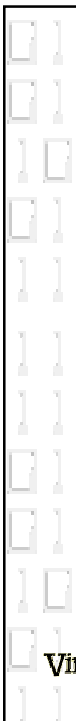




# 3D modeling & Animation

Nicholas F. Polys, Ph.D.



- geometry & appearance
- animation and time
- sensors
- environments



## Resource Reminder

- Specifications, models, tutorials:
  - [www.web3d.org](http://www.web3d.org)
- VRML Reference Manual
  - <http://www.cs.vu.nl/~eliens/documents/vrml/reference/BOOK.HTM>



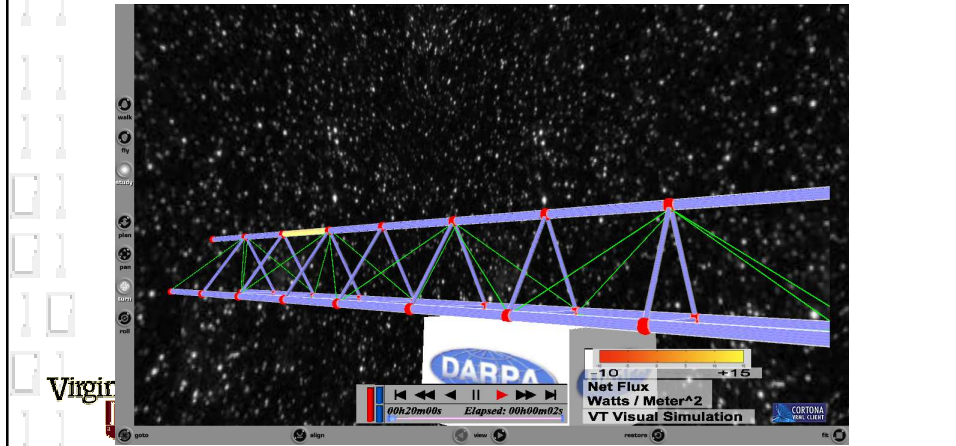
## FLUX Studio

- Save (flux format)
- Import / Export
  - VRML, X3D
  - Many others roundtrip
- File -> Pull Dependent Files Local
  - Collects resources for upload/publication



## Viewers, Browsers, Players

- Plug-ins vs. Stand Alone



## Building VEs

- **Navigation** - many control options - walk, fly, examine or none (engines may also support proprietary modes)
- **Viewpoints** - pre-defined camera positions

## Building VEs

- **Models** - primitives (box, sphere, cone, cylinder), extrusions, indexed face set (mesh), line set, point set, elevation grid and text (engines may also support proprietary spline and NURBS geometry)
- **Materials** - diffuse colour, specular, emissive, ambient, shininess, transparency, colour per vertex



## Hands-on with Flux Studio

Visual Perception, Cameras, & Navigation

- geometry
- appearances
- lighting



# VRML & X3D

## Scenegraph Basics

- Hierarchy of nodes (transformation graph)  
**Nodes / Elements**
- Events and attribute Data types (behavior graph)



# DAG!

## (Directed Acyclic Graph)

Important nodes - (Grouping in parent/children hierarchy):

- Transform { }
- Group { }
- Anchor { }
- Switch { }
- LOD { }
- Billboard { }



## Transform Node:

- translation  
(x y z coordinates, +Y is up,  
+Z is toward the user)
- rotation around an axis  
(x y z theta; language in rads, Flux in deg)
- scale (factor in x y z )
- children (other nodes)
- Boundingbox (helps rendering optimizations)



## Shapes

Contain:

- geometry
- appearance



## geometry


- Primitives
  - Box, Cone, Cylinder, Sphere
- Extrusion
- ElevationGrid



## Indexed Faces



- Coordinate { }
- coordIndex
- creaseAngle : shading across polygons edges of the mesh
- normals (for shape-dependent lighting control)
- solid
- colorPerVertex






## Mesh editing

- Convert to IFS and select vertices, edges, polygons



## Indexed Lines

- Like faces except:
  - Use emissive color
  - Typically not pickable
  - No creaseAngle, solid fields etc.





## High order shapes

- Swept surfaces
- NURBS



## Boolean Operations

On geometry:

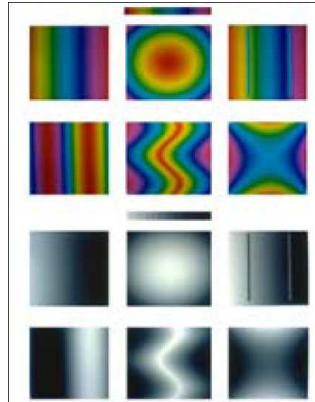
- Union
- Intersect
- Subtract (Extract)



## Features: Color

- Luminance channel  
(3x spatial acuity)
- Red / Green channel
- Yellow / Blue channel

The spectrum is not a perceptually linear sequence (not pre-attentive)!  
(Keller 1993; Ware, 2000)






## Shapes & Appearances

- **Appearance** { } and **Materials** { } :  
specular, emissive, and diffuse Colors  
in RGB, shininess, transparency,  
ambientIntensity
- creaseAngle : shading across polygons  
edges of the mesh
- normals (for shape-dependent lighting  
control)
- colorPerVertex



```
Material {
  RGB diffuseColor
    0.678, 0.169, 0.07
  specularColor
  shininess
}
```



