

The background features a dark, textured surface with a glowing blue and purple sphere in the center. A white Apple logo is positioned at the top of the sphere. The text "Worldwide Developers Conference" is overlaid on the image. The word "Worldwide" is in a gold, serif font. The word "Developers" is in a white, serif font and is enclosed in a white rectangular border. The word "Conference" is in a gold, serif font. The overall aesthetic is futuristic and tech-oriented.

Worldwide

Developers

Conference



Game Sprockets: Mac OS and Beyond!



Game Sprockets: Mac OS and Beyond!

Steven Glass

**Vice President,
Mac OS Engineering**



Game Sprockets Overview

Chris De Salvo

Technical Lead

What are Game Sprockets?

Two categories of technologies:

- Cool-ware
- Duh-ware



What is Cool-ware?

Stuff that adds neat new functionality

- **Hardware abstracted device input**
- **Dirty-rectangling**
- **Game-oriented networking**
- **3D sound**



What is Duh-ware?

Things we should have done a long time ago

- Good support for double-buffering
- An *easy* way to change screen resolution
- A way to hide the damn menu bar
- A *usable* network API
- Consistent joystick interface



The Components:

DrawSprocket



NetSprocket



InputSprocket



SoundSprocket



What's the Cost?

- **Get them for free**
- **Redistribute them for free**
- **Includes free license for dependent components**
 - **Sound Manager**
 - **Display Manager 2.0**





DrawSprocket: Graphics Context Management

Chris De Salvo

Technical Lead

DrawSprocket Features

- **Easy display mode changing**
- **Multiple monitor handling**
- **Gamma fading**
- **Double/triple buffering (HW & SW)**



DrawSprocket Features (*cont.*)

- **Dirty-rectangling, blitting, underlay support**
- **CLUT management**
- **Special goodies**



Display Mode Management

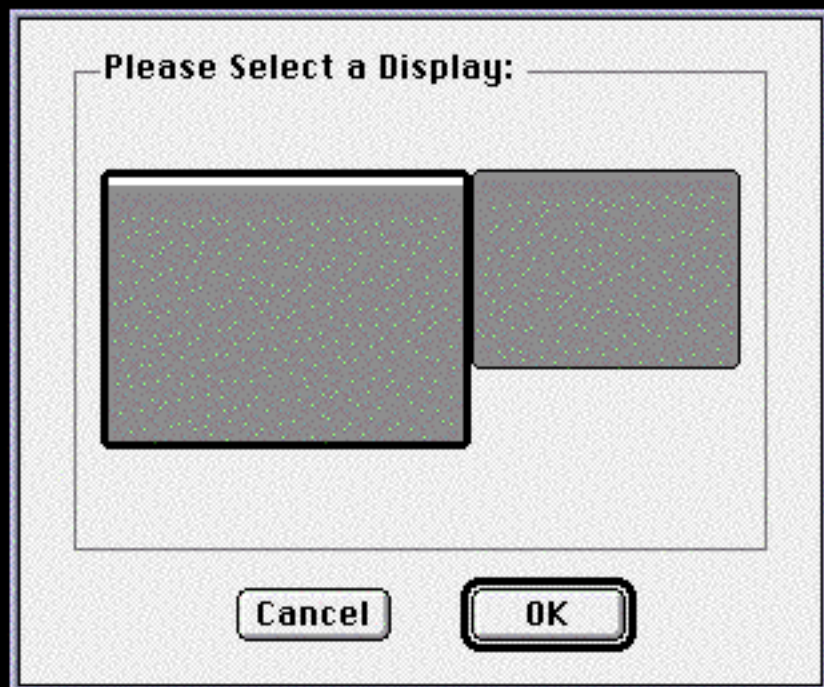
Find the best mode for your desired:

- Resolution
- Pixel depth
- Refresh rate
- Hardware options



Multiple Monitors

- **Create contexts on any monitor**
- **User interface so users can select a preferred monitor**



Gamma Fading

Uses color intensities, not RGB values

- **Fade to/from black or any other color**
- **Works with all devices, including direct devices**
- **Avoid uneven fading that occurs with indexed fades**
- **Fade any or all monitors at the same time**



Buffering/Blitting

Not hard, just tedious

- **Single, double and triple buffering**
- **Page flipping on supported hardware**
- **Blits that are color keyed, scaled, bilinearly interpolated**
- **Underlay compositing**
- **Dirty rectangling**



CLUT Management

“Give us a CSpecArray and we’ll do the rest!”

