

Portable Entertainment Development - the console approach

Paul Holman

Vice President, SCEE Research & Development

2011 Jordan Gaming Summit



Abbreviations

PSP™ (PlayStation®Portable) = PSP

PlayStation®3 = PS3

PlayStation®Suite = PS Suite

PlayStation®Vita = PS Vita

PlayStation®Certified = PS Certified



Agenda

- Welcome
- PS Vita Overview
- Traditional Console Development
- PS Suite
- Demo
- Q&A



Introducing PS Vita



PlayStation®Vita

“Designed for the Ultimate Portable Entertainment Experience”



Launches from **22nd February 2012:**

- €249.99 (RRP) for the Wi-Fi model
- €299.99 (RRP) for 3G/Wi-Fi

5 inch OLED screen: engineered for extremely low power consumption



Hardware

- Standard Iconic PlayStation Buttons
- Dual Analog Sticks
- Front Touch Screen
- Rear Touch Pad
- Front & Rear Camera
- Motion Sensors



Storage and Distribution

- PS Vita card
 - 2 GB or 4 GB
 - May contain some writable memory
 - (-5..10% for save data, patches etc)
- memory card
 - Cards will be large to support a variety of downloaded content
- Single Submission for both formats



Connectivity

- Wi-Fi and 3G/Wi-Fi Models
 - 3G/Wi-fi model has GPS
- Wi-Fi models can still determine location using less precise methods



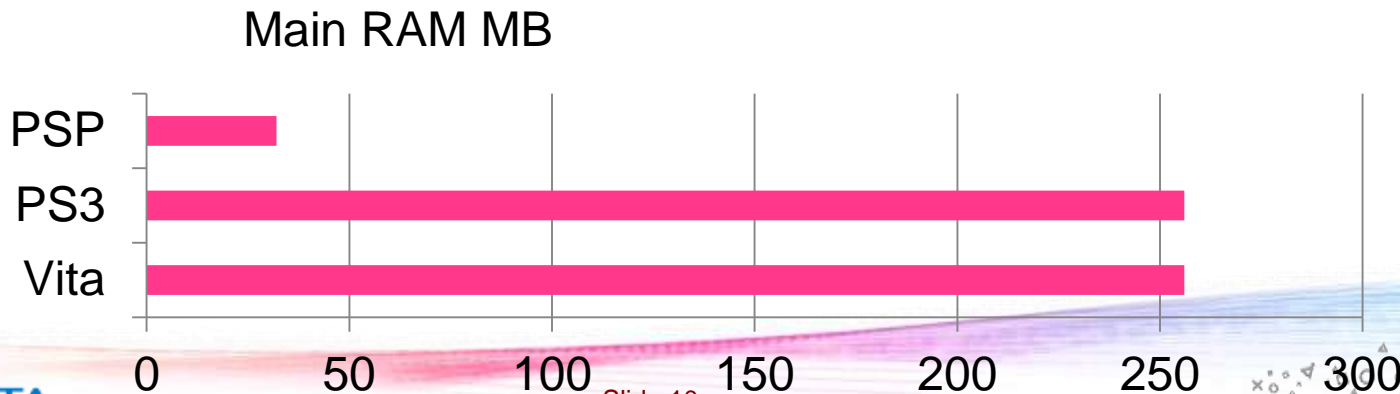
Hardware

- Custom SOC
- 4x Arm9 Core
 - High performance 32-bit cores
 - Advanced branch prediction
 - Full cache coherency between cores
 - NEON integer and floating Point SIMD Engine
- POWERVR SGX543MP4+ GPU
 - Multi-core
 - Dynamic Load Balancing
 - Tile Based Deferred Rendering (TBDR)
 - Earliest possible rejection of non visible pixels



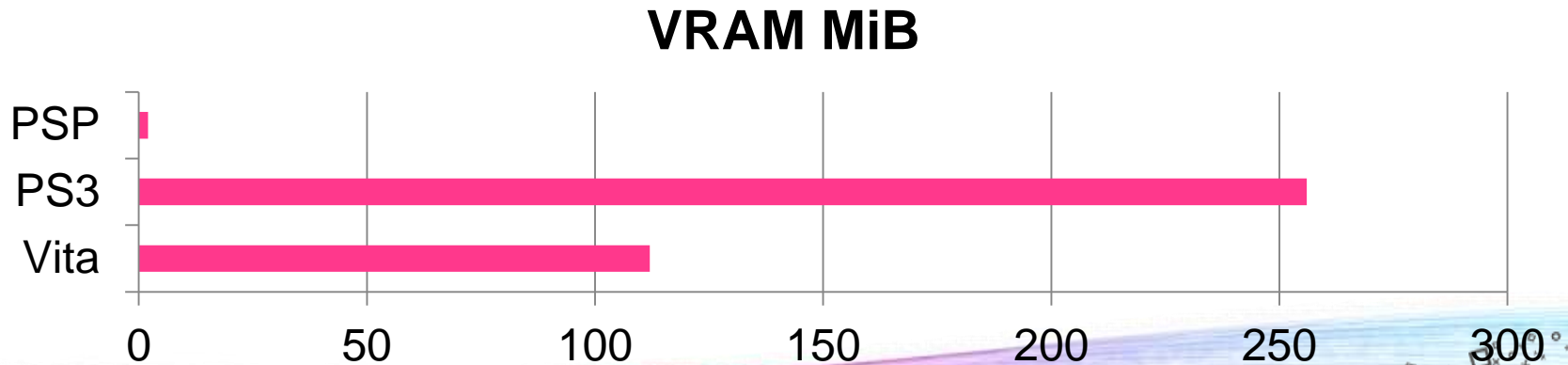
Platform Comparison: Main Memory

- PS Vita has lots of main RAM
 - 256MB is available to developers
- PS3 has 256MB of main RAM, but not all of that is available to applications



