M3G – Java Mobile 3D

Tomi Aarnio

Nokia Research Center

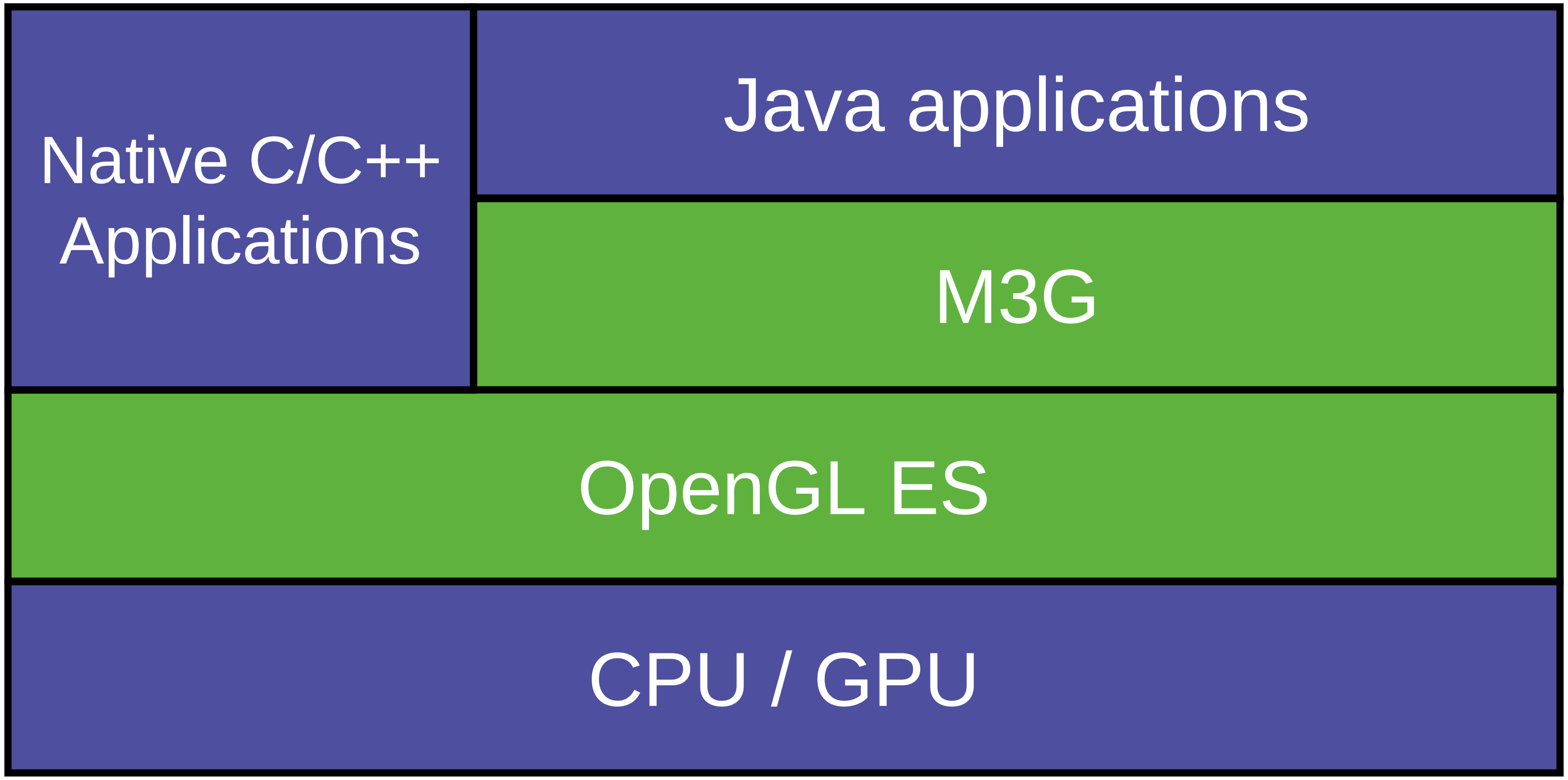
Agenda

- What is M3G
- What's new in 2.0

M3G – Mobile 3D Graphics API for Java

- Enables real-time 3D on mass-market phones
  - Came out in 2004, now almost universally adopted
  - Installed base somewhere between 500M-1B
- Retained mode API
  - OpenGL ES features wrapped into Java objects
  - Animation and scene graph layered on top

Mobile 3D Graphics APIs



Mobile Java

Pros

+ More widely available than any other platform
+ The <u>only</u> platform on many/most phones
+ Easy to write code that works

Cons

– Different devices have different APIs (and bugs)
– Latest hardware features not always available
– Performance not as good as in C/C++

NOKIA

M3G Design Principles

#1	Minimize the amount of Java code
#2	Do not require graphics hardware
#3	Enable easy content creation

Programming Model

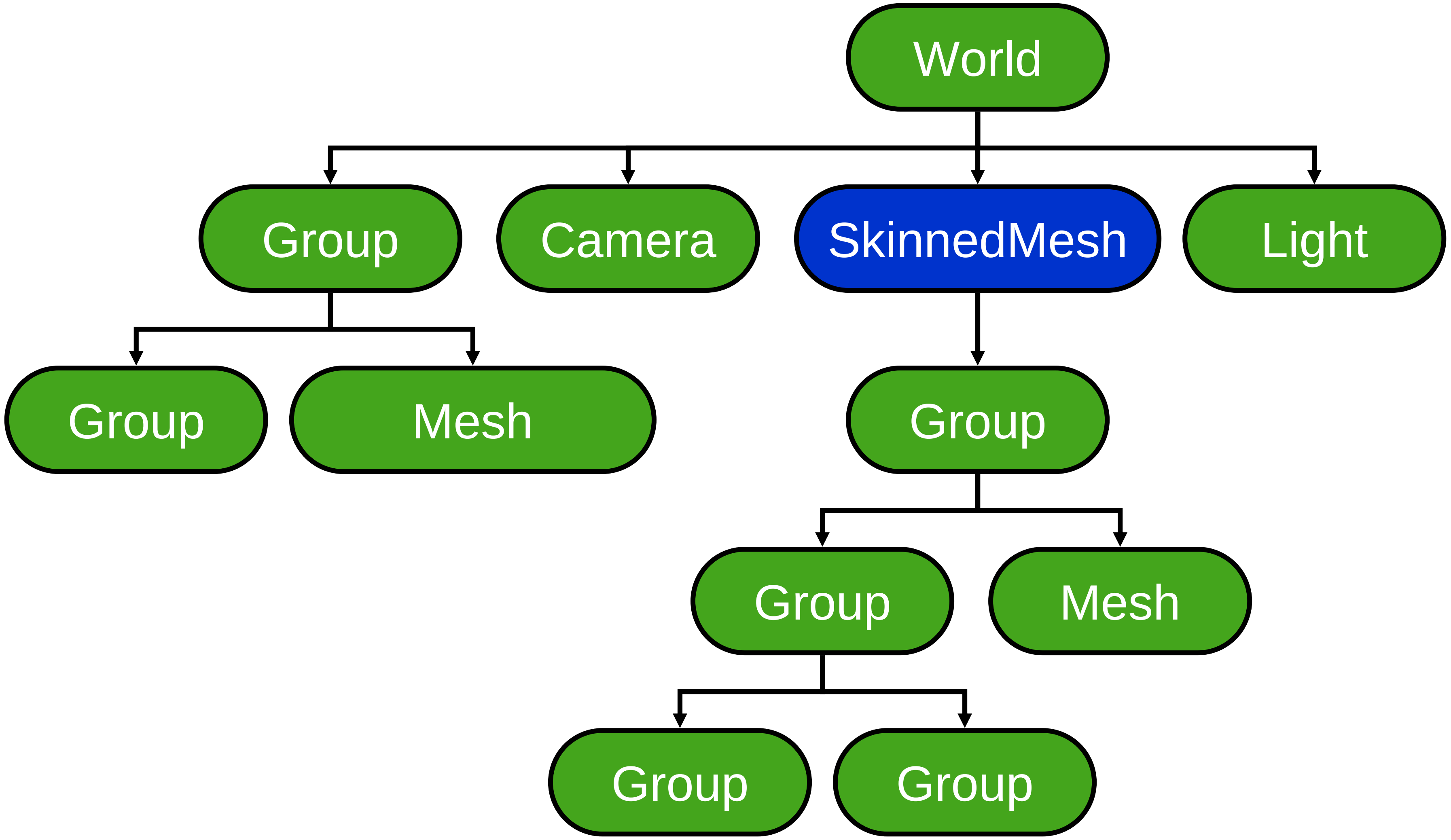
- M3G is not an "extensible" scene graph
  - No interfaces, events, or render callbacks
  - No threads; all methods are synchronous

- Scene update is decoupled from rendering
  - `render` ➔ Draw the scene, no side-effects
  - `animate` ➔ Update the scene to the given time
  - `align` ➔ Re-orient target cameras, billboards

