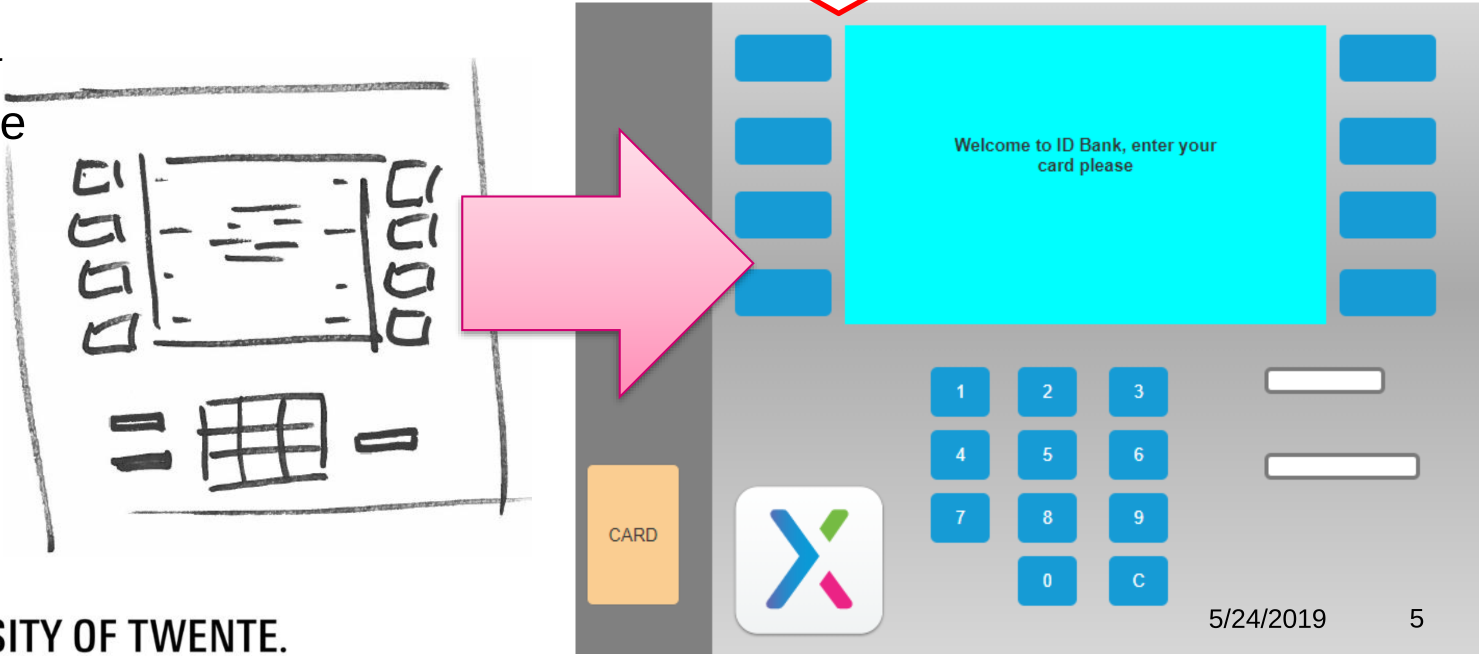
PROTOTYPING USERINTERFACES

FROM DESIGN TO PROTOTYPE

Software tools to make interactive demos & interfaces:

- Java
- Axure
- …

With Axure, you can publish a prototype online:

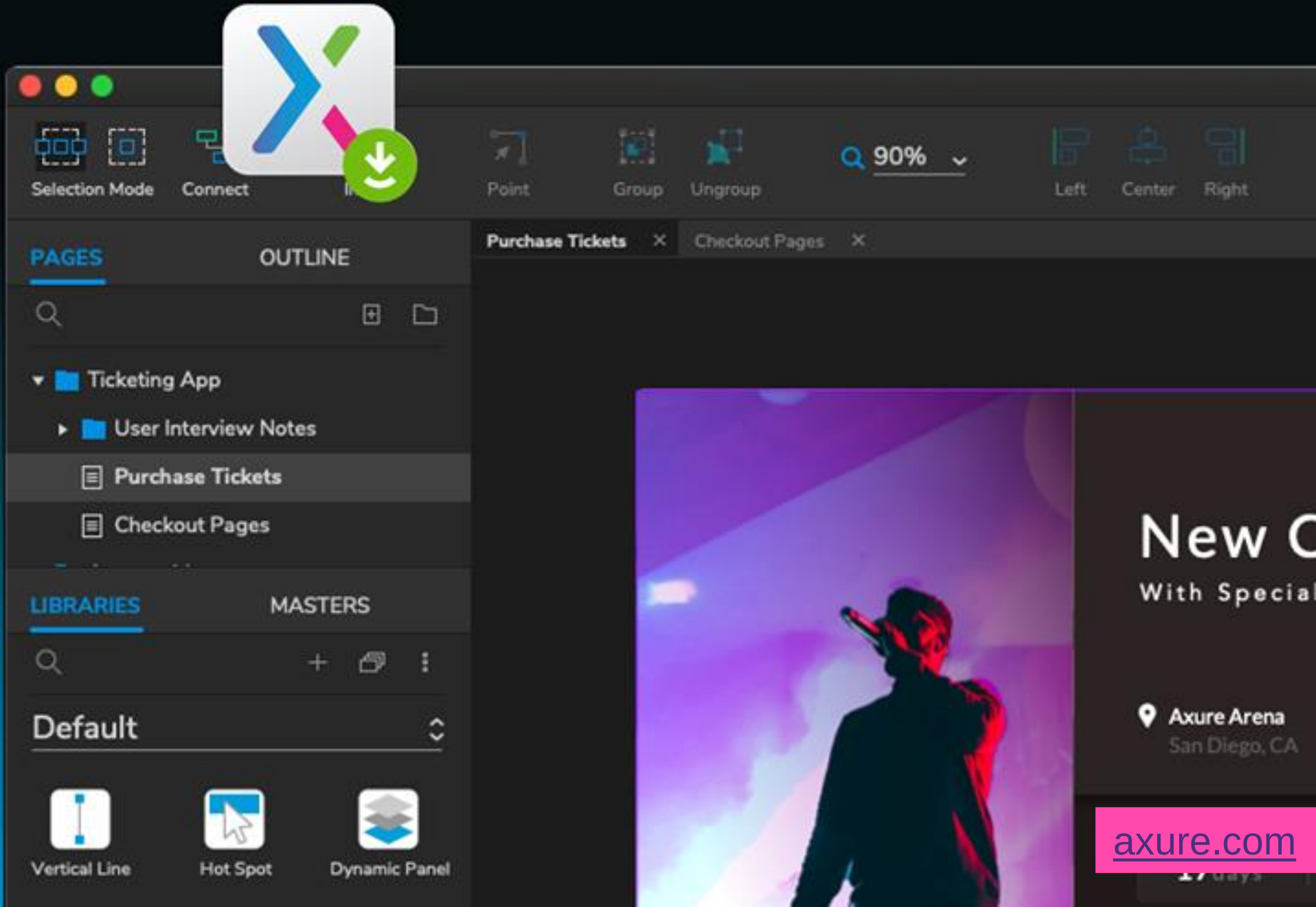


Welcome to ID Bank, enter your card please

AXURE

- Flexible, prototype apps, websites
- Create complex interactions
- Can be used for low- & high fidelity prototypes
- License available on Canvas:

  Application Development > Axure license



axure.com

LEARN

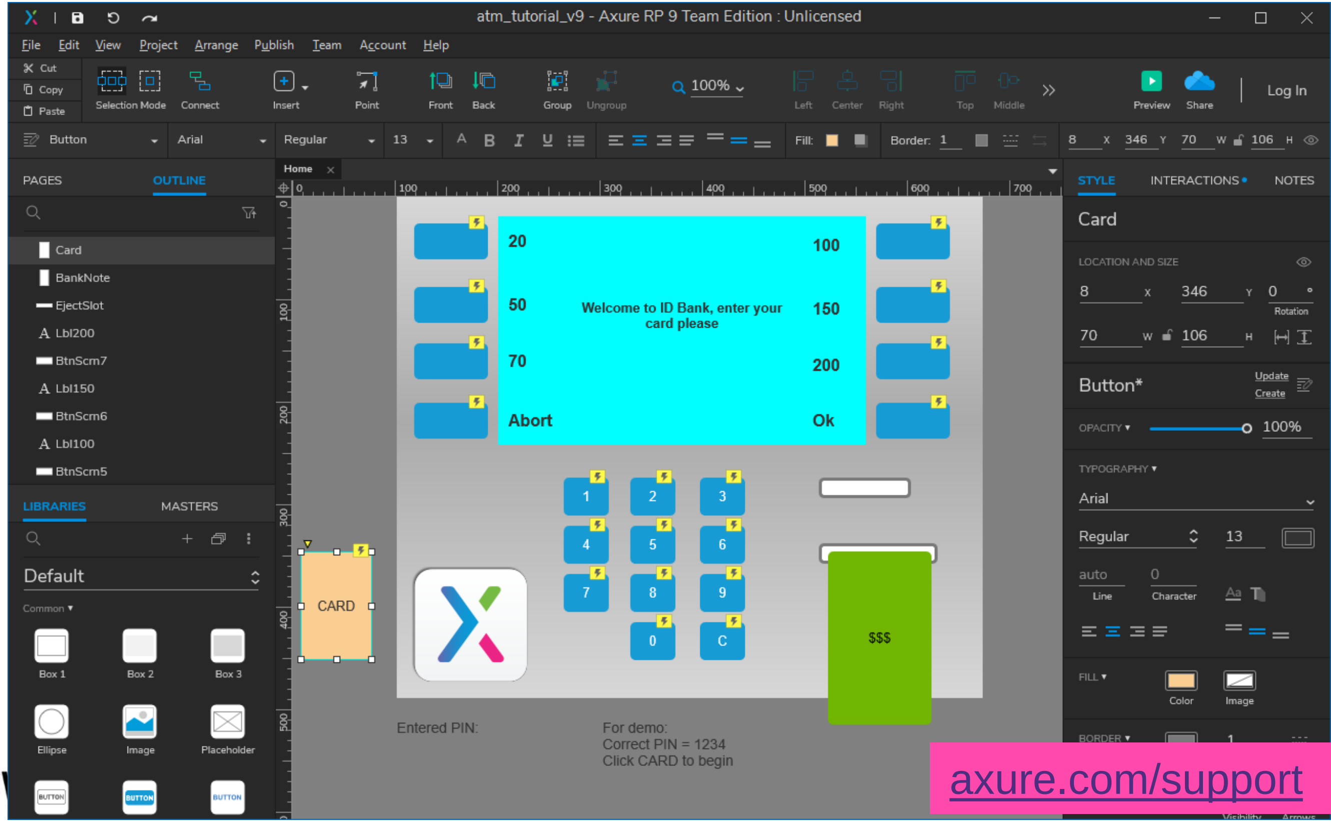
- Tutorials on axure.com/support
- Practice tutorial:

  Build a prototype of interface for ATM:

vanslooten.com/appdev > Additional Online Materials, UI Prototyping:

🔗 UI Prototyping with Axure tutorial - prototype an ATM

🔗 UI Prototyping with Java tutorial - prototype an ATM



atm_tutorial_v9 - Axure RP 9 Team Edition : Unlicensed

Welcome to ID Bank, enter your card please

Entered PIN:

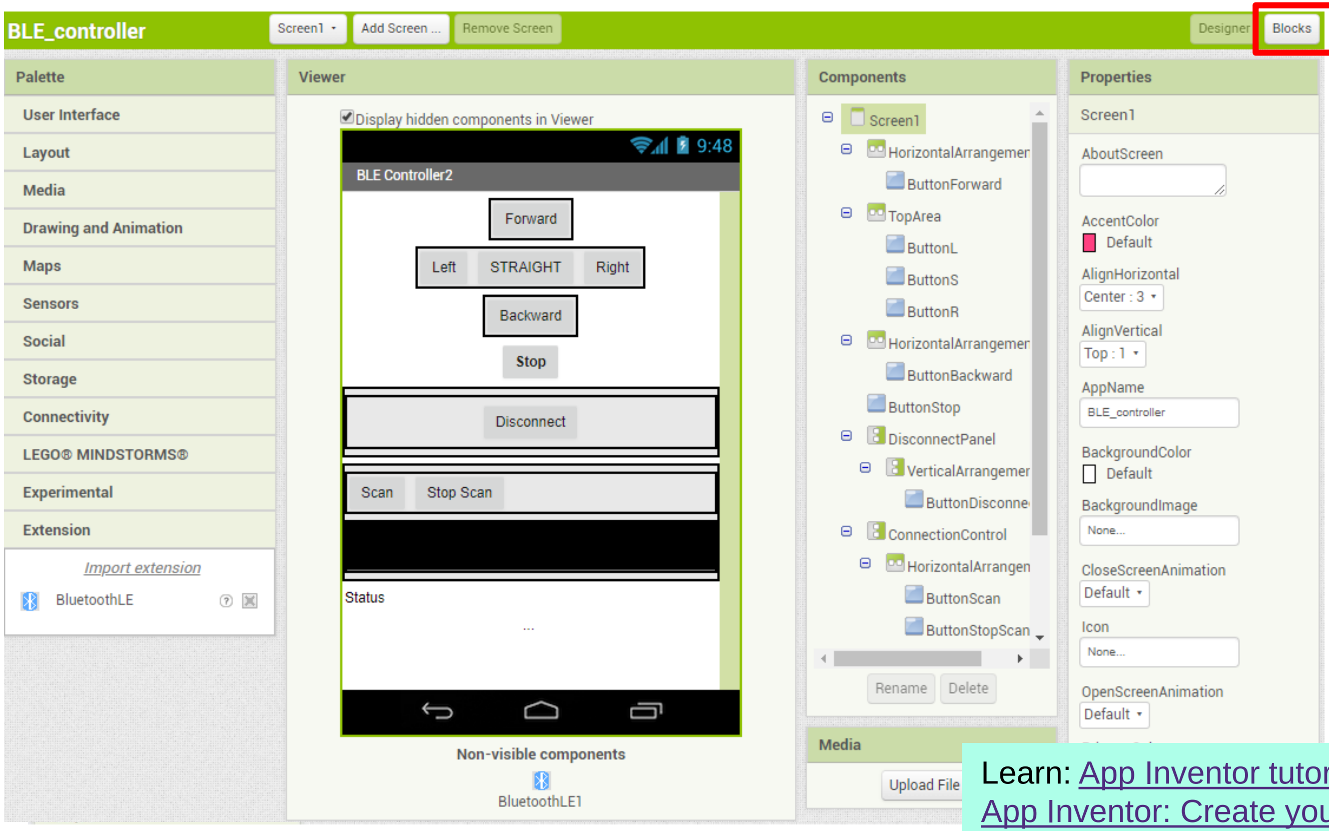
For demo:
Correct PIN = 1234
Click CARD to begin

axure.com/support

APP PROTOTYPING: APP INVENTOR

BUILD MOBILE APPS

ai2.appinventor.mit.edu



☑ Display hidden components in Viewer

Screen1

AboutScreen

AccentColor: Default

AlignHorizontal: Center : 3

AlignVertical: Top : 1

AppName: BLE_controller

BackgroundColor: Default

BackgroundImage: None...

CloseScreenAnimation: Default

Icon: None...

