

Welcome

To Advance through Presentation  
Use Page Up and Page Down Keys



99 | Worldwide  
Developers  
Conference



99 | Worldwide  
Developers  
Conference

# QuickTime for Java

William Stewart

Manager, Software Architect

QuickTime for Java

# QuickTime

- A multimedia “engine” and data format—Movie
- Cross-platform C API
- Support for a wide variety of media formats



# Java

- An object-oriented language
- A cross-platform run-time (Java VM)
- A cross-platform execution model
- Security model for downloadable execution



# QuickTime for Java

- Combines the benefits of both technologies
- QuickTime provides graphics, sound and media capabilities
- Java provides a cross-platform application framework and execution model



# Runtime Requirements

- QuickTime 3 or better
- Java 1.1 (or better) compliant VM
  - Mac OS Macintosh Runtime for Java (MRJ) 2.1
  - Win32 Java Runtime Environment (JRE)



# What Is It?

- A set of Java classes that represent most of the QuickTime API
- A framework that provides services to developers:
  - Java integration
  - Spaces and Controllers
  - Animation services



# QuickTime Applets

- Requires the use of an appropriate Java VM
- Win32:
  - Required to use Java Plug-in from Sun
- Mac OS:
  - Use MRJ Plug-in with Netscape, MRJ with Internet Explorer







99 | Worldwide  
Developers  
Conference

# Demo

## QuickTime Streaming Applet

# Creating the Movie

- Any QuickTime content can be presented :

```
myQTCanvas = new QTCanvas ();  
add (myQTCanvas);  
QTDrawable drawer =  
    QTFactory.makeDrawable (myURL);  
myQTCanvas.setClient (drawer, true);
```



# Capabilities

- For presentation of QuickTime
- For using QuickTime to create and edit movies
- Capturing media using QuickTime capabilities
- Used for movie authoring, applets, image processing and music tools



# Presentation

- Provides a runtime environment for the presentation of QuickTime content
- Allows applets to provide more complex interactions with QuickTime movies





99 | Worldwide  
Developers  
Conference

# Demo

Movie Presentation  
and Callbacks

# QT VR Callbacks (1)

- Installing Callbacks to interact with nodes:

```
Track vrTrack = myMovie.getQTVRTrack (1);  
QTVRInstance vr =  
    new QTVRInstance (vrTrack, myMC);
```

```
vr.setEnteringNodeProc (new EnteringNode(), 0);  
vr.setLeavingNodeProc (new LeavingNode(), 0);  
vr.setMouseOverHotSpotProc (new HotSpot(), 0);
```



# QT VR Callbacks (2)

- Installing Callbacks to interact with hot spots:

```
class EnteringNode implements  
QTVREnteringNode {  
    public int execute (QTVRInstance vr, int nodeID) {  
        //... do enter node actions  
        return 0;  
    }  
}
```



# Spaces and Controllers

- Provides an architecture that uses services of QuickTime
- A runtime presentation of both groups of QuickTime objects and an animation engine





# QTDisplaySpace

- Spaces that are concentrated in the presentation of media
- DirectGroup
  - Allows groups of QuickTime objects to be presented and controlled
- Compositor
  - An animation and compositing service





99 | Worldwide  
Developers  
Conference

# Demo

Composited Effects

# Constructing a Space

- Members of a space present their data  
`myGroup.addMember (myJavaText);`
- Controllers of a space control members  
`myController = new SWController  
    (myResponder, true);  
myGroup.addController (myController);`



# Animation

- QuickTime Sprites and SpriteWorlds
  - Provides a powerful compositing capability
- QuickTime TimeBases
  - Provides the capability to construct animations





