Imagination
TECHNOLOGIES

# Adding Advanced Shader Features and Handling Fragmentation

## How to Enable Your Application on all POWERVR Devices

**Contact: devtech@imgtec.com**

# Imagination Overview

- **Leading Semiconductor IP Supplier**
  - POWERVR™ graphics, video, display processing
  - ENSIGMA™ receivers and communications processors
  - META™ processors – SoC centric real-time, DSP, Linux
  - Licensees: Leading Semis and OEMs
  - #4 Silicon Design IP provider *


- **Innovative Consumer Product Manufacturer**
  - PURE digital radio, internet connected audio

- **Established technology powerhouse**
  - Founded:1985
  - Listed:1994-London Stock Exchange:IMG
  - Employees: more than 620 worldwide

** Source: Gartner IP Suppliers Report, March 2009*

**UK Headquarters**

**R&D**

**Sales**

# POWERVR Graphics

- **Unique Tile Based Deferred Rendering architecture (TBDR)**
  - Enhances performance
  - Reduces power consumption
- **MBX Series**
  - Fixed-function graphics acceleration
  - Widely adopted in mobile devices
- **SGX Series**
  - Programmable, shader-based graphics acceleration
  - Newer technology already available in 100+ platforms


Apple iPhone 3GS


Samsung Wave


Motorola Droid/Milestone


Samsung i8910HD


Sony Ericsson Satio


Vivaz


HTC Qilin (Dopod T8388)


Nokia N900


Vodafone 360 H1 (Samsung )


Emblaze Else


Palm Pre








Imagination
TECHNOLOGIES

# POWERVR Devices
# or
# “What’s the challenge for developers?”

# Hardware Fragmentation

- **Architecture Fragmentation**
  - Intel/AMD X86 versus ARM7/9/11/Cortex/MP, SHMobile, MIPS, X86, PowerPC, etc.
  - Feature Fragmentation
    - E.g. FPUs Optional (VFP), SIMD Extensions Optional (NEON)

- **Physical & Performance Fragmentation in Embedded Market**
  - From 200MHz ARM9 to GHz+ Multi-Core Cortex-A9
  - From unified memory with 16bit DDR bus to segmented memory with multiple 128bit buses to DDR3
  - From single pipeline GPUs to multi-core multi-pipeline GPUs
  - From QVGA to 1080p Screens (Mobile to HDTV)
  - From 16MB (feature phone) to 256MB (Smartphone) to GBs of memory (netbook)
  - Full keyboard to touch screen only

- **And this list is increasing…**

Apple iPhone 3GS | Samsung Wave | Motorola Droid/Milestone | Samsung i8910HD | Sony Ericsson Satio | Sony Ericsson Vivaz | HTC Qilin (Dopod T8388) | Nokia N900 | Vodafone 360 H1 (Samsung ) | Palm Pre

# And this list is increasing…

Imagination

Dell
Mini 10

Acer
Aspire One AO751H

Nokia
Booklet 3G

Sony
Vaio X-series

Sony
Vaio P-series

Asus
S121

Fujitsu
UH900

Archos 9

Sharp
Willcomm D4

Clarion
Mind

Archos 5


Asus
R50A

Apple
iTouch 32GB

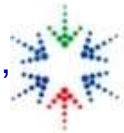
# Software Fragmentation

Imagination

- **OS & Windowing Frameworks**
  - iOS, Windows Phone 7, Android, WinCE, Linux (Maemo, LiMo, bada, Meego, X11 variants etc.), Symbian (S60/UIQ), Android, OS X, many RTOS (Nucleus, QNX, uiTron, ...)
- **This list is also increasing…**
- **What's it mean?**
  - Obviously many different ways of doing things and different capablities
    - File handling
    - Input handling
    - Initialisation code…
  - Less obvious things
    - Can't rely on STL being available on all devices
    - Compilers sometimes don't deal very well with templates and similar
    - Some devices don't have file systems, command line options…





LiMo Foundation

palm webOS™

# Standard APIs

- **Standard APIs help reduce fragmentation**
  - Khronos Group focus on Multi-Media API standardisation including:
    - OpenGL ES 1.1/2.0 for 3D Graphics
    - OpenVG 1.1 for Vector Graphics (Flash-like content)
    - OpenMAX for Video Decoding


- **Industry Standard APIs define a "common" language**
  - Must be translated to the language supported by the hardware in the software driver
    - Efficiency of this translation is variable - depends on standard & hardware capabilities
  - API definitions are not perfect
    - Always scope for variability and different interpretations – not always possible to fix



- **But extensions can be essential for good performance and aren't standard**
  - Must be checked for
  - Alternatives may require different art assets, rendering strategies

# The Impact of Graphics Acceleration
## Software versus Acceleration

Imagination

- **Advances like acceleration mean huge progress in mobile software expectations**

