

High-level 3D programming on Mac OS X using Coin and Sc2I

Marius Kintel ~ Karin Kosina



SYSTEMS IN MOTION

Sc21?



- 🔍 a soft, silvery-white metal
- 🔍 atomic number: 21
- 🔍 relative atomic mass: 44.96

Sc21




- a Mac OS X framework
- Objective-C++
- integration of Coin with Cocoa

Coin



- another Mac OS X framework
- C++
- a high-level 3D graphics library

Overview

-  Introduction
-  Sc21
-  Demo
-  Coin
-  Q & A

High-Level 3D Graphics

- Hierarchical structure (“scenegraph”)
 - abstractions
 - automatic caching
 - run-time efficiency
- Object-based instead of vertex-based
- Data-driven vs. application-driven

Coin Features



- data-driven scenegraph library
- uses OpenGL for rendering
- fully Open Inventor 2.1 compatible
- multi-platform (Mac OS X, Windows, GNU/Linux, other UNIX systems)
- VRML97 support
- Free Software

Coin vs. Open Inventor

- Open Inventor API:
 - designed by SGI
 - last version: Open Inventor 2.1
- Open Inventor implementations:
 - SGI's original Open Inventor
 - Open Inventor from TGS
 - Coin by SIM



Other High-Level 3D Toolkits

 Among others:

 Java3D

 OpenGL Performer

 VTK

 and more...

 None of them is well-integrated with Mac OS X!

Add-ons and Related Libraries

 SIM Voleon (volume rendering)

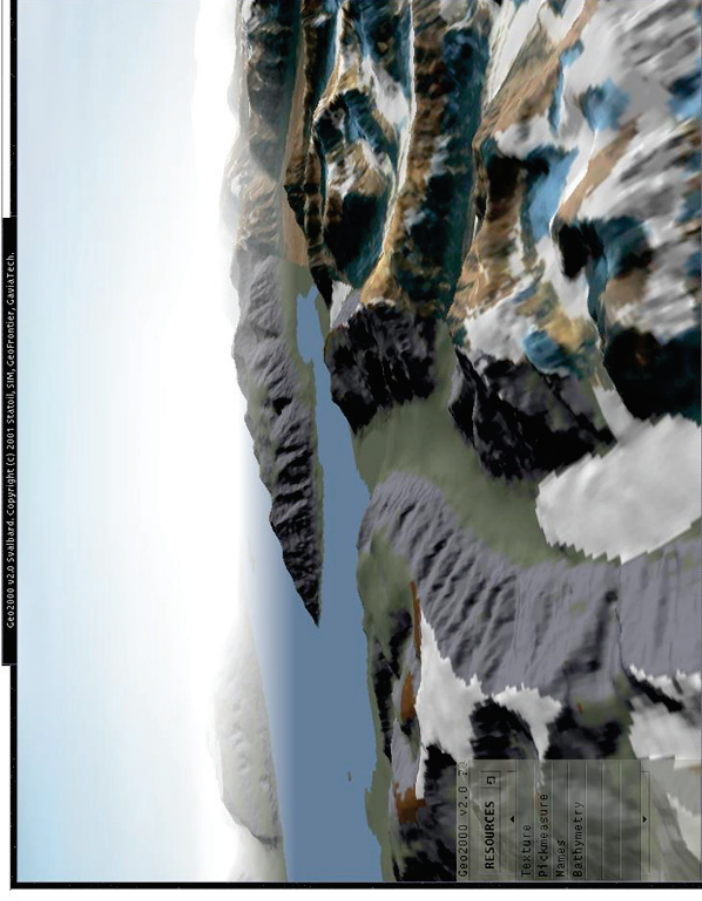
 SIM Scenery (terrain rendering)

 SoGUI libraries (GUI toolkit integration)

 SoQt (Qt)

 SoWin (Windows)

 SoXt (Motif)



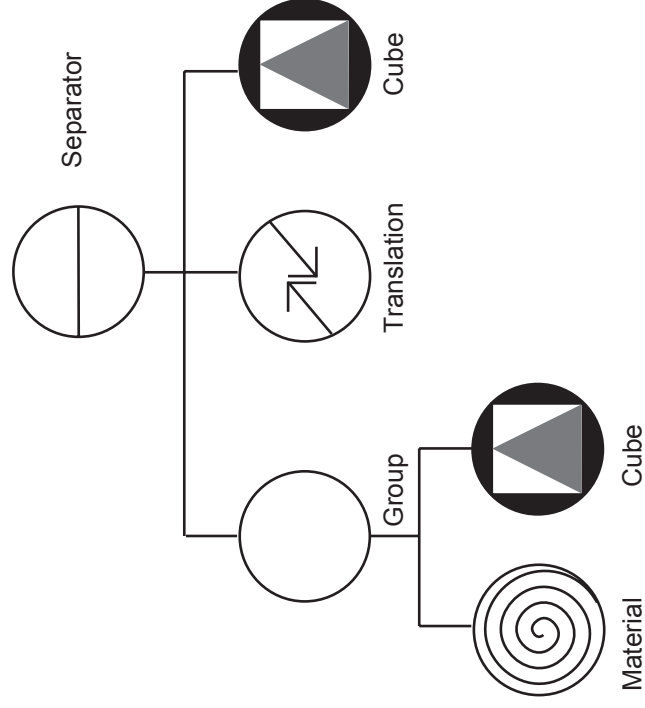
A Small Survey...

- Who of you is familiar with Open Inventor?
 - ... Coin?
 - ... other high-level 3D graphics toolkits?
- What language do you mostly use...?
 - What language would you like to use...?

Coin 101

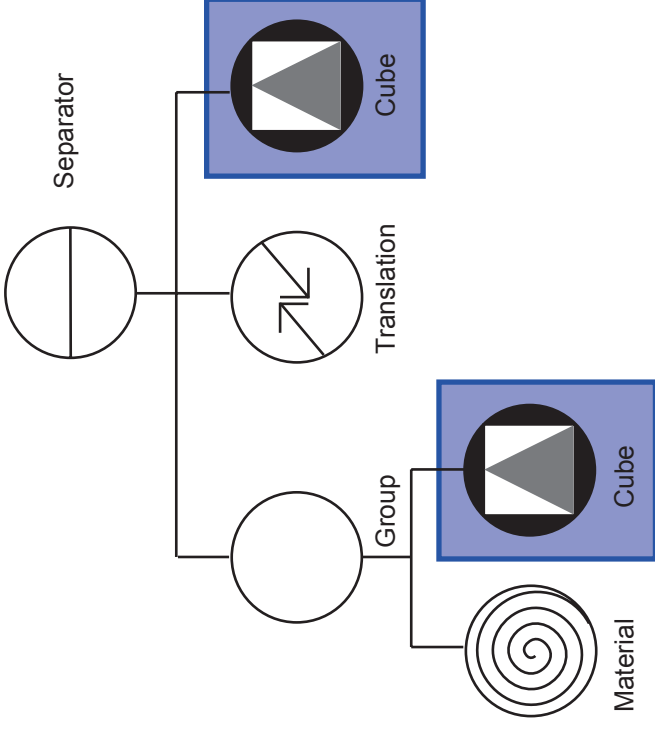
- A high-level introduction to Coin
- More on Coin programming later!
- Describing the scene:
Nodes in the scenegraph
- Rendering the scene:
Applying actions to the scenegraph

