Unity for PlayStation®Mobile

Erik Hemming
Unity Technologies
Mobile Mechanic / PlayStation Protégé
& Lead Developer of “Strategic Technology”

Agenda

- What is PlayStation®Mobile?
- What is Unity-for-PSM?
  - ‘Build & Run’
  - Performance
  - Rendering / Input / Script APIs / Plugins
- Publishing & In-App Purchase
- License
- Roadmap

What is PlayStation®Mobile?

PlayStation®Mobile Terminology

- PSM - a platform and a business model
  - Publisher license is open with annual fee (free as of now)
  - Allows self-publishing to PS Store
- PSM Runtime
  - Software framework running on the device
- PSM SDK
  - Development tools and APIs available to create content for PSM
- Unity-for-PSM
  - A different set of runtime & SDK also targeting the PSM platform

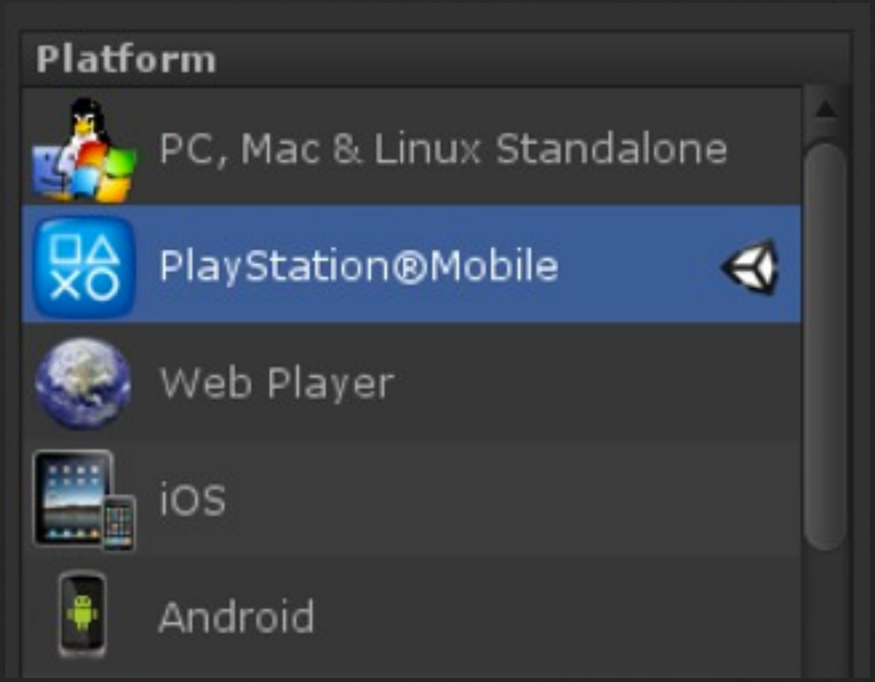
PlayStation®Mobile SDK

- PSM SDK utilizes the PSM runtime/toolchain developed by Sony Computer Entertainment Inc.
  - IDE (Editor / Debugger) - PSM Studio (customized MonoDevelop)
  - C# API geared towards game development
  - Runtime for Android (2.3+) and PS Vita (and emulator under Windows)
- Released in April, 2012 (open beta) - public release in October
- Based on the Mono platform (open source C# VM)
- Uses a “Development Assistant” app for debugging/testing content on Android/PS Vita

What is Unity-for-PSM?

Unity for PlayStation®Mobile

- A new way of targeting the same PSM platform
- Developed as a collaboration effort between Unity and SCE
- A new target platform from the Unity Editor
  - Similar to Unity's support for iOS/Android and the Unity Webplayer
- Uses a separate runtime (Unity runtime vs. PSM runtime)
  - Runs only on PS Vita
- Public Preview available now



Unity for PlayStation®Mobile

- Runs Unity authored content in a sandboxed environment
  - more on that later
- Doesn’t allow native code - only C# / UnityScript / Boo
- Separate runtime - doesn’t reuse any of the PSM runtime
  - Doesn’t run on top of PSM runtime - uses it’s own optimized Unity runtime
- Separate script API
  - provides the same Unity scripting API available on other platforms.
- Separate toolchain - but reuses the SEN ID from PSM platform.

