Montag, 1. März 2010
15:17

Koordinatensystem

Graphical Programming (1)

- programs to process and manipulate graphical objects and data
- early standards: PHIGS, GKS → grafische Kernsystem
- 2D, 3D graphics libraries
- level of abstraction
  - low-level API: OpenGL
  - scene graph API: Java3D

anderes Verwaltungsmerkmal

aufwendig

Graphical Programming (2)

- X-Windows → client & server → client
  - X-protocol, Xlib, window manager → setzt sich aus Befehlen d. X-protokolls zusammen
- OSF/Motif
  - widgets, gadgets, window manager → Fenster mit Funktionalität
- MS-Windows
- RenderMan
- OpenGL
- Direct3D

Pixar

Geometrie, Schattierung

realistische Bilderzeugung

high-level Programmierung

Graphical Programming (3)

- OpenInventor, VRML, X3D
  - scene graph oriented
  - engines, sensors, manipulators
- ~~IRIS~~ OPEN GL Performer → Echtzeitanforderungen
- Java3D

Ton, Lichteffekte

OpenGL / Windows System

What is OpenGL?

- OpenGL is a hardware-independent interface to the graphics hardware

OpenGL does not provide rendering context management functions or means to manage user input

→ Interface between OS and OpenGL needed:

- WGL / Microsoft Windows
- GLX / UNIX's X-Windows (direct and indirect rendering)
- GLUT / OS independent

Java3D

Scenegraph

- Hierarchical structure
- Corresponds to logical structure of the scene
- Easy design and manipulation of complex scenes

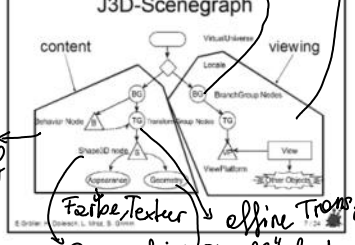
→ Animation eines Körpers bestehend aus mehreren Gliedmaßen leichter

1

Angabe d. Betrachterpositionation
z.B.: Abstand d. Augpunkts

Möglichkeit mehrere Betrachter zu definieren

J3D-Scenegraph



Verhalten nach Zeitablauf Einwirkung auf z.B. Tag

Farbe, Textur

affine Transf.

Geometrie: Oberfläche, Menge v. Polygonen → Punktelisten, Kantenliste

The Renderer

- Double buffering, RGB(A) → 2 Rasterbereiche jedes Pixel d. RGB u. Alphaband
- Infinite loop: Traverse Graph
- Take care of transparency
- Regions of influence of fog, lights, sound
- Optimized for rendering (display list)

Ausgabe reduzierte Abarbeitung

Vorteil affine Transf.: als Matrix darstellbar
10 Transformationen zu einer Matrix

Node States - Capabilities

- Node created: all operations legal
- *Live*: Node inserted into scene
  - *capabilities* specify permitted operations (r / w / modify)

  `setCapability(ALLOW_TRANSFORM_WRITE);`
- *Compiled* (optional): Optimized by renderer
  - modifications restricted
- Capabilities fixed if live or compiled

Java3D Node Types

- Group nodes
  - `addChild(...)`
  - BranchGroup
    - only detachable/relocable node
    - grouping of Shapes
  - TransformGroup
    - transform applied to all children

interner Knoten, Strukturierung zu Einheiten

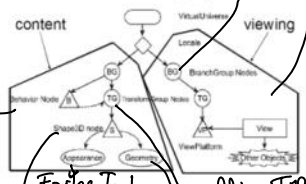
Hello3D! / Scenegraph

Hello3D! / Constructor

Angabe d. Betrachtungssituation
zB.: Abstand d. Augpunkts

Möglichkeit
mehrere Betrachter zu definieren

J3D-Scenegraph



Verhalten
nach Zeitablauf
Einwirkung
auf zB. TG

Farbe, Textur → affine Transf.

Geometrie: Oberfläche, Menge v. Polygonen
→ Punktelisten, Kanteliste

The Renderer

- Double buffering, RGB(A) → 2 Rasterbereiche, jedes Pixel d. RGB
- Infinite loop: Traverse Graph
- Take care of transparency
- Regions of influence of fog, lights, sound
- Optimized for rendering (display list)

Ausgabe
rekursive Abarbeitung u. Alphakanal

Vorteil affine Transf.:
als Matrix darstellbar
10 Transformationen zu einer Matrix

Node States - Capabilities

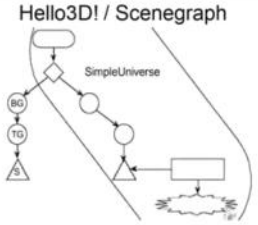
- Node created: all operations legal
- *Live*: Node inserted into scene
  - *capabilities* specify permitted operations (r / w / modify)
  - `setCapability(ALLOW_TRANSFORM_WRITE);`
- *Compiled* (optional): Optimized by renderer
  - modifications restricted
- Capabilities fixed if live or compiled

Java3D Node Types

- Group nodes
  - `addChild(...)`
  - BranchGroup
    - only detachable/relocable node
    - grouping of Shapes
  - TransformGroup
    - transform applied to all children

interner Knoten, Strukturierung zu Einheiten

Hello3D! / Scenegraph



Hello3D! / Constructor

```
...
Canvas3D c=new Canvas3D(null);
aWindow.add(c);
BranchGroup content=createSceneGraph();
SimpleUniverse u=new SimpleUniverse(c);
u.addBranchGraph(content);//now "live"
...
```