OpenScreenAnimation: Default

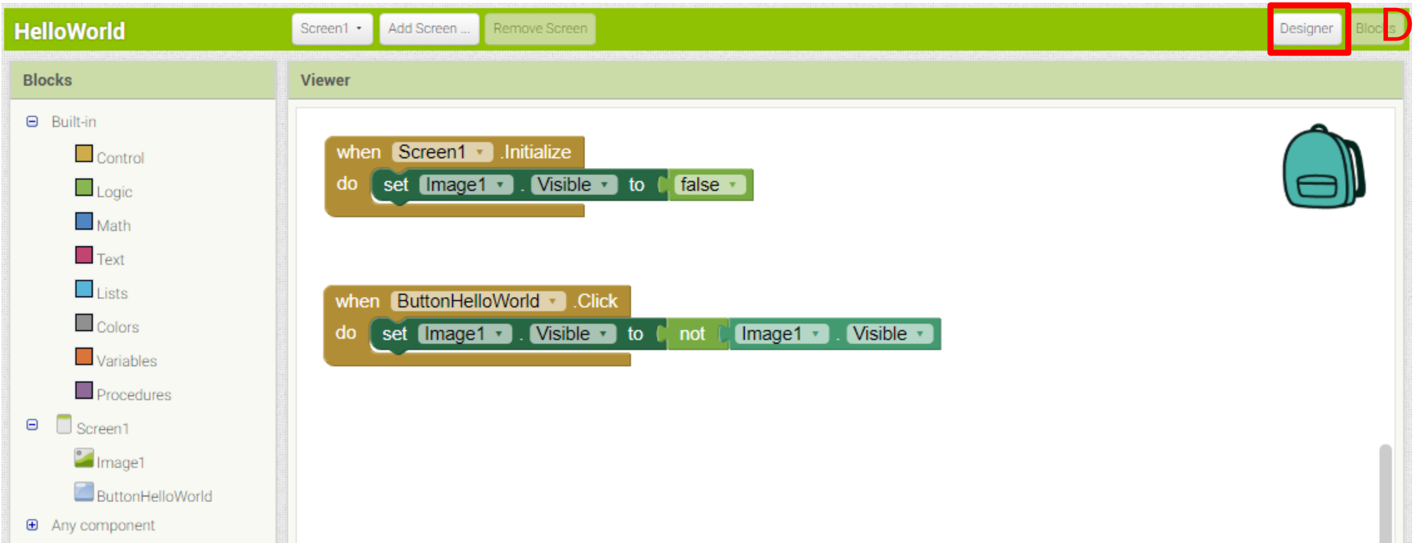
Program

Learn: App Inventor tutorial,
App Inventor: Create your own Android Apps

APP PROTOTYPING: APP INVENTOR

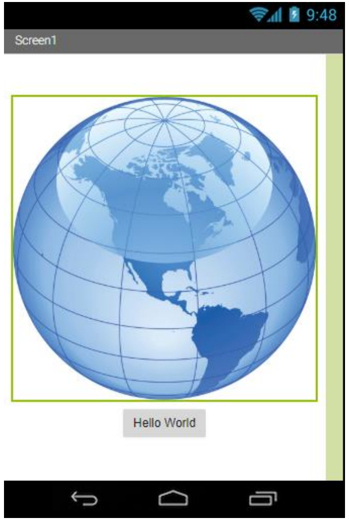
BUILD MOBILE APPS

ai2.appinventor.mit.edu



```
when Screen1 .Initialize
do  set Image1 . Visible to false

when ButtonHelloWorld .Click
do  set Image1 . Visible to not Image1 . Visible
```

Design

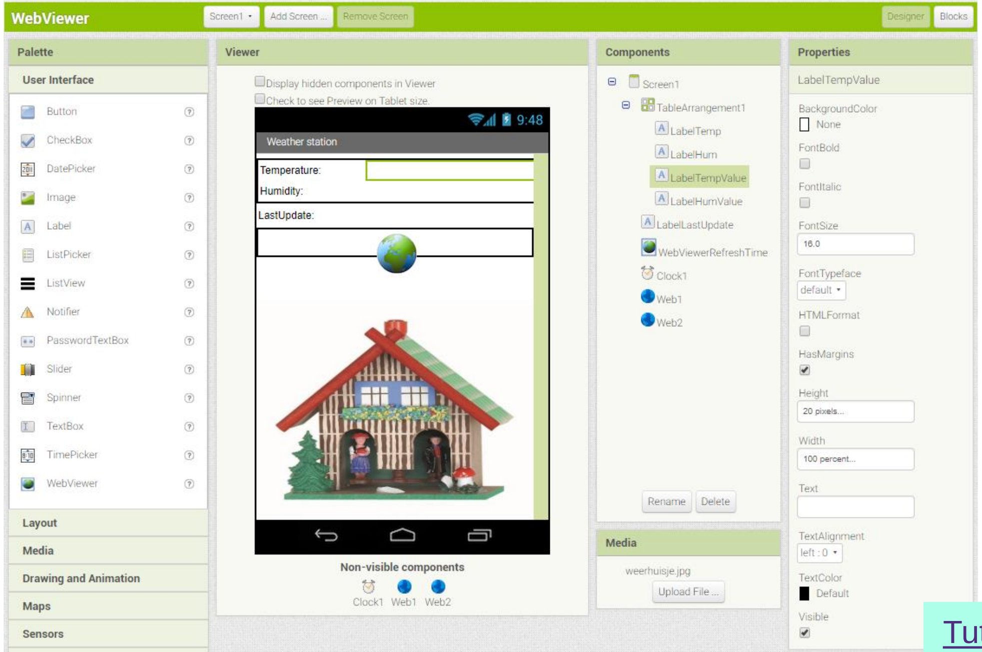


Learn: App Inventor tutorial,
App Inventor: Create your own Android Apps

APP INVENTOR TUTORIAL

BUILD AN APP FOR A CONNECTED WEATHER STATION

vanslooten.com/appdev > Additional Online Materials, UI Prototyping



□ Display hidden components in Viewer
□ Check to see Preview on Tablet size.

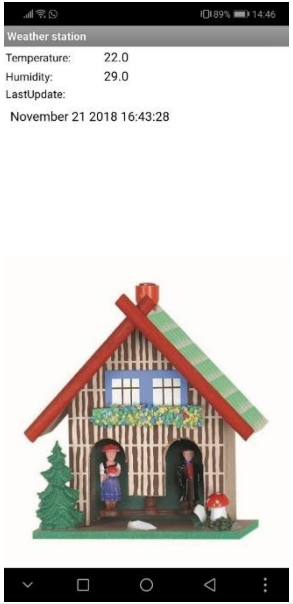
9:48
Weather station
Temperature:
Humidity:
LastUpdate:

Non-visible components
Clock1 Web1 Web2

Components:
- Screen1
  - TableArrangement1
    - LabelTemp
    - LabelHum
    - LabelTempValue
    - LabelHumValue
  - LabelLastUpdate
  - WebViewerRefreshTime
  - Clock1
  - Web1
  - Web2

Media: weerhuisje.jpg

Properties: LabelTempValue
- BackgroundColor: None
- FontBold: □
- FontItalic: □
- FontSize: 16.0
- FontTypeface: default ▾
- HTMLFormat: □
- HasMargins: ☑
- Height: 20 pixels...
- Width: 100 percent...
- Text:
- TextAlignment: left : 0 ▾
- TextColor: Default
- Visible: ☑



Weather station

Temperature:	22.0
Humidity:	29.0
LastUpdate:	

November 21 2018 16:43:28

Tutorial: Build an App with App Inventor which can display values of a connected sensor

More: Control an RGB LED from an Android App via Bluetooth

UNIVERSITY OF TWENTE.

FROM DESIGN TO CODE

- Small steps, iterate (while designing, already perform small tests)
- Test sensors, build small parts, write small test programs using examples
- Later on: put smaller parts together

Rules of thumb:

- Don't try to design everything up front
- Just start (its better to start with sloppy code full of mistakes, than to postpone and wait for a better design)
- Never write more than **10 lines** of code without testing

Making mistakes is the best way to learn

FROM DESIGN TO CODE: HOW TO TEST?

- Call a method, see result…
- Print statements!

Arduino/C++:

```
void setup() {
   // test methodX:
   object.methodX(); // what happens?
   Serial.println("MethodX just finished");
   // Check output of print-statements in Serial Monitor
}

void loop() {
   // get the reading(s) from sensor
   light = lightSensor.readRaw();
   Serial.print("light="); Serial.println(light);
}
```

Java:

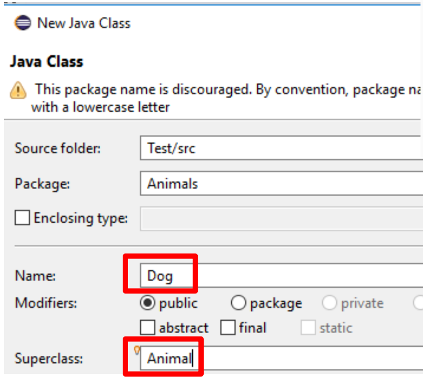
```
public MachineUI() { // constructor
   // test methodX:
   object.methodX(); // what happens?
   System.out.println("MethodX just finished");
   // Check output of print-statements in Console
}

void read() {
   // get the reading(s) from sensor
   light = lightSensor.readRaw();
   System.out.println("light="+light);
}
```

INHERITANCE

Head First: p59-69, 134-137 ***Aan de slag met*:** 8.1-8.4

- New class inherits from existing
- Existing: superclass
- New: sub/derived class
- Sub is often extension (with new/other methods/properties)

```
public class Dog extends Animal {
}
```

