

# Programmiermethodik

## Übung 7

Sommersemester 2010  
Fachgebiet Software Engineering

Andreas Scharf  
andreas.scharf@cs.uni-kassel.de

# Agenda

- **Vorstellung HA6**
- **Entwicklung von grafischen Oberflächen**
  - Mock-Ups
  - GUI Builder
  - Swing
  - SWT
  - Demo
- **Praktische Übung: Erstellen eines Mancala Login Screens**

# Vorstellung HA6 I

- **Vorbereitung**
  - Mancala Implementierung (eigene oder vom PM Blog)
  - Neue Version des Fujaba4Eclipse Plug-Ins herunterladen
- **Storyboards**
  - Storyboards zu 8 vorgegebenen Szenarien erstellen
  - Implementierung von `Pit::moveStones()` und `Mancala::checkEnd()` anpassen
- **Abgegeben werden soll exportiertes Eclipse Projekt. Inhalt:**
  - Fujaba Projekt mit Storyboards
  - Implementierung des Modells und der Methoden
  - Aus den Storyboards generierte Tests

# Vorstellung HA6 II

- **Beispiel**

*Steine bewegen, keine Umrundung*

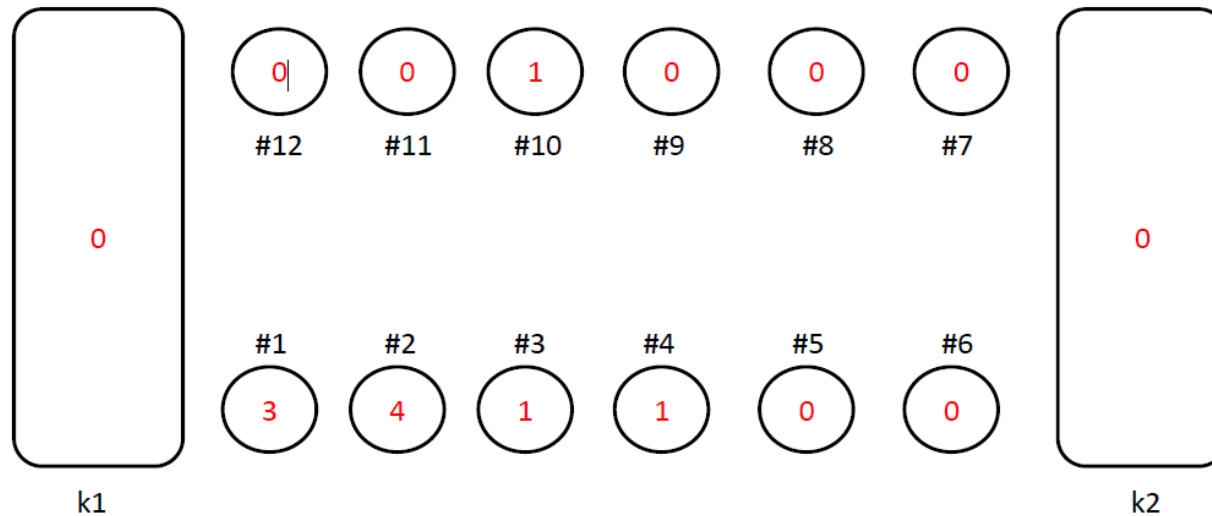


Abbildung 1: Steine bewegen, keine Umrundung

Spieler mit den *unteren* Pits ist an der Reihe. Aufruf von `Pit::moveStones()` auf Pit #1.

# Entwicklung von grafischen Oberflächen – Mock-Ups

- Bevor man anfängt zu coden: „Mock-Ups“
- In der Realität: Designer != Entwickler
- Erstellung von Mock-Ups kostet deutlich weniger Zeit
- Erleichtert die Implementierung, weil man schon weiß was herauskommen soll