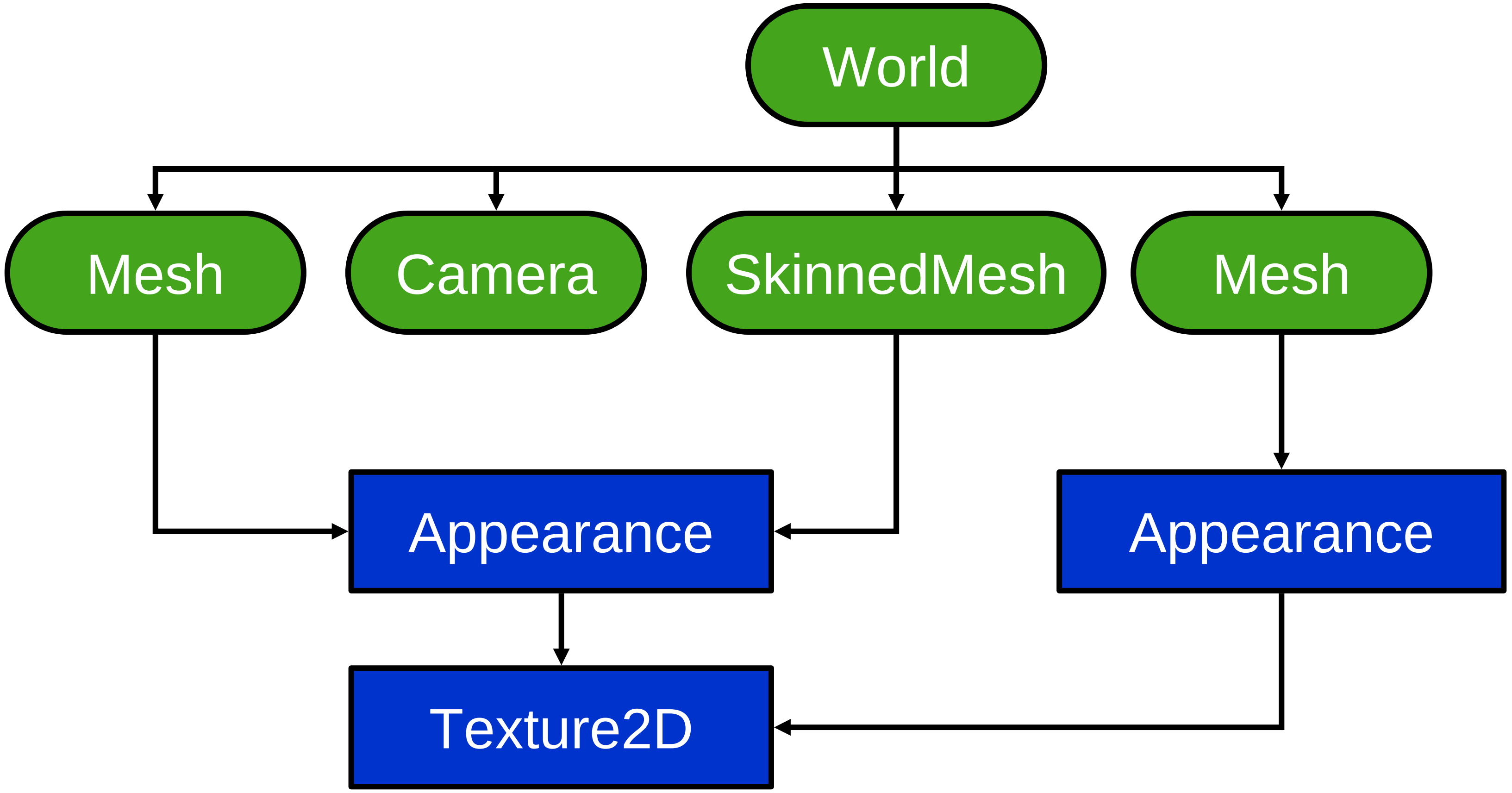
Main classes

Graphics3D	*3D graphics context* *Performs all rendering*
Loader	*Loads individual objects and entire scene graphs*
Mesh	*Encapsulates triangles, vertices and appearance*
World	*Scene graph root node*

Example scene graph

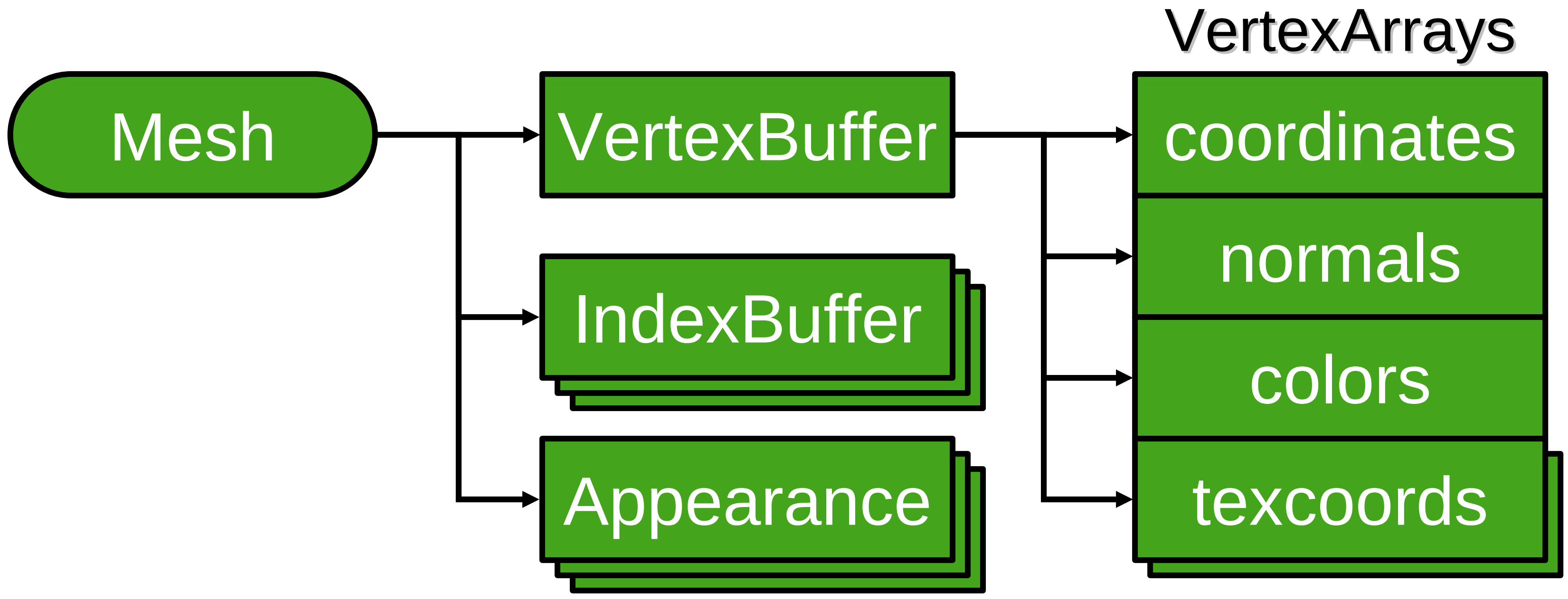


Components can be shared



Mesh

- One VertexBuffer, containing VertexArrays
- 1..*N* submeshes (IndexBuffer + Appearance)



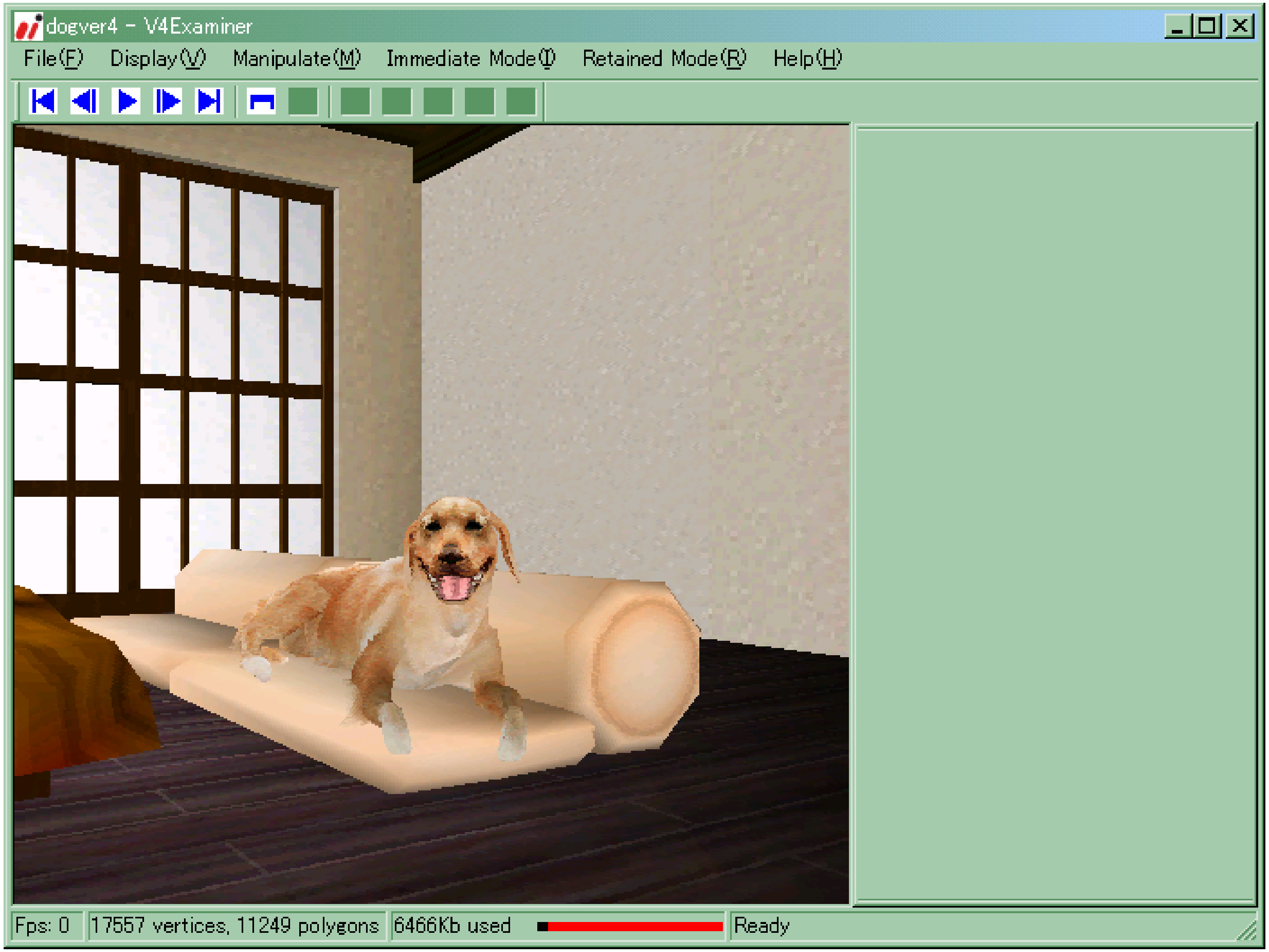
Simple animation player

```
world = (World) Loader.load(“/scene.m3g”)[0];
```