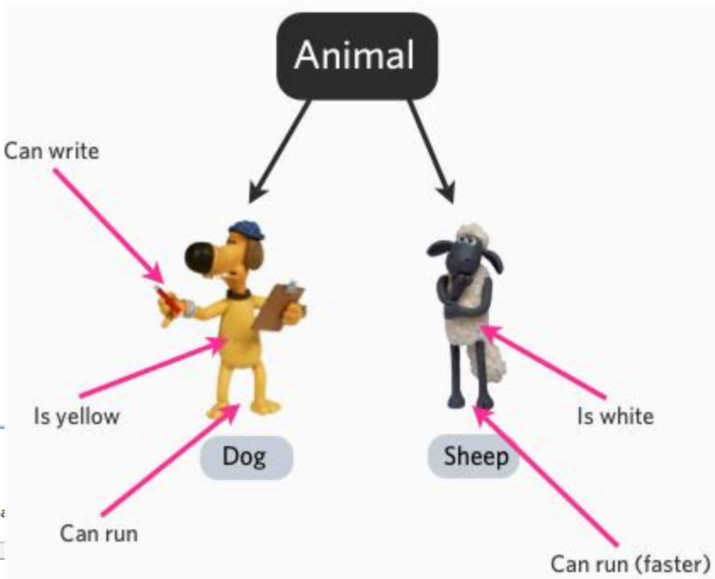


New Java Class

Java Class

⚠ This package name is discouraged. By convention, package na
with a lowercase letter

Source folder:	Test/src
Package:	Animals
□ Enclosing type:	
Name:	Dog
Modifiers:	● public ○ package ○ private ○ protected □ abstract □ final □ static
Superclass:	Animal

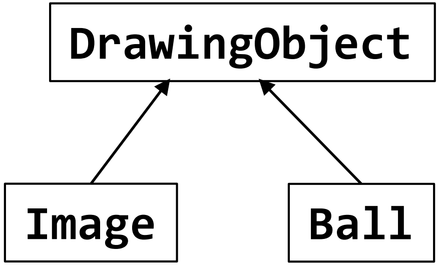


INHERITANCE

- ArrayList 'accepts' family members
- Who is who? **`instanceof`**

DrawingObject is superclass of Image and Ball

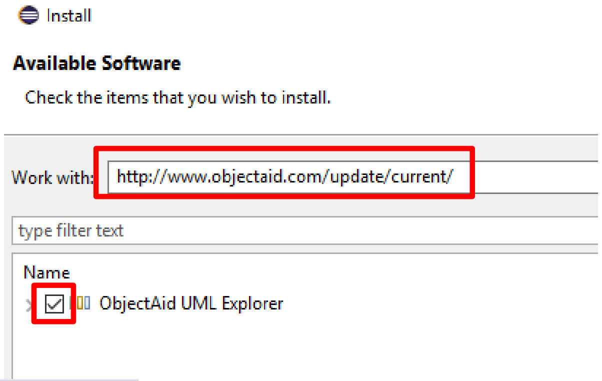
```
ArrayList<DrawingObject> list;

list = new ArrayList<DrawingObject>();

Image i = new Image();
Ball b = new Ball();
list.add(i);
list.add(b);

// what is item x in the list?
if ( list.get(x) instanceof Ball )
   System.out.println("It is a Ball!");
```



Is the element in the list of the type **Ball**? Or short: is this a **Ball**?

INHERITANCE: CLASS DIAGRAM

- Choose ***Help > Install New Software*** from menu
- Enter update site: http://www.objectaid.com/update/current/ (press Enter)
- Select "ObjectAid UML Explorer“
- Press Next (2x)
- Accept license, Finish



Install

Available Software

Check the items that you wish to install.

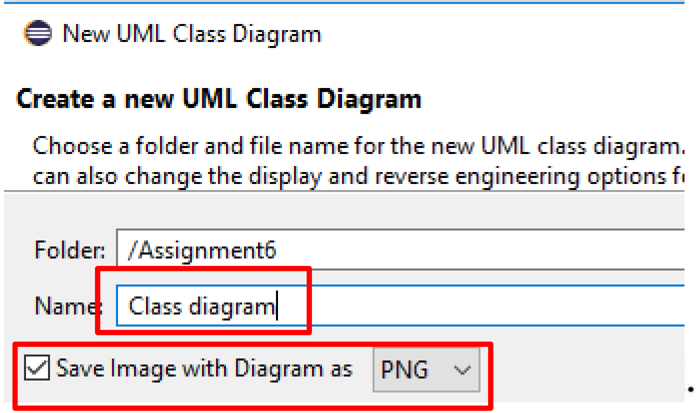
Work with: http://www.objectaid.com/update/current/

type filter text

Name

☑ ObjectAid UML Explorer

Use:
File > *New* > *Other*, choose *Object Aid UML Diagram* > *Class Diagram*



New UML Class Diagram

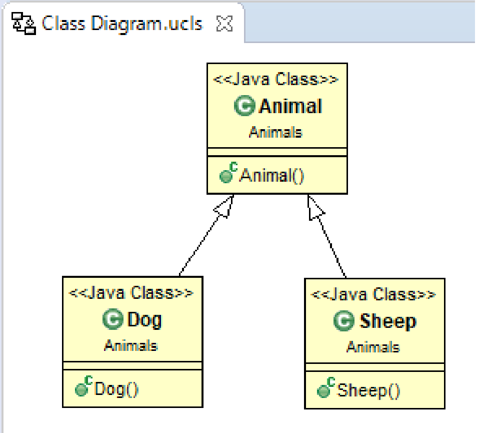
Create a new UML Class Diagram

Choose a folder and file name for the new UML class diagram. can also change the display and reverse engineering options f

Folder: /Assignment6

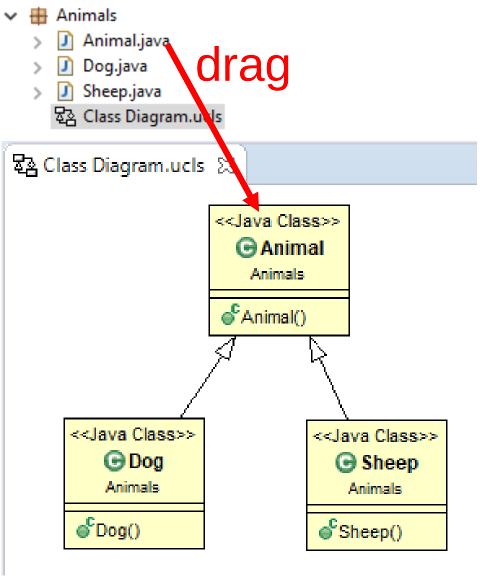
Name: Class diagram

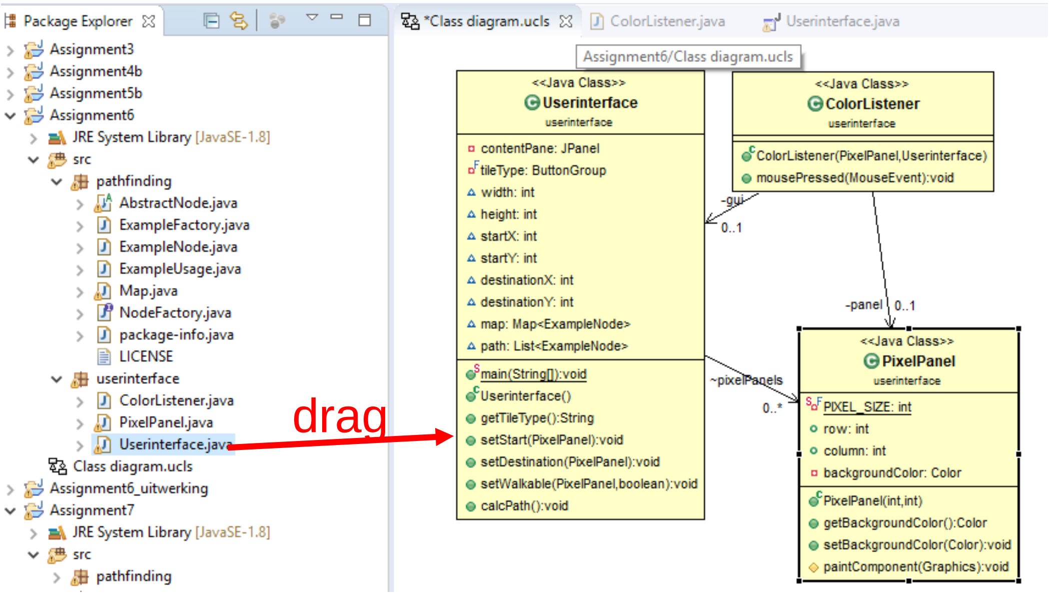
☑ Save Image with Diagram as PNG



INHERITANCE: CLASS DIAGRAM

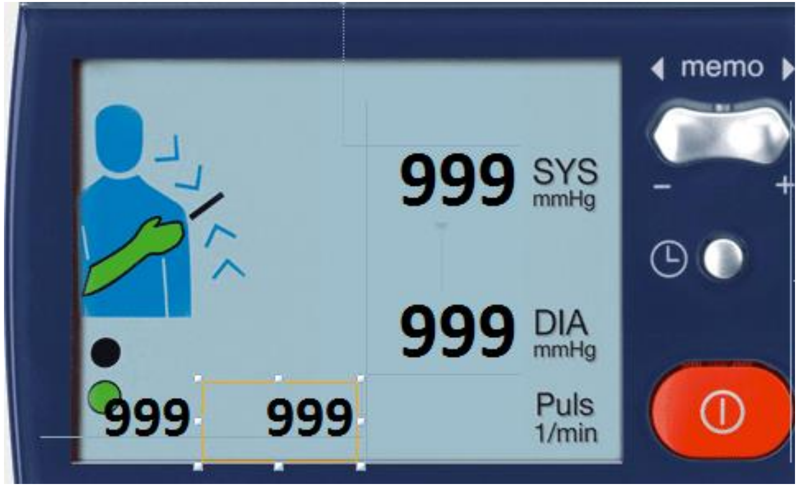
- Use: *File > New > Other*, choose *Object Aid UML Diagram > Class Diagram*
- Drag classes from package explorer into diagram