**Software**

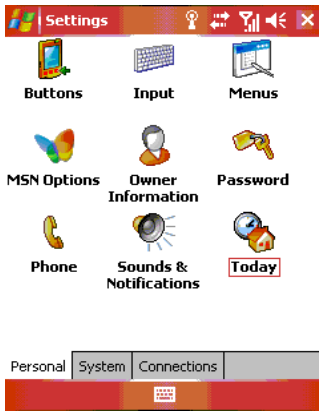
**CPU Only**


**OpenGL ES 1.1 Hardware**

**POWERVR MBX & SGX**

**OpenGL ES 2.0 Hardware**

**POWERVR SGX**

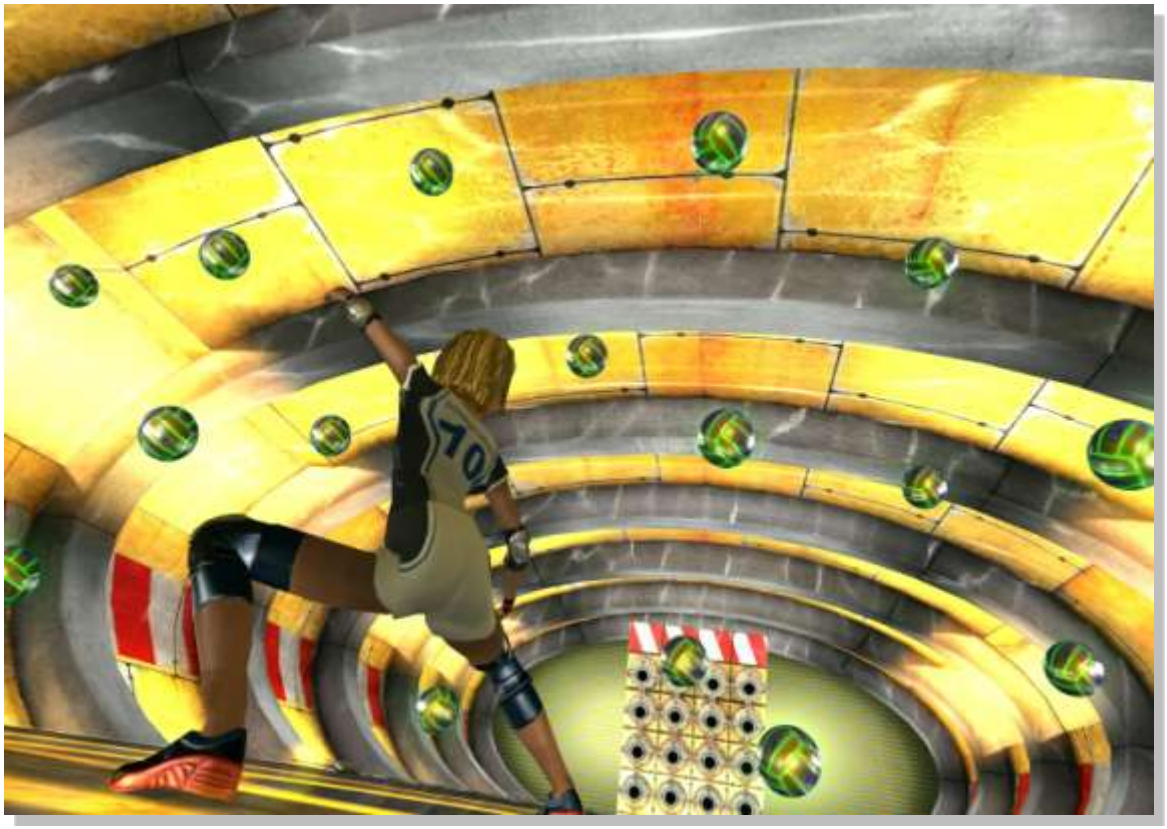




# High Quality Graphics

Imagination

- Advanced effects and complex scenes are now possible
- Capabilities of devices are growing very fast



- But this also means a wide range in device capabilities
- Lowest common denominator isn't good enough

# So…

- **Choose a single platform and develop for this**
  - + Do one platform well vs do many platforms badly
  - + Cost of development vs return may be better especially in short term
  - - Dependent on single device's market share
  - - Dependent on single device's future
  - - Moving to target more platforms may be difficult later

## OR…

- **Develop intelligently for multiple platforms**
  - + Larger market to target
  - + Less dependence on single device's success
  - + More future-proofed
  - - Complexity of development
    - - Time = Money = Resources

# How to Target Multiple Platforms?
## Design Practices

- **Abstraction Layers**
  - Write bulk of the code to your own unified API
    - Hides platform complexity
      - OS, screen sizes, memory allocations, initialisations, etc.
    - Reduces testing cost and time
    - Simplifies support of 3rd parties porting your code
- **Design Flexibility**
  - Create your software architecture with an eye on expansion and interaction
    - Plug-in Approach
      - Unify the interaction with other software components
    - Extension Approach
      - Build in interfaces for extra functionality
- **Scalability**
  - Build scalability into your engines from day one
    - Future proofing for more powerful, but yet-to-be released platforms
    - Tweak-ability before release or per game level
  - Flexible art asset chains
- **Automation**
  - Regenerating assets for multiple platforms by hand wastes time
  - Automatic nightly builds hi-light problems in common code quickly

Imagination
TECHNOLOGIES

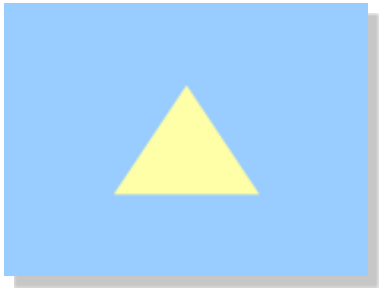
# POWERVR Developer Technology

## or

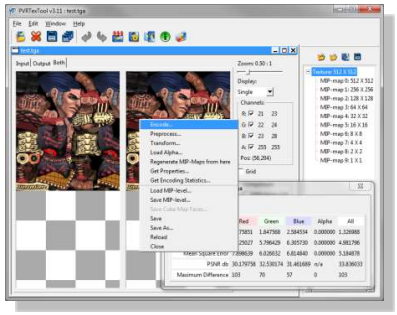
## “How do we do it?”

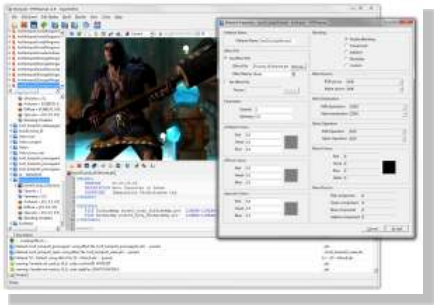
# Motivation

- **Requirement for software that has to run on and support numerous existing and future platforms**
  - Optimal development time
    - Minimize code duplication
    - Minimize duplicate skills
  - Maintenance time
  - Flexibility
    - Can't predict future platforms: OSs, APIs




- **Requirement for training software for specific graphics techniques**
  - Reduce distraction of unrelated operations in code
  - Coherence across multiple platforms to help learner