Node Types Overview



- Shape nodes (objects in the scene)
- Property nodes (OpenGL state)
- Group nodes

Node Types Overview

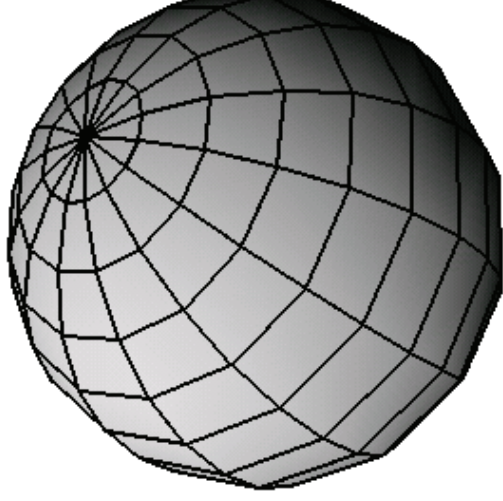


- Shape nodes
(objects in the scene)
- Property nodes
(OpenGL state)
- Group nodes

Node Types Overview

Shape Nodes

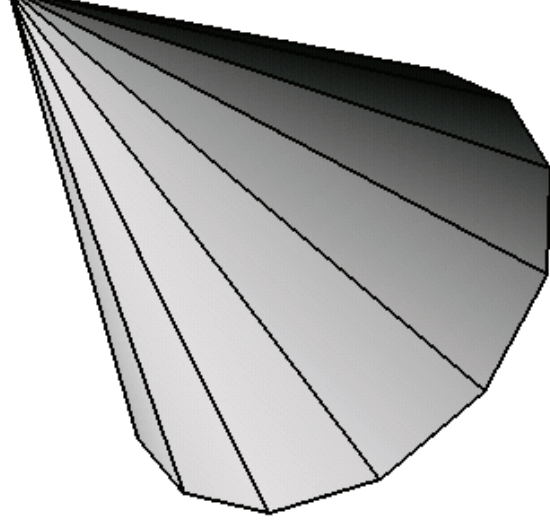
- Basic shapes
 - sphere, cone, cylinder, &c
- general shapes
 - facetset, lineset, pointset, &c.
- special shapes
 - text, nurbs, &c.



Node Types Overview

Shape Nodes

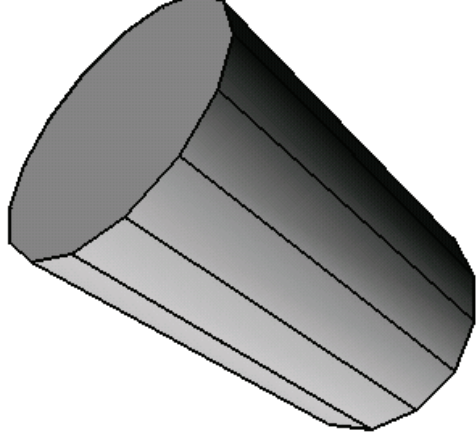
- Basic shapes
 - sphere, cone, cylinder, &c
- general shapes
 - facetset, lineset, pointset, &c.
- special shapes
 - text, nurbs, &c.



Node Types Overview

Shape Nodes

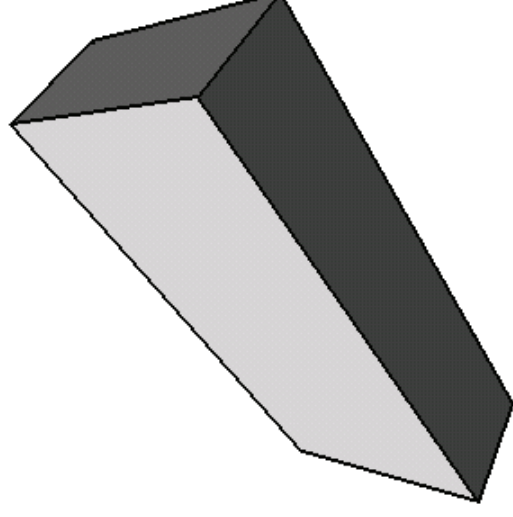
- Basic shapes
 - sphere, cone, cylinder, &c
- general shapes
 - facetset, lineset, pointset, &c.
- special shapes
 - text, nurbs, &c.



Node Types Overview

Shape Nodes

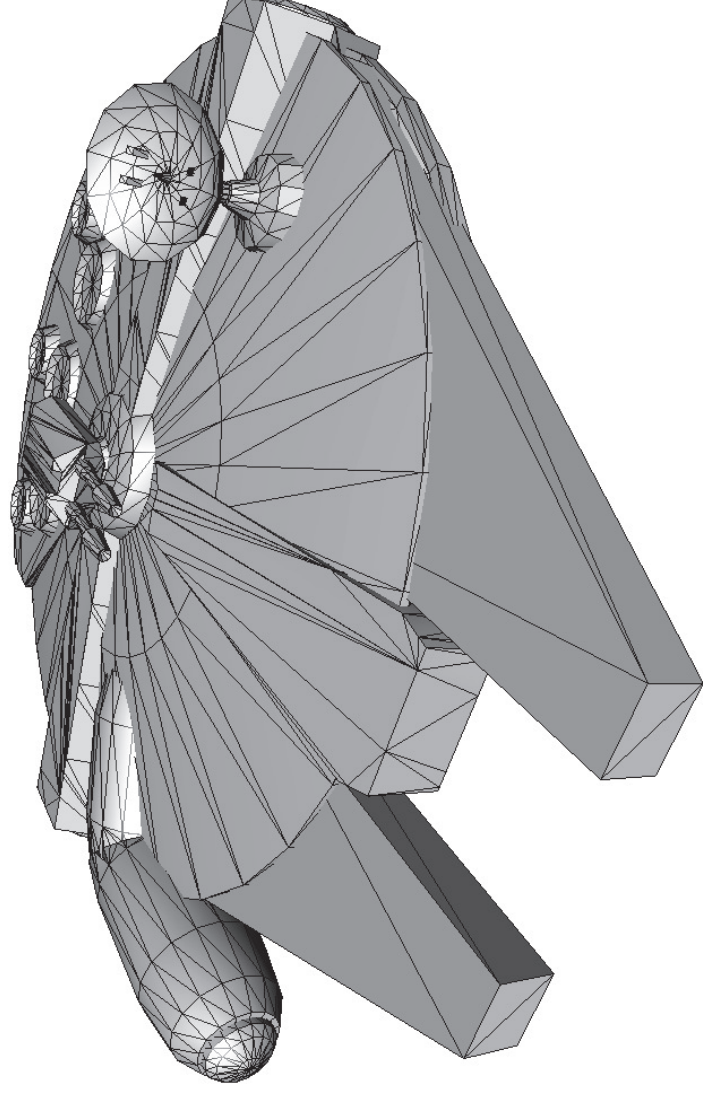
- Basic shapes
 - sphere, cone, cylinder, &c
- general shapes
 - facetset, lineset, pointset, &c.
- special shapes
 - text, nurbs, &c.