- **Get/set CLUT entries**
- **Automatically updates all back-buffers**



Special Extras

All the things you may not know how to do (properly)

- **Square the GrayRgn**
- **Leave Finder icons undisturbed**
- **Hide Control Strip and other floaters**



Special Extras (*cont.*)

- **Hide the menu bar**
- **Blitters are tuned for each piece of hardware**
- **Secret handshakes with some video drivers**



How a Game Uses DrawSprocket

Define, reserve, activate your draw context

```
while (playingAGame)
{
    Get the next back-buffer
    Render your frame
    Mark your invalid rectangles as dirty
    Swap in your new frame
}
```

Release and destroy your context



DrawSprocket Since 1.0

- **GoggleSprocket for stereoscopic viewing**
- **Custom blitters**
- **Custom AltBuffers**
- **Control over back-buffer scanline padding**
- **Keyboard short-cuts in UI**
- **Bug-fixes and performance enhancements**





Demo

Future Plans for DrawSprocket

- CFM-68k version!!!
- 2D hardware acceleration and effects
- Ability to switch between full-screen and windowed operation





InputSprocket: One-Stop Shop for Device Input

Chris De Salvo

Technical Lead

What InputSprocket Provides

- **Hardware abstraction layer for physical devices**
- **Software abstraction layer for non-physical input sources**
- **Consistent user interface for configuration**
- **Easy support for saving configuration sets**



Hardware Abstraction

What is currently supported

- Thrustmaster sticks, rudders and throttles
- CH Products game-pad, sticks, rudders and throttle
- Gravis sticks and game-pad
- Microsoft SideWinder 3D Pro



Hardware Abstraction (*cont.*)

What is currently supported

- **AppleJack (Pippin) controller**
- **100% Apple compatible mice**
- **Kensington mice and trackballs**
- **Keyboards**



Hardware Abstraction (*cont.*)

What else could you do?

- **Head-mounted display tracking**
- **Graphics tablets**
- **USB (actually, it already exists)**
- **Video jog/shuttle controls**



Hardware Abstraction (*cont.*)

What else could you do?

- **Audio mixing boards**
- **3D armatures**
- **Treadmills**
- **Just about anything else that you can physically connect to a Macintosh**



Software Abstraction

- **Speech recognition**
 - No client code changes!!
 - Recognized speech commands look just like button clicks



Software Abstraction (*cont.*)

What else could you do?

- Network controller
- Custom AppleEvent handler
- Input playback from a file



How a Game Uses InputSprocket

High-level versus low-level interface

- **High-level interface is “game-centric”**
 - Buttons for fire, jump
 - Axes for x, y and z
 - Utility keys
- **Low-level interface is “device-centric”**
 - References to individual input devices
 - Lists of physical elements on each device



“Typical” High-Level Usage

Define input needs, types and usages
Initialize high-level with your needs

```
while (playingAGame)
{
    poll current value of axes

    while (buttonEventsAvailable)
        get button events
}
```

Stop the high-level



Low-Level Usage

- **Get list of all devices**
- **Get list of elements for each device**
- **And then...**



Low-Level Usage (*cont.*)

... You have to do all the rest

- **User interface**
- **Configuration/saved sets**
- **Acceleration/non-linearity**
- **Chording/macros**
- **Data conversion**
- **Axis reversal**
- **Etc.**



InputSprocket Since 1.0

- **Greatly enhanced user interface**
- **Delta types**
- **Speech recognition support**
- **Drivers, drivers, drivers**
- **Bug fixes and performance enhancements**





Demo

Future Directions for InputSprocket

Some of these make great third-party opportunities...

- **Force-feedback**
- **Calibration**
- **USB**
- **More direct-to-device drivers**
- **Better keyboard driver user interface**
- **Graphics tablet usage types**





NetSprocket: Nothing But Net

Jamie Osborne

Lead Engineer

What NetSprocket Provides

Abstraction layer for client-server network services

- **It's a messaging system**
- **Handles all of the networking details for you**
- **Provides a default user interface for hosting and joining a game**
- **You don't have to know much about networking to use it**



Features

- **Easy-to-use API**
 - Centered around the game and its players
 - Add networking in a day
- **Support for AppleTalk and TCP/IP**
 - Use both in the same game
 - Open Transport-only
 - You don't need to know where a player is or what protocol they're speaking
- **Custom join request handlers**



Features *(cont.)*

- **Dynamic addition and deletion of players**
- **Per-message priority rating**
- **Synchronous or asynchronous operation**
- **Group support**
- **Client-server topology**



NetSprocket Since 1.0

- **CFM-68k version!!!**
- **Bug-fixes**
- **Improved human interface**
- **Auto-port selection for TCP/IP**