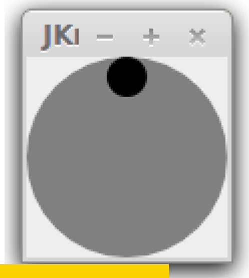




USER INTERFACES

- Add images to style UI elements
- Layout, layers
- Borders, icons
- Advanced UI elements
- Create your own UI elements

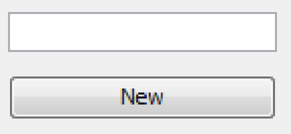


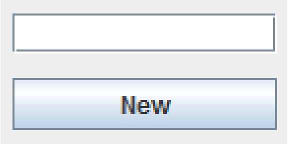


JKnob.java

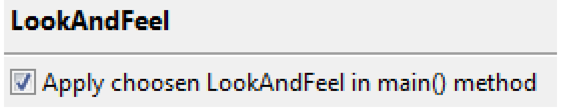
Example: RoundPlayPauseButton

LOOK & FEEL



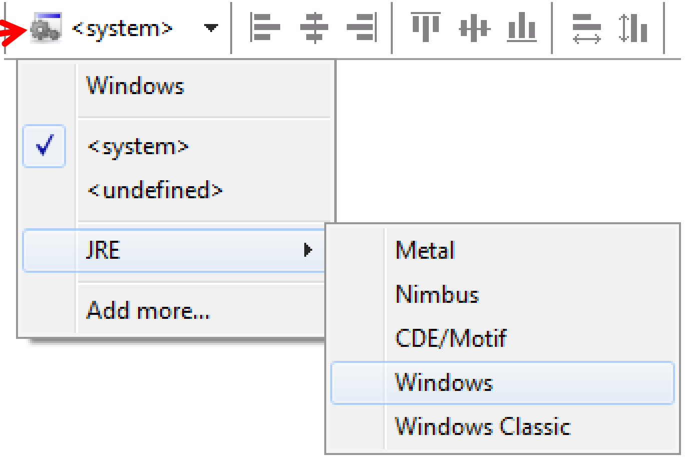


- Once: *Window > Preferences, WindowBuilder > Swing > LookAndFeel*
- Per application: Top of WindowBuilder

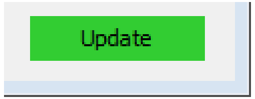


LookAndFeel

☑ Apply choosen LookAndFeel in main() method



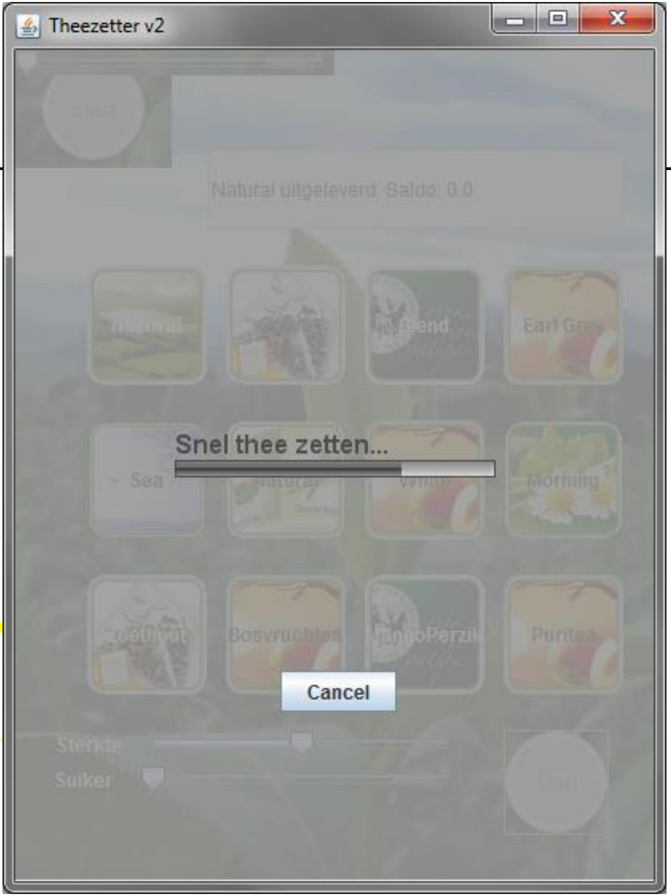
Flat-design button?:



```
JButton button = new JButton("Update");
button.setUI((ButtonUI) BasicButtonUI.createUI(button));
button.setBackground(new Color(50, 205, 50));
button.setBorder(null);
```

PANEL

- Is container
- Separate parts of UI
- Each own layout
- Turn on/off: **setVisible()**



Theezetter v2

Natural uitgeleverd. Saldo: 0.0

Snel thee zetten...

Panel with 12 buttons in *Grid Layout*

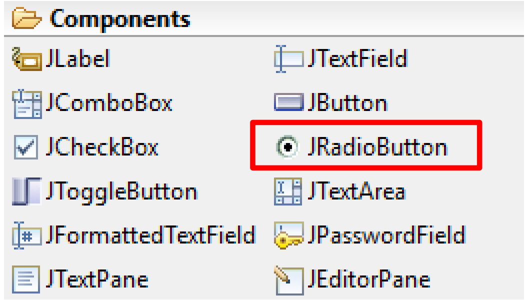
`jPanelMain`: main program
`jPanelProgress`: semi transparent progress bar

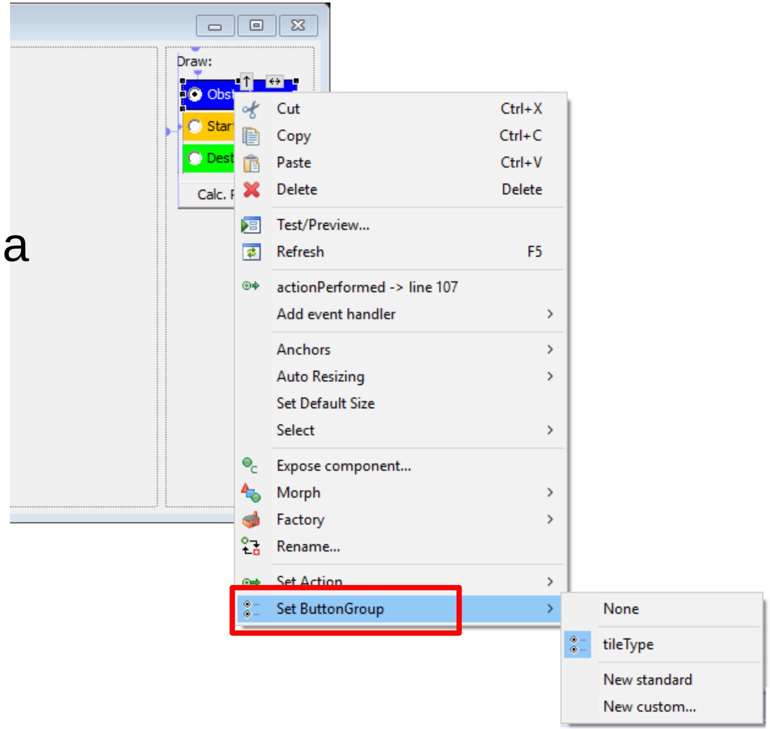
Example: Tea Maker

RADIO BUTTONS

Part of today's assignment

- Select **one** from set of buttons
- Add buttons
- Group them: by adding them to a **ButtonGroup**

