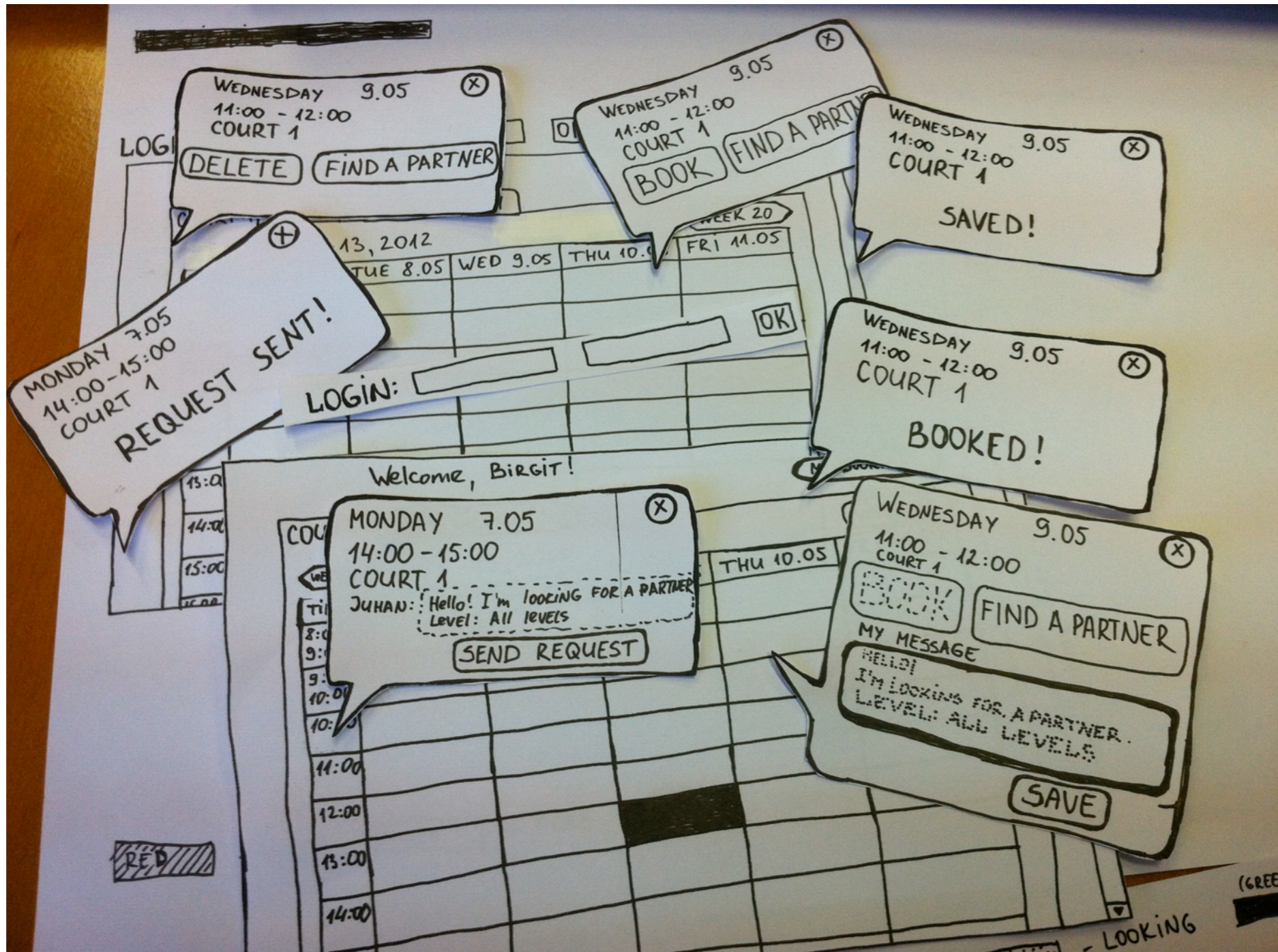


GUI Entwicklung

Softwarepraktikum 2016

Nikolas Havrikov

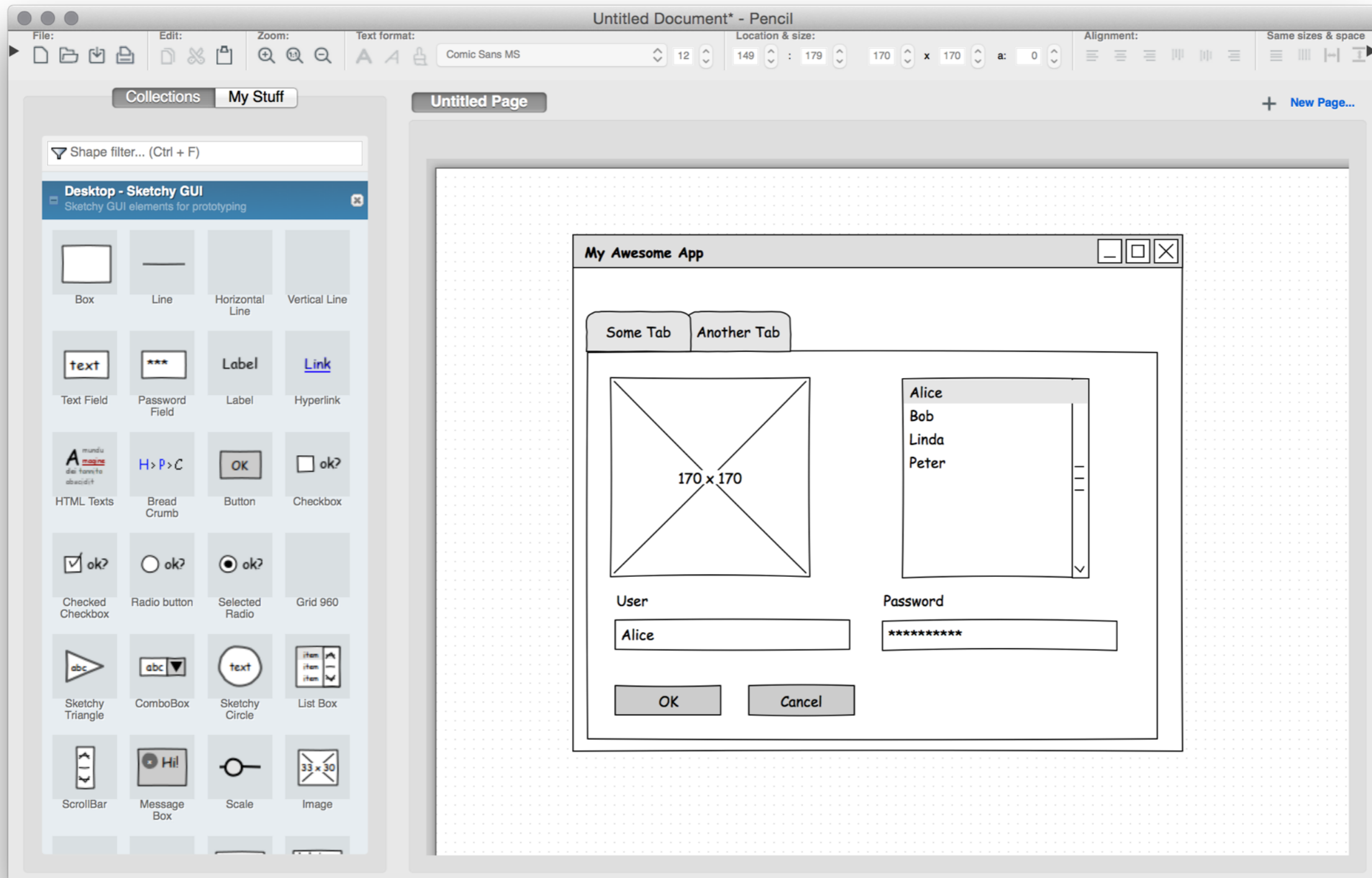