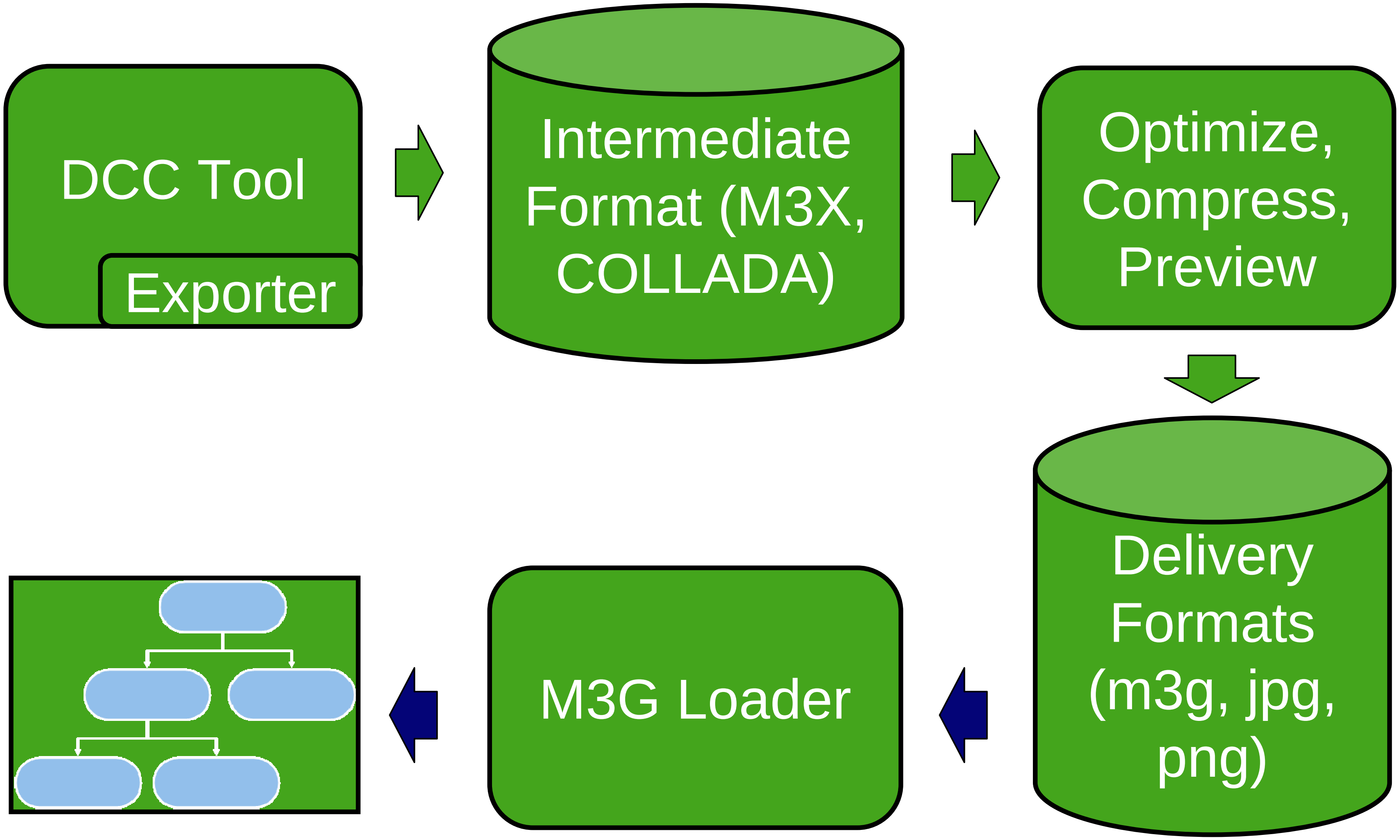
```
void paint(Graphics g) {
   world.animate(currentTime);
   graphics3d.bindTarget(g);
   graphics3d.render(world);
   graphics3d.releaseTarget();
}
```

NOKIA

犬友 (Dear Dog) Demo

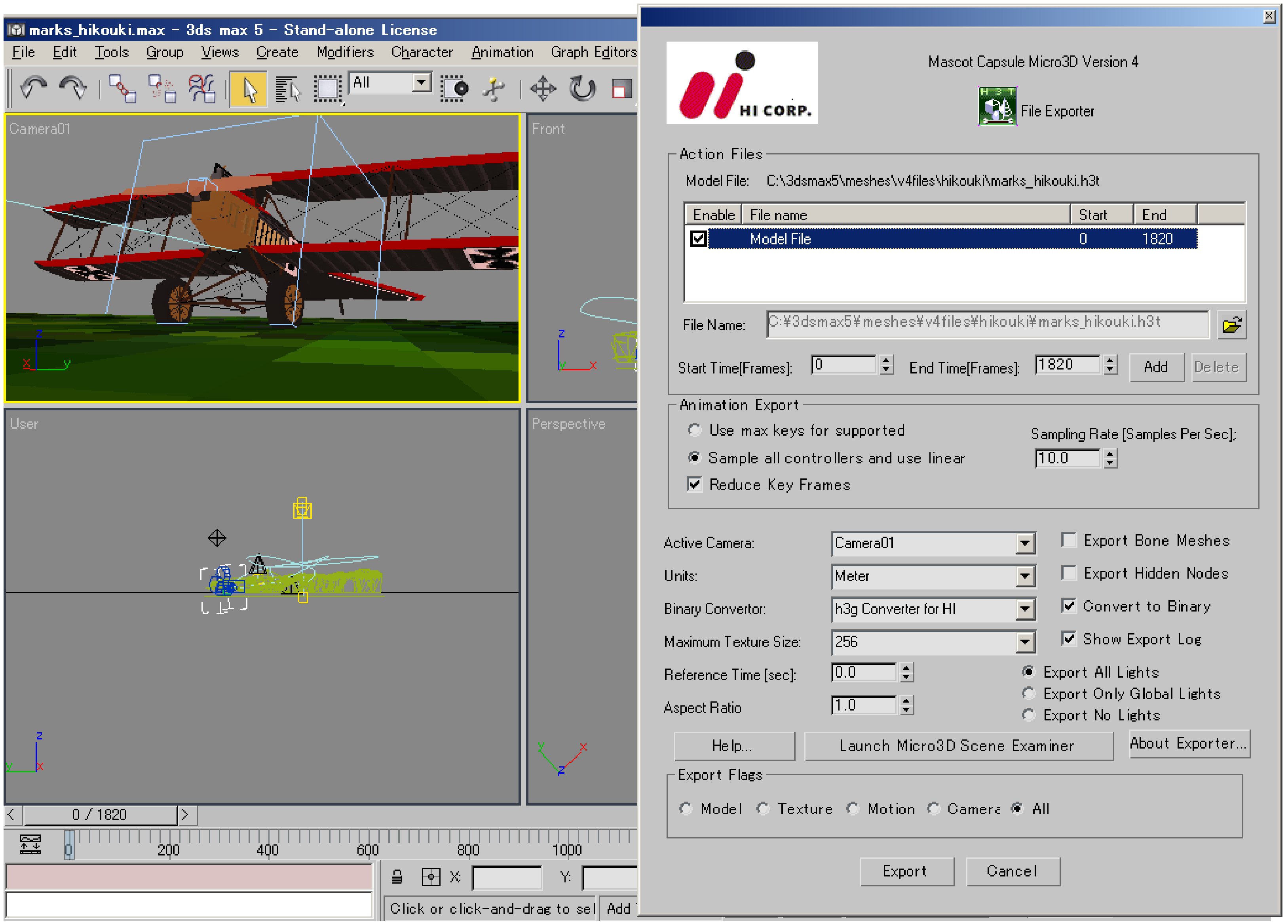

dogver4 - V4Examiner
File(F) Display(V) Manipulate(M) Immediate Mode(I) Retained Mode(R) Help(H)
Fps: 0
17557 vertices, 11249 polygons
6466Kb used
Ready

Creating art assets


DCC Tool
Exporter
Intermediate Format (M3X, COLLADA)
Optimize, Compress, Preview
Delivery Formats (m3g, jpg, png)
M3G Loader

Runtime Scene Graph

Mascot Capsule M3G Exporter


marks_hikouki.max - 3ds max 5 - Stand-alone License
File Edit Tools Group Views Create Modifiers Character Animation Graph Editors
All
Camera01
Front
User
Perspective
0 / 1820
0 200 400 600 800 1000
X: Y:
Click or click-and-drag to sel
Add
HI CORP.

Mascot Capsule Micro3D Version 4

File Exporter

Action Files

Model File: C:\3dsmax5\meshes\v4files\hikouki\marks_hikouki.h3t

Enable	File name	Start	End
✓	Model File	0	1820

File Name: C:¥3dsmax5¥meshes¥v4files¥hikouki¥marks_hikouki.h3t

Start Time[Frames]: 0 End Time[Frames]: 1820 Add Delete

Animation Export

Use max keys for supported [✓ Sample all controllers and use linear]

