Grafik - wozu?

- GUI – Grafische Benutzungsschnittstellen
  - Gehört zum Standardumfang des JDK
  - 2 Varianten: AWT und Swing
  - Konzeptuell sind beide Varianten gleich
  - Heute: Beispiel für AWT
- Zeichnen, Bilder, 3D-Modelle

AWT (Abstract Window Toolkit)

- Basierend auf den Elementen des Fenstersystems
- *Schnittmenge* der Fenstersysteme
- Alt
- Vergleichsweise schnell
- Läuft (halbwegs) problemlos in Internetbrowsern

Beispiele aus dem AWT-Tutorial.

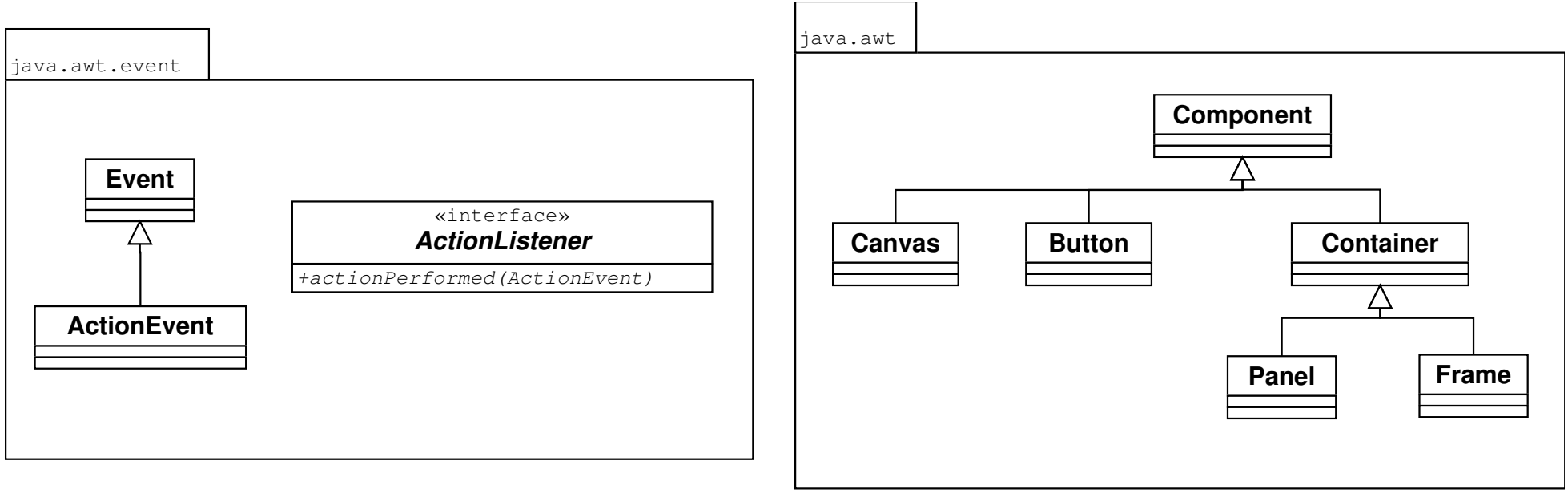
Swing

- Vollständig in Java implementiert
- Viele Features – “jeder mögliche grafische Schnickschnack”
- Neu
- Langsam
- Läuft nicht selbstverständlich in Internetbrowsern