PSM SDK
vs.
Unity-for-PSM
vs.
Unity PS Vita / Unity Android

PSM SDK vs. Unity-for-PSM

- Separate toolchain
  - Unity editor + PSM Add-on (provided by Unity)
  - PSM Toolset for Unity (provided by SCE)
  - MonoDevelop (still in development)
- Separate runtime
- Separate Development Assistant
- Different scripting API
  - Sce.PlayStation.Core.* vs. UnityEngine.*
- Only support for PS Vita

Unity PS Vita (business side)

- Need to incorporate (register as a company)
- Need to sign NDAs
  - Closed, development cannot be discussed publicly
- Need to purchase separate devkit hardware
  - With PSM no need for devkit - development with consumer unit
- Native PS Vita development
  - has a higher cost (fee/devkit/etc)
  - has more process (TRC)
  - and hard(er) to self-publish
- Unity-for-PSM does not have any of these requirements

Unity PS Vita (technical side)

- Full native access including full PSN access
- Full suite of performance and (native) debugging tools
  - Visual Studio Integration
  - Razor CPU/GPU performance tools
- Unity-for-PSM does not have any of these features
  - Instead Unity Profiler / MonoDevelop bridges this gap (somewhat).
- Other technical differences
  - Pre-compiled shaders vs. runtime-compile shaders
  - Mono script Ahead-Of-Time vs. Just-In-Time compilation

Unity-for-PSM vs. Unity Android

- Both uses JIT
- Both uses runtime compiled shaders
- But no native access with PSM
    - Instead similar to how the Unity Webplayer is implemented