Sampling Rate [Samples Per Sec]: 10.0

[✓] Reduce Key Frames

Active Camera: Camera01

Units: Meter

Binary Convertor: h3g Converter for HI

Maximum Texture Size: 256

Reference Time [sec]: 0.0

Aspect Ratio 1.0

[ ] Export Bone Meshes

[ ] Export Hidden Nodes

[✓] Convert to Binary

[✓] Show Export Log

[✓ Export All Lights] Export Only Global Lights Export No Lights

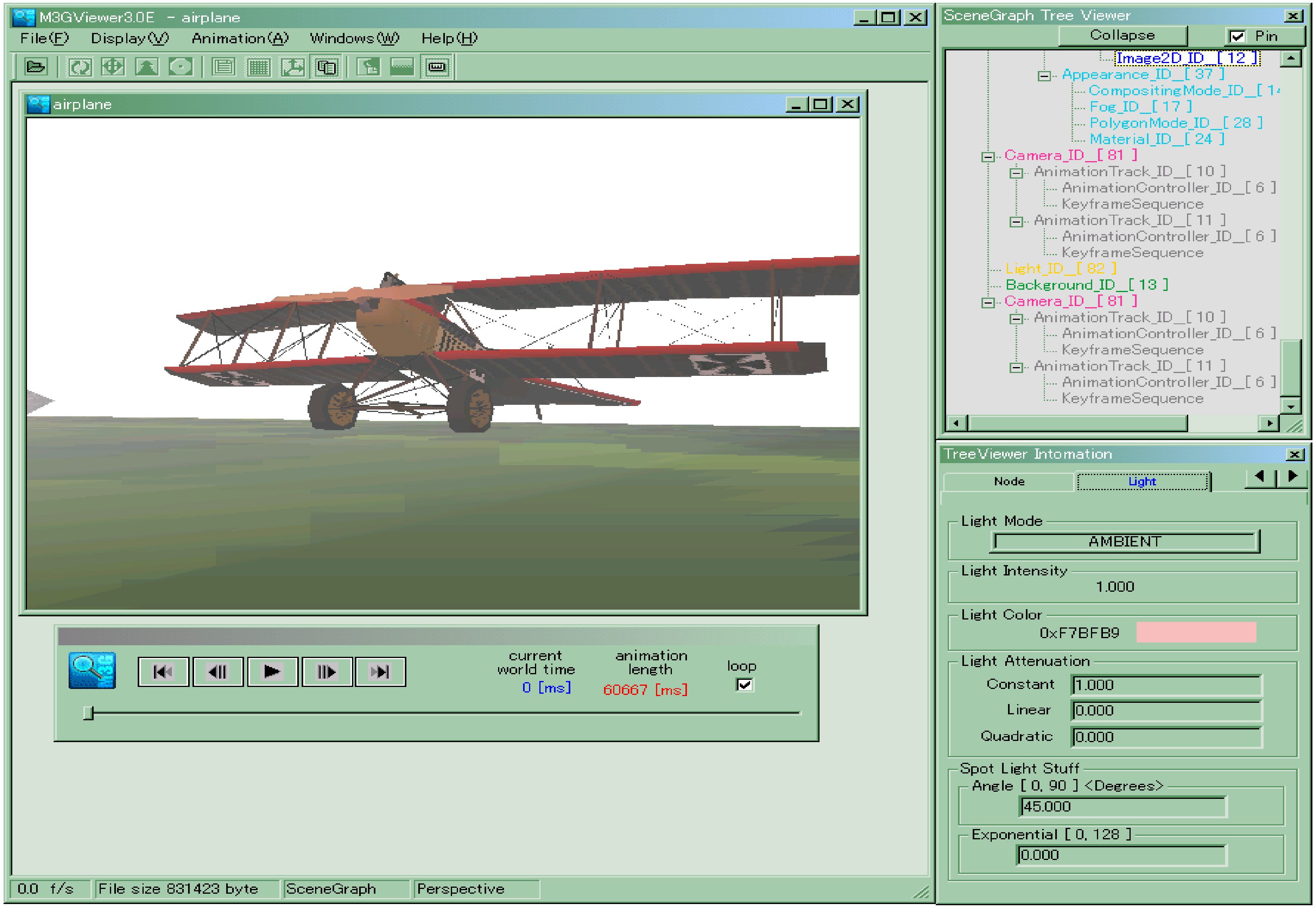
Help... Launch Micro3D Scene Examiner About Exporter...

Export Flags

Model Texture Motion Camera [✓ All]

Export Cancel

Mascot Capsule M3G Viewer



Light Mode: AMBIENT

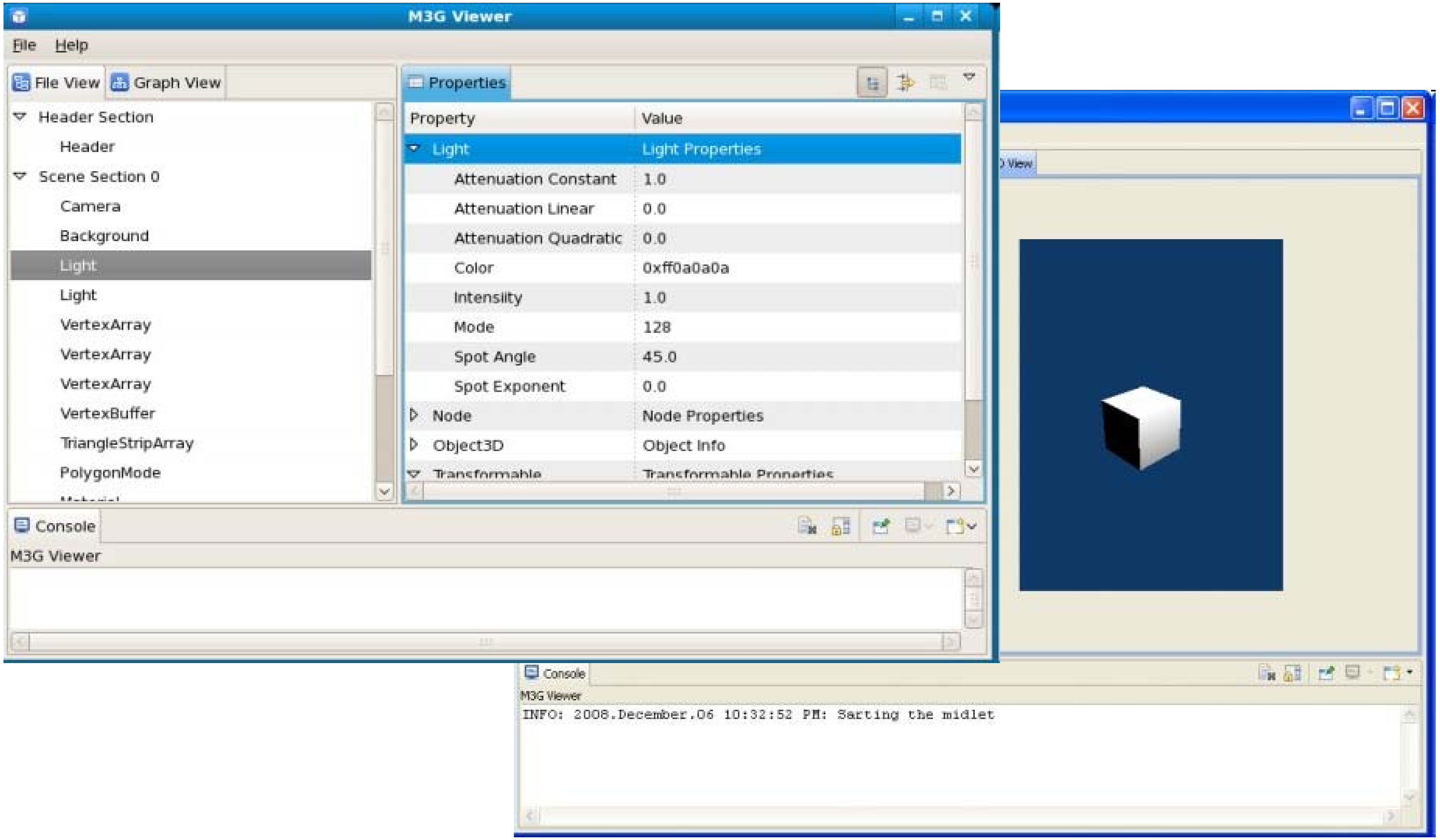
Light Intensity: 1.000

Light Color: 0xF7BFB9

Light Attenuation
- Constant: 1.000
- Linear: 0.000
- Quadratic: 0.000

Spot Light Stuff
- Angle [0, 90] <Degrees>: 45.000
- Exponential [0, 128]: 0.000

Wizzer Works M3G Viewer



- Header Section
  - Header
- Scene Section 0
  - Camera
  - Background
  - Light
  - Light
  - VertexArray
  - VertexArray
  - VertexArray
  - VertexBuffer
  - TriangleStripArray
  - PolygonMode

Property	Value
Light	Light Properties
Attenuation Constant	1.0
Attenuation Linear	0.0
Attenuation Quadratic	0.0
Color	0xff0a0a0a
Intensity	1.0
Mode	128
Spot Angle	45.0
Spot Exponent	0.0
Node	Node Properties
Object3D	Object Info
Transformable	Transformable Properties

```
INFO: 2008.December.06 10:32:52 PM: Sarting the midlet
```