Beispiele aus dem JDK1.4: SwingSet2.jar

AWT-Grundkonzept

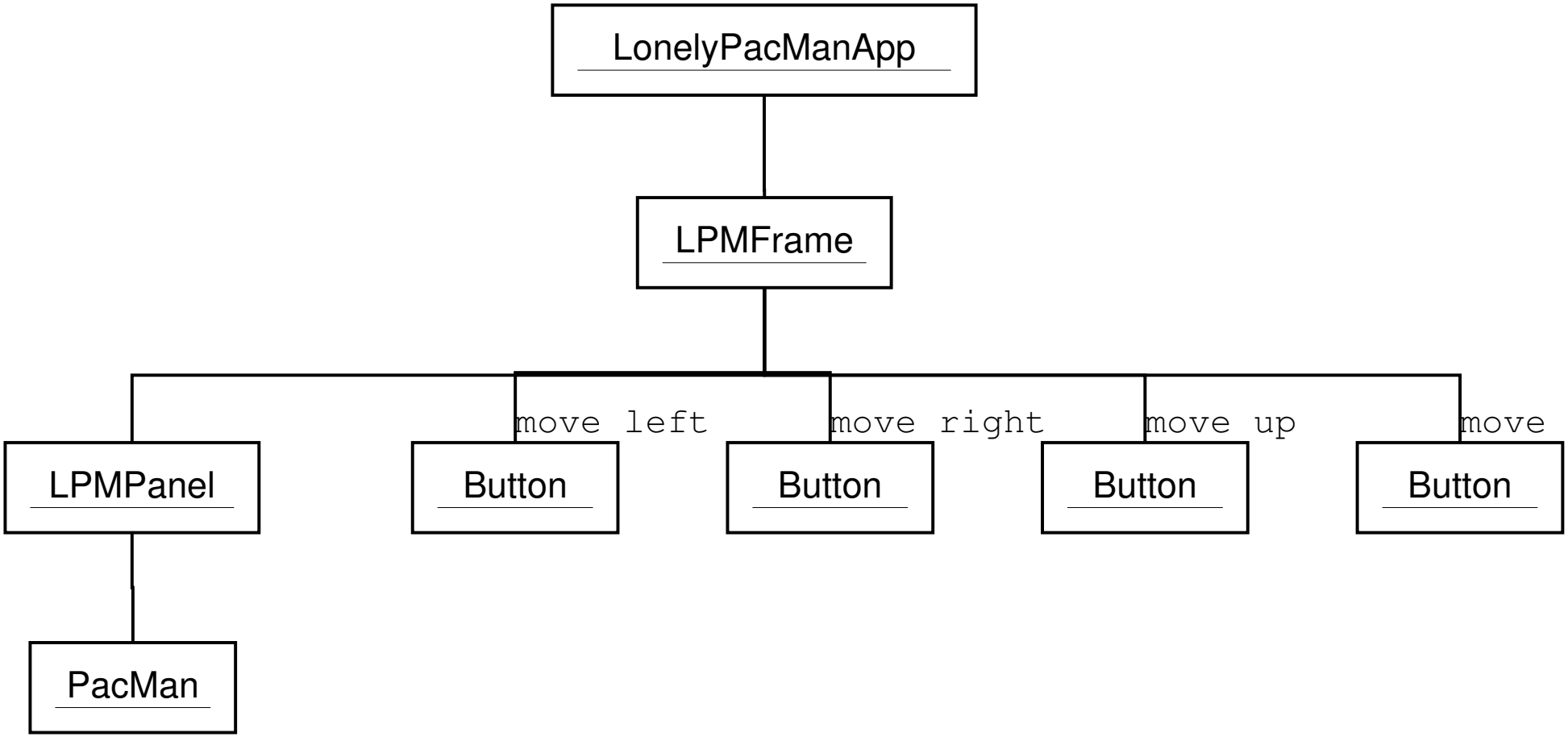
- Hierarchische Klassenstruktur für grafische Elemente
- Klassen werden direkt verwendet und/oder spezialisiert
- Pakete `java.awt` und `java.awt.event`:
  - `java.awt`: Grafische Komponenten
  - `java.awt.event`: Ereigniskonzept