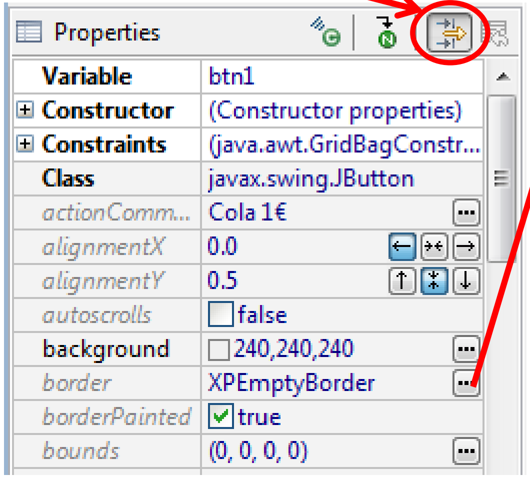




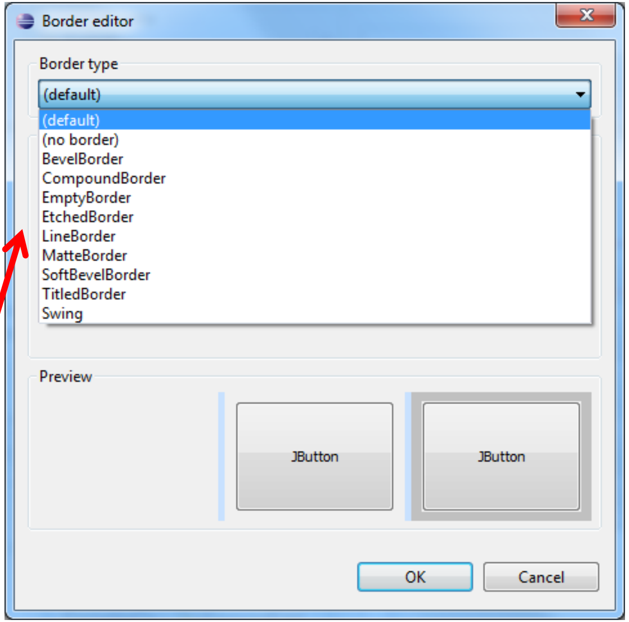
BORDERS

CAN BE SET ON (ALMOST) ALL UI COMPONENTS

- Assignment 6b: highlighted border
- Via advanced properties



Variable	btn1
⊞ Constructor	(Constructor properties)
⊞ Constraints	(java.awt.GridBagConstr...
Class	javax.swing.JButton
actionComm...	Cola 1€
alignmentX	0.0
alignmentY	0.5
autoscrolls	□ false
background	240,240,240
border	XPEmptyBorder
borderPainted	☑ true
bounds	(0, 0, 0, 0)

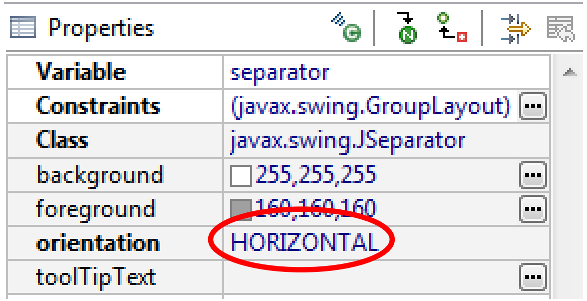


Border type: (default)

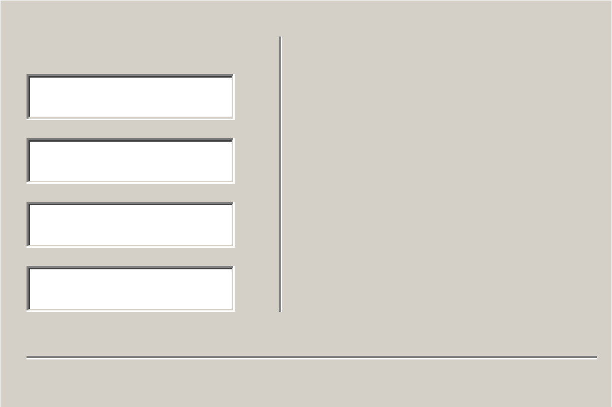
- (default)
- (no border)
- BevelBorder
- CompoundBorder
- EmptyBorder
- EtchedBorder
- LineBorder
- MatteBorder
- SoftBevelBorder
- TitledBorder
- Swing

DIVIDE/ORGANISE PARTS OF UI

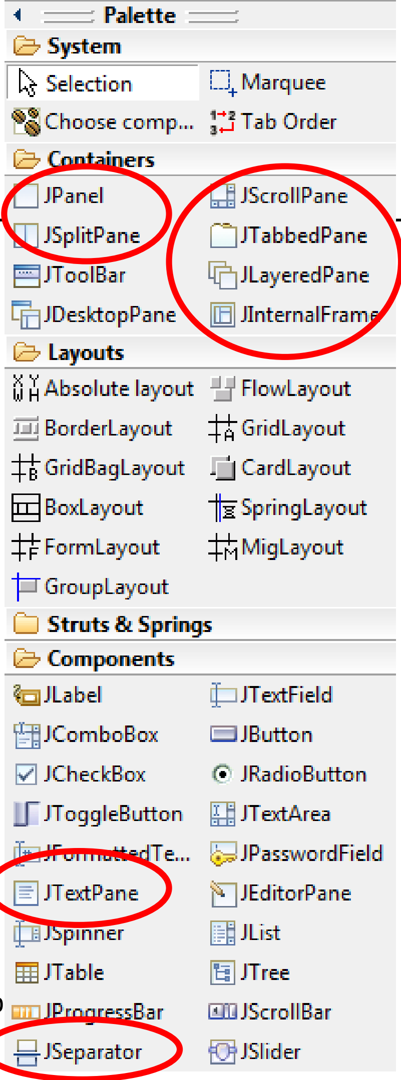
- Separators
- Tabs (Tabbed Pane)
- Split Pane
- Scroll Pane
- Layered Pane



Variable	separator
Constraints	(javax.swing.GroupLayout)
Class	javax.swing.JSeparator
background	□ 255,255,255
foreground	■ 160,160,160
orientation	HORIZONTAL
toolTipText	



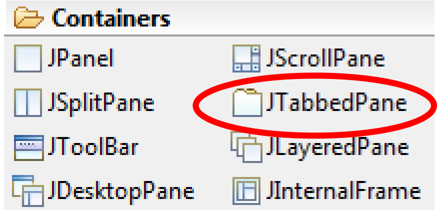
App

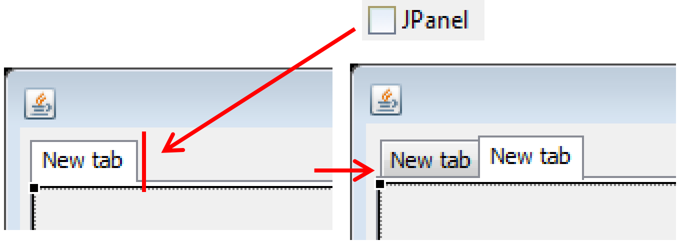


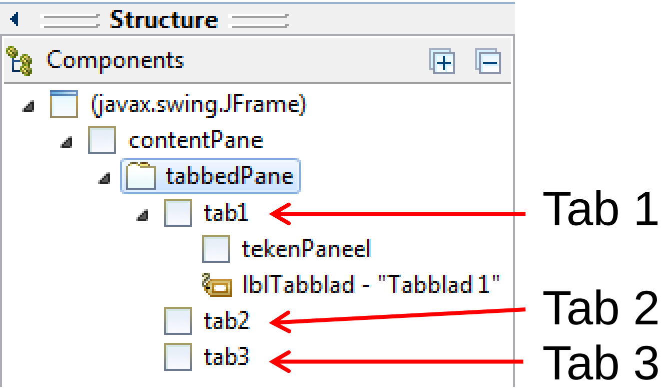
TABS

TABBED PANE

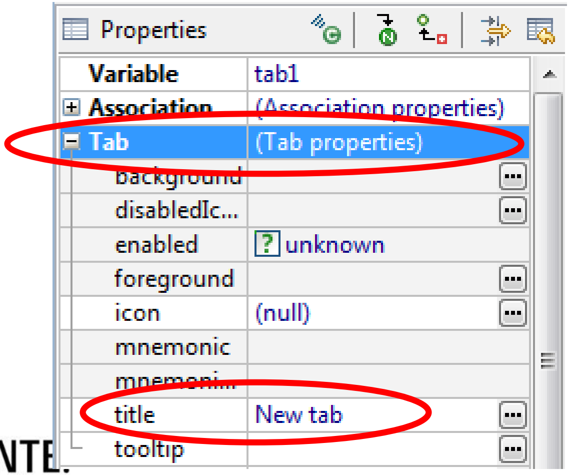
1. Place Tabbed Pane
2. Drag a Panel onto it (becomes 1st tab) (create user interface in that panel)
3. Add more tabs:
   - Place new Panel next to tab
   - If green plus-sign appears, release