Selected open source projects

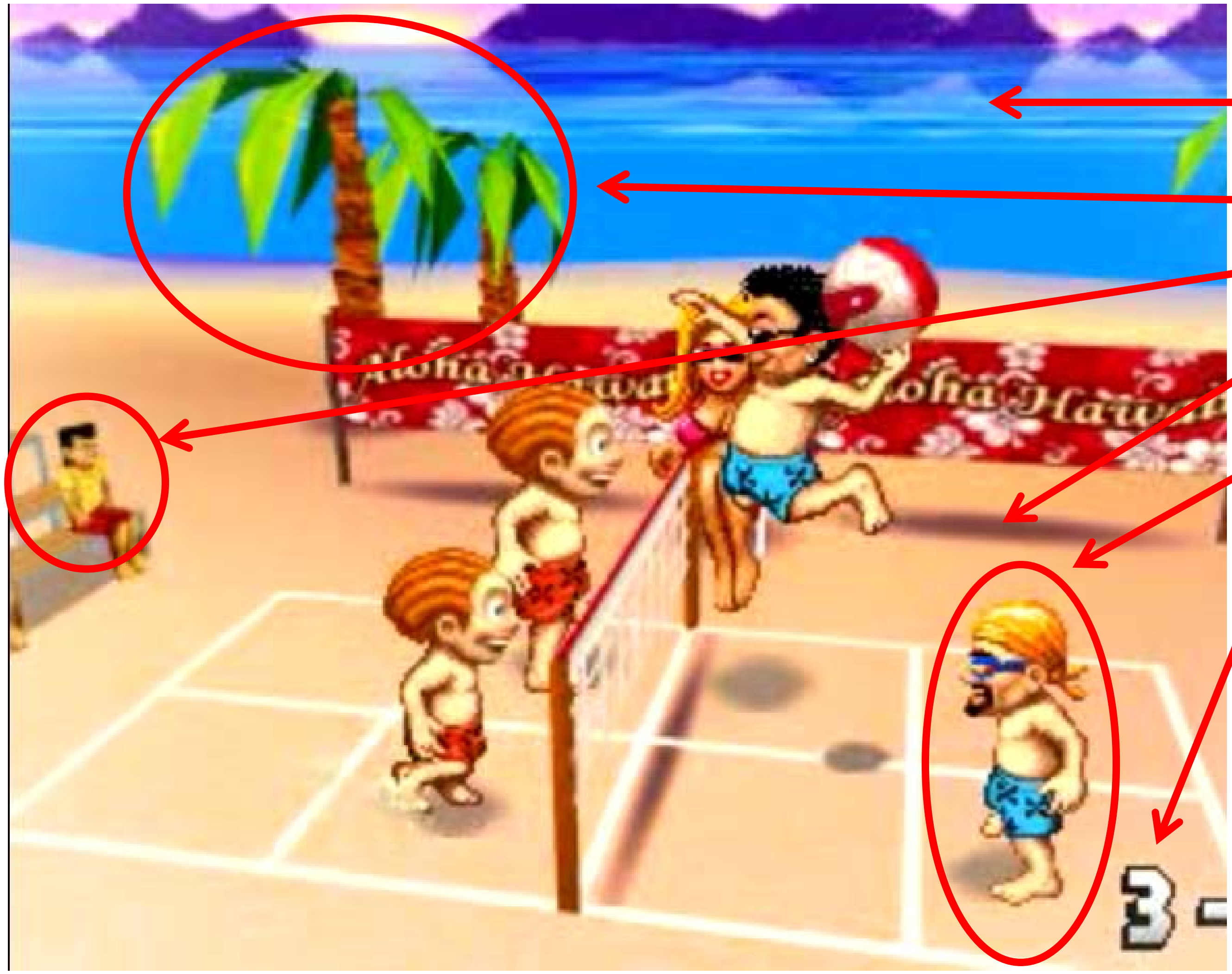
- www.wizzerworks.com
  - M3G Toolkit & Viewer for manipulating .m3g files
- m3x.dev.java.net
  - XML encoding of the .m3g file format + tools
- www.microemu.org
  - Java ME stack implemented on Java SE / Android
- lwuit.dev.java.net
  - Lightweight UI Toolkit, uses M3G for transition effects

Start developing!

- Choose IDE
  - www.eclipse.org
  - www.netbeans.org
- Choose SDK
  - forum.nokia.com/java
  - developer.sonyericsson.com/java
  - mpowerplayer.com/sdk
- Choose Exporter
  - www.m3gexport.com – Maya
  - www.mascotcapsule.com/M3G – Max, Maya, Lightwave, XSI
  - www.nelson-games.de/bl2m3g – Blender (open source)

Example Games

Playman Beach Volley – RealNetworks



2D backdrop

3D background

2D spectators

3D field

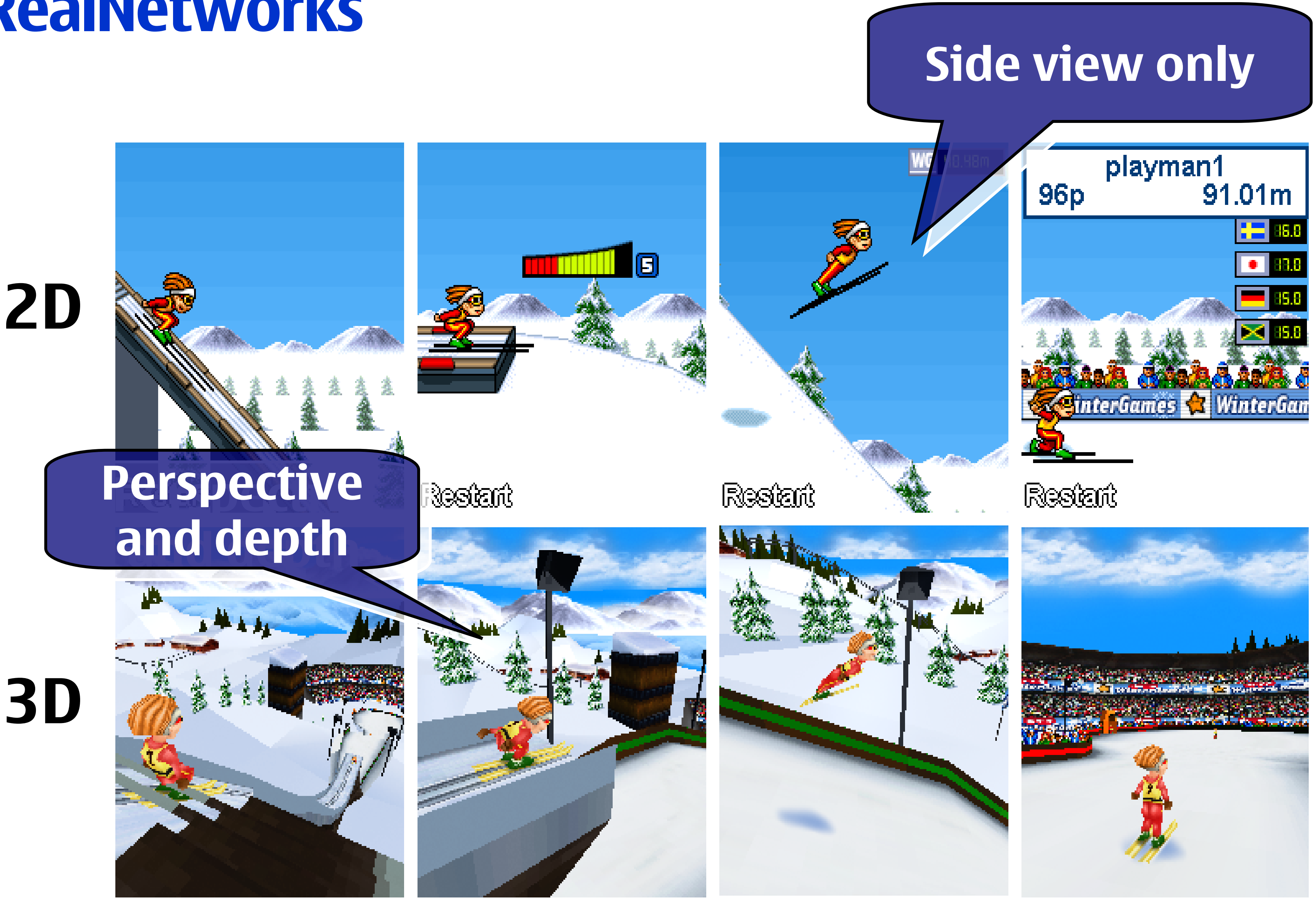
2D players

2D overlays

~7 layers of 2D and 3D!