99 | Worldwide  
Developers  
Conference

# Demo

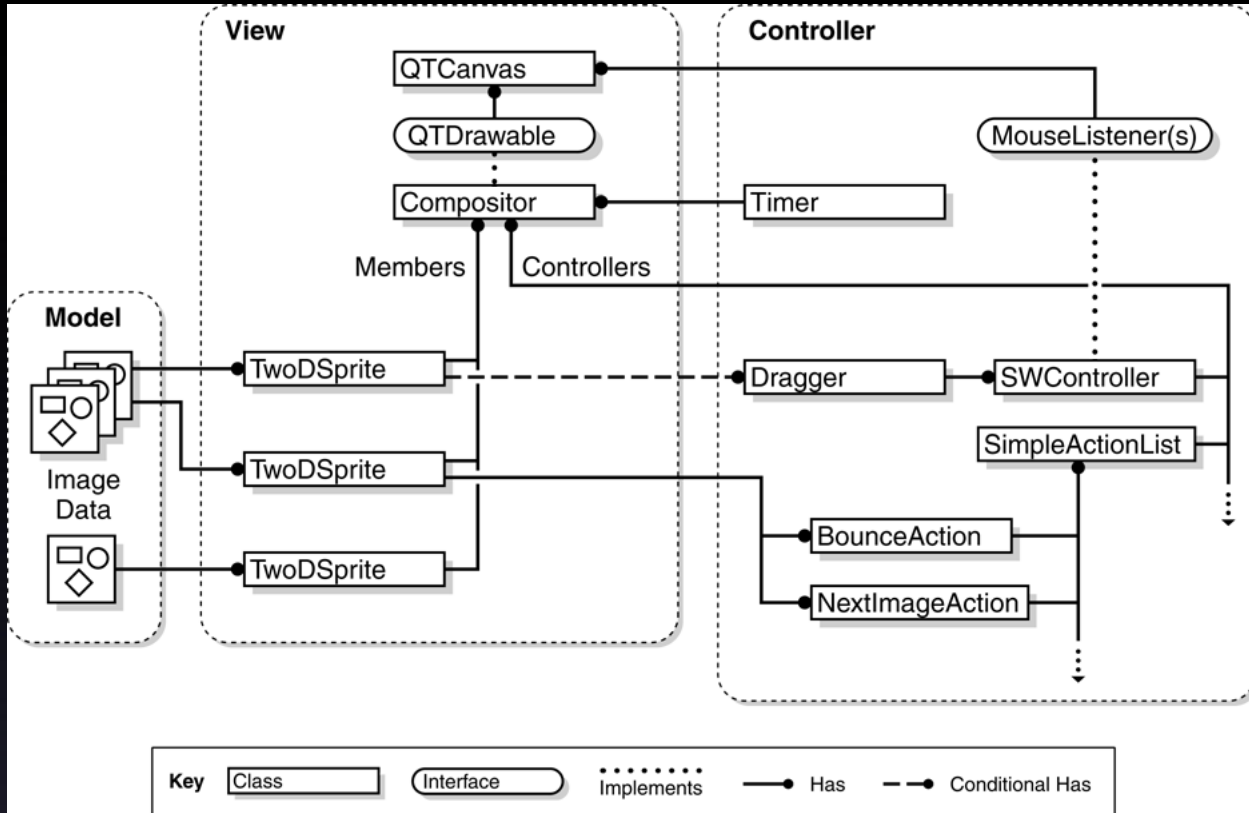
## DraggingSprites

# Model-View-Controller

- Separate the model (data) from its presentation and control aspects
  - Allows multiple views of the same data
  - Separates presentation from control
  - Data-hiding, abstraction and code reuse
  - Improve both generality and maintainability of classes



# MVC Diagram



# Custom Controllers

- Applications can define customised controllers for interacting with Movies
- Several examples in the QuickTime for Java SDK







99 | Worldwide  
Developers  
Conference

# Demo

## KeyController

# KeyController

- Add controller same as preceding
- Controller will be “wired-up” to Java event model by the Space

- Application just defines the actions:

```
public void keyPressed (KeyEvent e) {  
    switch (e.getKeyCode()) {  
        case KeyEvent.VK_UP:  
            player.setTime (player.getDuration());  
            ...  
    }
```





99 | Worldwide  
Developers  
Conference

# DynaMap

John Burkey

# DynaMap (1)

- Uses public-domain data bases as the source of the map
- Generates pixel data based on this terrain data
- QuickTime services used for rendering terrain
- Provides multiple views of same model data