Platform Comparison: CDRAM

- Plenty of CDRAM (also called VRAM)
 - 128MB total
 - Some of this is used for buffers
 - Textures and Vertex Data can also live in main RAM
 - Similar to PS3

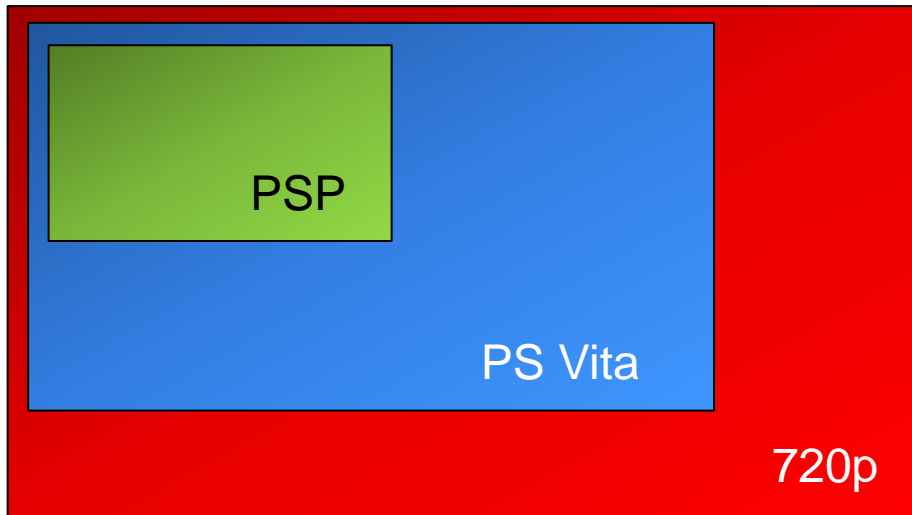


Platform Comparison: Clock Speeds

- PS Vita is a portable device
- Power use and battery life are as important as performance
- PS Vita performance will not be as high as PS3 because of this
- Some phones claim 1.5GHz+ speeds
 - This is not sustained and rarely, if ever, actually reached for more than a couple of seconds



Platform Comparison: Screen



PSP = 480x272

PS Vita = 960x544

720p = 1280x720

- High Resolution
- High Pixel Density
- Efficient MSAA
 - Multi-Sample Anti-Aliasing => improves image quality



Other Features

- Social Networking
 - Designed from ground up as a social networking device
- Location Based Gaming
 - Location Services
 - near



Developing for PS Vita



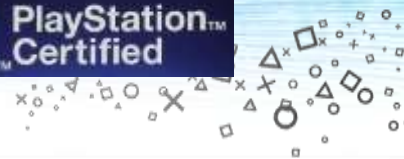
(1) Traditional Console Approach

- Programmers code in C/C++/Assembler
- Gain maximum performance from low-level libraries and hardware access
- Dedicated Development Hardware:
 - PSP DevKit: €1,200
 - PS3 DevKit: €1,700
 - PS Vita: TBC (Competitive)



(2) New PlayStation®Suite Approach

- Developers code in C#
- Code runs on a virtual machine – provides direct binary compatibility across a range of devices
- Development using a Windows PC and either
 - A consumer PS Vita
 - OR a PlayStation™ Certified Android Phone/Tablet device (more details later!)
 - OR PC Simulator



Summary

	“Traditional “ Approach	PlayStation Suite
Concept Approval	Yes	Lightweight (EULA)
Programming Language	C/C++/Assembler	C#
Technical Support	Included in costs of development hardware. Very strong technical support, consultancy & training.	Support is forum based
“Field of Use”	Games (some exceptions)	Games or Non-Games Application
Graphical Performance	High (low level libraries available, high levels of control)	Lower (runs on virtual machine) but OK for most genres
Sales Channel	Either as a PS Vita card and/or via main PlayStation®Store for PS Vita	PlayStation Suite area of PS Store for PS Vita and PlayStation Certified devices



Developing for PS Vita: “Traditional” Option



PlayStation®Vita Approach

- PS Vita was not created in isolation by hardware engineers in Japan
- Huge amounts of input from our 1st party as well as 3rd party studios (inc T&M partners) from a very early stage
- Feedback included:
 - Input mechanisms and gameplay features
 - Hardware – SOC: CPU and GPU
 - Software – Libraries and API design and implementation
 - Tools – design and implementation
- All aspects of PS Vita has been developed from the ground up with developers in mind
- In a very real sense it has been created by developers for developers



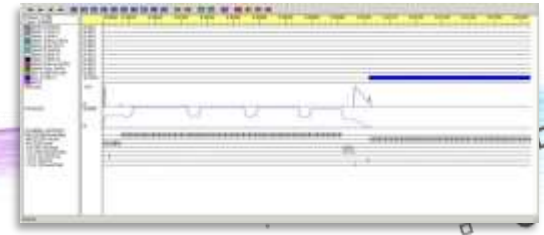
Friendly development platform

- Really nice platform on which to develop
- Hardware development kit is very nice
- Iteration time, tools, and documentation are great
- Performance is excellent



Game Development on PlayStation

- Largely C or C++ (some assembler)
- Windows based development environment
 - Cross compile to a hardware “target”
- Tools integrated with Visual Studio
- Performance tools from SN Systems
- Free Technical Support & Training