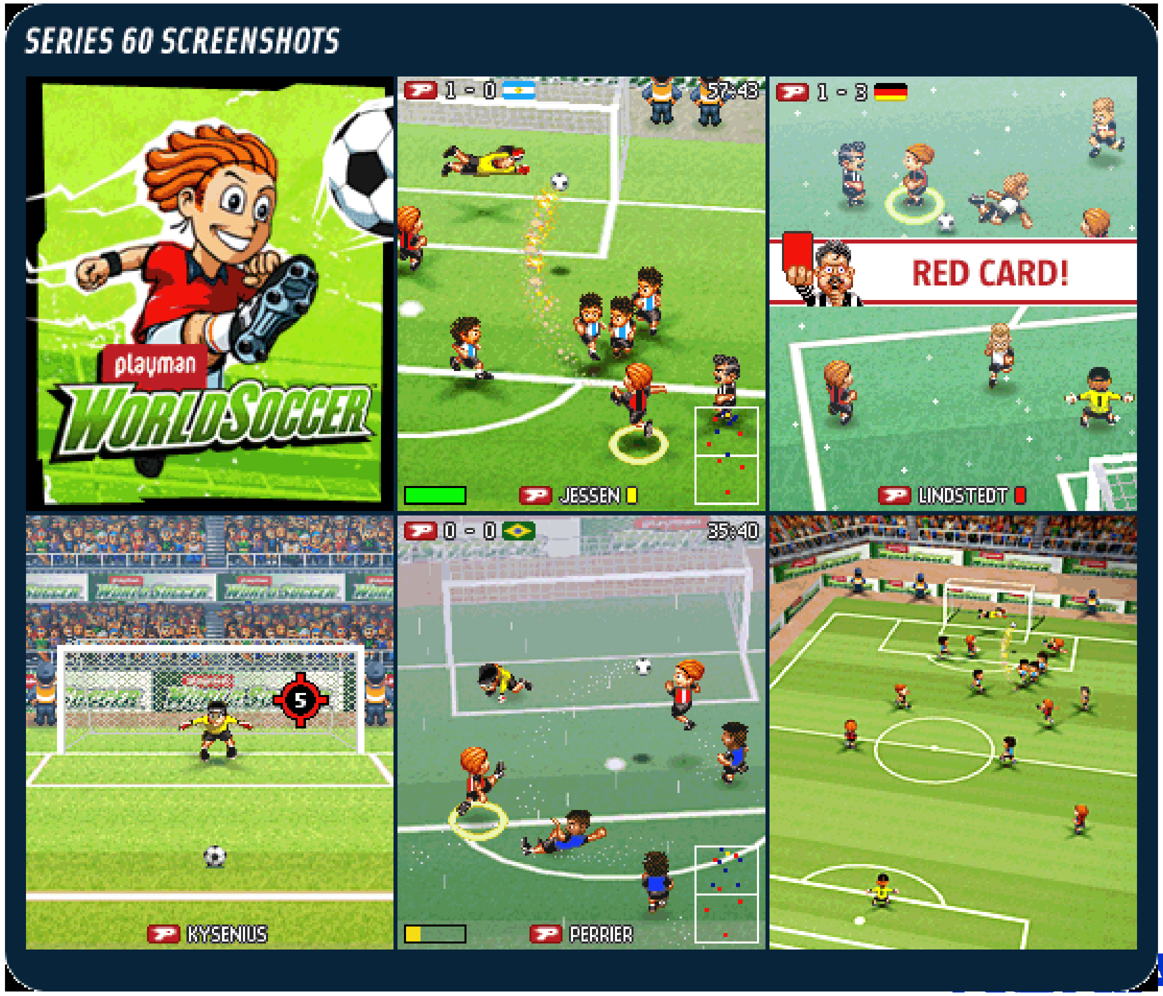
NOKIA

Playman Winter Games – RealNetworks

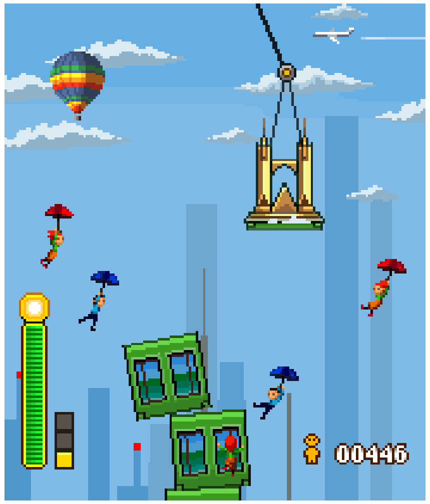


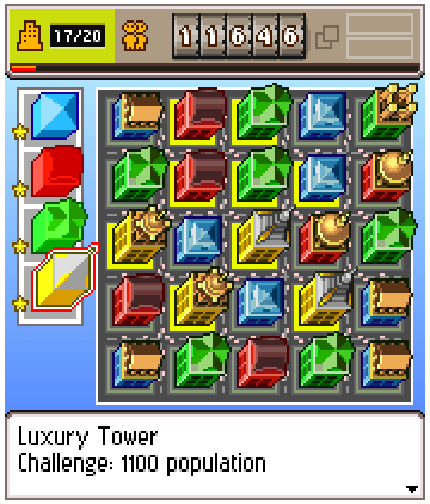
Playman World Soccer – RealNetworks

- 2D/3D hybrid
- Cartoon-like 2D figures in a 3D scene
- 2D particle effects etc.

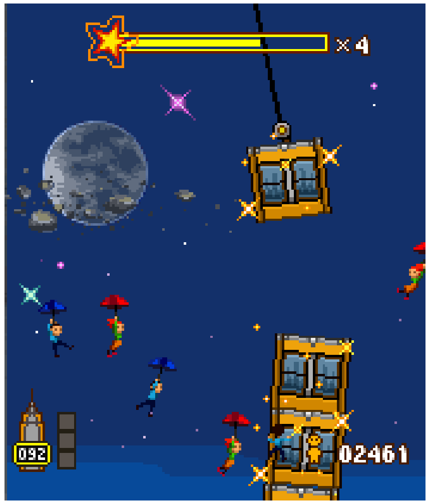


Tower Bloxx – Digital Chocolate





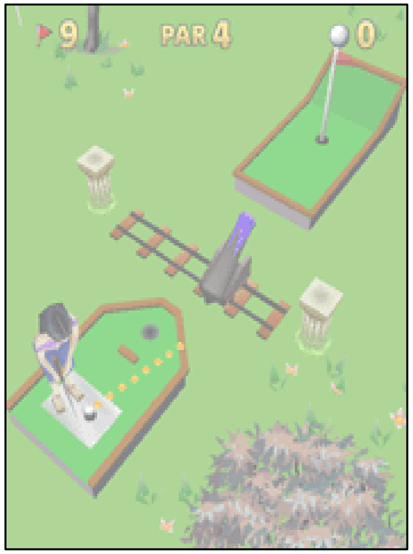
Luxury Tower
Challenge: 1100 population

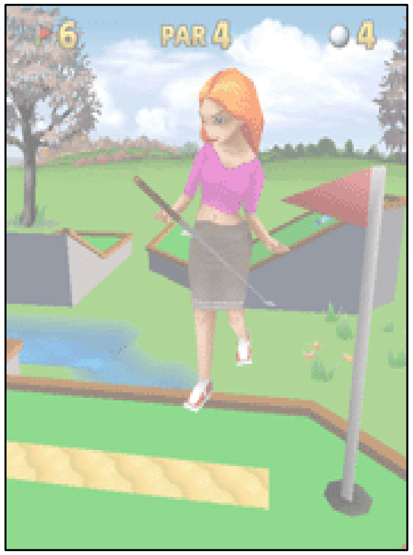


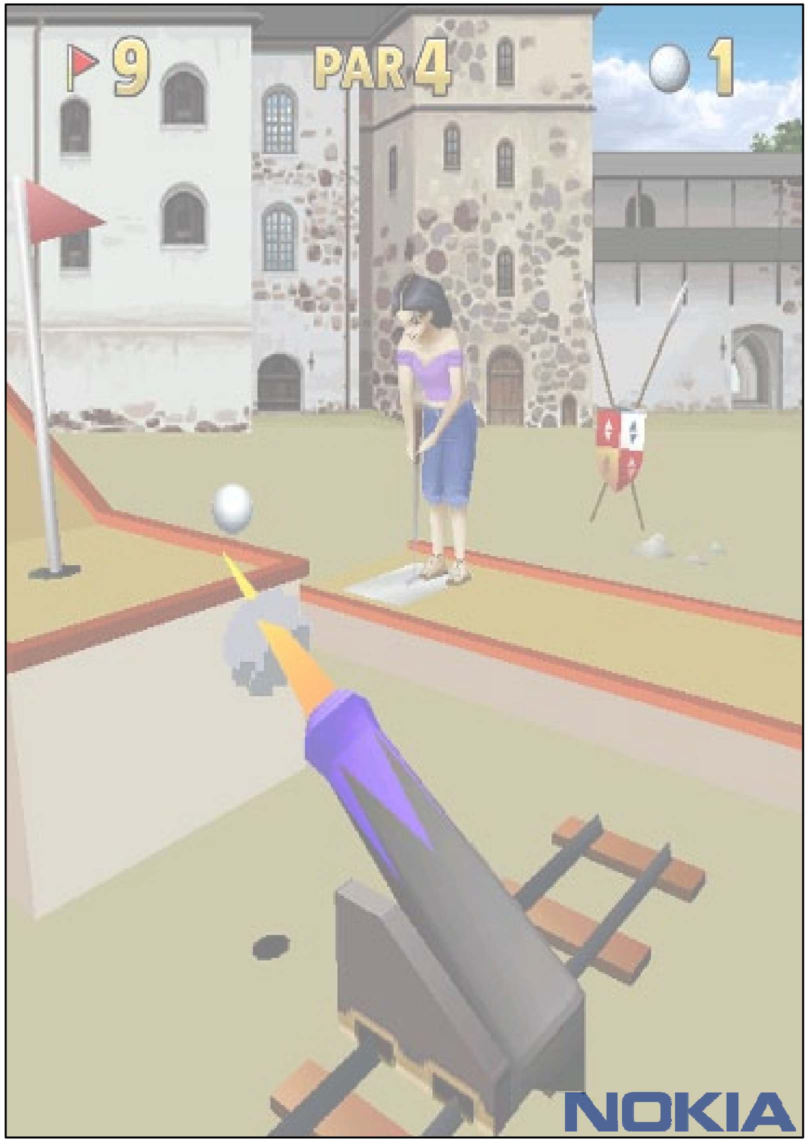
- Puzzle/arcade mixture
- 3D with 2D overlays and backgrounds

Mini Golf Castles – Digital Chocolate

- 3D with 2D background and overlays
- Skinned characters

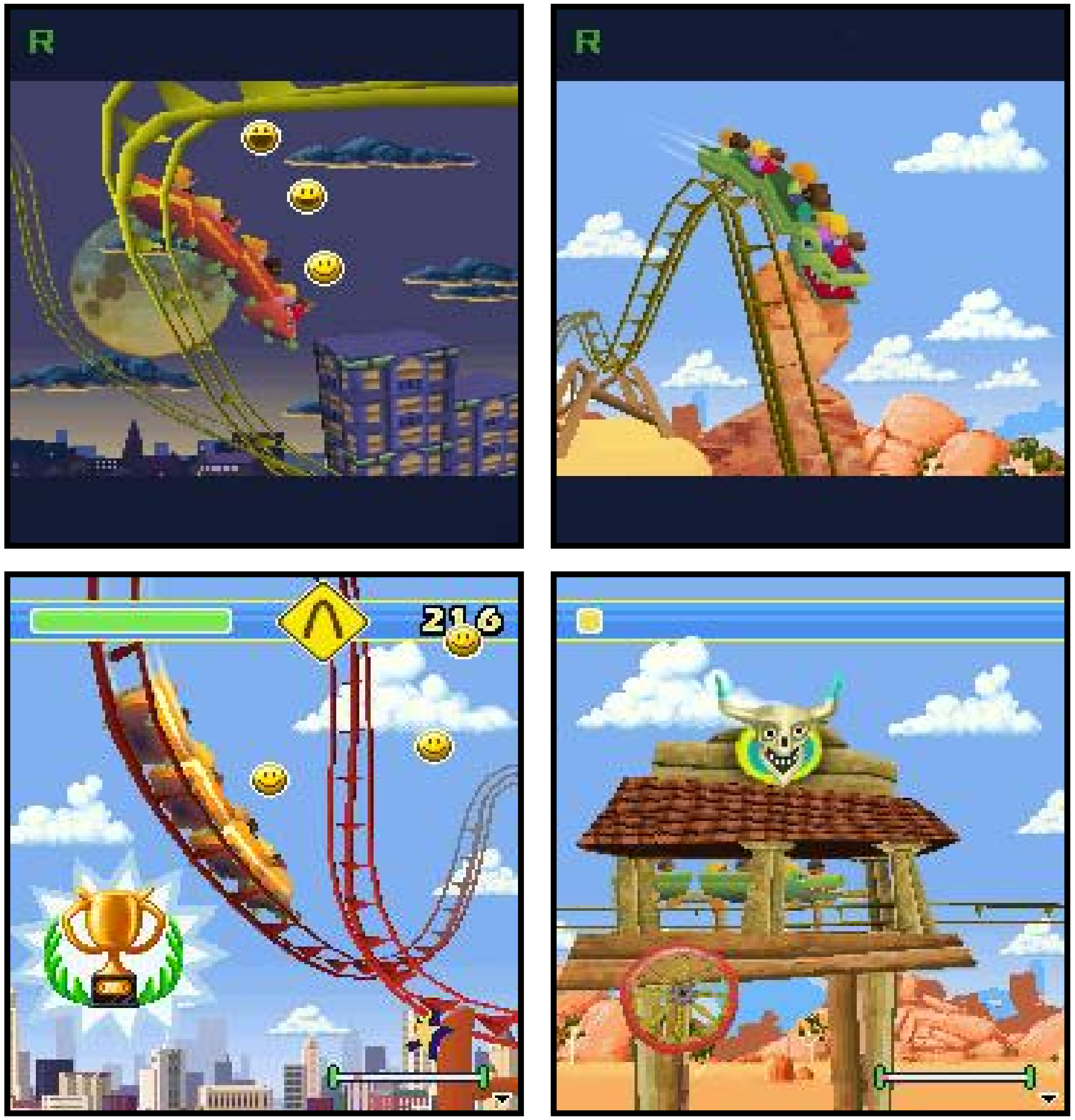






Rollercoaster Rush – Digital Chocolate

- 2D backgrounds
- 3D main scene
- 2D overlays



M3G 2.0

M3G 2.0

- Supercedes M3G 1.1
  - Adds programmable shaders in the high end
  - Improved features & perf also in the low end
  - Fully backwards compatible

- Work in progress
  - Get the Proposed Final Draft at www.jcp.org → JSR 297
  - Developer feedback can still make a difference!