Paper Prototyping



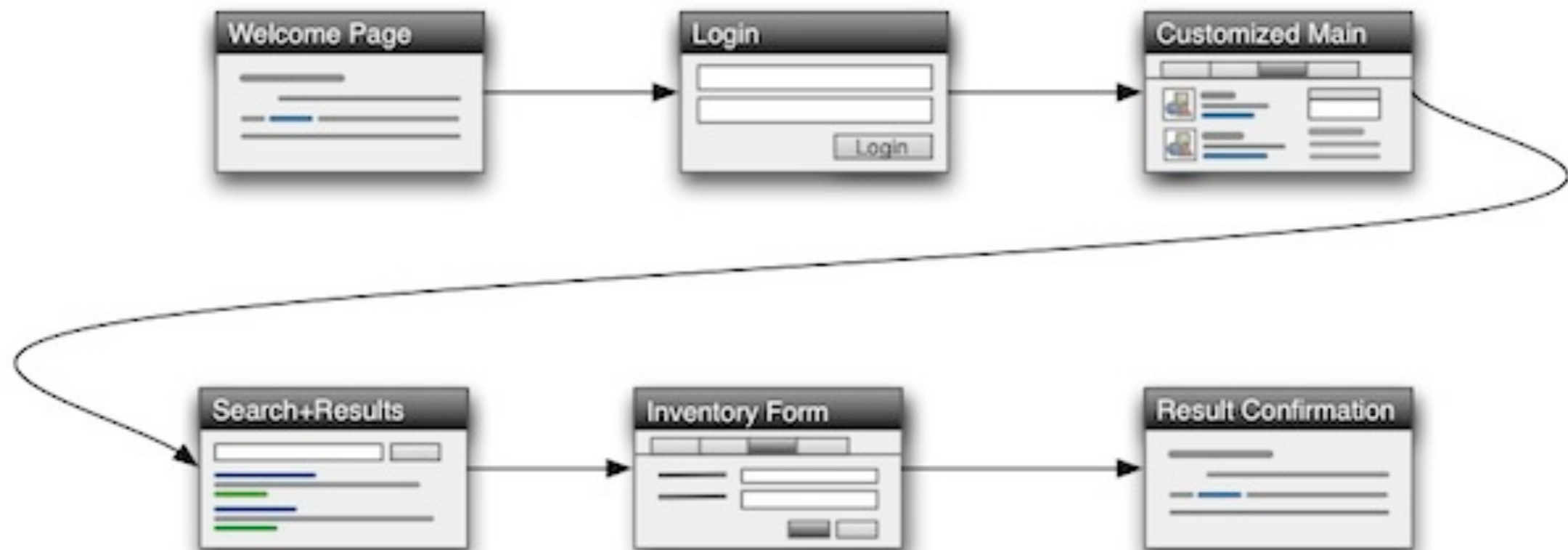
Vorteile

- Schnelles Erstellen eines Mockups ohne dazu programmieren zu müssen.
- Entdeckt frühzeitig Probleme im Design
- Ermöglicht das Einholen von User-Feedback vor dem Implementieren.
- Ermöglicht Zusammenarbeit von Teams aus mehreren Disziplinen.