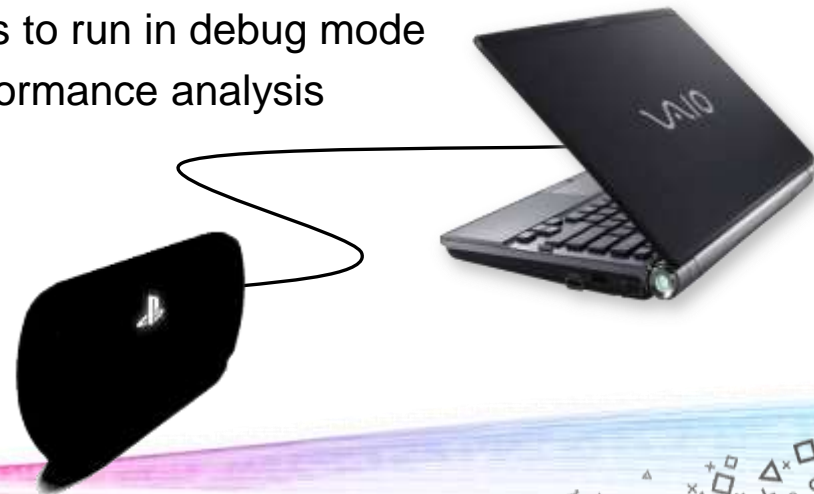
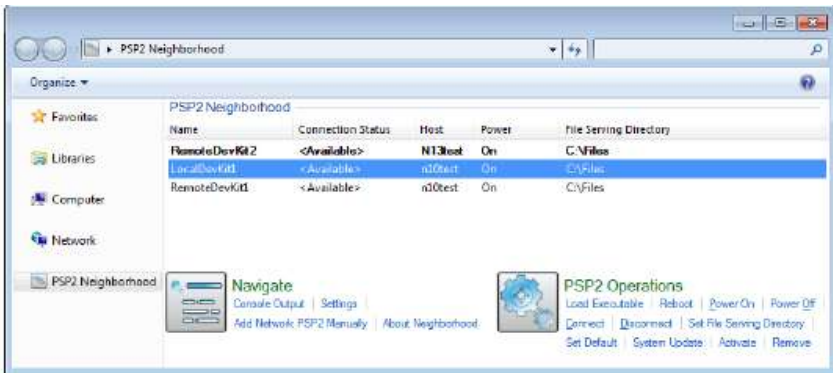
Titles using the traditional SDK

- A diverse range of titles
 - Biggest AAA Games down to mini games
- For the big games
 - Can reuse a lot of existing PS3/Xbox assets
 - Simplify models, shaders, textures etc
 - Have to write a platform specific layer
 - Most developers are already cross platform
 - Write once use in many games



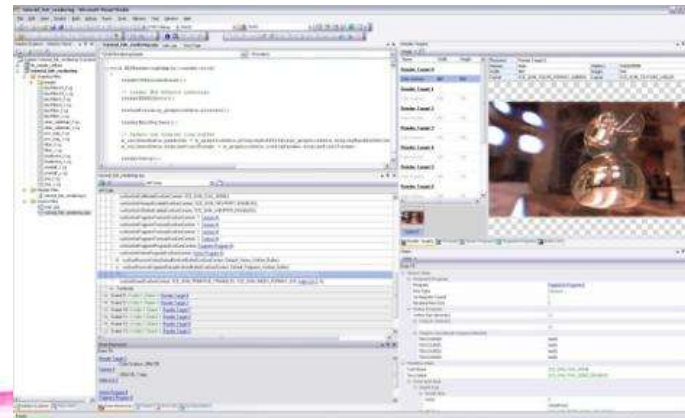
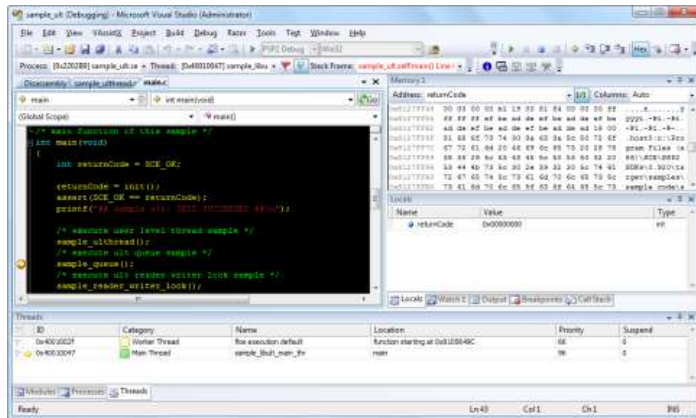
PS Vita Development Kit

- DevKit is designed to be **self contained**
 - Connect via USB to PC, but reachable across Ethernet
 - PS Vita kit is a single unit which closely resembles format of retail unit
 - Important for testing of location based services and input features such as Motion sensors
 - Extra memory allowing finished games to run in debug mode
 - Extra trace lines and counters for performance analysis



Development Tools

- Custom Development Tools
 - Uses Visual Studio IDE on Windows
 - Single Integrated Performance tuning tool for Graphics and CPU/Memory



Programming Model

- No unnecessary complications
- Tile-based processing handled behind the scenes
- Leaving you a familiar API interface



Shader Model 3.x+

- Familiar format of
 - Vertex / Fragment Shaders



Shader Model 3.x+

- Familiar format of
 - Vertex / Fragment Shaders
 - Geometry
 - Textures
 - Render Targets