NOKIA

Design Goals

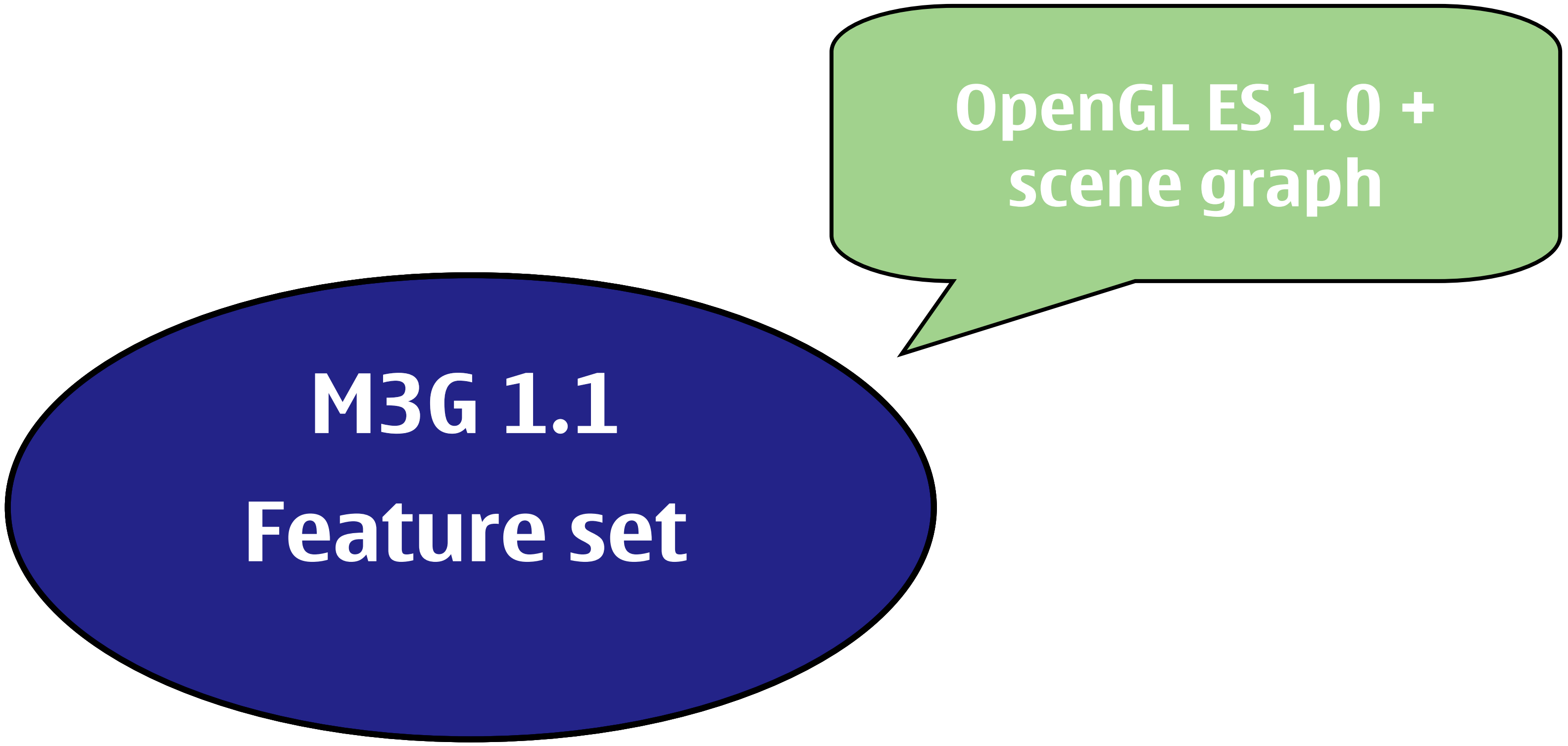
Target <u>all</u> devices

1. Programmable HW
2. No graphics HW
3. Fixed-function HW

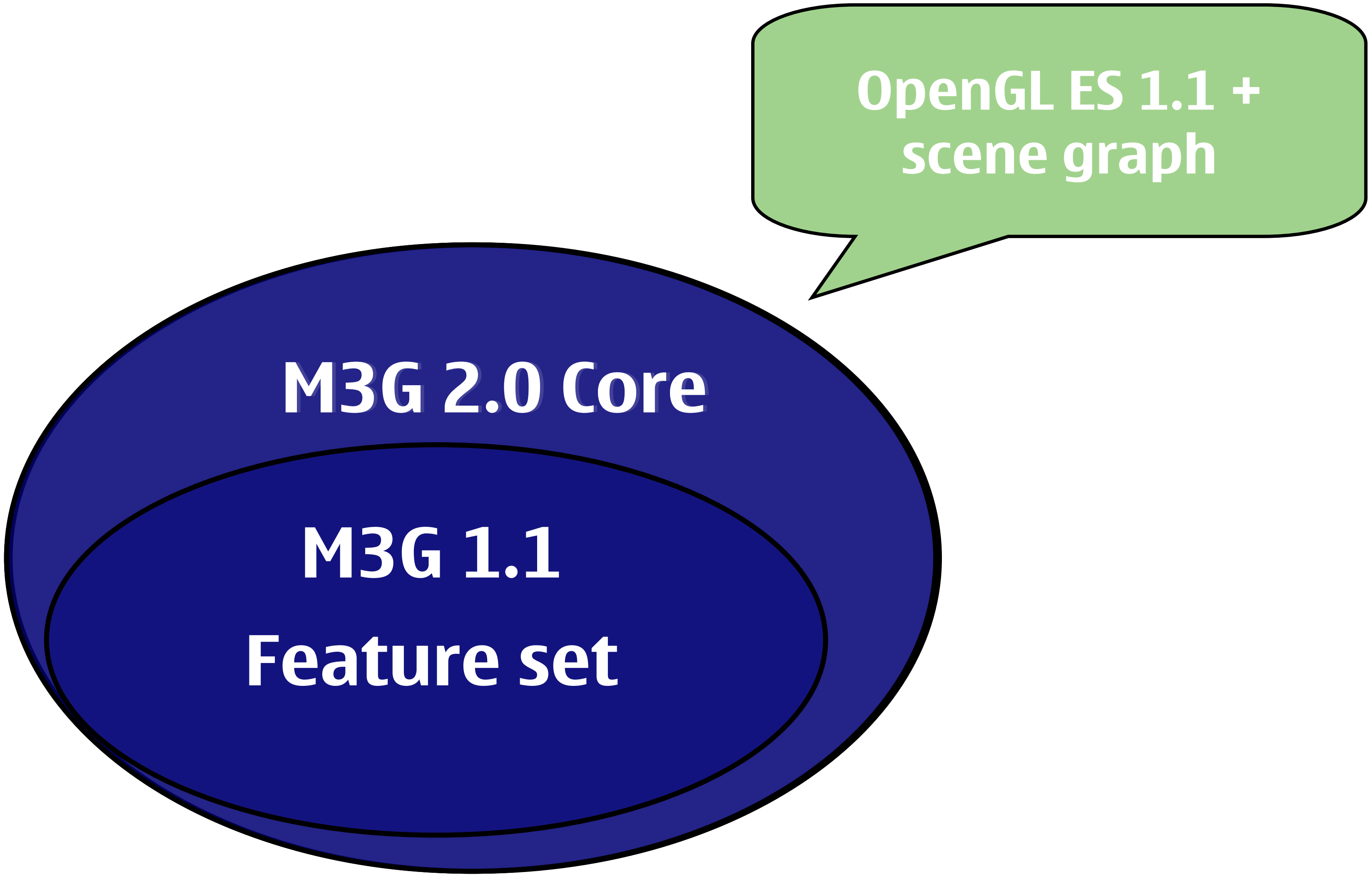
Enable reuse of

1. Assets & tools (.m3g)
2. Source code (.java)
3. Binary code (.class)

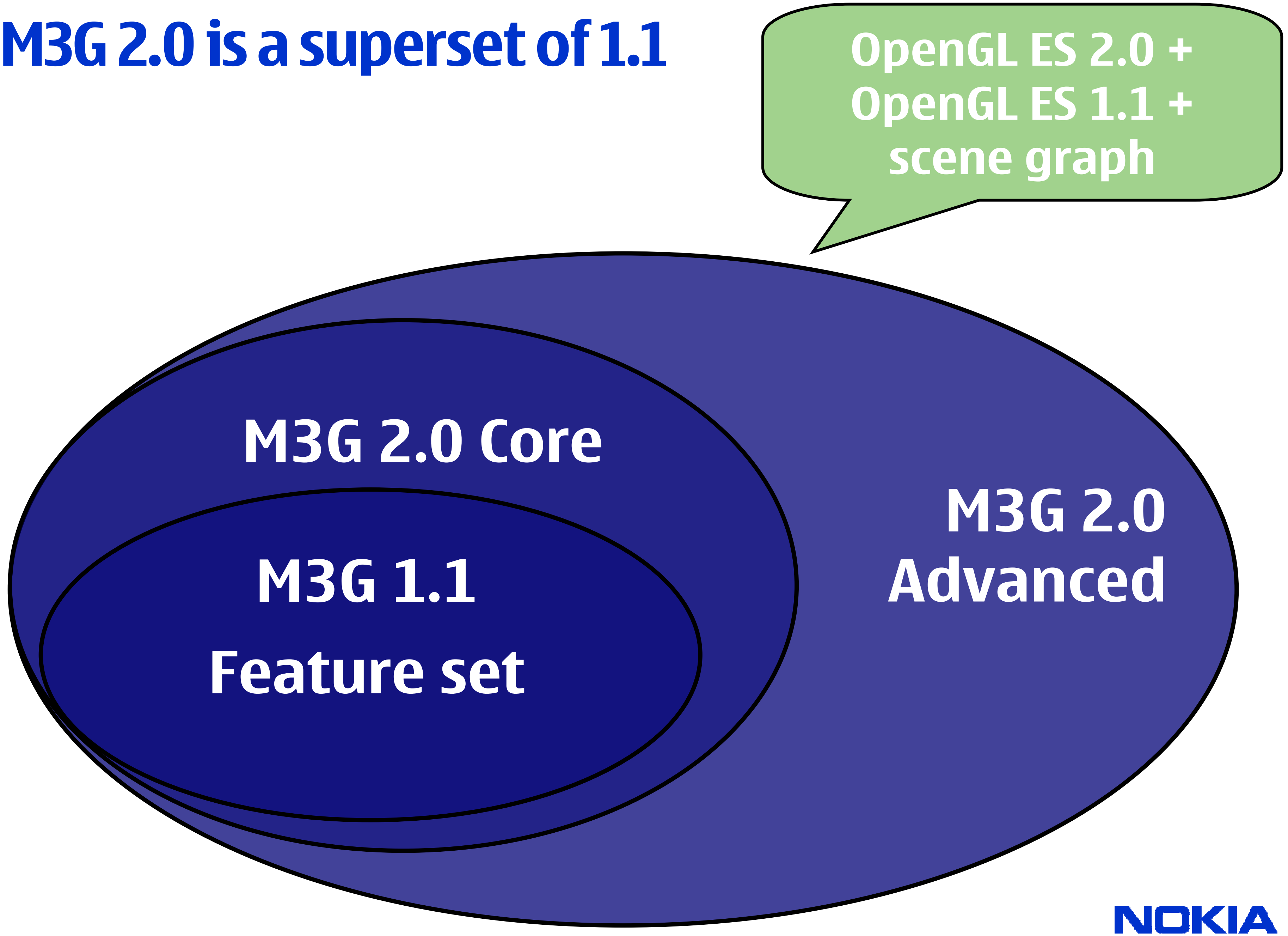
M3G 2.0 is a superset of 1.1



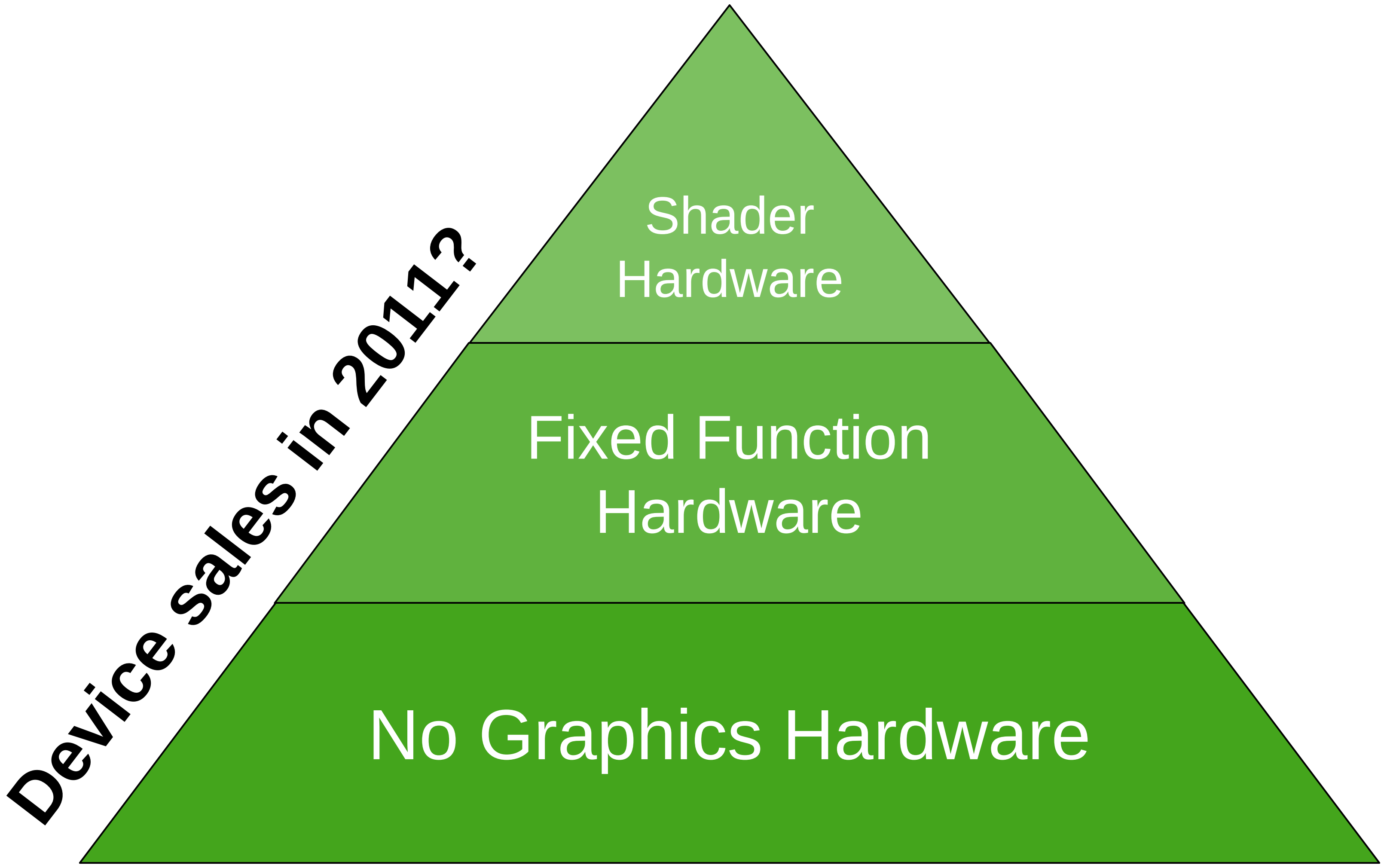
M3G 2.0 is a superset of 1.1



M3G 2.0 is a superset of 1.1



Why Not Shaders Only?



New Core features due to popular demand

- Optimized mesh deformation & animation
  - Morphing and skinning on the same mesh
  - Morph targets applied on a subset of the base mesh
  - Multichannel keyframe sequences
  - Animation event tracks
- Scene graph
  - Bounding volume hierarchies (boxes and spheres)
  - Neatly encapsulated multipass render-to-texture effects
  - Transparent objects can be sorted back-to-front
  - Lots of convenience methods

NOKIA

New Core features due to popular demand

- Improved texturing
  - Compressed textures, JPEG
  - Non-power-of-two sizes
  - Video textures
  - Bump mapping

- New primitive types
  - Point sprites, lines
  - Float/half vertices

Level of Detail (LOD)

- A Group node can select one of its children