Node Types Overview

Shape Nodes

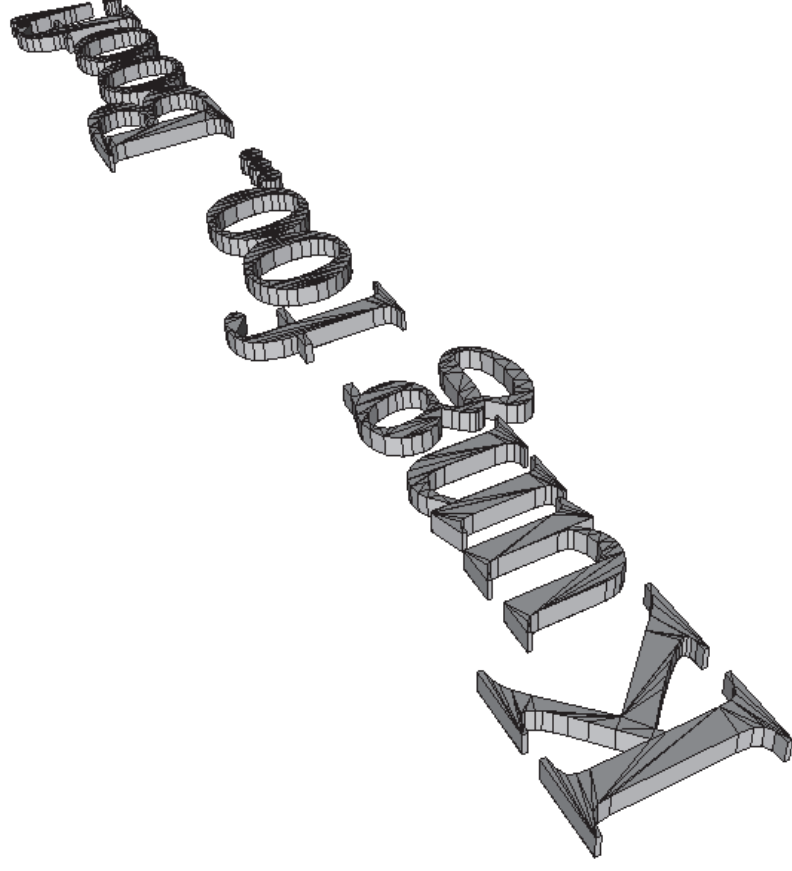
- Basic shapes
 - sphere, cone, cylinder, &c
- general shapes
 - facetset, lineset, pointset, &c.
- special shapes
 - text, nurbs, &c.



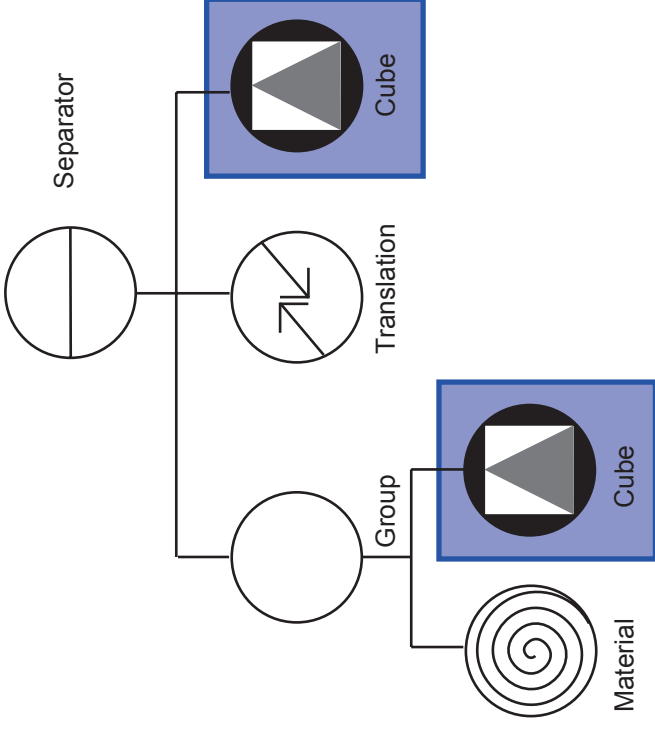
Node Types Overview

Shape Nodes

- Basic shapes
 - sphere, cone, cylinder, &c
- general shapes
 - facetset, lineset, pointset, &c.
- special shapes
 - text, nurbs, &c.

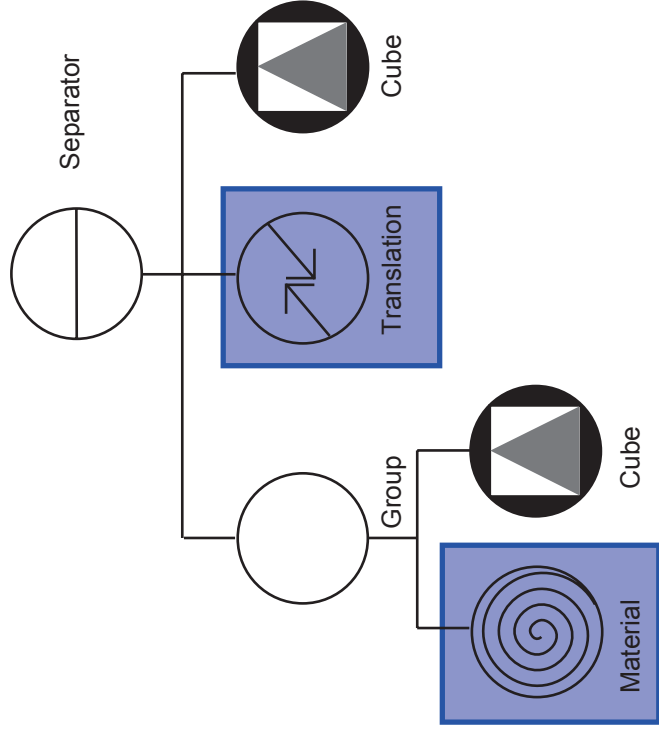


Node Types Overview



- Shape nodes
(objects in the scene)
- Property nodes
(OpenGL state)
- Group nodes

Node Types Overview



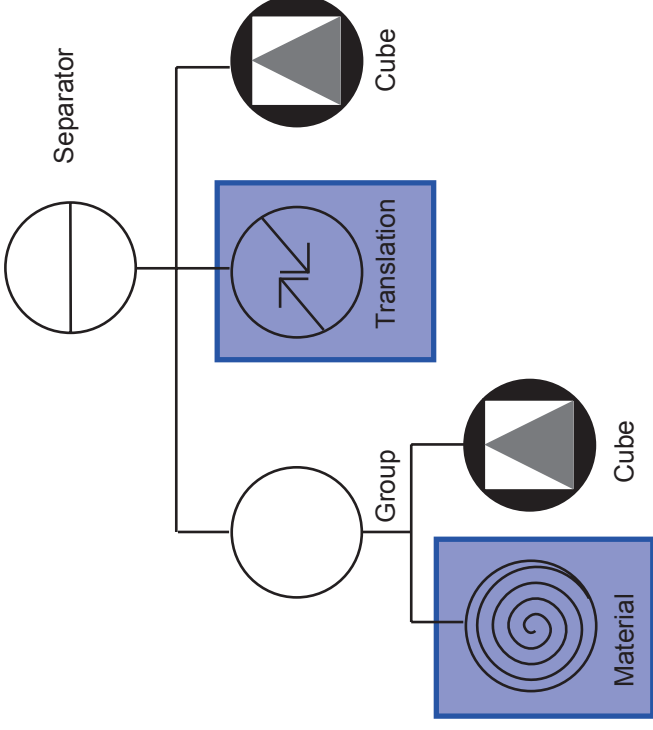
- Shape nodes (objects in the scene)
- Property nodes (OpenGL state)
- Group nodes