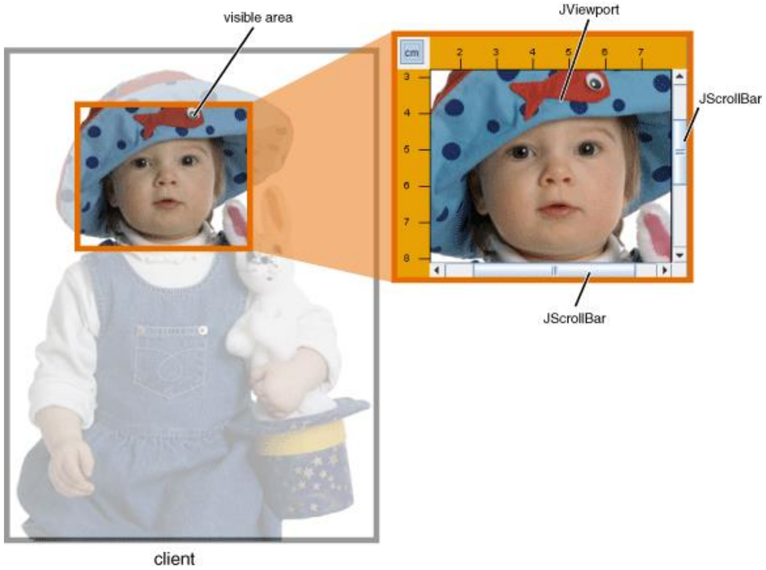
- (javax.swing.JFrame)
  - contentPane
    - tabbedPane
      - tab1
        - tekenPaneel
        - lblTabblad - "Tabblad 1"
      - tab2
      - tab3

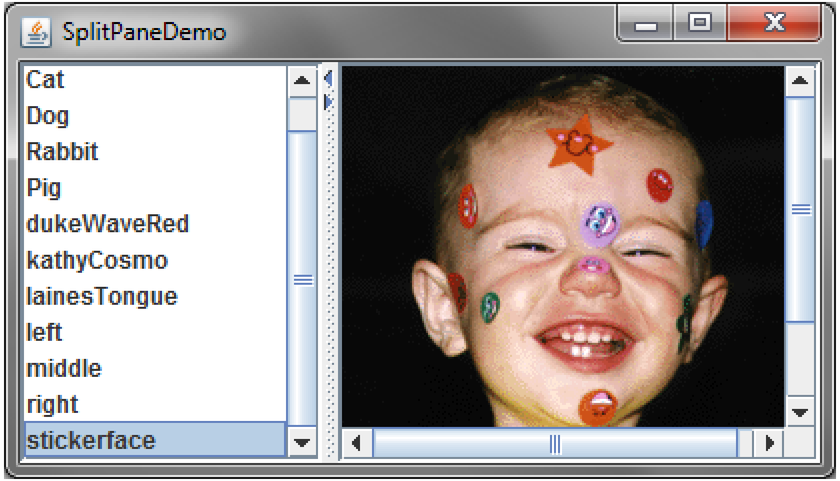


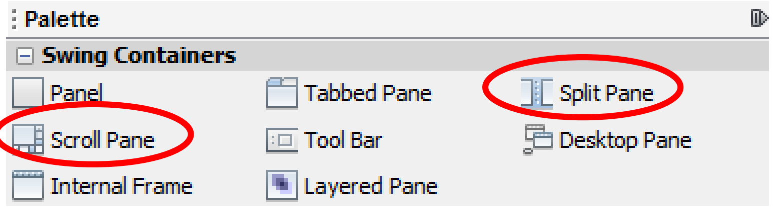
Variable	tab1
⊞ Association	(Association properties)
⊟ Tab	(Tab properties)
background	
disabledIc...	
enabled	? unknown
foreground	
icon	(null)
mnemonic	
mnemoni...	
title	New tab
tooltip	

Read & demo: How to Use Tabbed Panes

SCROLL & SPLIT PANE



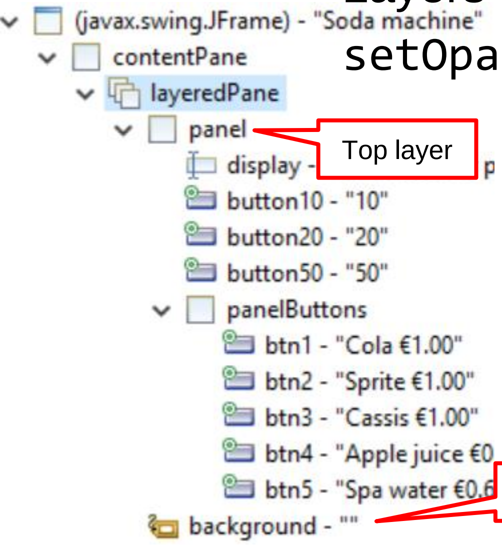




Read & demo: How to Use Split Panes, Scroll Panes

LAYERED PANE

- Build user interface in layers
- It is possible to turn layers on and off
  `setVisible()`
- Layers can be transparent and overlap
  `setOpaque()`

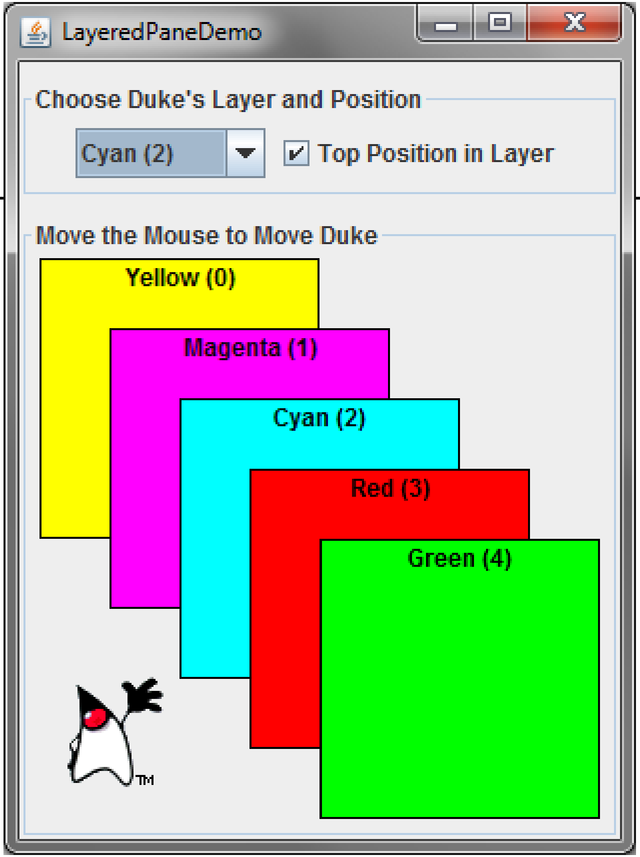


```
(javax.swing.JFrame) - "Soda machine"
  contentPane
    layeredPane
      panel
        display - p
        button10 - "10"
        button20 - "20"
        button50 - "50"
      panelButtons
        btn1 - "Cola €1.00"
        btn2 - "Sprite €1.00"
        btn3 - "Cassis €1.00"
        btn4 - "Apple juice €0
        btn5 - "Spa water €0.6
    background - ""
```

Top layer

Bottom layer

Background of a label is transparent by default. If you want to assign a background color, make it opaque first:
`label.setOpaque(true)`