Shader Model 3.x+

- Familiar format of
 - Vertex / Fragment Shaders
 - Geometry
 - Textures
 - Render Targets
- Reduce asset size, but keep same visual fidelity
- Can be up and running in a few weeks



Other Libraries to Simplify Development

- Vector Maths and Geometry
- Rigid Body Physics Simulation
- Animation
- Facial Recognition



Dynamic Lighting



Rich Environment Detail



Water Effects



Post Processing

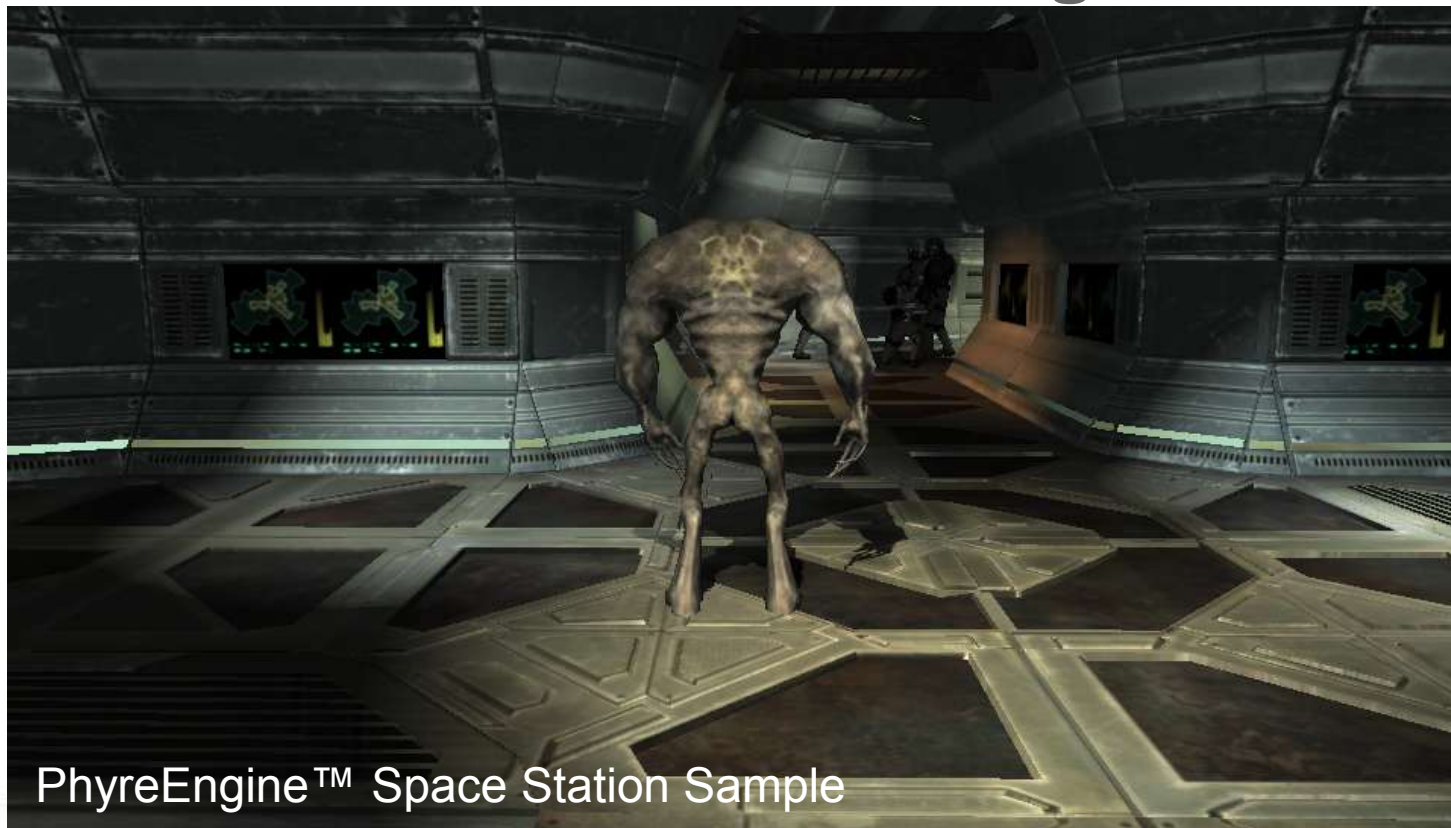
Motion Blur

Colour Correction

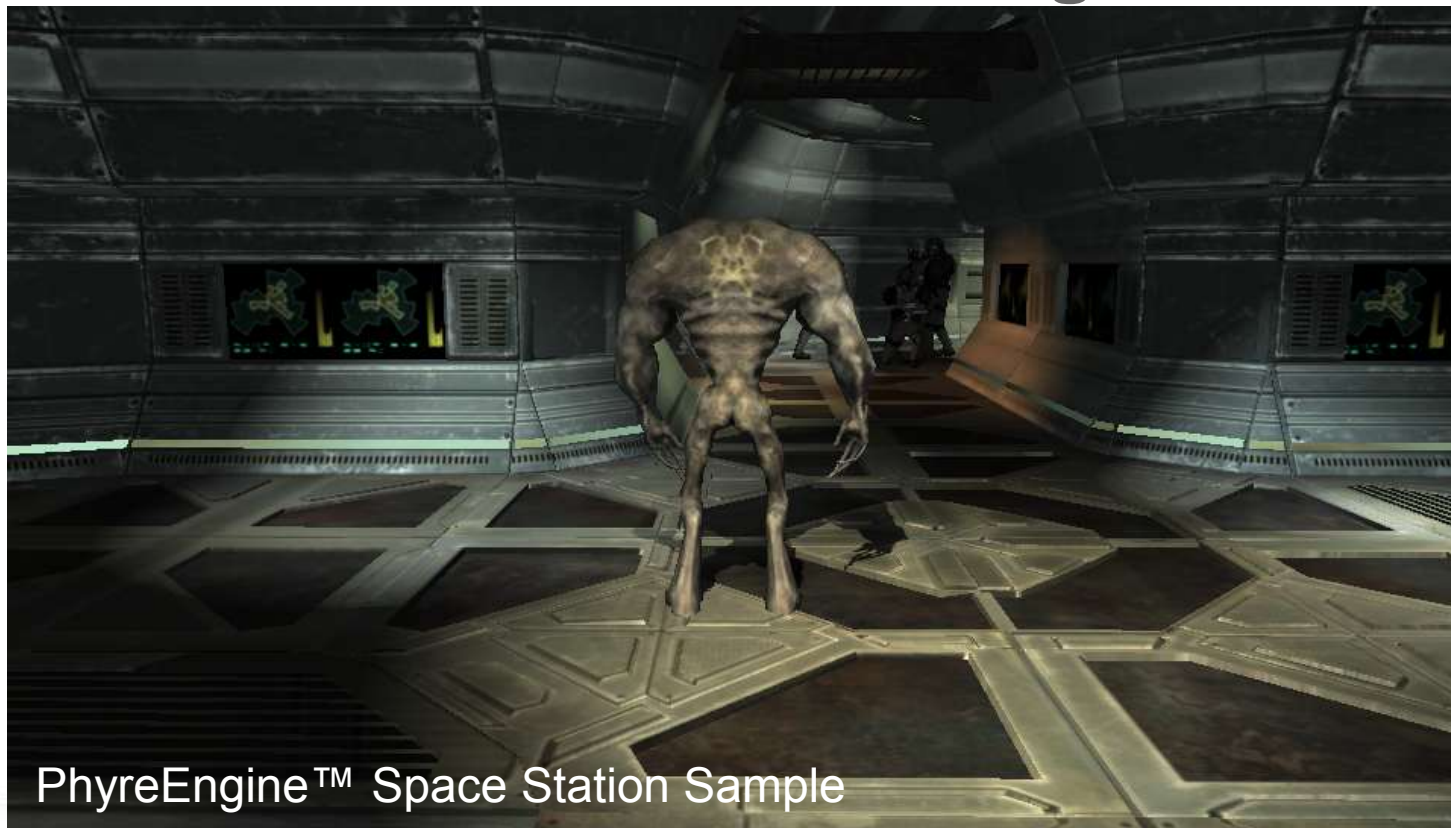
Bloom

Depth Of Field

Deferred Rendering



Deferred Rendering



PhyreEngine™ Space Station Sample





- PlayStation optimised game engine as source
 - PSP (version 2.6), PS Vita (version 3.0), PC & PS3 (2.7 and 3.0)
 - Used in fIOW™, GripShift™, Savage Moon, Flower and Burn Zombie Burn! and more than 60 other titles globally
- Provided as source
 - Portable, reusable
- Free to use, either full or in part
- Partner Programme
 - Integrating with middleware to ensure developers can choose the best solutions



Tools & Middleware Partners

- AiLive
- Allegorithmic
- Audiokinetic
- Autodesk
- Blitz Tech
- CRI
- Epic Games
- Firelight Technologies (FMOD)
- GameSpy Technology
- Havok
- NaturalMotion



- Nvidia
- Premium Agency
- RAD Game Tools
- RakNet
- SpeedTree
- Terminal Reality
- Trinigy
- Umbra
- Vicious Cycle
- Web Technology



VICIOUS ENGINE



RakNet
See You Online





PlayStation®Network



Some stats

- 55.5 Million PS3s
 - As of 30th September 2011
- 80% of users connected from PS3
- Over 1.91 billion items downloaded