Node Types Overview

Property Nodes

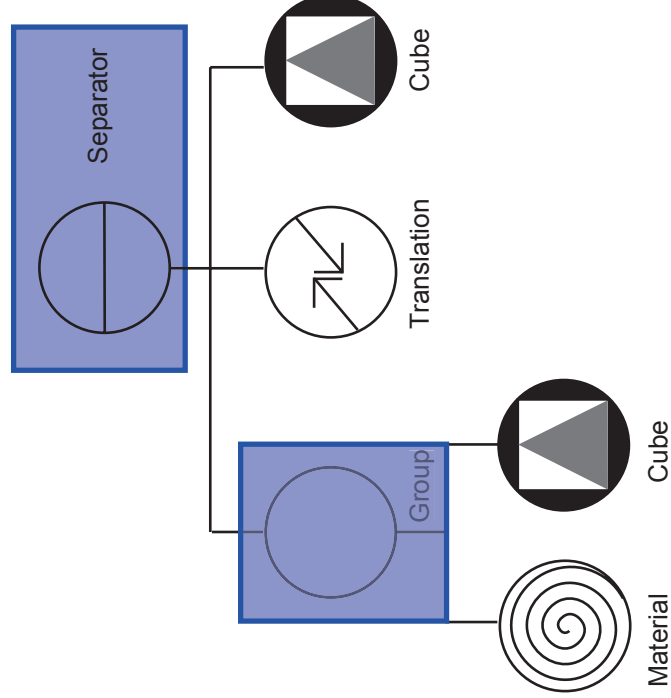
- OpenGL state
- materials, texture, lighting
- transformations
- Coin state
- geometric complexity
- font

Node Types Overview



- Shape nodes (objects in the scene)
- Property nodes (OpenGL state)
- Group nodes

Node Types Overview



- Shape nodes (objects in the scene)
- Property nodes (OpenGL state)
- Group nodes

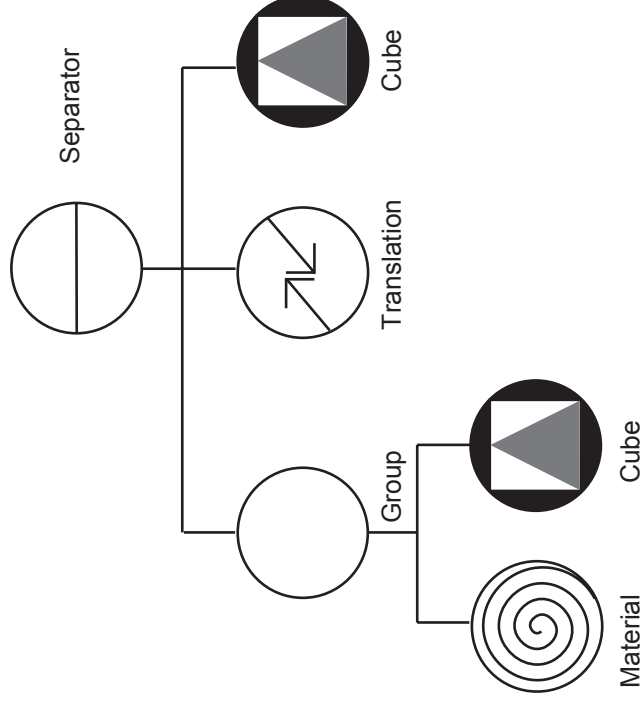
Node Types Overview

Group Nodes

- Containers to collect groups of objects (nodes) into subtrees
- Types:
 - SoGroup
 - SoSeparator
 - SoSwitch
- More later...

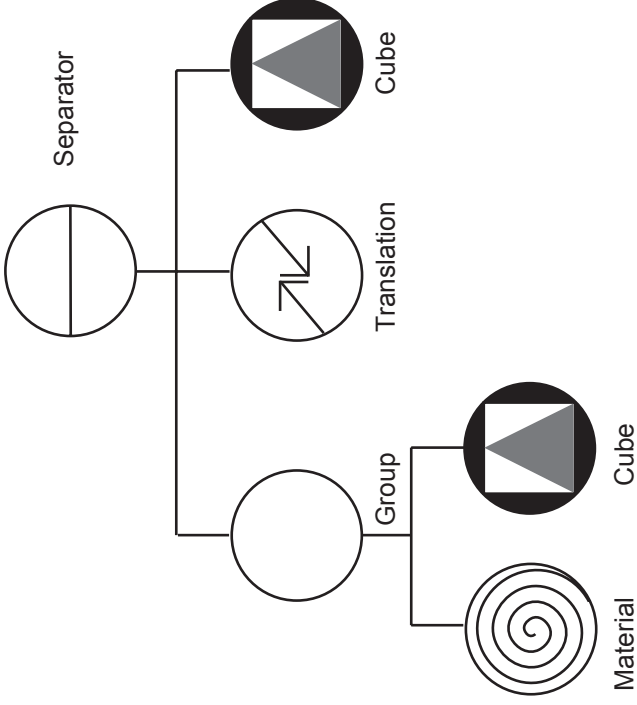
Scenegraph Description

- Separator {
 Group {
 Material {
 diffuseColor 1 0 0
 }
 Cube {
 }
 }
 Transform {
 translation 3 2 0
 }
 Cube {
 }
}



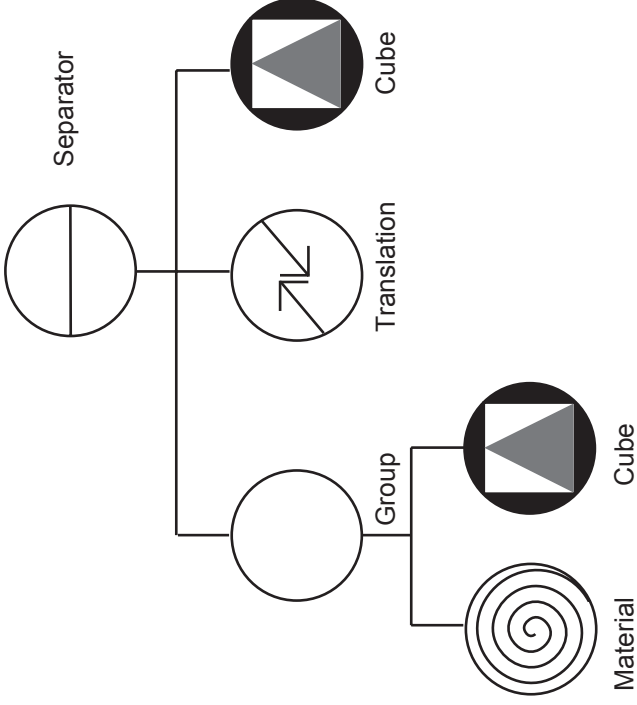
Actions

- Scenegraph is traversed from top to bottom and from left to right
- Each node can react to the action (behaviour depending on node type)
- Rendering the scene is an action