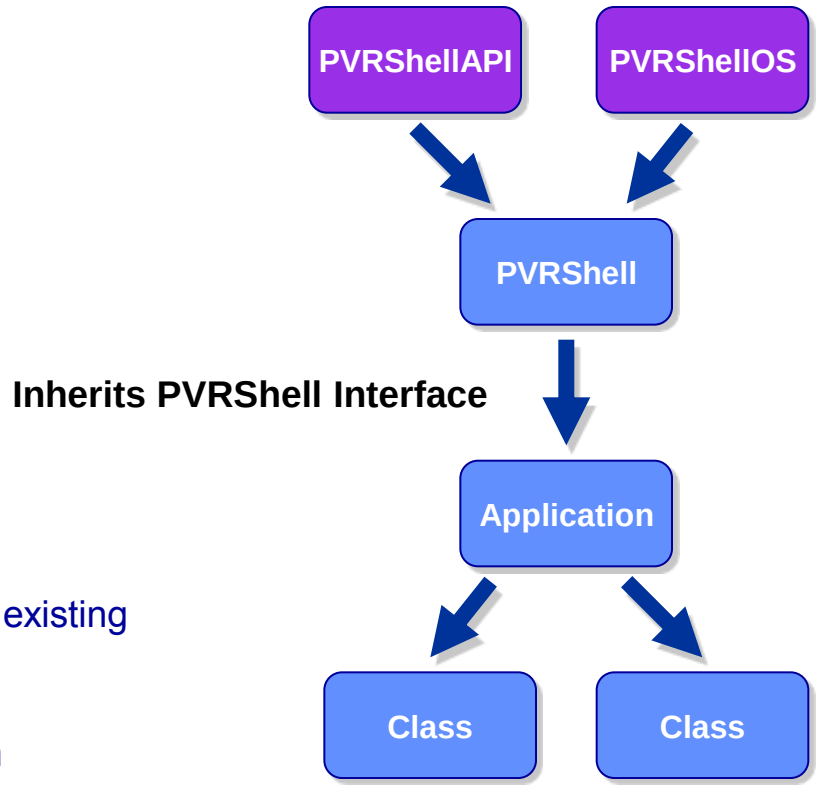





- **Scale of task**
  - Currently approaching 100 target platforms
  - OSs in double figures
  - Multiple graphics APIs and windowing systems

# PVRShell Abstraction Layer

- **Shields developer from common OS operations**
  - API & window setup
  - File handling
  - Input gathering
- **Other useful features**
  - Screenshots
  - Command line options




- **When we get a new Platform**
  - Create OS and/or API specific file
    - Use example code from platform provider
    - Use existing PVRShell code from “closest” existing platform
    - Hooks into existing PVRShell structure
  - Allows 30+ demos can to run on a new platform (sometimes within a day)
    - Usually no need to edit application code at all

# PVRShell for Developers

- **Source code**
  - Full commented, source code is available that allows customisation:
    - Extension of features for specific projects
    - Extensions for specific platforms
- **Handles environment setup**
  - Configurable through PVRShell interface
- **Direct access to standard APIs for drawing code**
  - minimal wrapper overhead
  - Minimal restrictions
- **Easily bypassed for specific situations**
  - Platform specific code is not impossible
  - Helps for prototyping and proof of concepts
- **Free and with almost no licence requirements**
  - We ask that distributed source code preserve our copyright
  - Use in binaries has no such restriction
    - We'd appreciate a credit or use of our logo
  - All other source code has the same conditions

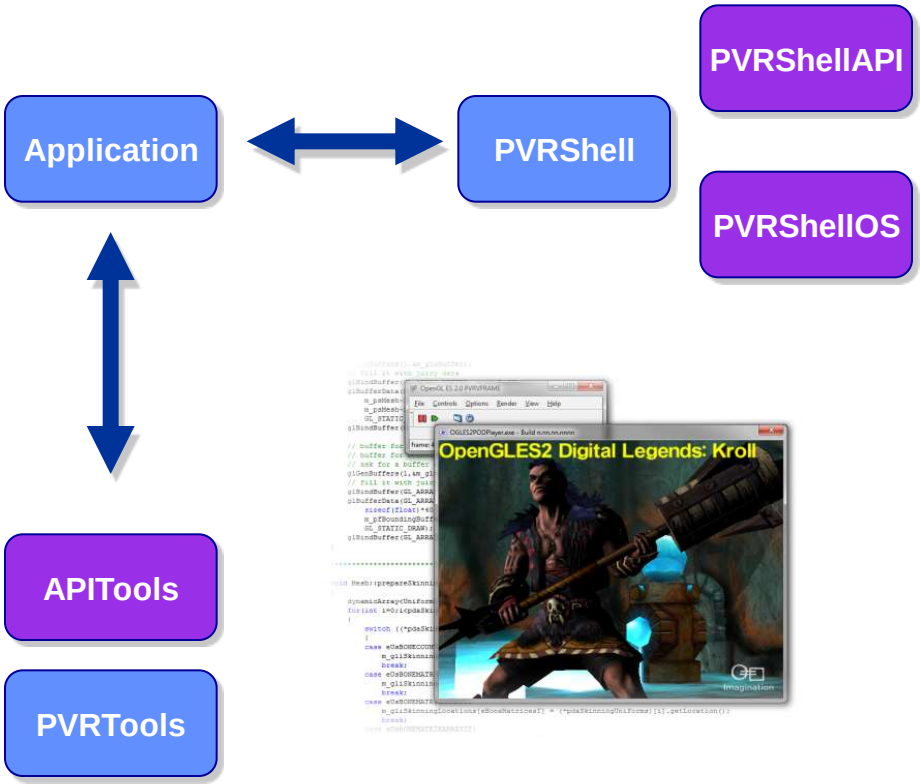
# PVRShell & PVRTools

## PVRTools

- OS independent
- API versions with dependent files
- Much common code shared between APIs
- Code that is API specific generally is of similar structure across APIs and diffs well

- **What's in the PVRTools?**
  - Texture loading
  - Scene/model loading
  - Effects runtime
  - Vector/Matrix library
  - Shadow volume calculation
  - String class…