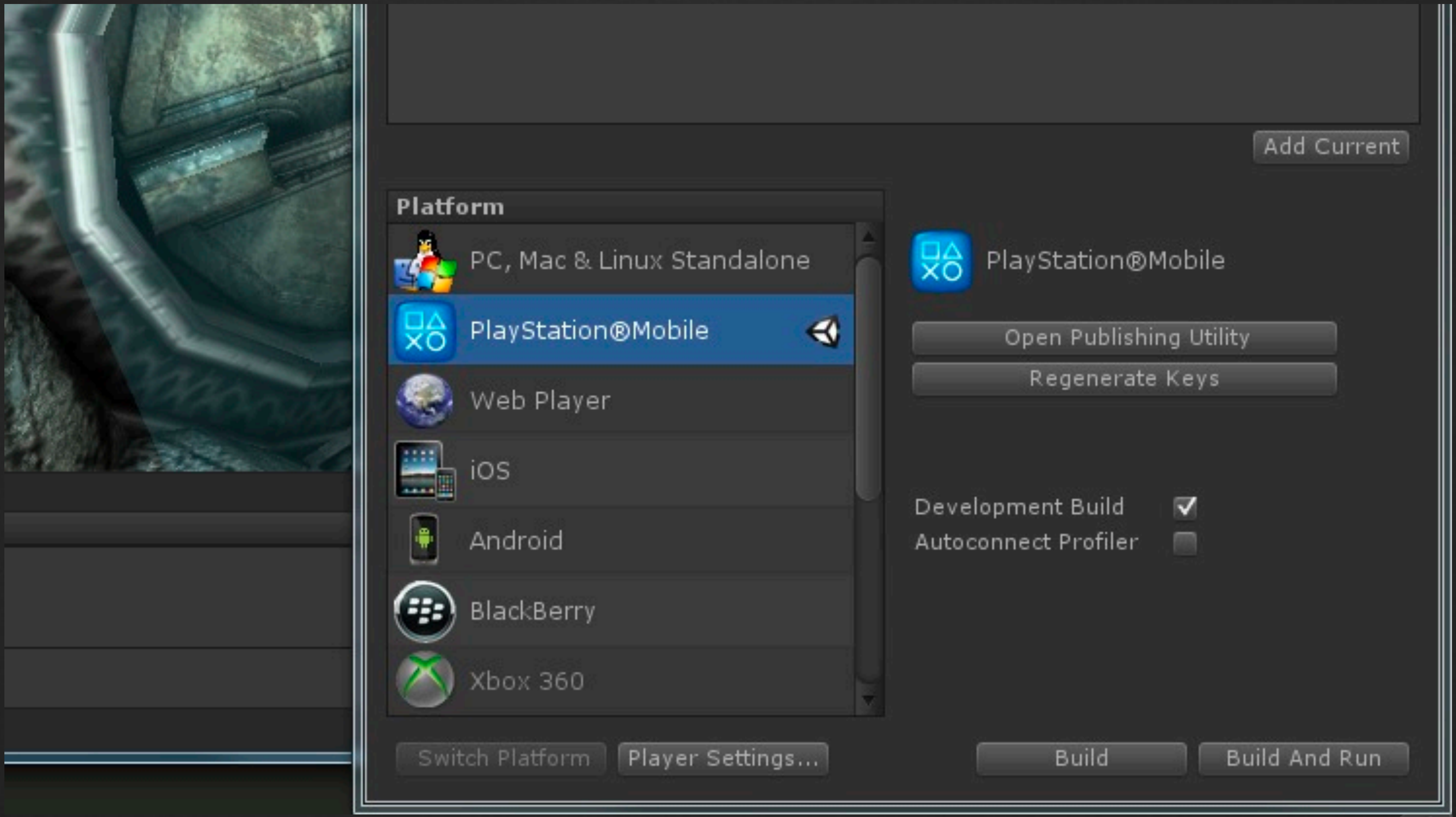
Unity for PlayStation®Mobile

True Portable Gaming Experience

- Dedicated Physical Controller
  - Dual analog sticks enable a wider range of game genres to be brought into the portable experience.
- Optimal form factor
  - Well designed oval form factor fits comfortably in user's hands.
- Multi-Touch 5-inch Display
  - 5-inch display with high definition can provide users deep and immersive gaming experience.
  - Back touch enables new gameplay styles
- And easily supported out-of-the-box with Unity