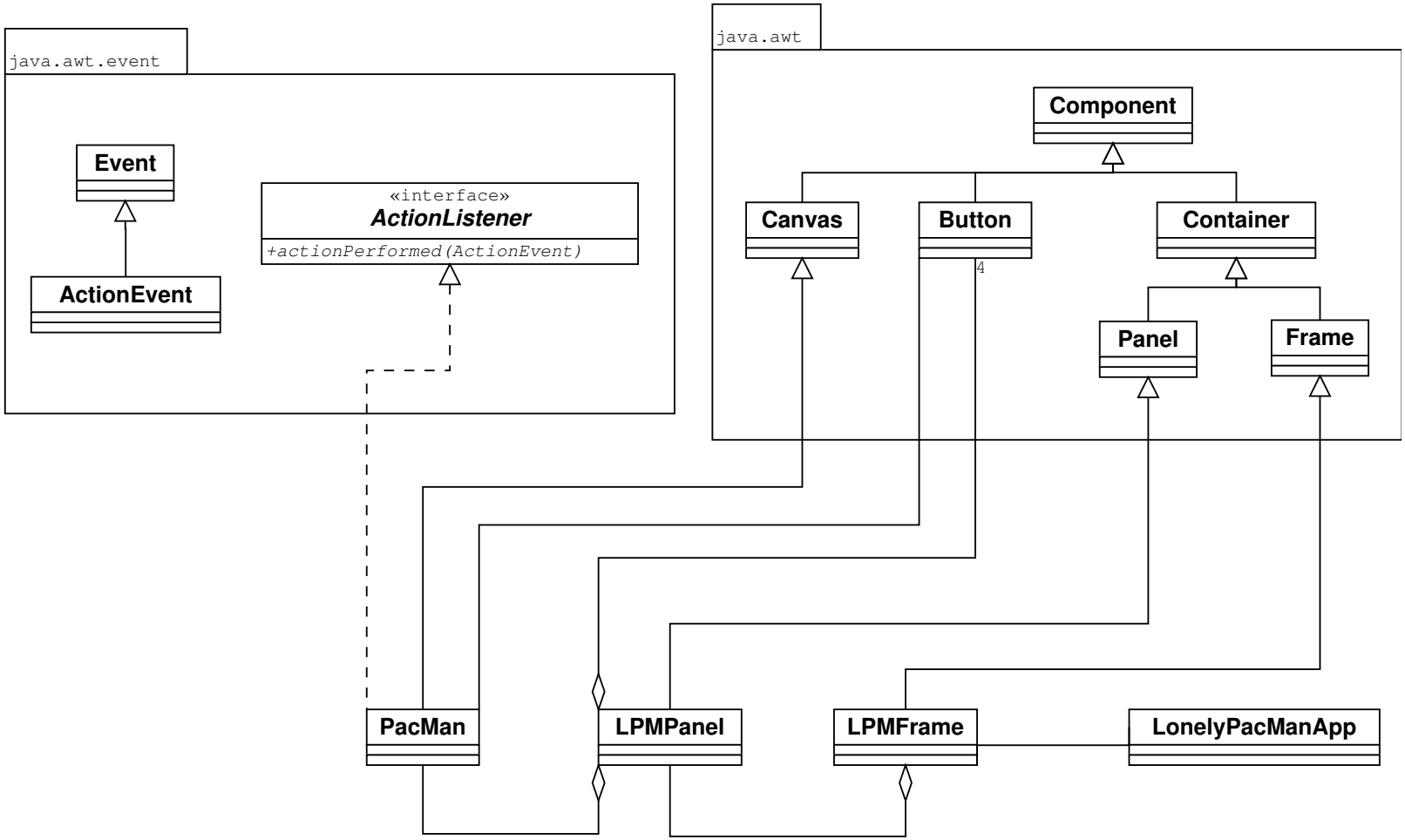
Vordefinierte AWT-Klassen.

Beispiel: Einsamer PacMan

Der einsame PacMan ohne seine Freunde.



Objekte für den einsamen PacMan.



Klassen für den einsamen PacMan.

Schritt 1: Grundgerüst

- Applikationsrahmen anlegen
- Hauptfenster spezialisieren
- Buttons, Panel einfügen

Java 1.1-API-Dokumentation

Schritt 2: Layout des Hauptfensters

- Layout-Unterstützung durch `LayoutManager`
- Hier: vordefiniertes `BorderLayout`
- Beachte: ist Standard für `Frame`-Objekte

Schritt 3: PacMan entwerfen

- Panel spezialisieren
- Hier: *kein* Layout, das macht PacMan selbst
- PacMan aus `Canvas` spezialisieren
- `paint()` redefinieren
- PacMan einfügen

Schritt 4: Interaktion realisieren

- Ereignisbasiert: PacMan lauscht auf Button-Ereignisse
- PacMan wird ein `ActionListener`
- PacMan als Lauscher registrieren

Weitere Grafik-APIs

Im JDK integriert:

- Java2D: Zeichnen von Punkten, Linien, Rechtecken etc.