- **Free**
  - Same licence as all other source code

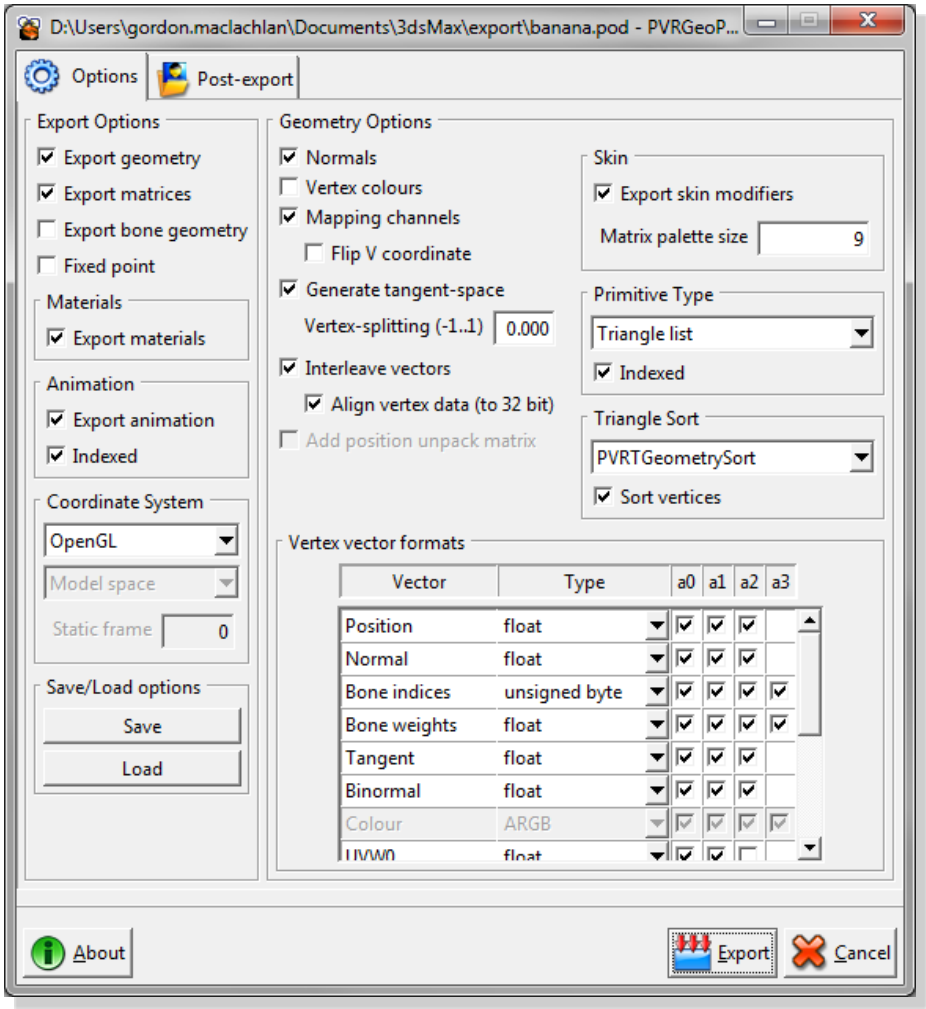


Imagination
TECHNOLOGIES

# Asset Creation

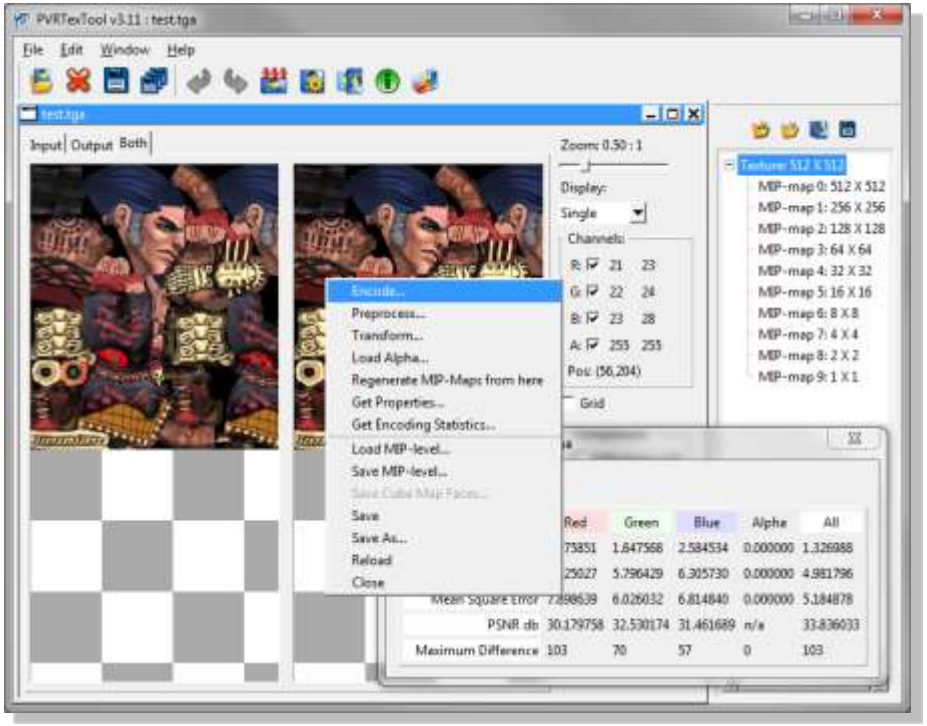
# PVRGeoPOD & Collada2POD

- **What is POD?**
  - POD is the POWERVR Object Data format
  - Optimised deployment format for POWERVR hardware
  - Runtime and examples provided in source code in the PVRTools
- **PVRGeoPOD**
  - POD exporter plug-in for 3D Studio Max, Maya and Blender
- **Collada2POD**
  - Provides conversion from the Khronos Collada interchange format to POD
  - GUI and command line versions of the utility are available for Windows, Linux and Mac OS