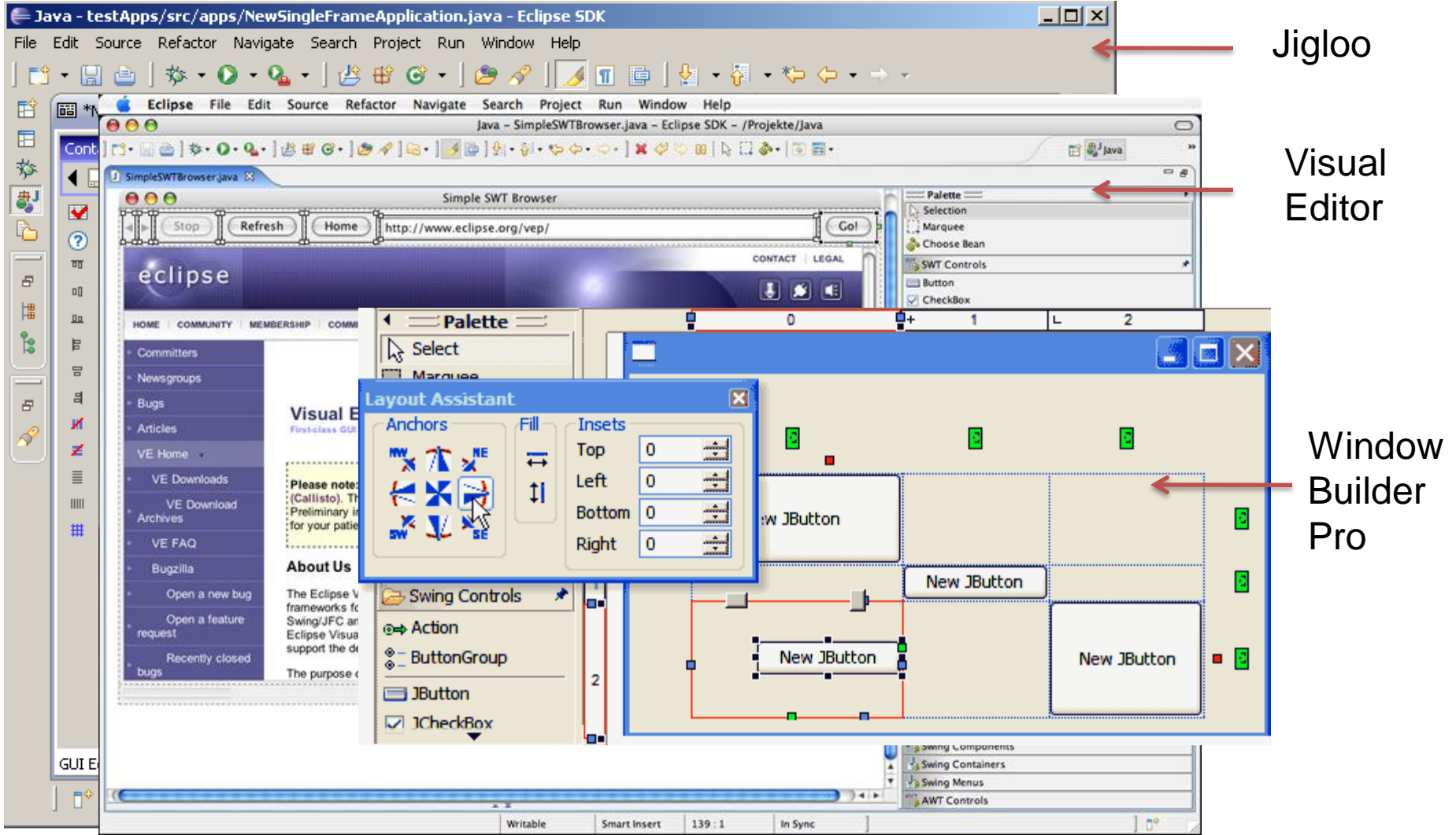


Quelle: <http://media.konigi.com/notebook/iphone-mockup.jpg>

# Entwicklung von grafischen Oberflächen – GUI Builder I

- **Problem bei grafischen Oberflächen: Aufwendig in der Entwicklung**
- **Abhilfe sollen GUI-Builder schaffen ( WYSIWYG Prinzip)**
- **Kommerzielle Tools (SWT/Swing):**
  - Jigloo
  - Window Builder Pro
  - ...
- **Freie Tools (SWT/Swing):**
  - Visual Editor (Eclipse 3.2, veraltet, wird nicht mehr gepflegt)
- **Generieren (meist hässlichen) SWT/Swing Code**