Beispiel aus dem JDK1.4: Java2Demo.jar

Optionale APIs:

- JAI (Java Advanced Imaging): Bildbearbeitung

- Java3D: Darstellung/Interaktion von 3D-Modellen

Java3D

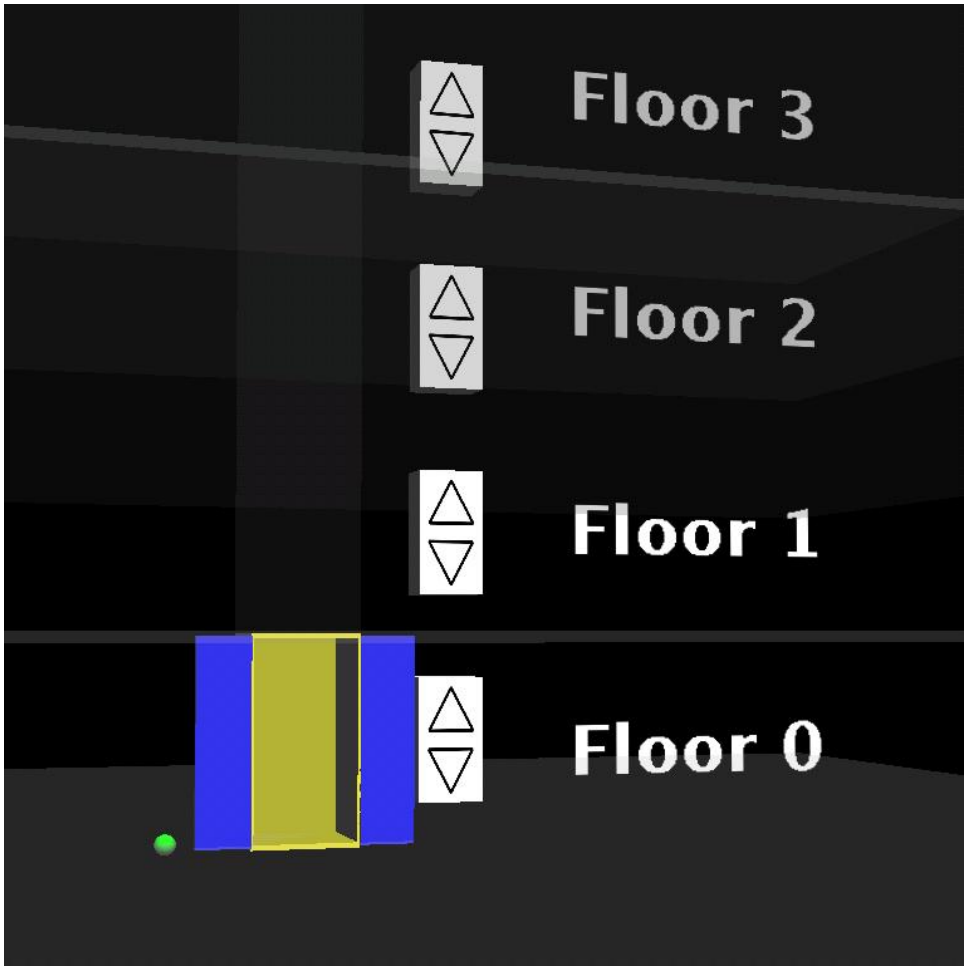
- Interaktive 3D-Modelle bzw. “virtuelle Welten”
- Grundgedanke ähnlich zu AWT/Swing:
  Klassenhierarchie für Komponenten und Interaktionen
- Unterschied:
  Interne Datenstruktur ist eine dreidimensionale virtuelle Welt
- Anbindung an GUI:
  Spezielle `Canvas`-Objekte präsentieren eine *Ansicht der virtuellen Welt*

Modellierung der virtuellen Welt durch *hiearchischen Szenengraph*:

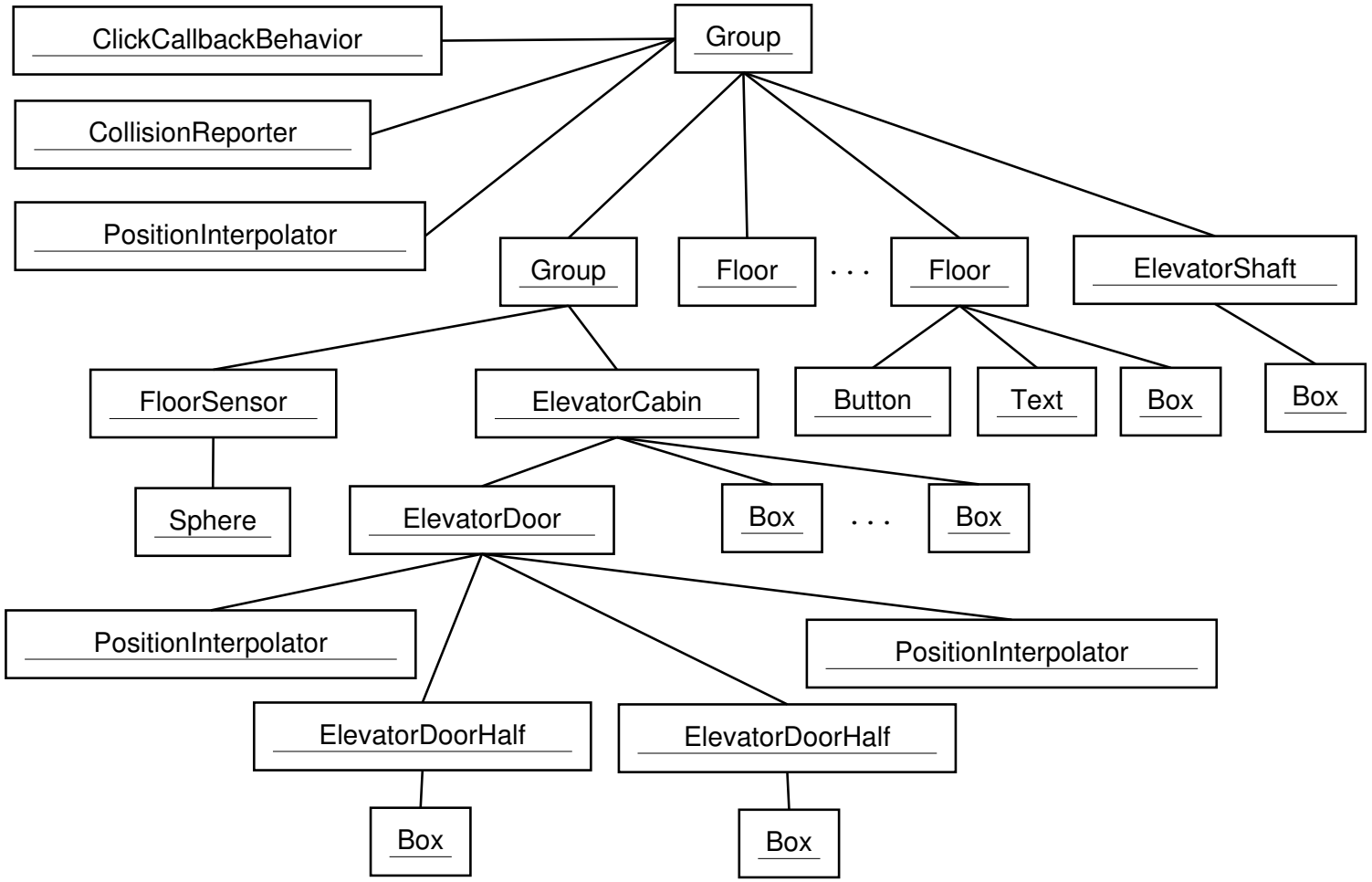
- Geometrie: Primitiven wie Würfel, Kugel sowie Kompositionen daraus
- Interaktive Elemente: Sensoren
- Aktive Elemente: Animationen, Abläufe

Alle Komponenten stehen als vordefinierte Java-Klassen zur Verfügung.

Beispiel: Fahrstuhl



Fahrstuhlapplikation



Szenengraph der Fahrstuhlapplikation

Ende.