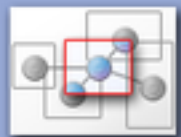


Piccolo2D @ JUGM

ein Referat von Marcus Rohrmoser,
dem Kopf hinter aladin-software.mobi.



Piccolo2D
Structured 2D Graphics Framework

JUGM

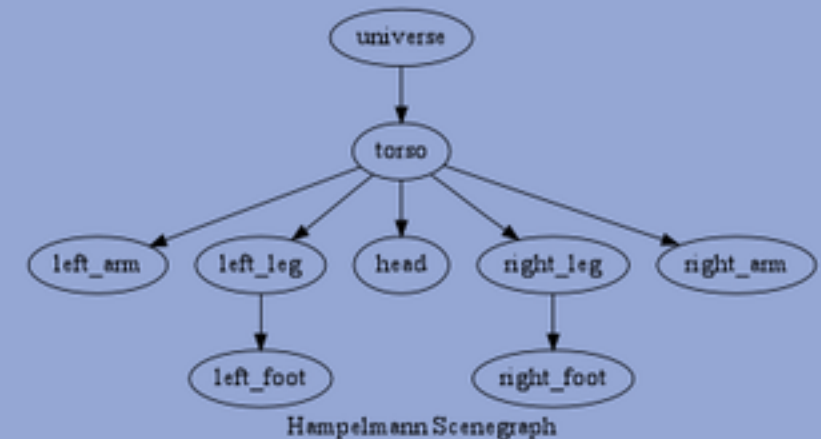
Was ist Piccolo2D?

- 2D Scenegraph API für Java und .Net,
- Spezialitäten:
 - Zooming,
 - Animation,
 - Mehransichtigkeit – z.B. gleichzeitige Detailansicht *und* Vogelperspektive,
- sehr kompakt – wenige Klassen, jar ~80 KB.



Exkurs: Scenegraph

- Datenmodell von Zeichnungsbausteinen („universe“)
- meist als **Baum** repräsentiert,
- reine Lehre:
 - Zeichnen der Elemente per **Depth-First Traversierung**
 - Knoten gruppieren und manipulieren (z.B. **Koordinaten Trafo**),
 - Blätter enthalten Graphikelemente (Kurven, Bitmaps, etc.),



Wobei hilft mir Piccolo2D?

- zur Laufzeit synthetisierte (komplexe) 2D Graphiken,
- veränderliche Bildinhalte,
- veränderliche Bildausschnitte und Effekte.
- Bildelemente finden (z.B. mit der Maus klicken) aka. „Object Picking“

Wie benutze ich Piccolo2D?

- jar einbinden (ideal per Maven Dependency)
- PCanvas in eine Komponente (JFrame) einfügen
- Scenegraph aufbauen und in PCanvas einhängen
- Controller als Event Handler in PCanvas einhängen

Bsp: Hampelmann Treiber

```
public static void main(final String[] args) {  
    final JFrame frame = new JFrame();  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
    final PCanvas canvas = new PCanvas();  
    frame.add(canvas);  
    // Scenegraph:  
    final Model model = new Model();  
    final PNode world = createScene(model);  
    canvas.getLayer().addChild(world);  
    // Controller:  
    canvas.addInputEventListener(new Controller(model));  
  
    frame.setSize(600, 700);  
    frame.setVisible(true);  
}
```


Bsp: Hampelmann Scenegraph

```
private static final PNode createScene(final Model
    final PNode torso;
    PNode tmp;
    final PNode universe = new PNode();
    universe.addChild(torso = load("/torso.png"));
    universe.addChild(load("/hand.png"));
    torso.addChild(load("/kopf.png"));
    torso.addChild(load("/linker_arm.png"));
    torso.addChild(tmp = load("/linkes_bein.png"));
    tmp.addChild(load("/linker_fuss.png"));
    torso.addChild(load("/rechter_arm.png"));
    torso.addChild(tmp = load("/rechtes_bein.png"));
    tmp.addChild(load("/rechter_fuss.png"));
    return universe;
}
```

Bsp: Hampelmann Controller

```
public void mousePressed(final PInputEvent arg0) {  
    p0.setLocation(arg0.getPosition());  
    pull0 = pull;  
    // super.mousePressed(arg0);  
}  
  
public void mouseDragged(final PInputEvent arg0) {  
    arg0.setHandled(true);  
    final double _dy = arg0.getPosition().getY() - p0.getY();  
    setPull(pull0 + _dy / dy);  
}
```


Exkurs: Design Patterns & MVC

- ZUI Patterns - vor allem
 - Zentrale Klassen
 - Koordinatensysteme
 - Zoom Kontext
- MVC naheliegend:
 - View = Scenegraph + PCanvas + PCamera
 - Controller = Event Handler
 - Model = Custom Classes

Woher kommt Piccolo2D?



Wer benutzt Piccolo2D?

unter anderem:

- <http://phet.colorado%2eedu/>
- <http://jcurl.org/wiki/JCurlShotPlanner>

Eine umfassendere Liste:

- <http://piccolo2d.org/applications/>

Vielen Dank

für Ihre Aufmerksamkeit.

Quellen und Handout zum Nachlesen gibt's hier:

<http://alsw.mobi/go/jugm>

Lizenz

[Creative Commons by-sa](#)