# Entwicklung von grafischen Oberflächen – GUI Builder II



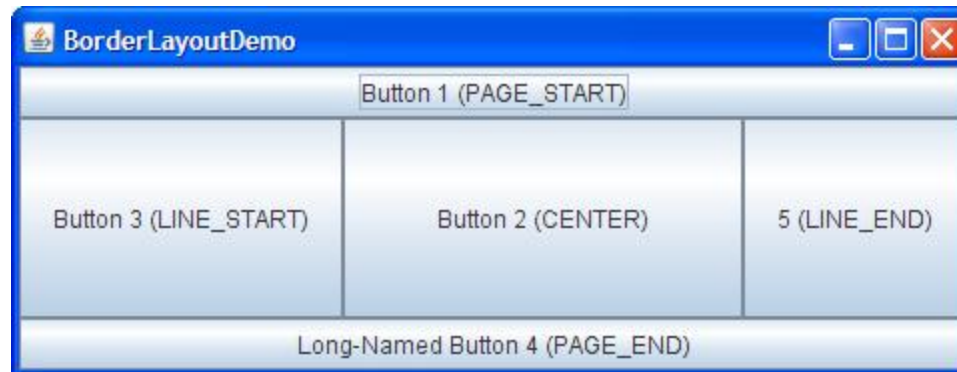
# Entwicklung von grafischen Oberflächen – Swing I

- **Swing ist seit Java 1.2 Bestandteil der Java Runtime**
- **Baut auf dem Abstract Window Toolkit (AWT) auf**
- **Swing Komponenten werden direkt von Java gerändert**
  - Funktioniert auf allen Plattformen
  - Sieht überall gleich aus
  - NICHT nativ
- **Verschiedene Look&Feels**
  - Windows
  - Linux
  - Mac
  - ...



# Entwicklung von grafischen Oberflächen – Swing II

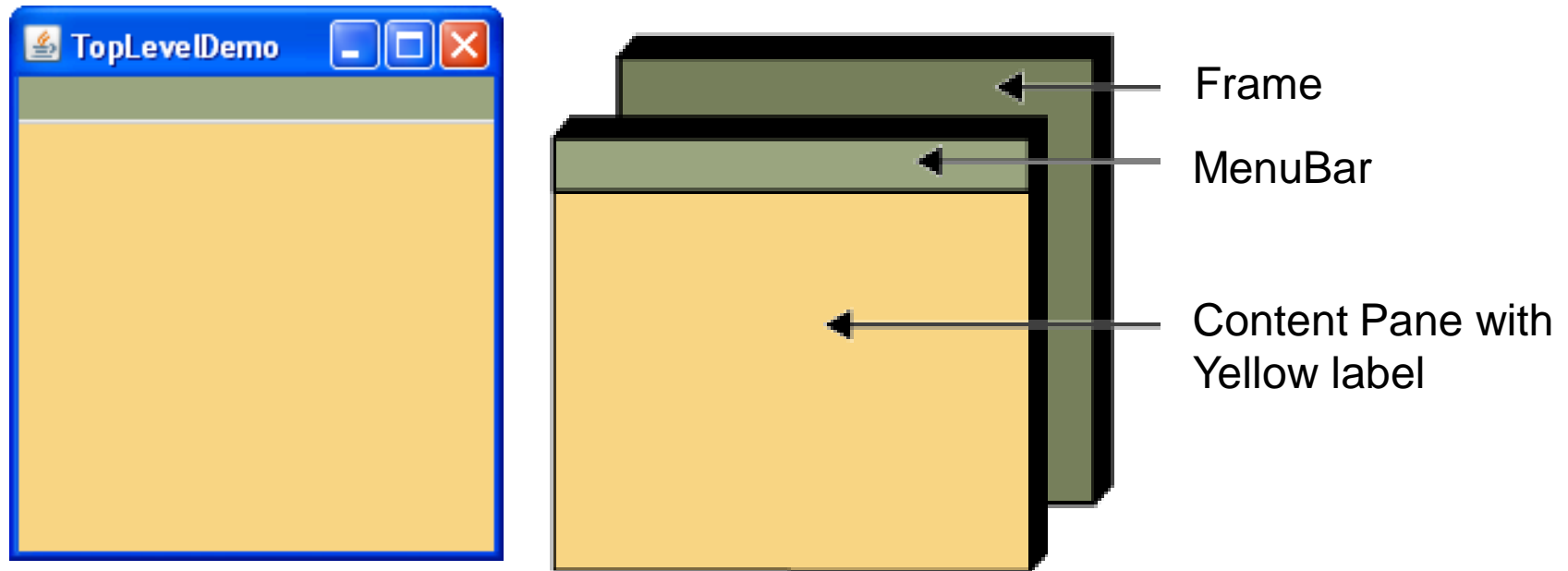
- Referenz: <http://java.sun.com/docs/books/tutorial/uiswing/>
- Einstieg: JFrame (Top-Level Container)
  - Repräsentiert ein Fenster
  - Kann Inhalt aufnehmen
  - Standard Layoutmanager: BorderLayout