 - As of October 2011



Evolution of PlayStation®Network

SCEE
Research &
Development



- Back in 2006 a very limited feature set
- Now grown into a fuller featured network services platform
- We still allow our developers to use their own services if they want



PSVITA
PlayStation Vita



Introducing PlayStation®Vita

- Wi-Fi - 802.11b/g/n
 - WEP, WPA-PSK, WPA2-PSK
- Location
 - Built in GPS (3G/Wi-Fi model only)
 - Wi-Fi location service support
- All the existing PlayStation Network features, and more
 - Increasing community on the console and in games



near

- Allows users to discover each others:
 - Locations
 - 5 Recently played games
 - Gifts that was registered with the above 5 recently played games
- Exposes users to unknown games
- Increases titles popularity



Gifts

- Does not have location information
- Made out of 3 parts:
 - Condition: Unit, Radius, duration, receiver restriction, probability
 - Information: Icon, name and description
 - Body part: up to 100KB of binary data
- 1 per game



Use cases

- Gifting challenges to nearby users
- Instead of items, place challenges
- Award user with in-game rewards if the challenge is completed
- Gifting in-game items
- Place one or multiple items in the gift



Party

- Group players together
 - Different party groups for a game/genre
 - Discuss the game through text and voice chat
 - Doesn't need players to be playing the same game at once
- Party provides
 - Online player grouping for friends
 - Cross title experience
 - Voice chat and messaging between party members