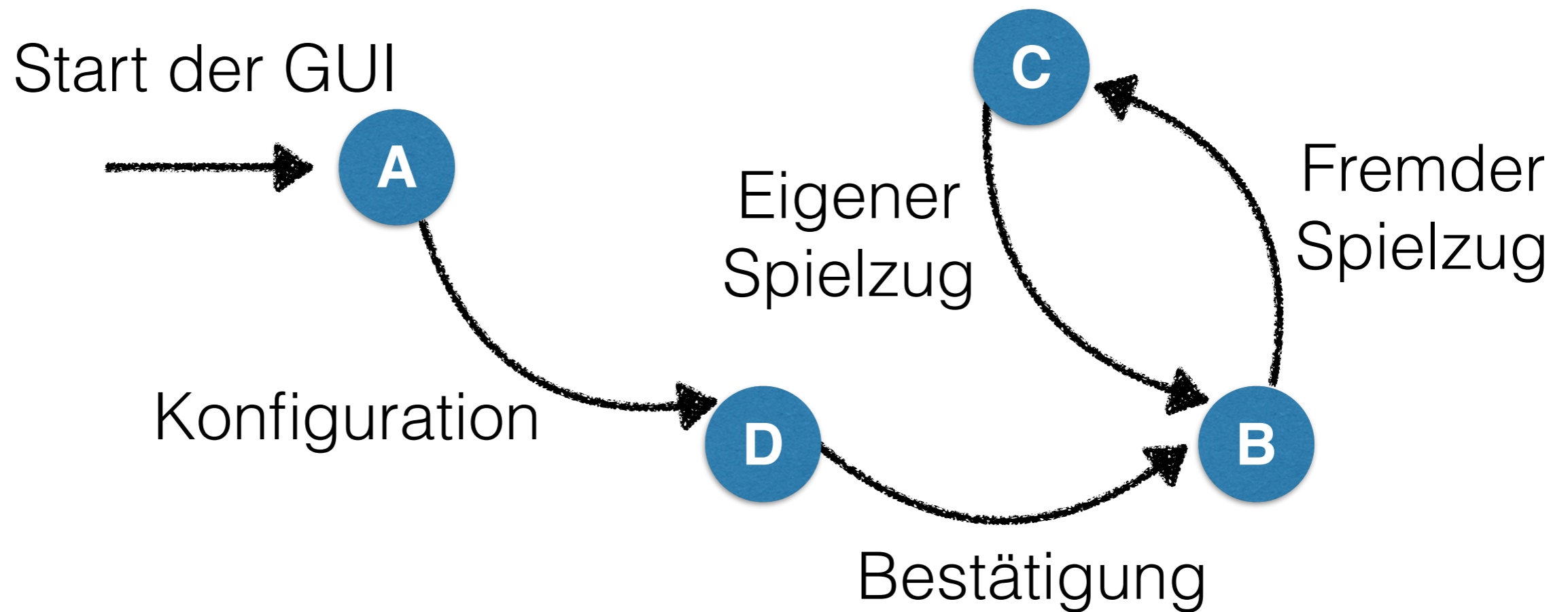
Wireframing



Storyboards



Status Diagramme



Status Beschreibungen

- A. **Uninitialisiert:** Fenster: *Main*. Panels: *Main:Control*. Alle Buttons gesperrt, *Menu>File>(Connect, Quit)* verfügbar.
- B. **Warten auf Mitspieler:** Command Buttons nicht nutzbar, Spielfeld präsent in Panel *Main:Map*. Spielstände in Panel *Main:Info* angezeigt.
- C. **Eigener Spielzug:** ...
- D. **Warten auf Konfiguration:** ...

Status Übergänge

Start der GUI: Erstellung des Hauptfensters (*Main*) mit Panel *Main:Control*.

Konfiguration: *Menu>File>Connect* öffnet Dialog: *Config*

Bestätigung: Dialog: *Config:Connect-Button* + gültige Einstellungen schließt Dialog und erstellen Panel *Main:Main* und *Main:Info*.

...

Interaktionen

Backend zu GUI

Empfange Moved Event: Animiere den Positionsübergang des Characters in der GUI.

Empfange Winner Event: Setze GUI in End-Status und veranlasse Anzeigen des Gewinnerteams.