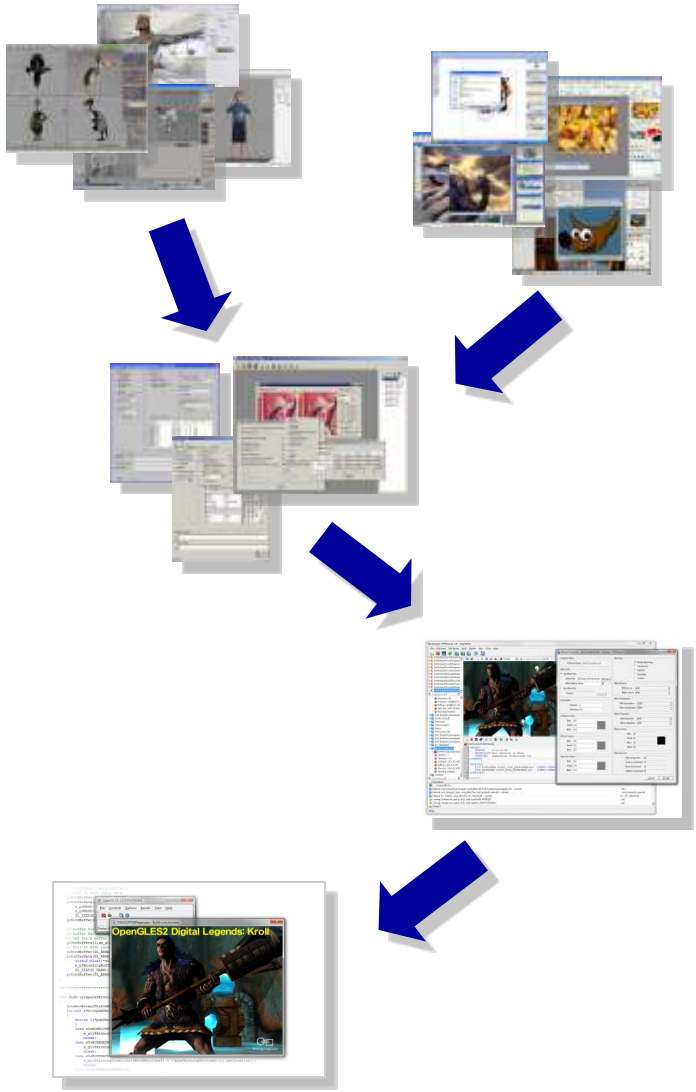
# PVRTexTool

- **What are PVR files?**
  - Texture container format
  - Single line loading code in the PVRTools
- **Processes and compresses textures**
  - Command line and GUI versions of the utility available for Windows, Linux and Mac OS
  - Exporter plug-in for Adobe Photoshop
  - PVRTexLib library for direct integration into toolchains
- **PVRTexTool features**
  - Opens, displays and compresses the most widely used texture formats, including all OpenGL ES (1.x, 2.0), Direct3D (9, 10 & Mobile) and OpenVG standard formats
  - Pre-processing support
    - Normal map generation, colour bleeding, border addition, high quality scaling algorithms and sky box optimisation
  - Support for PVRTC, DXT/S3TC and ETC compression formats



# Why these formats?

- **Formats are designed for deployment to multiple mobile platforms**
  - Optimised for size and speed
    - Texture compression and triangle ordering optimisation not available elsewhere
- **Proven and tested on the many platforms we support with great success**
  - Full runtimes are provided as source code in the PVRTools
  - API for use is streamlined and uniform across platforms
    - Loading through PVRTools library is usually a single line
    - Same or similar line on every platform
- **Other formats:**
  - PFX: shader effect format
    - As seen in PVRShaman and multiple POWERVR demos
  - PVG: vector graphics format
    - for OpenVG



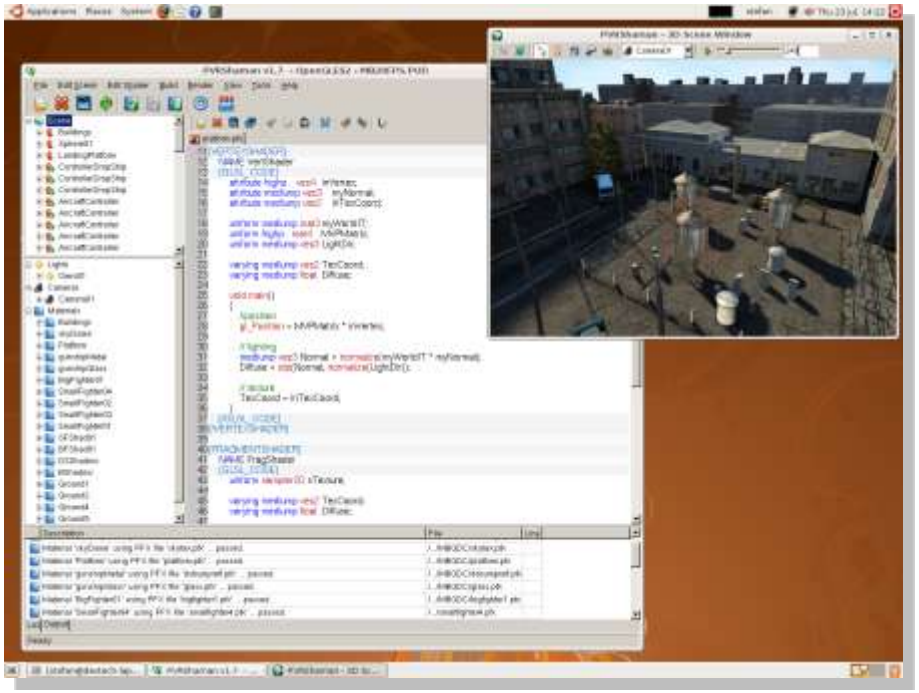
Imagination
TECHNOLOGIES

# Prototype

# PVRShaman

- **Integrated shader development environment**
  - An IDE that allows rapid prototyping of shader effects on Windows. Linux and Mac OS
  - Combines the functionality of multiple POWERVR utilities
  - See results of shader changes as you implement them

- **PVRShaman features**
  - Work with POD models or import Collada files
    - Scene navigation, object selection
    - animated and free camera modes
    - wire-frame mode, overdraw analysis etc
  - Material editing
    - Edit shaders through.PFX effects
    - Add .PVR textures
  - Integrated PVRUniSCoEditor & Compiler
  - Multi-window configurations and multi-screen support
  - Support for DirectX, OpenGL and OpenGL ES (1.1 & 2.0) graphics APIs



# PVRVFrame

- **POWERVR OpenGL ES emulator**
  - OpenGL ES (1.1 & 2.0) emulation on Windows and Linux