Virginia Tech

## Building VEs

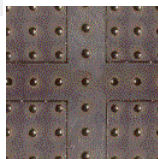
- **Textures** - support for JPEG, GIF, PNG and MPEG1 video. (engines may also have proprietary support for Flash, RealMedia, AVI, multi-texturing and environment mapping)
- **Lighting** - directional, point or spotlight
- **Environments** - background, and fog, hyperlinks (anchor), inlines

# Textures

- ImageTexture {} with (or without) alpha channels can be applied and mapped to geometry as fixed or animated maps.
  - Standard formats: .png, .jpg,
- MovieTexture {}
- TextureTransform {} ...
- PixelTexture {}



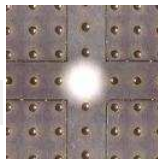
## MultiTexture {}



Base Texture



+ Lightmap



= Result

Blending operations  
specified via  
mode field



# Lighting

## Lighting Nodes:

- `on, intensity, ambientIntensity, color`
- `Pointlight {attenuation}`
- `DirectionalLight {}`
- `Spotlight {direction, beamWidth, cutOffAngle}`
- **AMD 1: SFBool global**



# Lighting

- Lights have color!
- Directional Lights
  - ‘Scoped’ by scenegraph
  - sibling rule
- `webfolder/vrml/lightingbasis.wrl`



## Existential Perception

What is my relation to this environment?

What can I do in this world?

What do my senses tell me?

- Viewpoint {fieldOfView}
- NavigationInfo {avatarSize, headlight, visibilityLimit, type, speed}
- Timesensor {cycleInterval }



## Environmental effects

- Background {}: colors and textures give a context for the environment
- TextureBackground {transparency}
- Fog {type color visibilityRange}
- LocalFog {} & FogCoordinate {} (...x3d only)



## Time & Interactivity

- Keyframed animation
- Functional animation
- Events are ROUTED between nodes



## Auditory Perception

- **Sound** {}
- **AudioClip** {}
- **MovieTexture** {}
  - pitch
  - intensity
  - Spatialized Audio (doppler effect)
  - Standard formats:  
.wav, .midi, .mp3, mpeg-1



## Building VEs

- **Performance** - LODs (levels of detail), visibility distance culling
- **Animation** - animate position, rotation, scale, points, colour and much more. Scope for many separate animations in one world all with different time lines and triggered by different events. Almost every attribute can be animated.
- **Sensors** - sense user activity such as touch, drag, keypress (plane, cylinder, sphere, and key sensors). Environmental sensors include time, proximity, and visibility.



## Building VEs

- **Scripting** - Interfaces directly with ECMAScript; also with Java, the web browser (DOM) and any programming language residing on the client/server
- **Routes** - scripts, animations and object properties can be "wired" together in an infinite number of ways to create any effect





## Interpolators

- Position
- Orientation
- Scale
- Others:
  - Colors
  - Coordinates



## Sensors

- TouchSensor
- Dragsensors
  - Plane
  - Cylinder
  - Sphere
- ProximitySensor
- VisibilitySensor



## Frame Rate


- Threshold for perceiving continuity:
  - flicker < 50 Hz
  - > 24 fps looks smooth & plenty interactive
- Flicker & Attention can lead to change blindness (Simmons, 2000)
- `Browser.getCurrentFrameRate()`
- Implementing `X3DPerFrameObserverScript`
  - `public void prepareEvents (){}`





## HTML Integration

- `<EMBED SRC="vrmf/composed.wrl" WIDTH="450" HEIGHT="300">`
- The newer lets you specify a `classid=""` attribute  
`<OBJECT >`
- Anchor
  - url
  - description
  - parameter "target=\_blank" or some named frame






- webfolder/index.html and
- embedded.html



## Characters

- Hanim
- Flux Studio
- Avatar Studio – Canal plus
- <http://www.vrmlworlds.com/software/avatastudio/>



## New X3D technology

- Scene Access Interface (SAI)
  - Connect with external applications (eg Java, COM)
- Document Object Model (DOM)
- AJAX [www.ajax3d.org](http://www.ajax3d.org)



## Prototypes

- Encapsulating scenegraph branches
- For reuse
- The instance attributes are declared on the interface
  
- Slider example



# Scripting

## Client side

- ECMA Script
  - Loosely typed
  - Non-compiled
  - Basic objects such as Math, Date, Browser
- Java
  - Fully typed
  - Compiled
  - Industrial-strength classes



# Server Side

## • Internet and Local Resources

- MIME Types
- For VRML
  - x-world/x-vrml .wrl
  - Content-encoding: x-gzip .gz
- For X3D
  - AddType model/x3d+xml .x3d
  - AddType model/x3d+vrml .x3dv
  - AddType model/x3d+binary .x3db
  - AddEncoding gzip .x3dvz
  - AddEncoding gzip .x3dbz

## • Anchor, Inline



# Getting online content

```
urlString =
```

```
'http://server.vt.edu/gateway/section_query.pl?param=yes
```

```
Browser.createVrmlFromURL(urlString,self,'isAdded');
```

Hexunit code example (server not live)