Interaktionen

GUI zu Backend

Main:Control:Connect#click

Erstelle neue Verbindung zum
Server

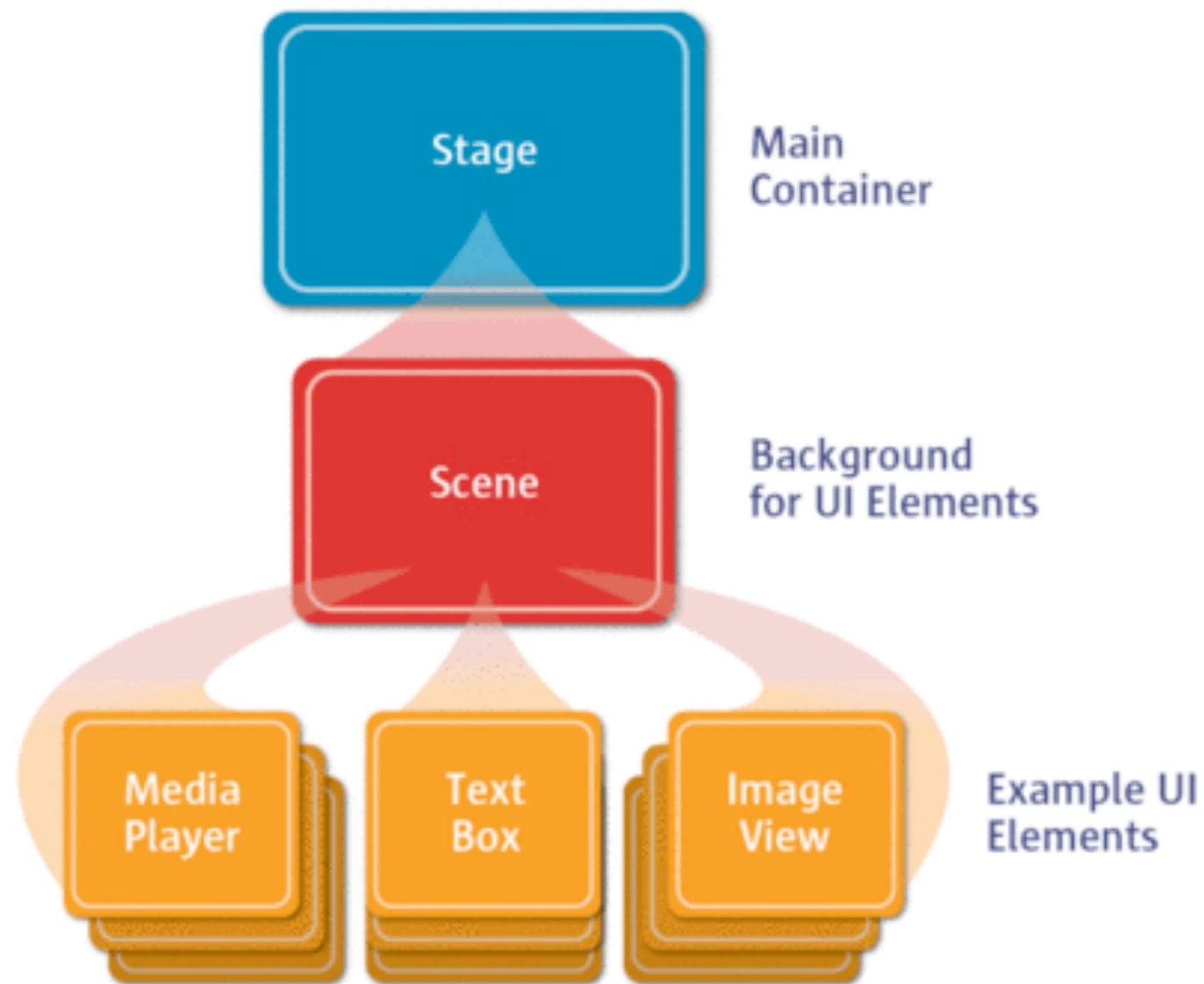
Main:Main:Move#click

Sende einen Move Command
zum Server

