SIGGRAPH2006

SIGGRAPH2006

OpenKODE Developer Perspectives

Timo Suoranta

Futuremark Corporation

- Established in 1997
- HQ in Espoo, Finland
- Private and profitable
- Products
  - Industry standard benchmark software
  - Performance related web applications
  - Custom technology demos
- Mission

to increase growth within IT industry by showing the performance of new PC and mobile technology, simultaneously taking into account the end user satisfaction

Agenda

- *Motivation*
- Futuremark engines
- Content path

Motivation - Variations

SIGGRAPH2006

- Handheld market ISVs need to cope with variations:
  - OS
  - Display capabilities
  - Audio capabilities
  - Controls
  - Performance

Where handheld devices come from?

SIGGRAPH2006

- Operator
- Mobile device manufacturer
- Integrator
- Operating system
- CPU IP
- OpenGL ES IP, OpenVG IP

Motivation - APIs

SIGGRAPH2006

- We can not change the actual hardware variations. Hardware variations are good for the consumer.
- From developers perspective it is the APIs – not the hardware – that matters.
- Good API helps developers to live happily with the hardware variations.

What about Java?

- Java and JSRs are good in theory
- ISVs need to cope with varying implementation bugs
- Performance varies a lot and is often far from optimal
- Even when these get fixed, there will always be some amount of native code development

Agenda

SIGGRAPH2006

- Motivation
- *Futuremark engines*
- Content path

Futuremark Mobile Engines

- Developed in house since 2003
- Used in multiple demos, custom benchmarks, SPMark04, 3DMarkMobile06
- Two content paths:
  - Avid Softimage XSI
  - NewTek LightWave

Futuremark Mobile Engines Requirements

- Highly portable
- Single source for all platforms, ANSI C
- Flexible source code sharing with partners
- Version controlling

Futuremark Mobile Engines Design: Layers

- Application

  Demos, Benchmarks

- Engine

  OpenGL ES e., OpenVG e.

- OS Abstraction

  Desktop Windows

  Nokia Series 60 Symbian

  Dell Axim X50v/X51v

Futuremark Mobile Engines Design: Configuration Variables

SIGGRAPH2006

• CPU data type:	Floating or fixed point
• GPU data type:	Floating or fixed point
• Skinning:	CPU or GPU
• OpenGL API:	Desktop or ES
• Primitive type:	Triangle strips or lists

- Various OpenGL ES extensions

Futuremark Mobile Engine OS Abstraction Layer

- Roughly equals OpenKODE APIs:
  - Memory
  - File IO
  - Input
  - Timer
  - Graphics
- But is also an application framework.

Application framework requirements

- Startup code
- Default configuration discovery
- Event handling
- Threads
- Timer procedures
- Library loading
- Power management

OS Abstraction: Application entrypoints

- Int main(int argc, char** argv) ?
- create()
- configure()
- init()
- deinit()
- update()
- exit()

Experiences

SIGGRAPH2006

- With layers and configuration variables, the code is very portable
- Even a single binary can work on different hardware thanks to standard OpenGL ES library naming
- Changing OS abstraction layer API means every implementation needs to be updated

Agenda

- Motivation
- Futuremark engines
- *Content path*

Content path

SIGGRAPH2006

- Very important component
- Prefer data driven solutions
- Intermediate data format
- Do as much as possible in the content path and minimize runtime processing
- Try embedding engine inside the content path, or setup realtime update link

Content path: Budgets

- Frames per second
- Vertex processing per frame
- Fragment processing per frame
- State changes per frame
- CPU processing per frame
- Texture memory, texture sizes
- Vertex and index buffer memory

Summary

- OpenKODE APIs will make source portable software easier
- Application frameworks and middleware can be written on top of OpenKODE – Less or no platform specific code needed
- COLLADA shows the right direction for content paths

Thank You !

- Questions ?
- (Demos)