- `jFrame.getContentPane()` liefert Container

# Entwicklung von grafischen Oberflächen – Swing III

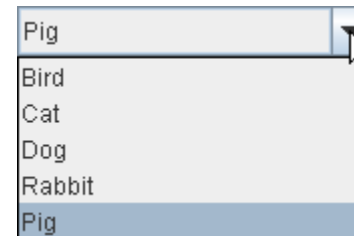
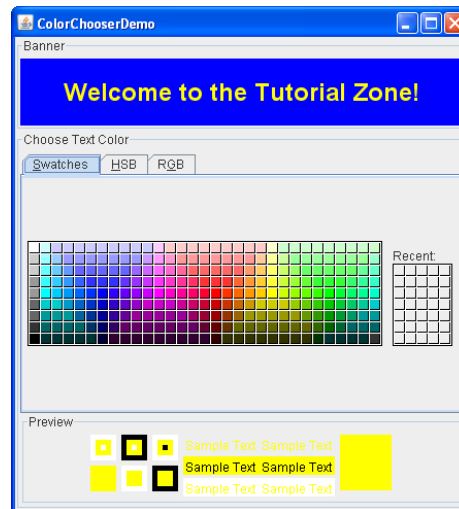
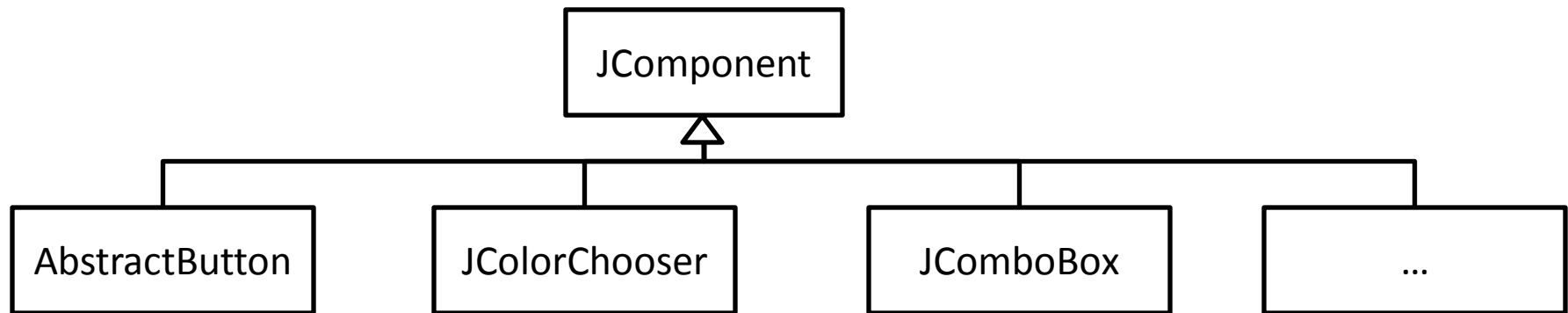
- Aufbau eines JFrames



- `jFrame.getContentPane().add(yellowLabel, BorderLayout.CENTER)`

# Entwicklung von grafischen Oberflächen – Swing IV

- Alles (bis auf die Top-Level Container) leitet von JComponent ab

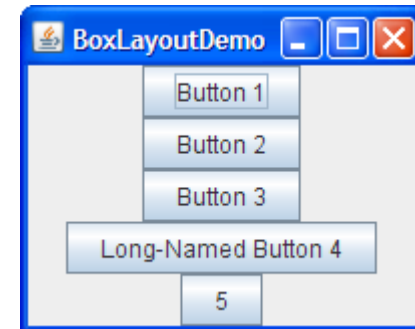
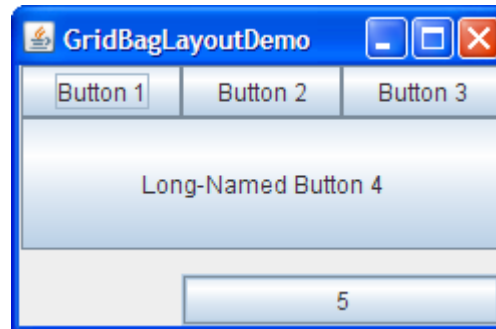


# Entwicklung von grafischen Oberflächen – Swing V

- **JComponent stellt alle grundlegenden Features zur Verfügung**
  - Tooltips
  - Rahmen
  - Look&Feel
  - Layout
  - DnD
  - ...