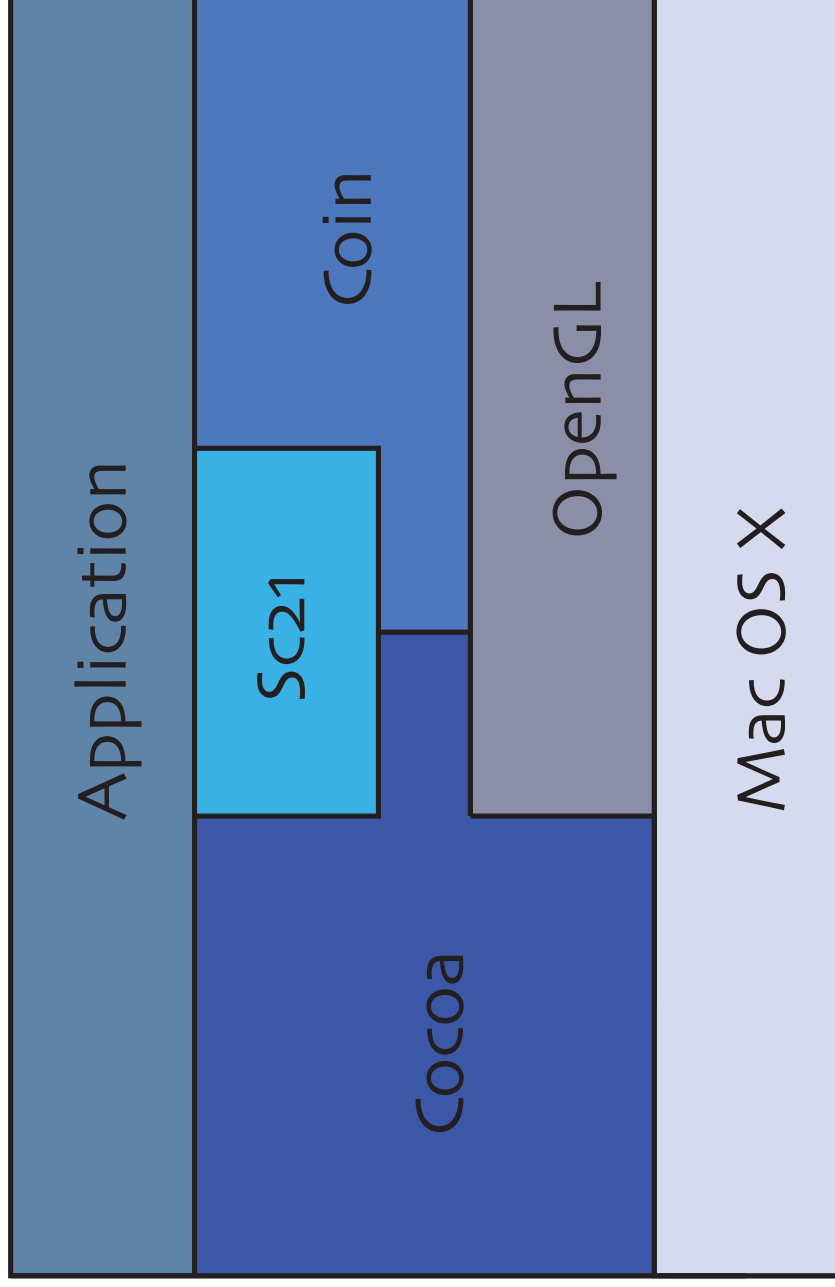


GLRenderAction Behaviours

- Group nodes traverse their children
- Shape nodes draw their geometry
- Property nodes set the OpenGL state
 - usually replacing the current state
 - exception: transformations (concatenated) and light sources (added)








Coin/Sc21 State Of The Union



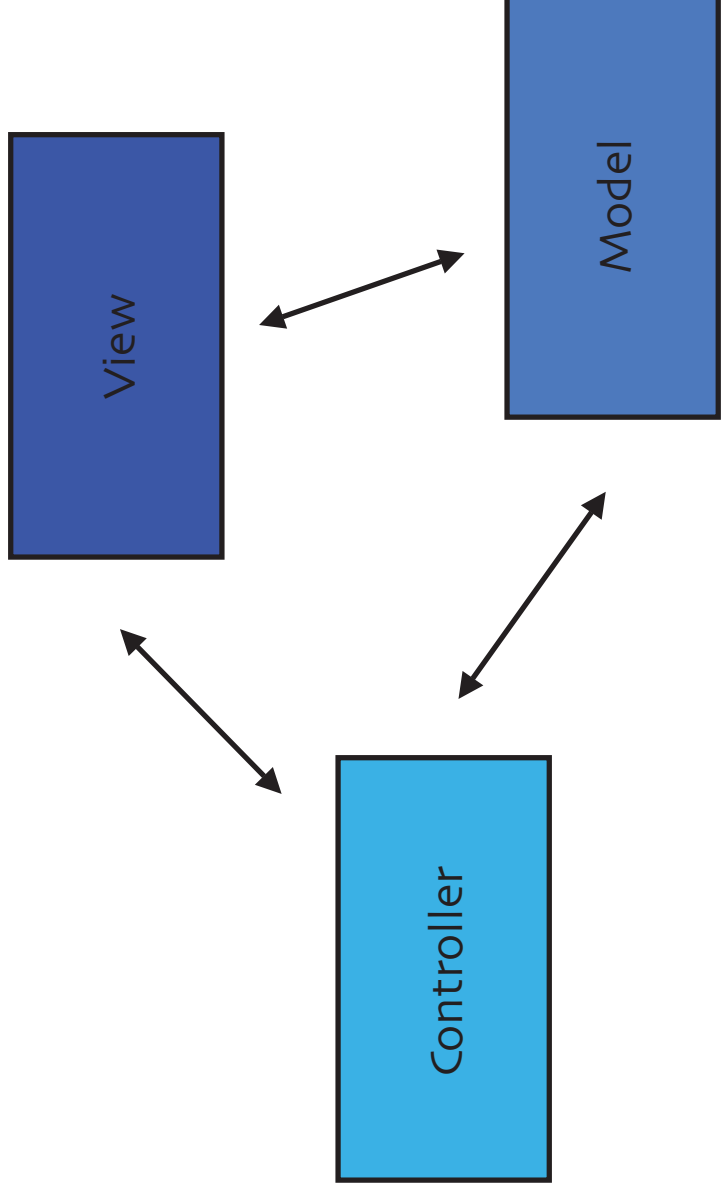
Cocoa in 2 Minutes :)

- Cocoa is one of the two basic frameworks on Mac OS X (together with Carbon)
- The language of Cocoa is Objective-C
 - Extension to C
 - very strongly run-time based
 - reflection, message-passing
 - what OO was meant to be :)

