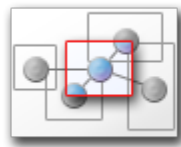




= Piccolo2D @ JUGM ein Referat von Marcus Rohrmoser, dem Kopf hinter aladin-software.mobi.



Piccolo2D

Structured 2D Graphics Framework



Anmerkung zu dieser Wiki Seite:

Manches hier sieht in Layout und Farbe ein wenig seltsam aus, da es in erster Linie bei der [Präsentation](#) passen muß.

Als Präsentationswerkzeug habe ich das [Dokuwiki S5 Plugin](#) gewählt, da's mir am einfachsten schien und komplett ohne Flash/Office/etc. auskommt. Ein moderner Web-Browser genügt.

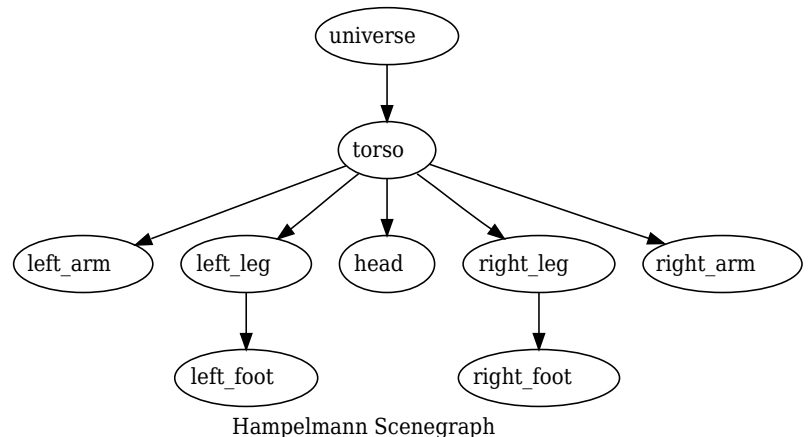
== Was ist Piccolo2D? * 2D Scenegraph API für Java und .Net, * Spezialitäten:



- Zooming,
- Animation,
- Mehransichtigkeit – z.B. gleichzeitige Detailansicht *und* Vogelperspektive,

* sehr kompakt – [wenige Klassen](#), [jar ~80 KB](#).

* benutzt intensiv [java2d](#) (ohne Compositing) und v.a. [AffineTransform](#), * ist [recht alt](#) und wurde in 2 1/2 Sprachen (Java/C#/C# Mobile) implementiert ⇒ konzeptionell sehr ausgereift, == 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.),

* in 3D üblich, 2D eher selten.

== 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, world));

    frame.setSize(600, 700);
    frame.setVisible(true);
}
```

* PCanvas * PCamera

== Bsp: Hampelmann Scenegraph

```
private static final PNode createScene(final Model m) {
    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);
}

```

* [PInputEventListener](#) * [PInputEvent](#)

== 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?



als SVG Bild, ergo sinnvoll klickbar.

== 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](#)  

* Präsentation: http://wiki.mro.name/_export/s5/jugm/p2d * Quellen zum Referat:
<http://github.com/mro/jugm-p2d/> * p2d Homepage: <http://piccolo2d.org> * p2d Maven Repo:
<http://repo2.maven.org/maven2/org/piccolo2d/> * p2d JavaDocs:
<http://www.piccolo2d.org/doc/piccolo2d.java/release-1.2.1/apidocs/> * p2d Metriken:
<http://www.ohloh.net/p/piccolo2d>

From:

<https://wiki.mro.name/> - □

Permanent link:

<https://wiki.mro.name/jugm/p2d?rev=1263089046>

Last update: **2010/01/10 03:04**