- **PVRVFrame features**
  - Allows applications to be developed and debugged without POWERVR hardware
  - Full support for OpenGL ES (1.1 & 2.0)
  - Supports all MBX and SGX extensions
    - Choose a profile for a specific core

Imagination
TECHNOLOGIES

# Iterate & Deploy

# How much should you automate?

Imagination

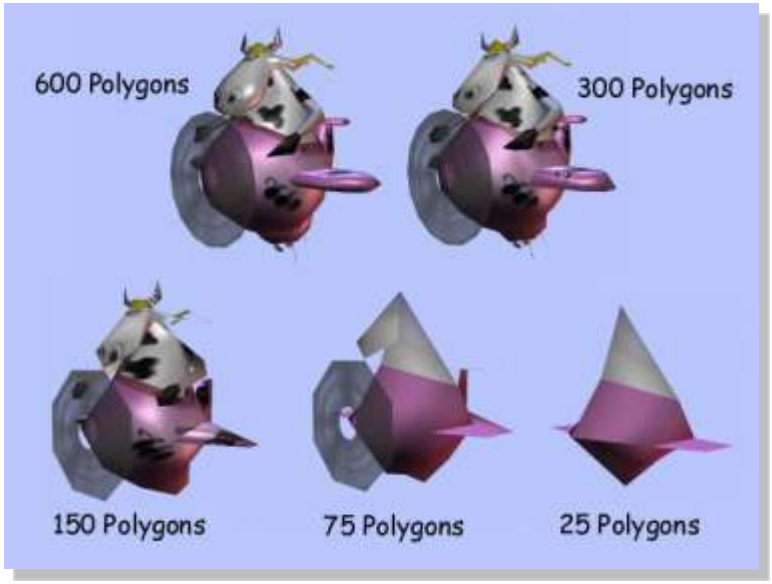
- **In short: *as much as you can!***

- **Examples:**
  - Nightly builds of *every* platform
    - That's every API and every OS
  - Texture encoding
    - Command line version of PVRTexTool allows scripting
    - PVRTexLib library allows direct integration into content creation tools
  - Project/makefile regeneration
    - List assets and files for a project
    - Run system
    - Produce projects/makefiles for all platforms automatically

- **For each new platform**
  - Add a template project
  - Tweak asset lists for texture size, weight of shaders, complexity of models
  - Rerun system

# Scalability

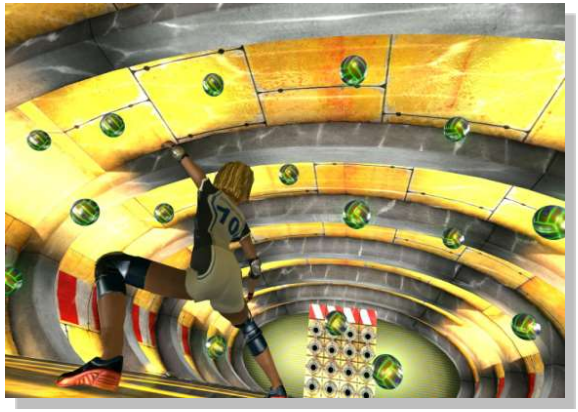
- **Look at desktop PC titles**
  - Level of detail for models
  - Resolution of textures
  - Complexity of shaders
  - Number of particles, lights
  - Type of shader




- **Run-time vs compile-time**
  - On mobile, often platform strengths/weaknesses can be provided for at build-time
  - Automation of asset creation will allow per-platform assets

# Performance

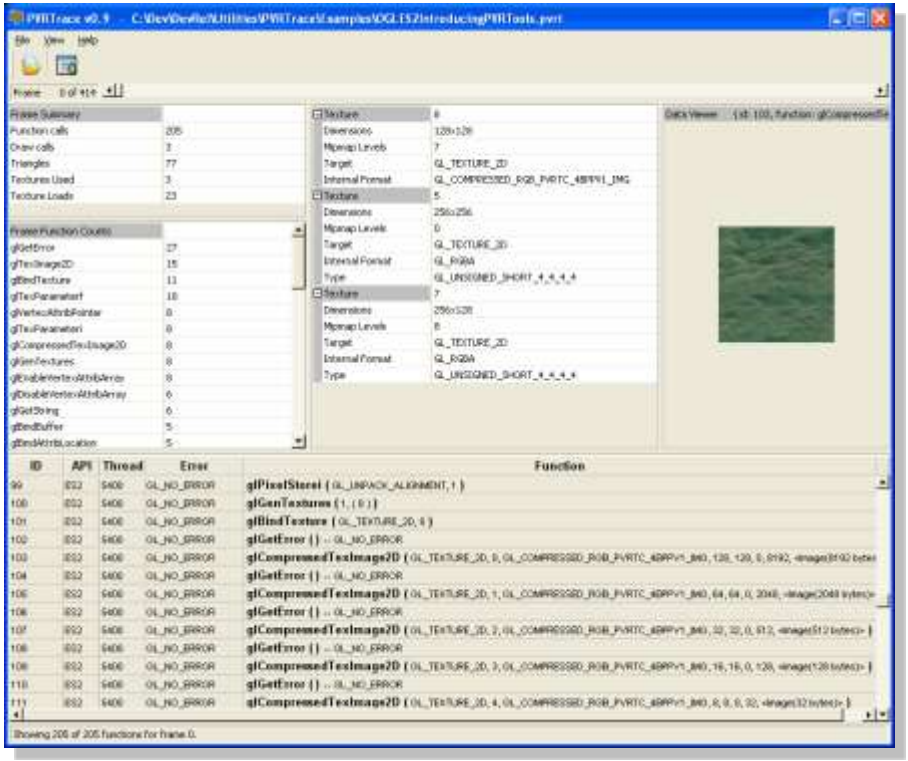
- **Read the documentation**
  - Our documentation is on our website and in our SDKs
- **Look at the examples**
  - The examples in our SDK are designed show how to implement a technique optimally
- **Use the tools we give you**
  - PVRTrace
  - PVRTune
- **Ask the experts**
  - POWERVR Developer forum: http://imgtec.com/forum/
  - Direct email to the POWERVR Developer Technology team: devtech@imgtec.com
- **Do these things at the beginning of a project!**
  - Not at the end
  - Do you want to have to re-write most of your rendering code in the week you're supposed to ship?