  - Based on their size in screen pixels
  - Similar to mipmap level selection

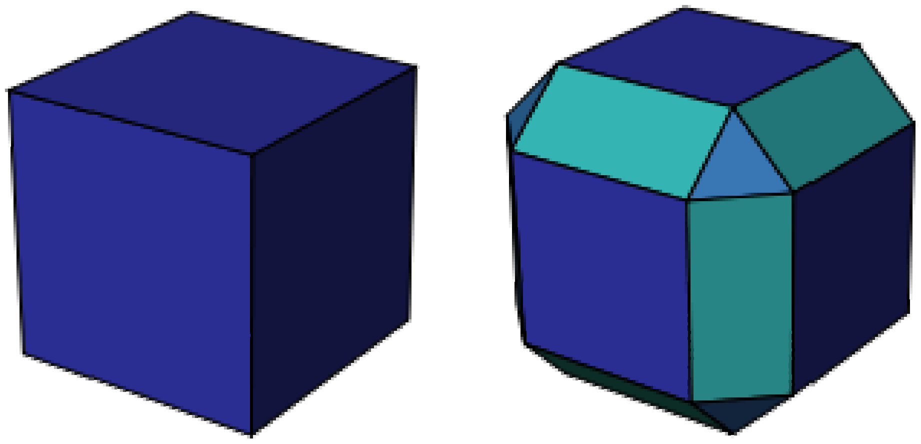
- Formally
  1. Compute ***s*** = pixels per model-space unit
  2. Select the node whose ideal scale $\boldsymbol{s_i}$ satisfies

$$\max\{ s_i \mid s_i \leq s \}$$

Collision Detection

- Each Node can have a collision volume
  - k-DOP = Discrete Oriented Polytope
  - AABB with corners & edges chopped off

- `world.collide(…)` to find all collisions



Simple vertex shader

```
#pragma M3Gpositionattrib(myVertex)
#pragma M3Gvertexstage(clipspace)

void main() {
  m3g_ffunction();
  gl_Position = myVertex;
}
```

Declare attribute semantics via #pragmas

Built-in function for morphing, skinning, model-view-projection

Result passed to the fragment shader

Summary

Summary

- M3G enables real-time 3D on mass-market phones
  - Easy to use, high performance scene graph API
  - Installed base somewhere between 500M-1B
  - Grab the tools and start developing!

- M3G 2.0 is under development
  - Adds programmable shaders in the high end
  - Improved features & perf also in the low end
  - Fully backwards compatible

M3G – Java Mobile 3D

Tomi Aarnio

Nokia Research Center