→ Ausgabebereich
→ Verbindung zu Fenster
→ Viewingteil
→ anhängen

2

Hello3D! Content

```
Public BranchGroup createSceneGraph(){
  BranchGroup objRoot=new BranchGroup();
  Transform3D spin=new Transform3D();
  Transform3D tmpspin=new Transform3D();
  spin.rotX(Math.PI/4.0d);
  tmpspin.rotY(Math.PI/5.0d);
  spin.mul(tmpspin);
  TransformGroup objTrans=new
              TransformGroup(spin);
  objRoot.addChild(objTrans);
  objTrans.addChild(new ColorCube());
  return objRoot();}
```

E.Gröller, H. Doleisch, L. Mroz, S. Grimm 13 / 24

↳ ergibt linken Teilbaum

Java3D Node Types

- Shape3D
  - defines object within the scene
  Contains:
  - Geometry
    - polygon related information
  - Attributes
    - material definition
    - rendering mode (wireframe, ...) → Transparenz, Schattierungsinfo

E.Gröller, H. Doleisch, L. Mroz, S. Grimm 14 / 24

[1 2 3] ... linear

Geometry Nodes

→ Vierecke

E.Gröller, H. Doleisch, L. Mroz, S. Grimm 15 / 24

→ Punkte werden neu mehrfach abgespeichert

[1 2 3 4]

Geometry Nodes

Geometry definition:

- coordinates
- optional: normals
- optional: RGB(A) color
- optional: texture coord. (u/v[/w])
- for indexed types: indices

Aufbringen eines Rasterbilds

E.Gröller, H. Doleisch, L. Mroz, S. Grimm 16 / 24

gekrümmte Fläche

Dreieck untersch. Normalvektoren

wichtig: Texturzuordnung in Form v. Koordinaten

innerer Punkt eines △s: lineare Kombination

Helper Classes

- com.sun.j3d.utils.*
  - SimpleUniverse:
    fast viewing model setup
  - Mouse-> Transform mapping
    for interactions
  - Simple geometrical Objects
    (cube, ...)

E.Gröller, H. Doleisch, L. Mroz, S. Grimm 17 / 24

Helper Classes

- javax.vecmath.*
  - Tuple/Point/Vector 2/3/4D d/float
    with corresponding operations
  - Matrix 3/4D d/float
  - Quaternion classes → 3 für Werte Rotationsachse, 1 für Winkel
- javax.media.j3d.Transform3D
  - 4D matrix + helper func. for CG:
    orth/persp. Projection,
    scale, rotate, translate, ...

→ Translationen einfacher

E.Gröller, H. Doleisch, L. Mroz, S. Grimm 18 / 24

3

Normalvektoren

geschlossene Objekte

Appearance

Defines attributes for rendering
geometry of an object

- polygon attributes: pts/wirefr./fill
- culling
- rendering attrib.: z-buff., α-blend.
- transp./color (if not per vertex)
- material (reflection coefficients)
- texture images

überblenden v. Info

→ Pixel wird nur teilweise überdeckt

E.Gröller, H. Doleisch, L. Mroz, S. Grimm 19 / 24

Lighting

Light source nodes require Bounds
delimiting the area of influence

- Ambient
- Directional (infinitely distant) z.B. Sonne
- PointLight (spot light) Geometrieangabe d. Lichtes

E.Gröller, H. Doleisch, L. Mroz, S. Grimm 20 / 24

Behaviors

Reaktion auf Interaktion

"Events" for the scenegraph

- Action executed if criteria met:
  - time triggered, kbd & mouse
    events, picking of objects,
    collision, frames elapsed
- Boolean combination of criteria
- Bounded area of relevance
- processStimulus() → vor jedem Rendern

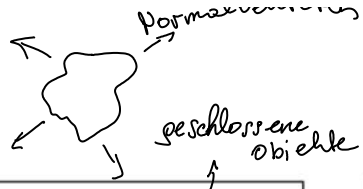
↳ welches Verhalten ist aktuell

E.Gröller, H. Doleisch, L. Mroz, S. Grimm 21 / 24

Behaviors

- Influence transform. / geometry
- Examples:
  - Mouse controlled navigation
  - Interpolator: time->value
  - ProximitySensor
  - Collision reaction → plastischer od. elastischer Stoß
  - Picking: Click ->
    list of Shape3Ds below mouse

E.Gröller, H. Doleisch, L. Mroz, S. Grimm 22 / 24



Appearance

Defines attributes for rendering geometry of an object
- polygon attributes: pts/wirefr./fill
- culling
- rendering attrib.: z-buff., α-blend.
- transp./color (if not per vertex)
- material (reflection coefficients)
- texture images

überblenden v. Info
→ Pixel wird nur teilweise überdeckt

Lighting

Light source nodes require Bounds delimiting the area of influence
- Ambient
- Directional (infinitely distant) z.B. Sonne
- PointLight (spot light) Geometrieangabe od. Richter

Behaviors

Reaktion auf Interaktion

"Events" for the scenegraph
- Action executed if criteria met:
  - time triggered, kbd & mouse events, picking of objects, collision, frames elapsed
- Boolean combination of criteria
- Bounded area of relevance
- `processStimulus()` → vor jedem Rendern
  ↳ welches Verhalten ist aktuell

Behaviors

- Influence transform. / geometry
- Examples:
  - Mouse controlled navigation
  - Interpolator: time->value
  - ProximitySensor
  - Collision reaction → plastischer od. elastischer Stoß
  - Picking: Click -> list of Shape3Ds below mouse

Viewing Model

→ Betrachtungs-Standpkt.

- Supports multiple Canvas3D -> stereo rendering, (Cave!)
- Support for tracking.
- Detailed description of viewer's eyes/ears configuration

Java3D - Further Material

Java 3D API specification documentation & tutorial:

`https://java3d.dev.java.net/`