LayeredPaneDemo

Choose Duke's Layer and Position

Cyan (2) ☑ Top Position in Layer

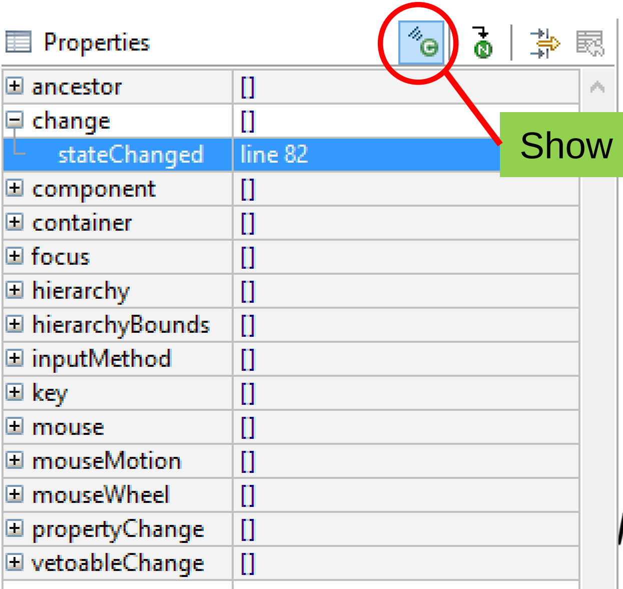
Move the Mouse to Move Duke

Example & demo: Assignment 5b & How to Use Layered Panes

EVENTS: LISTEN TO KEYSTROKES

KEYS PRESSED ON KEYBOARD

- Double-click element: propertyChange
- Change tab: stateChanged



Properties

⊞ ancestor	[]
⊟ change	[]
stateChanged	line 82
⊞ component	[]
⊞ container	[]
⊞ focus	[]
⊞ hierarchy	[]
⊞ hierarchyBounds	[]
⊞ inputMethod	[]
⊞ key	[]
⊞ mouse	[]
⊞ mouseMotion	[]
⊞ mouseWheel	[]
⊞ propertyChange	[]
⊞ vetoableChange	[]

Show Events

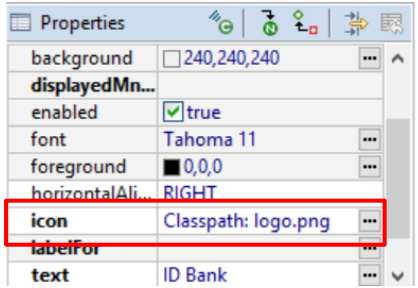
```
GameGUI extends JFrame implements KeyListener {
```

```
@Override
public void keyPressed(KeyEvent e) {
  System.out.println("key=" + e.getKeyCode() );
}
```

Example: **catch-the-ball game**

MEDIA: IMAGES AND SOUNDS

- Images
  - Use as icon*
  - Or draw in a panel**



Properties

background	□ 240,240,240
displayedMn...	
enabled	☑ true
font	Tahoma 11
foreground	■ 0,0,0
horizontalAli...	RIGHT
icon	Classpath: logo.png
labelFor	
text	ID Bank

```
// read image from file:
File file = new File("images/"+filename);
try {
    image = ImageIO.read(file);
} catch (Exception e) {
    System.err.println("Unable to read "+filename);
    return;
}
// draw image:
if (image!=null)
    g.drawImage(image, x, y, panel);
```

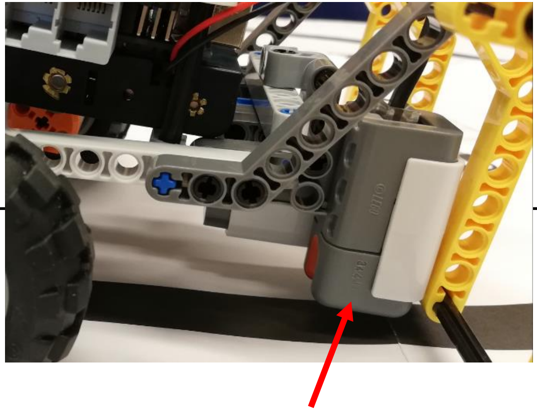
- Sound
  - Use class **PlayClip**, part of example

```
PlayClip sound = new PlayClip();

// check if basket catched something:
if (checkCatched(basket.getX(), basket.getY()))
     sound.play("sound/Ding.wav");
```

Example: **catch-the-ball** game,
* atm-tutorial,
** appendix assignment 2

ARDUINO: LINE FOLLOWER

USING ONE DOWN-FACING NXT LIGHT SENSOR



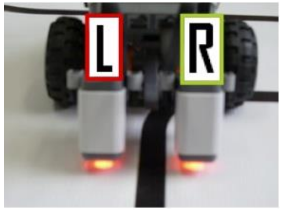
- Start with EVSHield-example **nxt_light_reflected** to measure light values
- Add to setup():

```
// start driving
evshield.bank_a.motorRunUnlimited(SH_Motor_Both, SH_Direction_Reverse, 15);
```

- Add to loop():

```
if (light<600) { // off track
   evshield.bank_a.motorStop(SH_Motor_1, SH_Next_Action_Float );
}
else { // on track
   evshield.bank_a.motorRunUnlimited(SH_Motor_Both, SH_Direction_Reverse, 15);
}
delay(50);
```

For higher precision, use 2 sensors, each next to the line



Linefollower.ino

DECLARATION AND INITIALIZATION

Need in assignment

```
// find path from 0,0 to 40,40
List<ExampleNode> path = myMap.findPath(0, 0, 40, 40);
```

- Declare path as a class-variable
- Initialize it in a method
- How?

```
public class MyClass {
   List<ExampleNode> path;
```

```
public void someMethod {
   path = myMap.findPath(0, 0, 40, 40);
}
```

PATHFINDING… CLEAR THE PATH

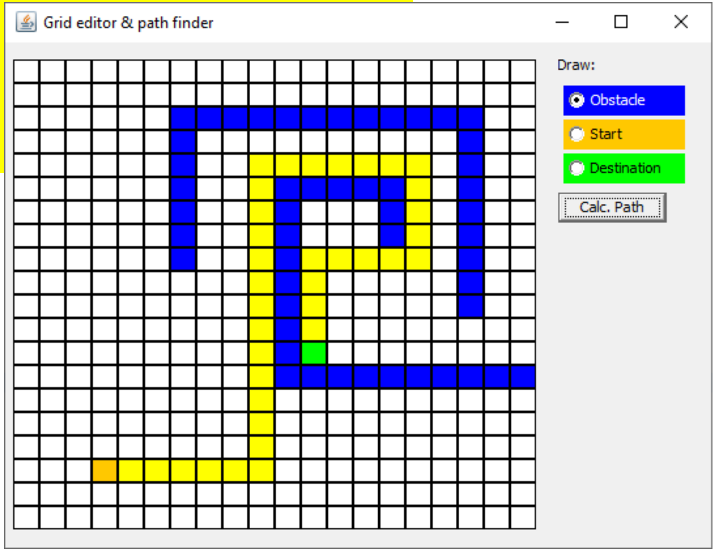
Need in assignment

Is there a path?

`size()-1`: Loop over all tiles in the path, except the last (to prevent clearing the destination tile)

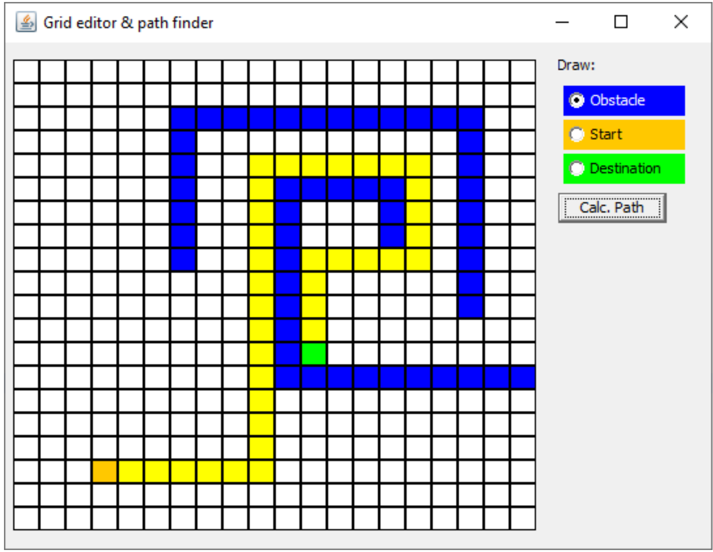
```
if (path != null) { // remove previous path
  for (int i = 0; i < path.size()-1; i++) {
    PixelPanel p = pixelPanels[path.get(i).getxPosition()][path.get(i).getyPosition()];
    if (p.getBackgroundColor()==Color.yellow) { // if shown as path
           p.setBackgroundColor(Color.white);
    }
  }
}
```

Re-color only tiles which are yellow



ASSIGNMENT #6

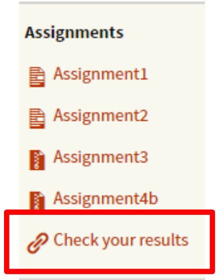
- Create an interactive map editor with path-finding capabilities
- Java assignment



Checkpoint

- Next assignment will further extend this, so to do assignment 7, you need 6

Check assignments results:



Assignments

- Assignment1
- Assignment2
- Assignment3
- Assignment4b
- Check your results

No lecture next week,
next lecture Tuesday June 4th

UNIVERSITY OF TWENTE.

Slides, assignments etc @ vanslooten.com/appdev