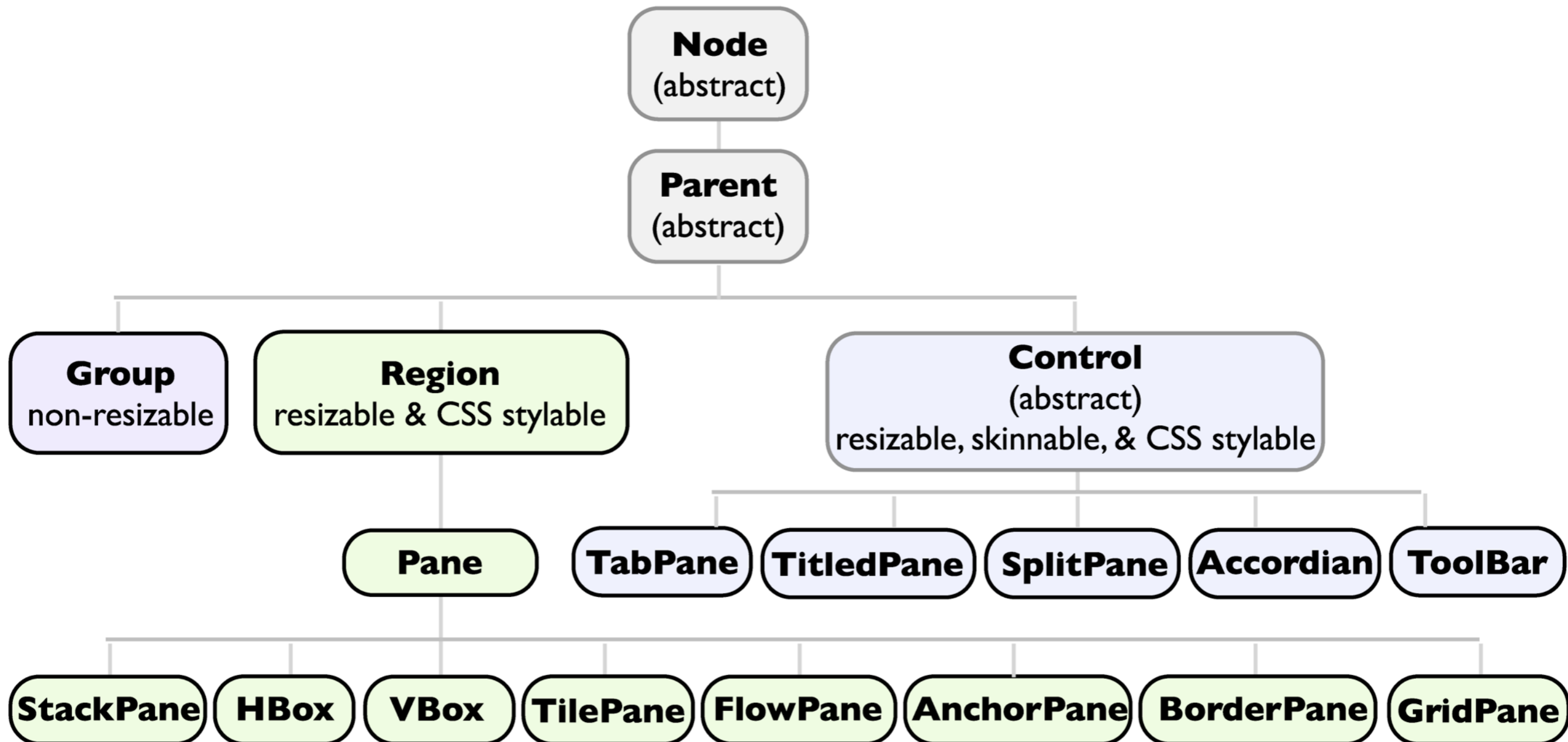


- Deklaratives GUI Framework
- Trennung von Code und Design (FXML)
- Nur enthalten im Oracle JDK!
- Eclipse Integration: [e\(fx\)clipse](#)
- IntelliJ IDEA Integration: JavaFX Plugin