LiveArea™ and Activity

Index mode



 ↑↓ Tap

Live mode



↔ Flick



↔ Flick



 ↑↓ Tap

Game mode



LiveArea Zones



Content
Information Zone
--> Live Item

Communication Zone
--> Activity from publisher



LiveArea and Activity

- An updatable interface for launching games
- A way for players of games to discuss their progress
- Updatable through game APIs



System Activity

- Automatically added:
 - Game start
 - Trophy acquisition
 - Store purchase
 - Store rating



Publisher Updates

- Ability for publishers to push data to both zones
 - Maintenance announcements
 - Informing users of new DLC
 - Online gaming events
 - News about game to encourage users to continue playing



Summary



PSVITA

PlayStation Vita



Summary

- Created from the ground up for developers



Summary

- Created from the ground up for developers
- Rich set of interfaces



Summary

- Created from the ground up for developers
- Rich set of interfaces
- Hardware does the hard work



Summary

- Created from the ground up for developers
- Rich set of interfaces
- Hardware does the hard work
- Shader Model 3.x+
- Powerful and intuitive tools



Summary

- Created from the ground up for developers
- Rich set of interfaces
- Hardware does the hard work
- Shader Model 3.x+
- Powerful and intuitive tools
- Developer Friendly



Developing for PS Vita: PlayStation Suite



Concept

- “Application Store” Model

- An open environment
 - » New approach for SCE, targeting all ranges of developers (traditional developers to independent developers)
- Casual games as well as non-game applications
 - » Lightweight content approval process
- Distinct from existing PSN content but available on all PS Certified devices



- PlayStation Suite SDK

- Cross platform / cross device, binary level compatibility
- Helping to address device fragmentation issues



PlayStation Certified

“PlayStation Certified” is a logo to identify devices that are able to run PS Suite content

- Primarily smart phones and tablets
 - Not just Sony devices
 - Not just Android devices – PS Vita
 - Not just future devices – PS3* etc.



SDK Licensing

- Target the full range of Developers – Homebrew all the way up to professional
- PS Suite SDK
 - C# as primary programming language
 - Program will sit on top of a virtual machine which is supported by PS Certified Android devices, PS Vita etc.
 - Includes 3D graphics libraries for games and UI Toolkit for non-game application development
- Nominal fee to cover some of our platform operational costs



Content Submission and Approval Process

- Objectionable content – Guidelines and takedown procedures
- Review / Approval – Exploring the balance between quick iteration times and a curated experience
- Publishing Process – Self service, single submission



PS Suite area of PlayStation Store

- Separate storefront that spans platforms
- Focus on content discovery and promotion
- Revenue share will be competitive with other platforms
- In-game purchases will be allowed (with revenue share)
- Capability to tune for market conditions
 - » Sales metrics
 - » Price adjustment
- Return policy is under discussion



Developer Community

- Community-driven, forum-based support
- Sharing-friendly
 - Create demos, samples, or libraries and freely share them with anybody
 - Talk about PS Suite development openly and freely with anybody on forums, blogs, etc.



Roadmap

- Closed Beta release (“tech preview”) - November 2011
 - » Includes core technology and all libraries
 - » Android version
 - » PC-based Simulator for use without an Android device
 - » Gather feedback from the community
- Open for business - date to be announced





Technical Overview



Programming language

- Main programming language is C#
 - » Modern, high-level programming language
 - » Runs on a Mono virtual machine which provides direct binary compatibility across devices
 - » Compile a single .exe file, then run it on any of the supported devices without modification
- Native code is not allowed
 - » Provides better security, more predictable output, and cross-platform portability
 - » Performance is excellent for a wide range of games and applications
- SCE may support other languages in the future
 - » Based on user feedback and if it makes sense...