# PVRTrace

- **Graphics API call tracing**
  - Record API calls made by OpenGL ES applications
  - Analyse trace data using the PVRTrace GUI utility on Windows, Linux and Mac OS

- **PVRTrace features**
  - Trace emulated applications running PVRVFrame, as well as POWERVR-based devices
  - Review calls made for each individual frame
  - Filter calls by type (draw, shader, buffers etc)
  - Search for particular calls by name
  - Use the Data Viewer to observe the contents of data passed to GL calls, such as textures and shaders

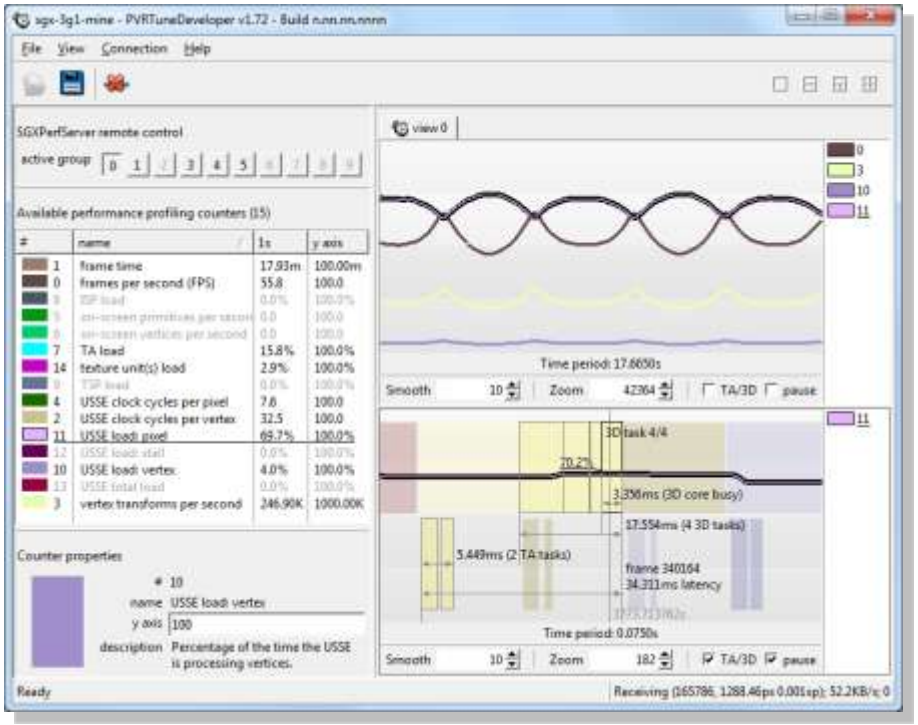


# PVRTune

- **POWERVR remote performance analyser**
  - Remote analysis client available for Windows, Linux and Mac OS
  - Uses a per-platform SGXPerfServer host application

- **PVRTune features**
  - Performance analysis device library combined with a remote analysis tool
  - Has very little impact on application performance
    - Uses driver level counters and hardware debug registers
  - Uses a network connection to transfer information between host & client applications
  - PC display of performance statistics
    - Accurate FPS counter
    - Vertex and pixel throughput
    - Batching & render state information
    - Parameter buffer & texture memory usage
    - Bandwidth usage
    - USSE utilisation & cache utilisation