# Entwicklung von grafischen Oberflächen – Swing VI

- **Absolute Positionierung vs. LayoutManager**
- **LayoutManager**
  - Positionierung
  - Verhalten bei Größenänderung des Fensters
- **Immer einen LayoutManager benutzen!**
- **Beispiele**
  - BorderLayout
  - BoxLayout
  - GridBagLayout



# Entwicklung von grafischen Oberflächen – Swing VI

- **Verwendung von LayoutManagern**

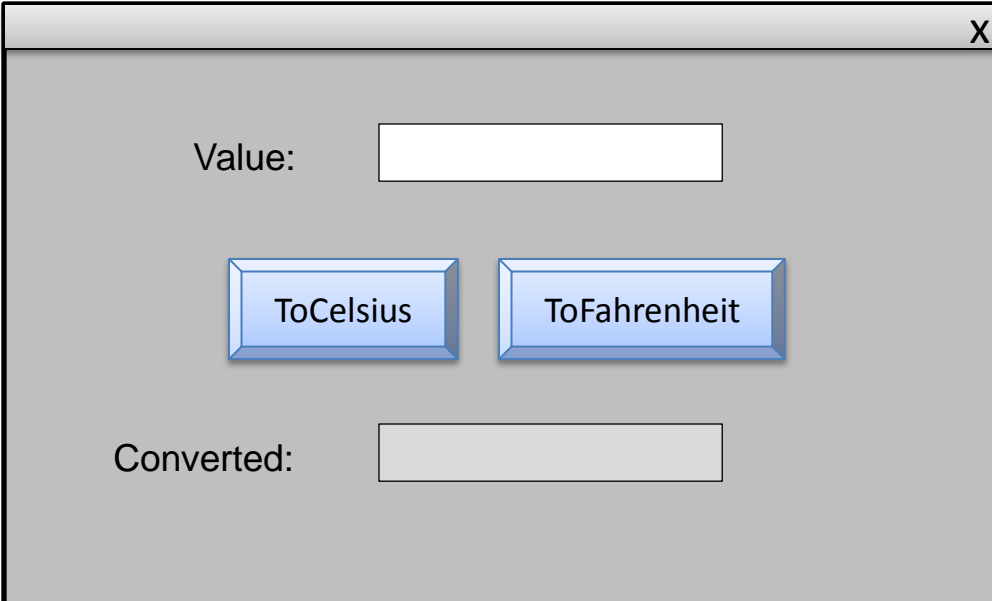
- Layout wählen (<http://java.sun.com/docs/books/tutorial/uiswing/layout/visual.html>)
  - `BoxLayout boxLayout = new BoxLayout(jComponent, BoxLayout.PAGE_AXIS)`
- LayoutManager auf Container anwenden
  - `jComponent.setLayout (boxLayout)`
- Kinder hinzufügen
  - `jComponent.add(new Label(`

# Entwicklung von grafischen Oberflächen – SWT

- **Standard Widget Toolkit (SWT)** <http://www.eclipse.org/swt>
- **Wurde 2001 von IBM für Eclipse entwickelt**
- **NICHT Bestandteil der Java Runtime**
  - Bibliothek (inkl. nativen Bestandteilen) müssen mit ausgeliefert werden
- **Abstrahiert von nativer grafischer Benutzeroberfläche**
  - Einmal coden, überall nativ laufen lassen (theoretisch)
- **(Unter Windows) deutlich schneller als Swing**
- **Im Gegensatz zu Swing „schwergewichtig“, wegen der Verwendung von nativen Komponenten (statt sie selbst zu zeichnen)**

# Entwicklung von grafischen Oberflächen

- **Demo: „Celsius – Fahrenheit converter“**
- **Umrechnungsformel:**
  - Celsius in Fahrenheit =  $(( T_{\text{Celsius}} \times 9 ) / 5 ) + 32$
  - Fahrenheit in Celsius =  $( T_{\text{Fahrenheit}} - 32 ) \times 5 / 9$
- **Mock-Up**



The mock-up shows a window with a title bar containing an 'X' button. The window has a light gray background. At the top, the text 'Value:' is followed by a white rectangular input field. Below this, there are two blue buttons with white text: 'ToCelsius' on the left and 'ToFahrenheit' on the right. At the bottom, the text 'Converted:' is followed by another white rectangular input field.



# Praktische Übung

- Entwicklung eines Mancala Login Screens
- Mock-Up

X

## Mancala

Spieler 1:

Spieler 1:

Los

Beenden

- Zeigt das Ergebnis einem Betreuer

**Ende**

**Schönes WE!**