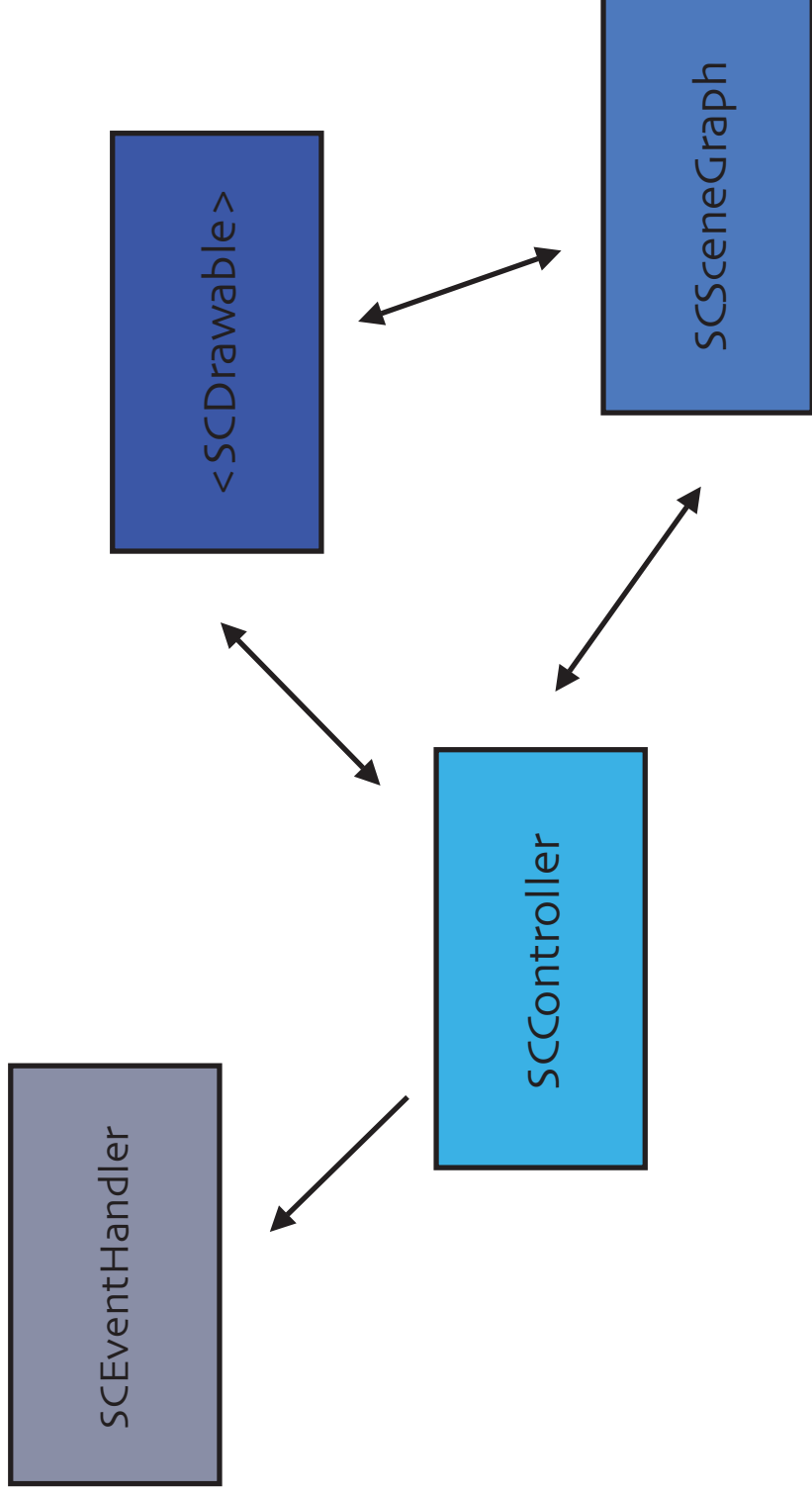
Cocoa in 2 Minutes :)

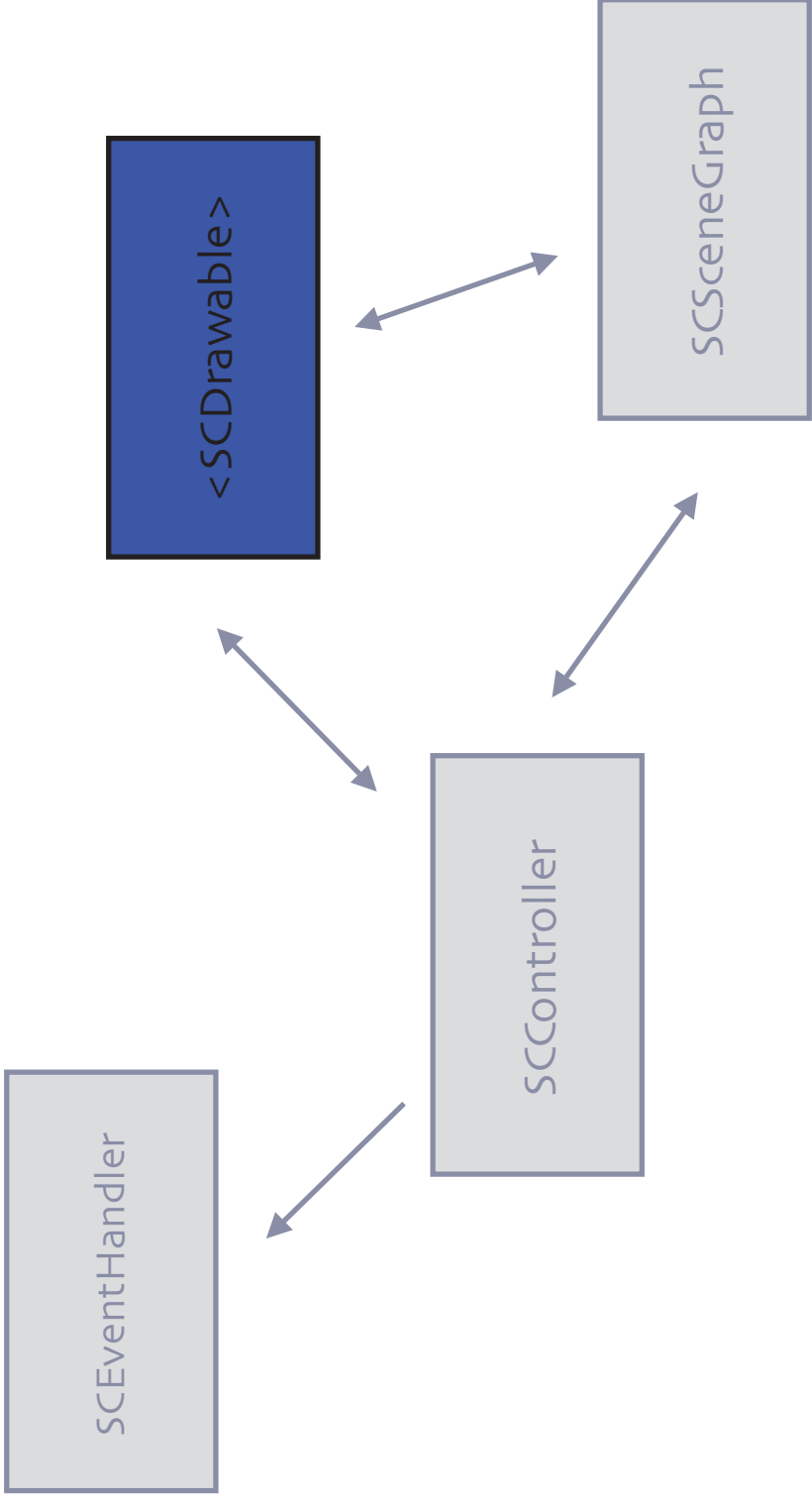
-  NIB files
-  serialized objects
-  graphical tool to build user interfaces:
Interface Builder
-  Foundation.framework
-  AppKit.framework