Other Networking Options

- **Open Transport**
 - Control of topology
 - Better memory management (bytes rather than messages)
- **NetManage WinSock**
 - Cross-platform API
 - Available with Metrowerks CodeWarrior
- **PowerPlant Network Classes**
 - Easy-to-use, but requires PowerPlant
 - TCP/IP-only





Demo



SoundSprocket: Here, There and Everywhere

Tim Carroll

Apple DTS

What SoundSprocket Provides

API for 3D sound placement

- **What is 3D sound?**
 - Filtering on a sound that makes it appear to have a specific position and velocity in space
 - Not “expanded stereo”
- **Makes sound an even more engaging aspect of gaming**
 - Increases sense of immersion
 - Provides additional cues in game play



How It Works

Simulation of sound physics

- **Source modeling**
- **Atmospheric effects**
- **Environmental reverb**
- **Doppler effect**
- **Head shadow**



How It's Implemented

Complete solution for 3D sound

- **Software pre-mix filter component**
 - Plugs into any Sound Manager channel
 - Modifies any sound played on that channel
- **Low-level message interface**
 - Controls filter component parameters
- **High-level API for setting up messages**
 - Handles the math
 - Optional



Software Filter Component

- **Installed per-channel as a pre-mix filter**
 - Unlocalized channels are not affected
 - Sound channels may be reused
- **Runs on any Power Macintosh**
- **“Quality knob” allows you to tune CPU load vs. quality tradeoff on the fly**
 - Best quality—true 3D sound
 - Least CPU load—panning, amplitude and Doppler effects



Low-Level Messages

- **Sent via Sound Manager to filter**
- **Built for speed and future growth**
- **Listener-centered polar coordinate space**



High-Level API

- **Fills in low-level messages**
- **Allows listener and multiple sources to move independently in Cartesian coordinate space**
- **Velocities may be computed automatically**
- **Convenient for most game use**
- **Compatible with (but doesn't require) QuickDraw 3D**



Additional Effects

- **Reverberation**
- **Large sources e.g., a fountain**
- **Directional sources e.g., a trumpet**
- **Ambient effect**



Using SoundSprocket

Use mono sound effects

- **Create a listener**
- **For each sound entity**
 - Create a sound channel
 - Install the SoundSprocket filter on the channel
 - Create a sound source



Using SoundSprocket (*cont.*)

Use mono sound effects

- **In the game loop**
 - Update the listener's position
 - For each sound entity
 - Update the sound source's position
 - Function call to fill in the low-level messages
 - Send the messages to the sound channel



SoundSprocket Since 1.0

- Part of QuickTime
- Part of QuickTime VR 2.0 SDK





Case Study: Marathon Trilogy

Alex Rosenberg

Code Monkey

Sprockets Integration With Bungie's Marathon Trilogy

Teaching old code new tricks

- Marathon engine was several years old
- All graphics, sound and input was custom code
- Took one weekend to get Draw, Input and Sound Sprockets integrated



Sprockets and Marathon (*cont.*)

Particularly cool because:

- Engineers had never seen the Marathon code before
- Version 1.5 now supports just about every input device made
 - And will support all future InputSprocket devices





The Future of Sprockets

Chris De Salvo

Technical Lead

The Future of Sprockets

- **#1 priority is to make sure that existing Sprockets-based games continue to work**
- **Use existing resources to move the technologies forward**
- **We're deciding where the Sprockets will be best taken care of in the new organization**
- **Pushing the message that we're more than just a games API**



What About a Rhapsody Port?

Currently under investigation, but in the meantime...

- **Most DrawSprocket features are in the new NSDirectScreen class**
- **Input functionality will definitely be present**
- **SoundSprocket likely to be moved over**



Where to Get More Info

Official site:

<http://devworld.apple.com/dev/games/>

mac-games-dev mailing list:

<http://www.solutions.apple.com/ListAdmin/>

Developer feedback:

sprockets@adr.apple.com

Online training from Apple:

<http://devworld.apple.com/dev/devtraining.shtml>



Where to Get More Info (*cont.*)

*Free online course from
Apple Developer University*

Usenet

comp.sys.mac.programmer.games

Unofficial site:

<http://www.unsupported.com>

Unofficial FAQ:

<http://www.viperware.com/licorice>





Q&A

The background features a dark, textured surface with a glowing blue and purple sphere in the center. A white Apple logo is positioned at the top of the sphere. The text "Worldwide Developers Conference" is displayed in a stylized, golden-yellow font. The word "Worldwide" is at the top, "Developers" is in the middle, and "Conference" is at the bottom. The word "Developers" is enclosed in a white rectangular box. The overall aesthetic is futuristic and tech-oriented.

Worldwide

Developers

Conference