- **Availability**
  - Currently available under NDA only

Imagination
TECHNOLOGIES

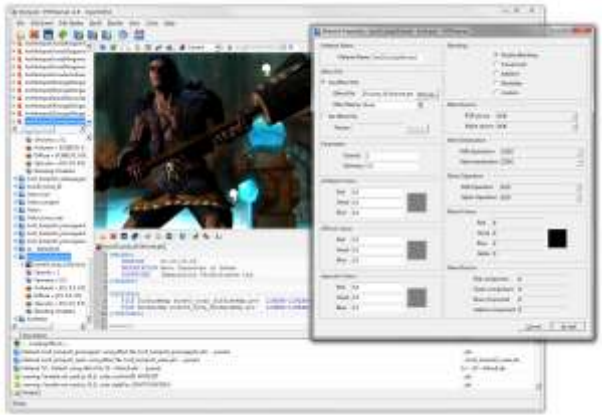
# The POWERVR *Insider* SDK

## or

## “How do you get this stuff?”

# What is the POWERVR *Insider* SDK?

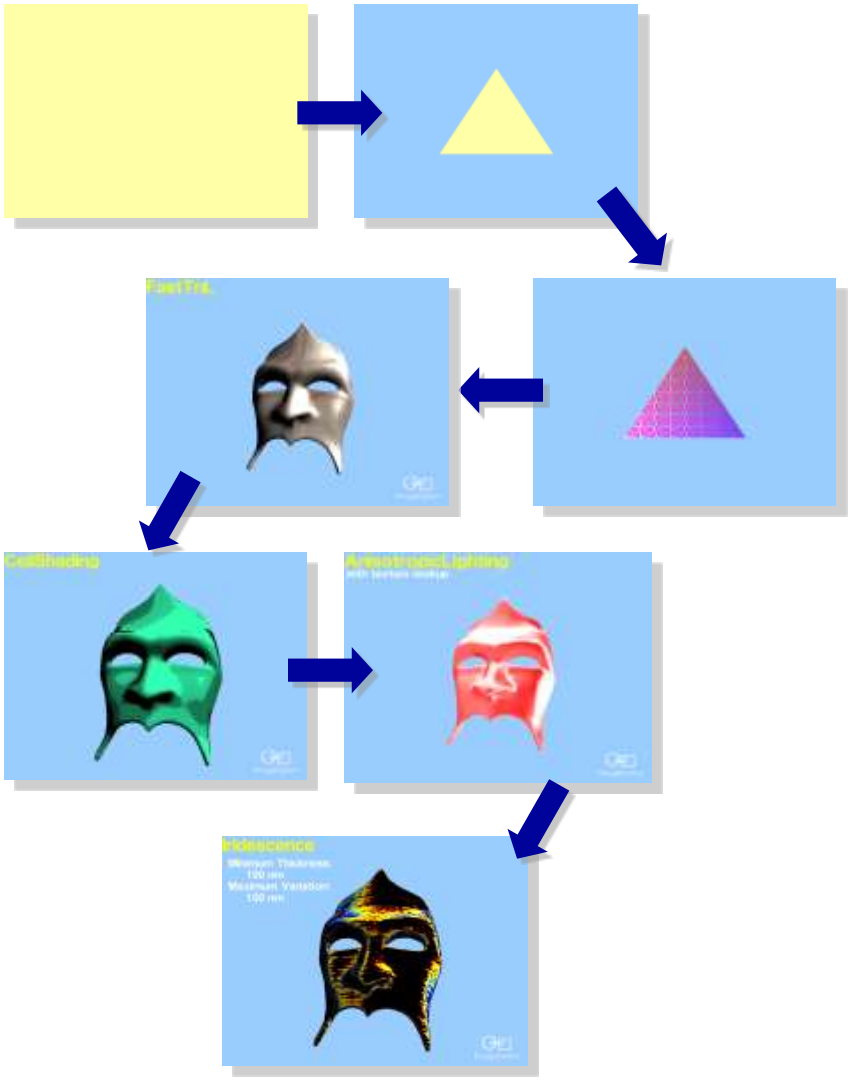
- **POWERVR Software Development Kit**
  - Utilities, tools, tutorials, demos and documentation
  - Designed to enable developers and customers to produce applications that make optimal usage of POWERVR hardware
- **Available for all POWERVR enabled platforms, for free!**
  - 70+ platform configurations
  - Multiple operating systems
    - Symbian, Linux, iPhone OS, WinCE and Android, with more to come
  - Multiple graphics APIs
    - OpenGL, OpenGL ES (1.x, 2.0), OpenVG and Direct3D (9, 10 and Mobile)

# Training Courses
## Source Code

- **Single technique tutorial approach**
  - Developed using partner and developer feedback
  - Each course targets a single topic to ensure simplicity
  - Wide range of supported graphics APIs
    - DX9/10, D3DMobile, OpenGL, OpenGL ES (1.1 & 2.0)
  - Gradual learning curve
    - From initialisation code, to rendering a simple triangle, to advanced shader effects
    - Tutorials have a similar implementation for each graphics API

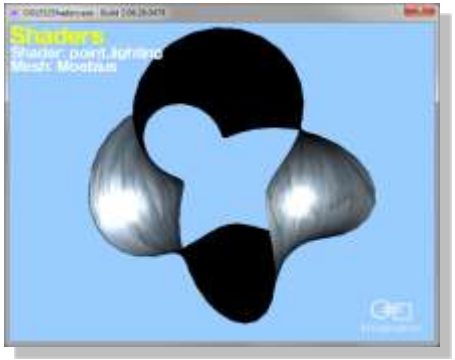
- **Optimised examples**
  - Fully commented source code and projects
  - Standardised implementation using the PVRShell framework and PVRTools
  - Demonstrates how to use POWERVR cores effectively



# Demos
## Source Code

- **More complex techniques, same clear approach**
  - Fully commented source code and projects
  - Optimised to make the most of the POWERVR hardware
  - Implement a combination of techniques to achieve robust, efficient 3D scenes
  - Uses the same standardised implementation as the Training Courses
  - Wide range of supported graphics APIs
    - DX9/10, D3DMobile, OpenGL, OpenGL ES (1.1 & 2.0)


- **OpenVG (1.0.1 & 1.1)**
  - Supported with its own set of demos
  - Same simple approach and implementation





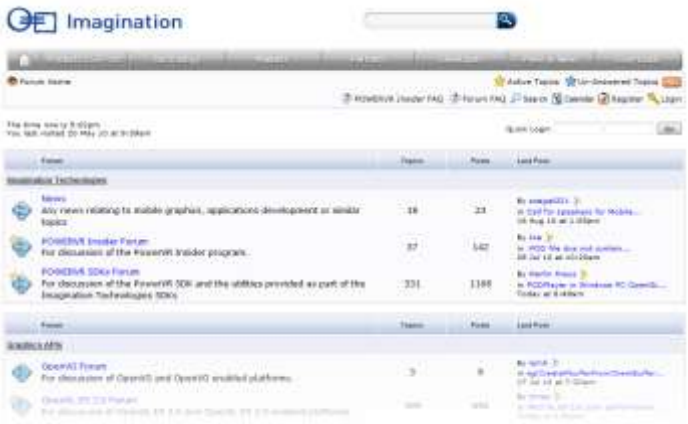

# POWERVR *Insider Ecosystem*
## Industry Leading Collateral and Support Resources

Imagination

### Specialised Website
### www.powervrinsider.com


### Focussed Newsletters

### Technical Support Forum




### Partner Promotional Flyers

# POWERVR *Insider* Ecosystem
## Enabling faster time to market

Imagination

- **Key Objectives**
  - Faster Time to Market by enabling partners to tap into capable resource database
  - Lower cost to adopting POWERVR SGX Graphics
  - Off the shelf POWERVR optimised solutions – maximum performance, power optimised
  - Network of experienced service companies to handle customisation requests
- **Targeting all key embedded focus areas**
  - GUI, Navigation, Browsers, Gaming, Service Providers, Benchmark Companies, etc.
- **Just a small selection of the hundreds of companies working with POWERVR:**

# Where to get the POWERVR SDK, Documentation & Support


- **Available through the POWERVR *Insider* Programme**
  - Click on “Developers” on the Imagination website: http://www.imgtec.com
  - Free to join!
- **Benefits of being a POWERVR *Insider***
  - Access to the SDK downloads
  - Documentation – performance guidelines, explanations of POWERVR technologies and utility user guides
  - FAQs
  - Developer forums
  - Direct email contact to POWERVR Developer Technology: devtech@imgtec.com
  - Partner Program
  - Newsletter
  - Cross-promotions through press announcements and print/online media
  - Tradeshow Partnership

Imagination
TECHNOLOGIES

# Adding Advanced Shader Features and Handling Fragmentation

## How to Enable Your Application on all POWERVR Devices

**Contact: devtech@imgtec.com**