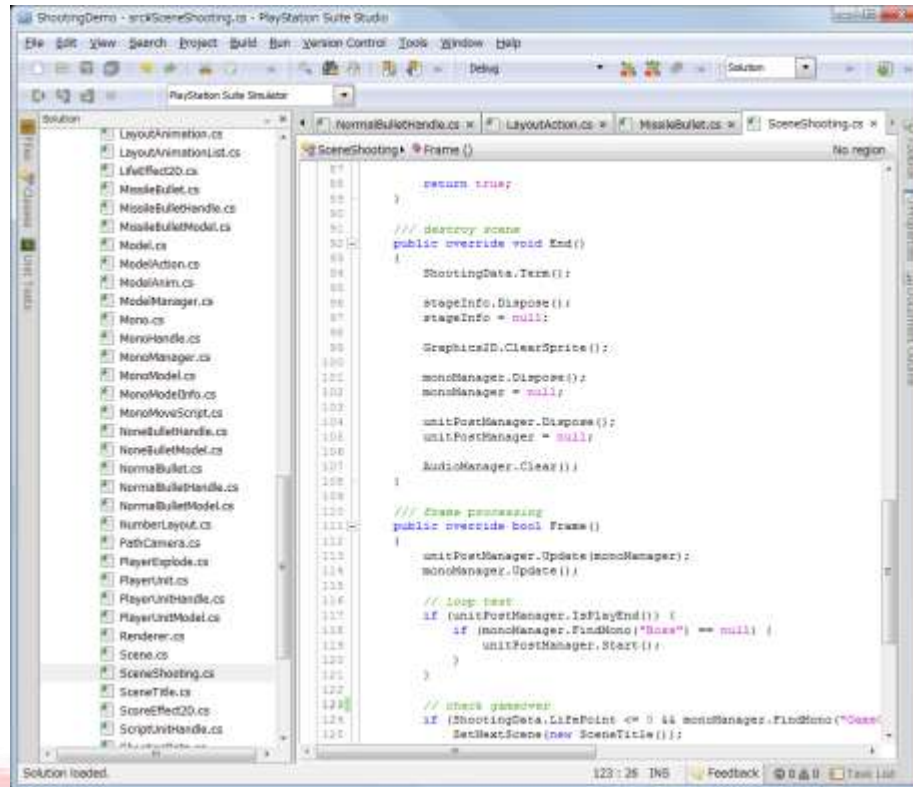


Demo

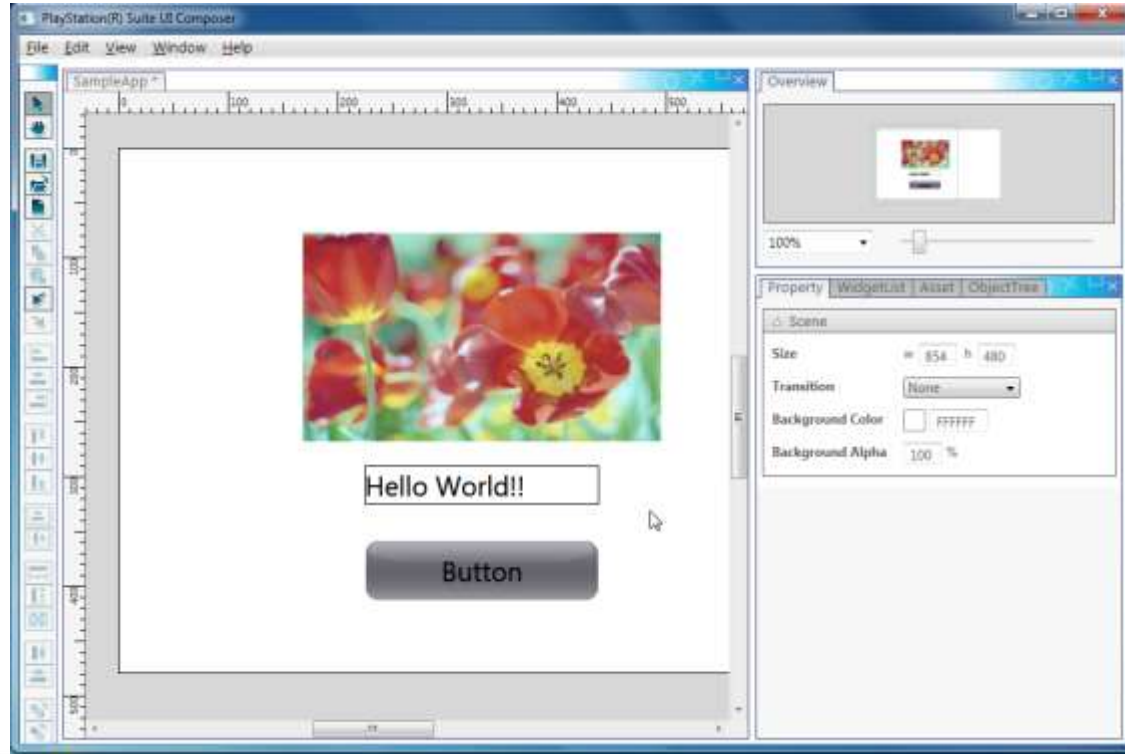
- Devices shown (the same .exe is running on every device!)
 - PC Simulator
 - PS Vita
 - Sony Ericsson Xperia Arc
- Tools
- PS Suite Studio (IDE)
- PS Suite UI Composer



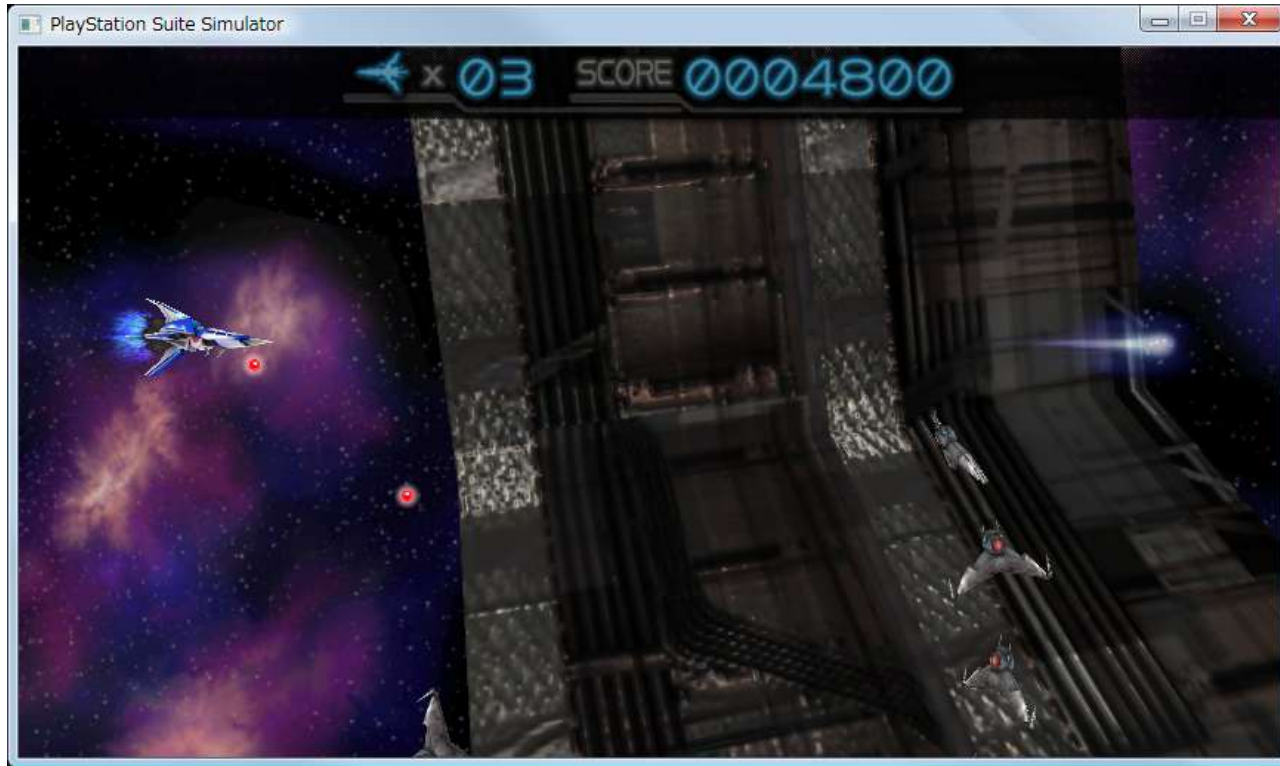
PS Suite Studio (IDE)