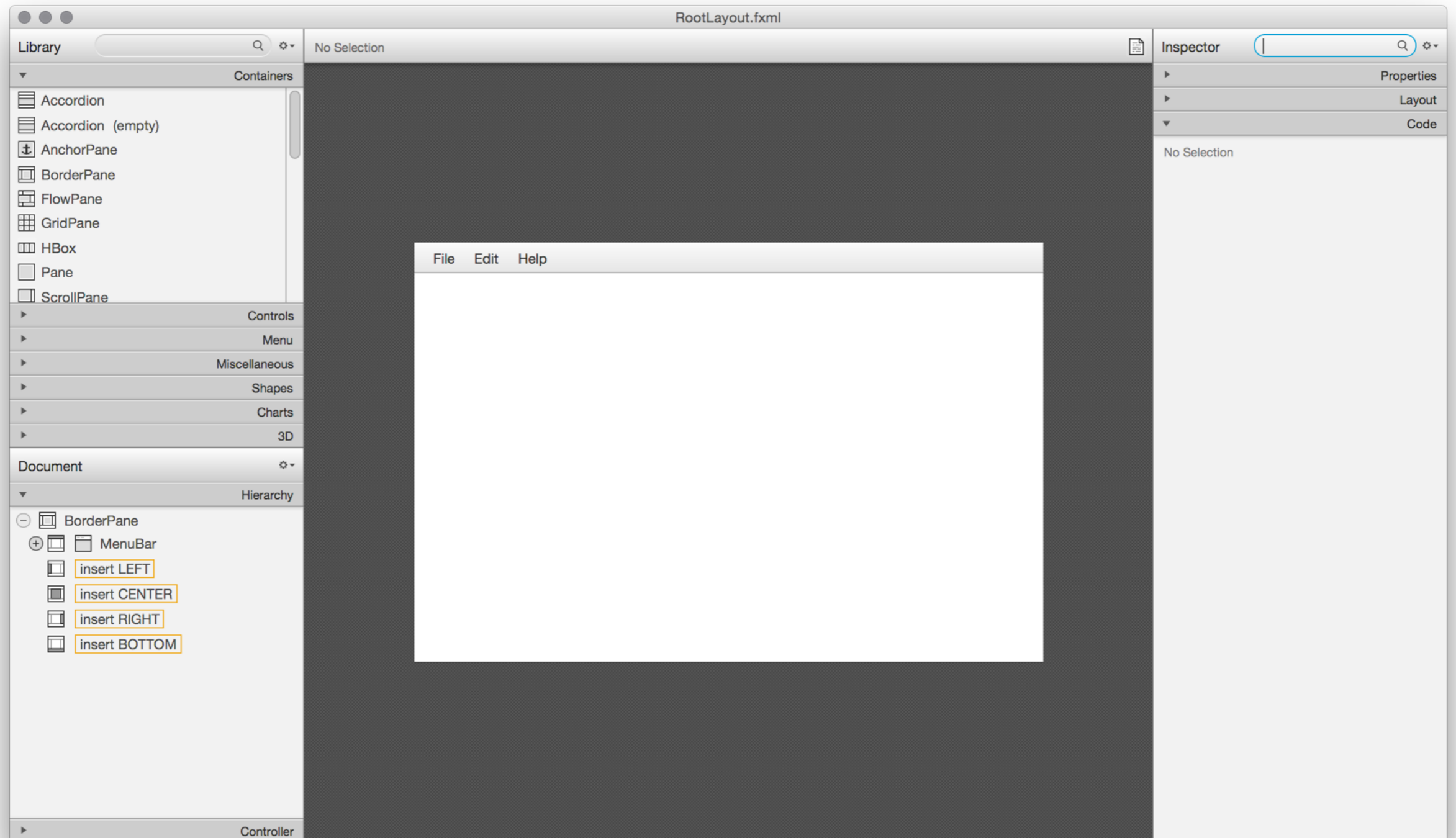
JavaFX UI Elemente



JavaFX UI Elemente



Demo: Scene Builder



Demo: FXML Verknüpfen

- `@FXML` Annotation für Felder und Methoden
- Automatische Initialisierung
- Manuelle Initialisierung mit `initialize()`

Demo: Selectors

Finden von GUI Komponenten:

- `Button`: Komponenten vom Typ `Button`
- `.myClass`: Komponenten mit `styleClass` `myClass`
- `#myId`: Komponenten mit ID `myId`

Demo: CSS Styling

- Ändern der Darstellung von Nodes
- Auswählen von Nodes mit Selectors
- Verfügbare Eigenschaften: <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/doc-files/cssref.html>

Nebenläufigkeit

- JavaFX ist **nicht Thread-Safe!**
- Lange Berechnungen und I/O können UI blockieren
- UI Änderungen aus anderen Threads mit `Animations` oder `Platform.runLater()`

Demo: Images und Shapes

- Geometrische Formen in der API
- Stylebar mit CSS
- Verknüpfung mit Events
- Bilder in der API

Nützliche Links

- <http://code.makery.ch/library/javafx-8-tutorial/>
- http://docs.oracle.com/javase/8/javafx/get-started-tutorial/get_start_apps.htm