# DynaMap (2)

- QuickTime also used for UI elements
- Use of controllers to drag UI elements
- Enables applications a simple but powerful API to build complex UI



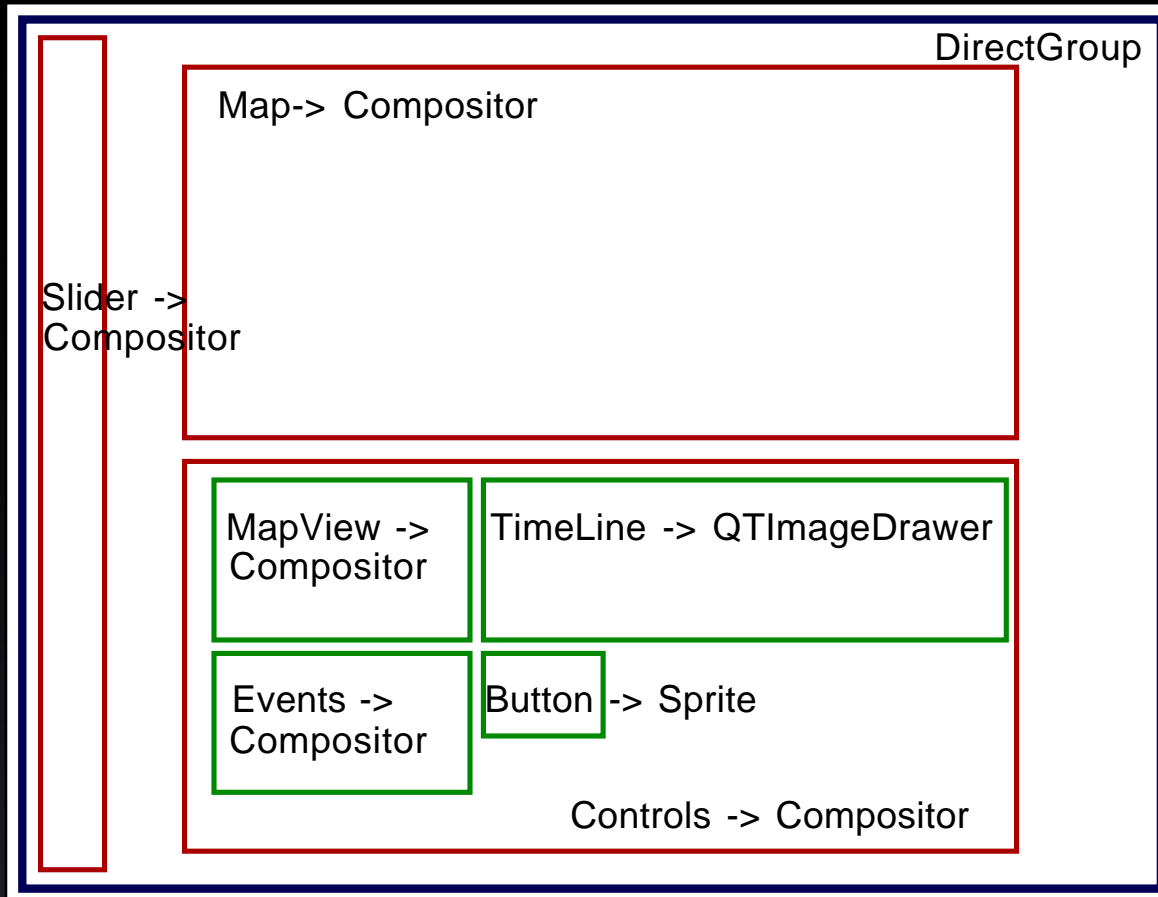


99 | Worldwide  
Developers  
Conference

# Demo

## DynaMap

# Map Spaces





# Q&A

<http://www.apple.com/quicktime/qtjava/>

Book:

QuickTime for Java  
Developer's Reference

JavaOne Gold Sponsors





Think different.<sup>TM</sup>



Welcome

To Advance through Presentation  
Use Page Up and Page Down Keys



99 | Worldwide  
Developers  
Conference