Sc2I Overview



Sc2I Overview





SCDrawable



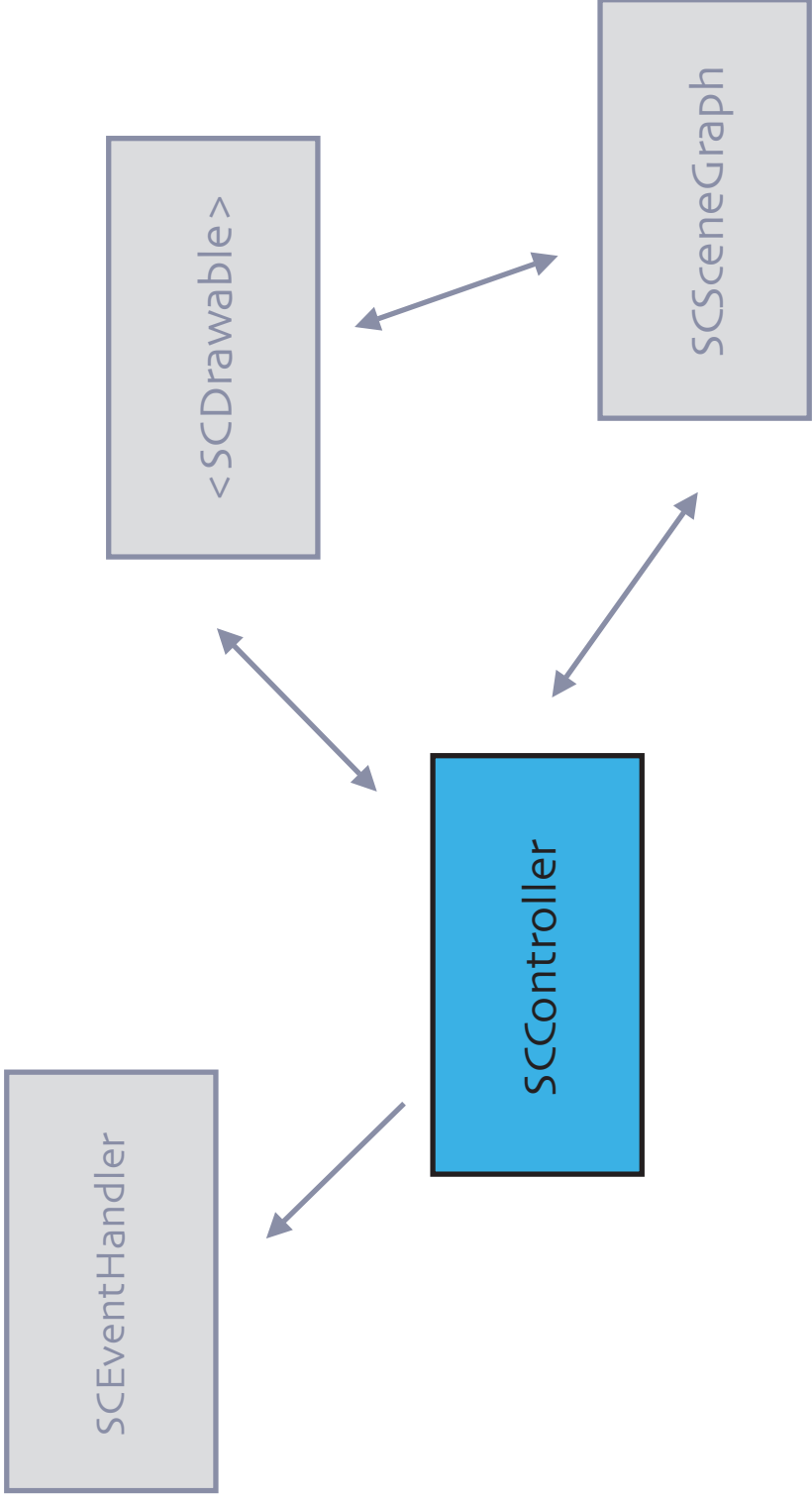
formal protocol

```
@protocol SCDrawable <NSObject>  
- (void)display;  
- (NSRect)frame;  
@end
```

SCDrawable



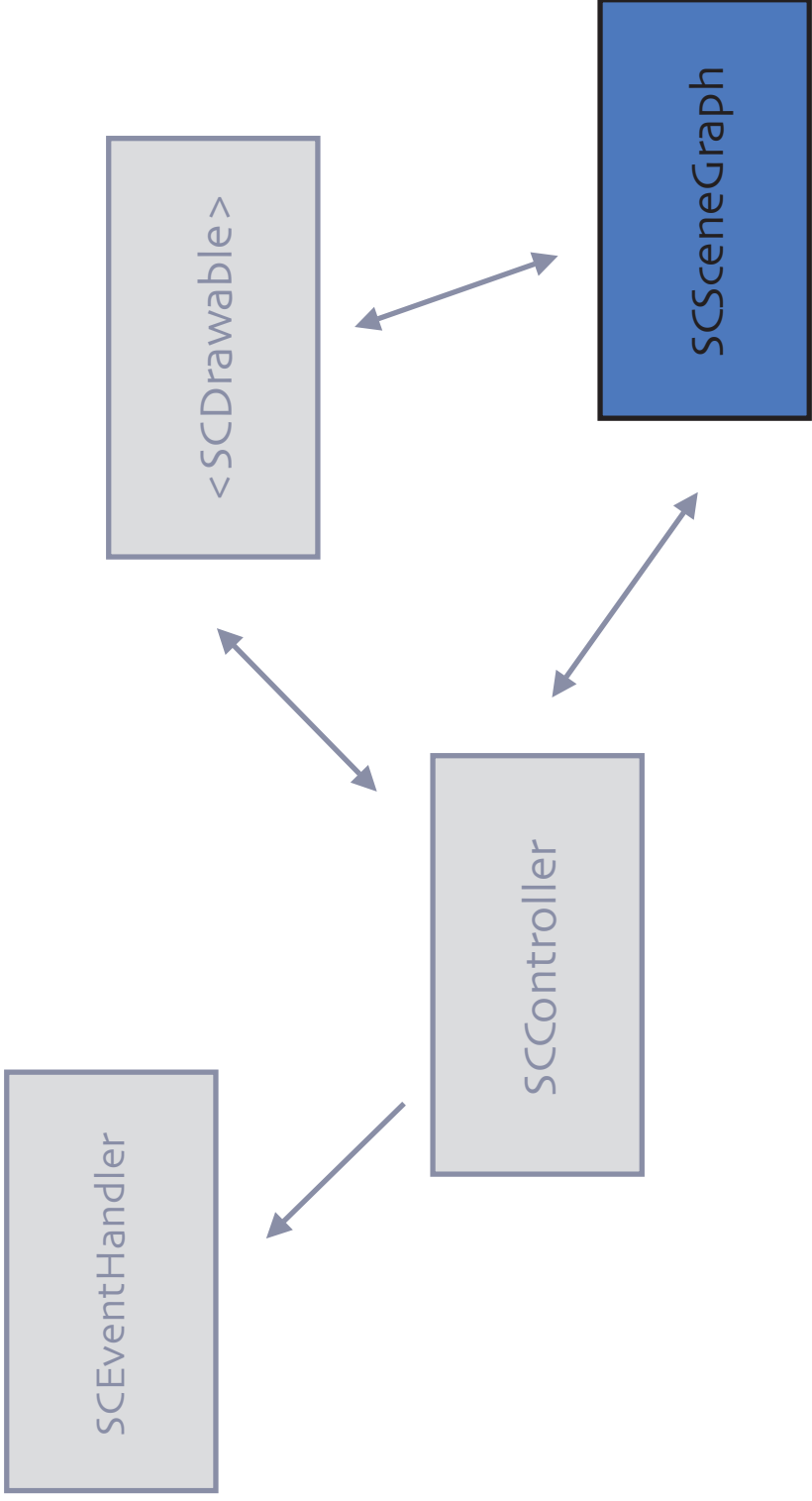
- SCView class
- implements <SCDrawable>
- behaves like NSView
- events forwarded to SCController::handleEvent
- Fullscreen rendering?
- just implement the protocol!