PS Suite UI Composer (UI Design Tool)



Screenshot – PC Simulator



Virtual machine

- Runs on a Mono™ virtual machine
- A custom implementation of the Common Language Infrastructure (ECMA-335)
 - » <http://www.ecma-international.org/publications/standards/Ecma-335.htm>
- Base Class Library (BCL) is API compatible with the core of the Portable Library project with some extensions to provide a better developer experience
 - » <http://msdn.microsoft.com/en-us/library/gg597391.aspx>
- Supported items
 - » Basic types like Array, String, Collections
 - » File I/O
 - » Threading
 - » Sockets
 - » Http
 - » Xml
 - » Etc...



- Style is very straightforward
 - Class-structured
 - Simple
 - Easy to understand

```
// teapot ↵
vbTeapot = new VertexBuffer( meshTeapot.VertexCount, ↵
                             meshTeapot.IndexCount, ↵
                             VertexFormat.Float3, ↵
                             VertexFormat.Float2 ); ↵

vbTeapot.SetVertices( 0, meshTeapot.Positions ); ↵
vbTeapot.SetVertices( 1, meshTeapot.TexCoords ); ↵
vbTeapot.SetIndices( meshTeapot.Indices ); ↵

// shadow map(simple) ↵
shaderShadowMap = new ShaderProgram( "shaders/Simple.cgx" ); ↵
shaderShadowMap.SetAttributeBinding( 0, "a_Position" ); ↵

// texture ↵
shaderTexture = new ShaderProgram( "shaders/Texture.cgx" ); ↵
shaderTexture.SetAttributeBinding( 0, "a_Position" ); ↵
shaderTexture.SetAttributeBinding( 1, "a_TexCoord" ); ↵
texColorMap = new Texture2D( "data/renga.png", false ); ↵
texColorMap.SetWrap( TextureWrapMode.Repeat ); ↵
```



Core APIs

- Graphics
 - OpenGL ES 2.0 equivalent
- Audio
 - SoundEffect
 - Bgm
- Input
 - Game Pad
 - Touch
 - Motion
 - PersistentMemory
- Imaging
 - Image Processing
 - Font
- VectorMath
 - Vector/Matrix calculation
- Environment
 - Clipboard
 - CommonDialog (TextInput, etc...)
 - Shell
 - SystemEvent
 - SystemParameters
 - Storage



Shader language

- Developers can write their own custom shaders (vertex and fragment)
- Enables modern graphics programming across all platforms
- Basic shader samples are contained in the SDK



High-level APIs

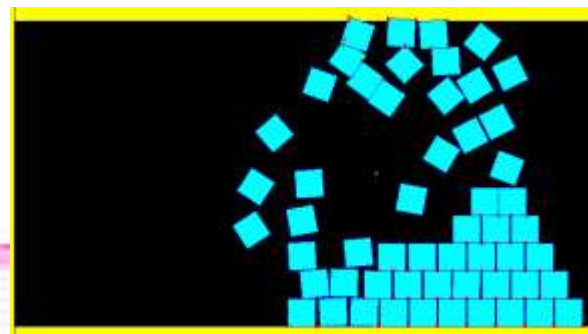
- Model
 - Model rendering/animation



- UI Toolkit
 - Widget set



- Physics
 - 2D-based physics library



Technology Roadmap

Features coming in the future :

- Visual Studio integration
- 2D game engine

Features that are under discussion (not yet scheduled)

- Support for additional programming languages
- Low-level audio API
- Camera API
- Location API

Other features to be driven by community feedback



Conclusions



Conclusions

- PS Vita is an exciting new platform for developers
- Two complementary development paths:
 - Traditional console approach
 - PS Suite approach
- Which one will you choose ?



Any Questions ?



Useful Links

- **tpr_registration@scee.net** - 1st port of call in becoming a registered developer
- **<http://www.worldwidestudios.net/xdev>** - For registered developers who wish to propose titles to SCEE XDEV
- **SCEA PubFund** - Please contact your SCEE account manager who will reach out to SCEA Developer Relations who oversee PubFund (open to all developers globally)
- **<http://research.scee.net/>** - SCEE R&D public site
- **<http://www.playstation.com/pss/closedbeta/>** - PSS Closed Beta