SCController



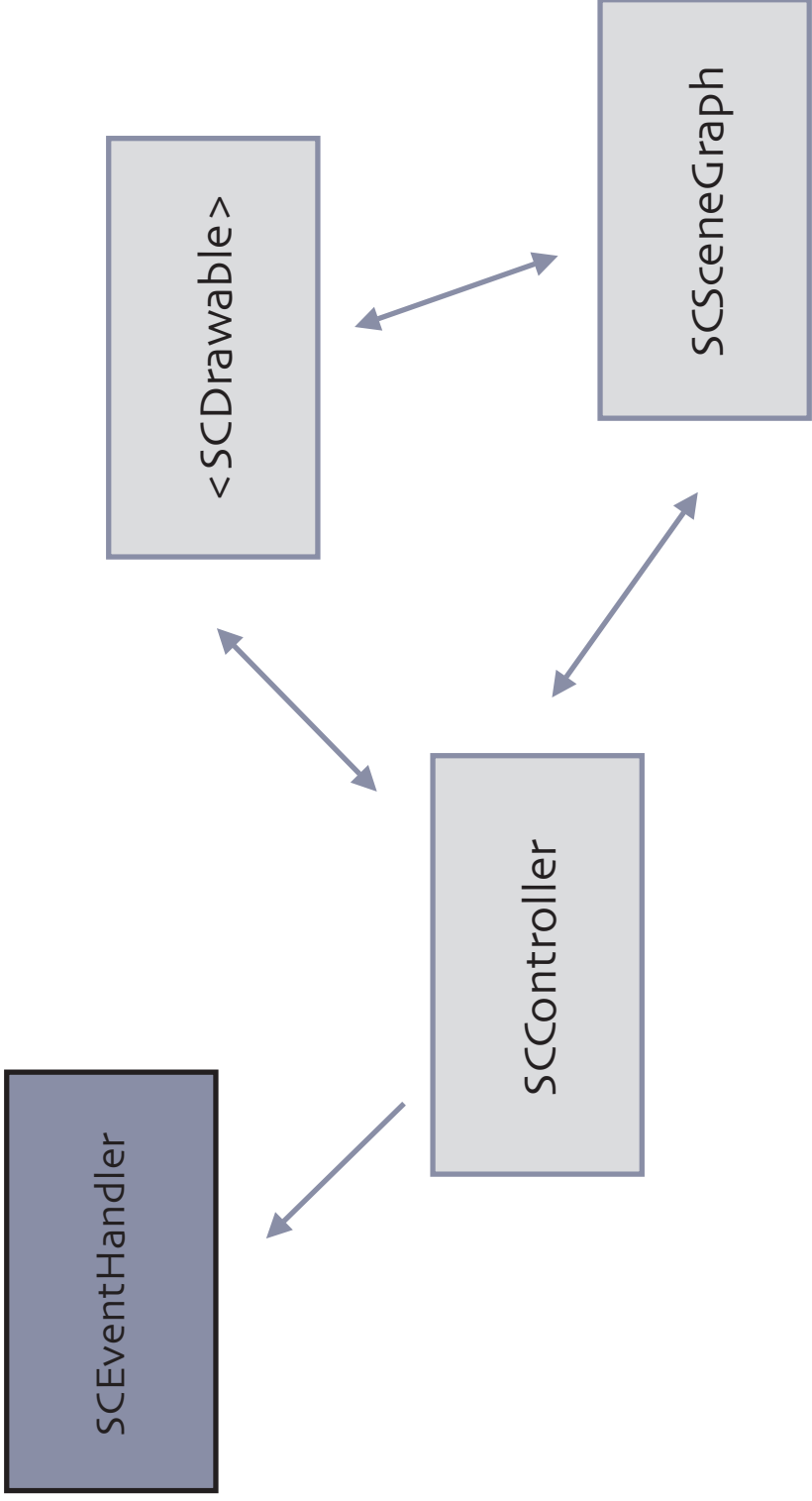
- Coin rendering
 - draws into its SCDrawable
- Event handling
 - SCView sends events to handleEvent:
 - SCController passes them on to its SEventHandler



SCSceneGraph



- Abstraction for the Coin scenegraph
- File I/O
- Superscenegraph creation
- checkbox in IB palette
- delegate for more control



SCEventHandler



- Will handle NSEvents occurring in our SCView
- Chain of event handlers
- Built-in event handlers:
 - SCExaminerHandler
 - SCCoinHandler

SCEventHandler



- Superclass for event handlers
- Two important methods:
 - (BOOL)controller:(SCController *)controller
handleEvent:(NSEvent *)event;
 - (void)controller:(SCController *)update;

Demo

[Sc21 release]

- 📌 You now know all there is to know about Sc21 :)
- 📌 Give it a spin!
- 📌 <http://www.coin3d.org/mac/Sc21>
- 📌 public beta of Sc21
- 📌 sample code
- 📌 documentation



Licensing

- Part of Coin3D (Coin, Sc21, SoGUI libs)
- GPL for Free Software development
- PEL (Professional Edition License) for all other uses
 - developer license (no run-time fees, royalties, &c.)
 - full source code available
 - \$ 2K / developer and year
- Inquiries to <sales@sim.no>

Summary

Where to go from here...



Coin/Mac and Sc2I website

<http://www.coin3d.org/mac/>



API documentation

<http://doc.coin3d.org>



The Inventor Mentor

The Inventor Toolmaker



Coin3D Mailinglist:

coin-discuss@sim.no

Q & A

Systems in Motion

Booth # 932

<http://www.sim.no>

Karin Kosina
kyrah@sim.no

Marius Kintel
kintel@sim.no