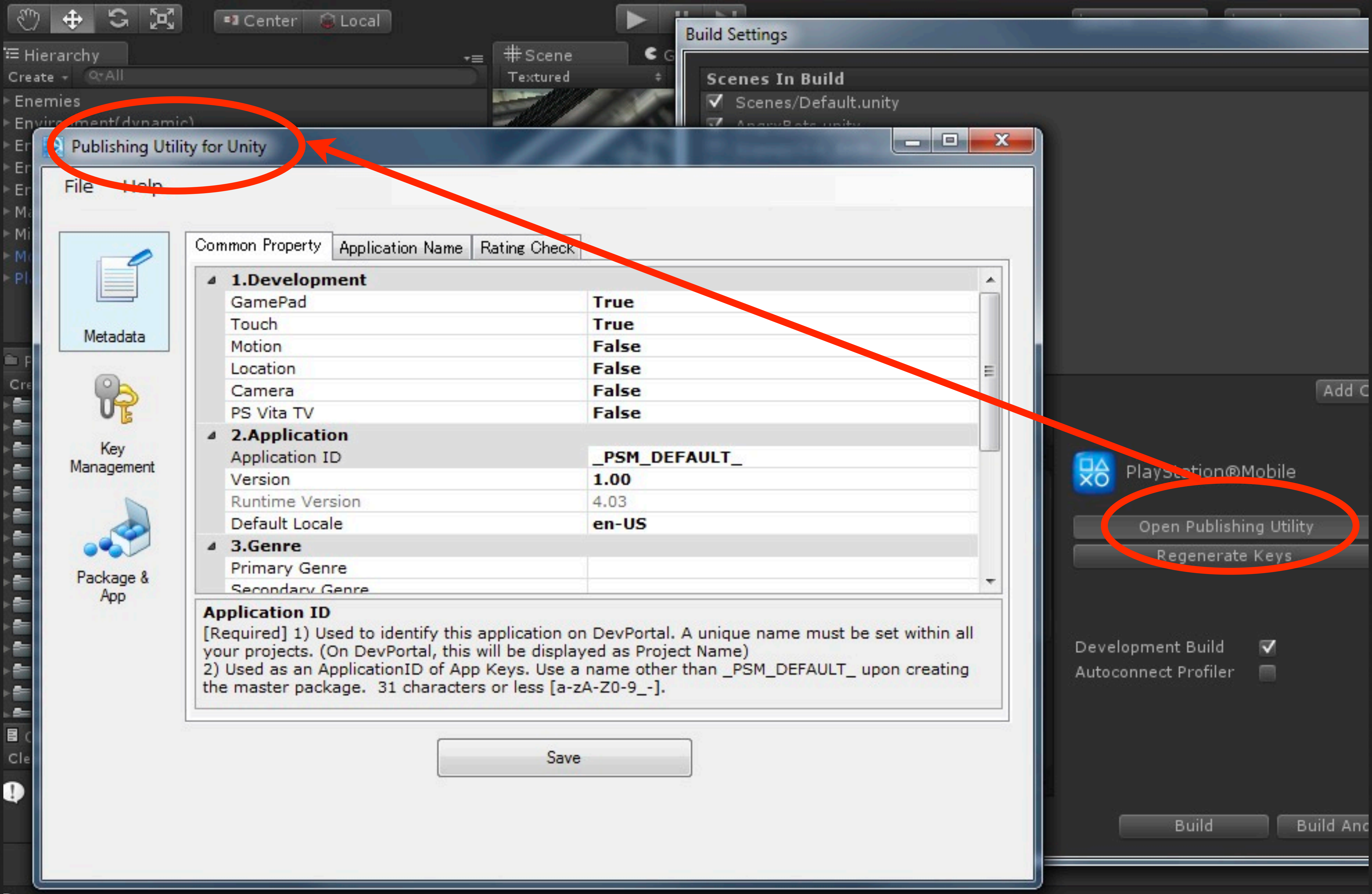
Unity's support for PlayStation®Mobile

- PSM adds a new platform



Development Build ☑
Autoconnect Profiler □

PSM Tool Set for Unity

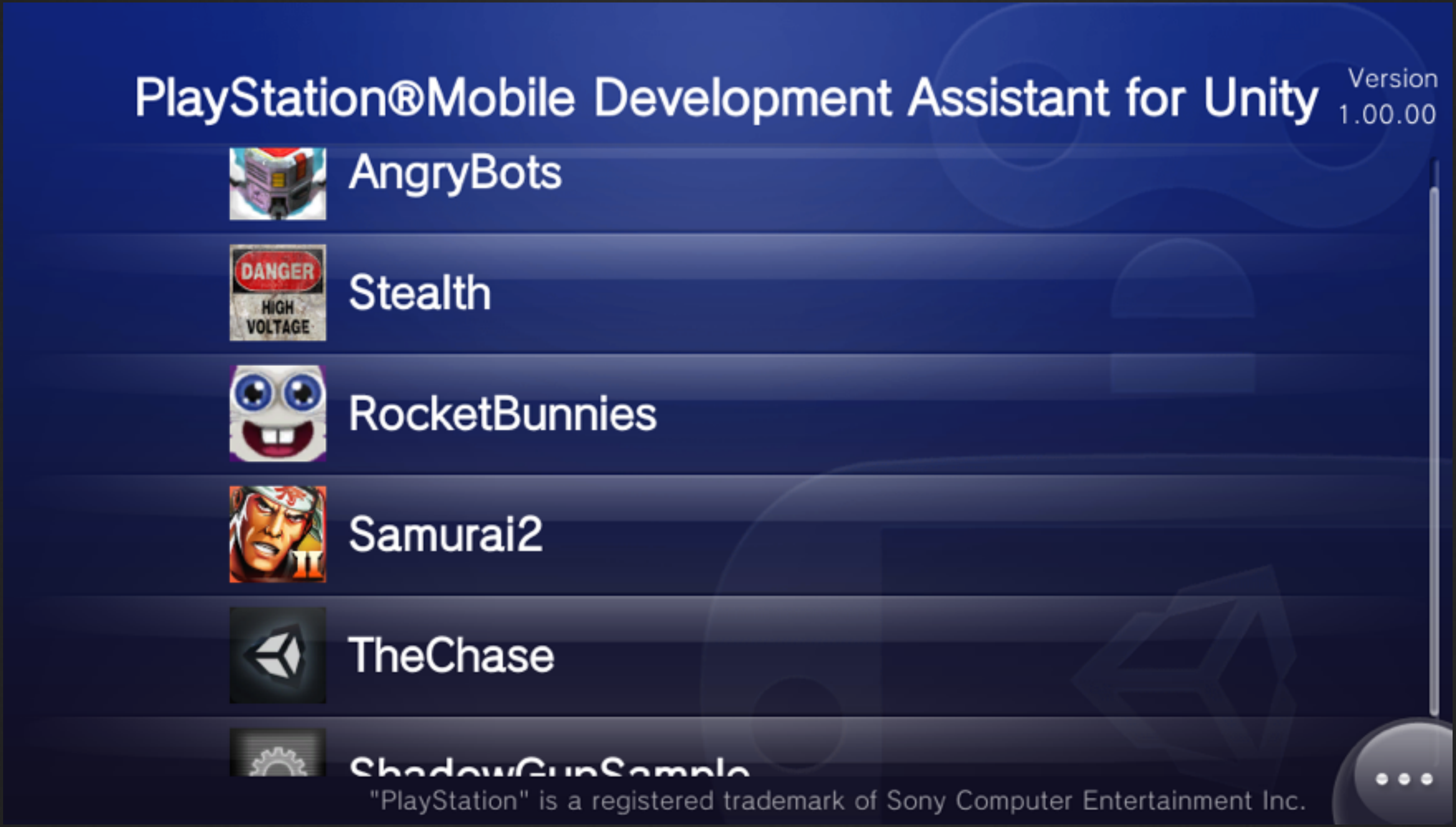


1.Development	
GamePad	**True**
Touch	**True**
Motion	**False**
Location	**False**
Camera	**False**
PS Vita TV	**False**
2.Application	
Application ID	**_PSM_DEFAULT_**
Version	**1.00**
Runtime Version	4.03
Default Locale	**en-US**
3.Genre	
Primary Genre	
Secondary Genre	

Application ID
[Required] 1) Used to identify this application on DevPortal. A unique name must be set within all your projects. (On DevPortal, this will be displayed as Project Name)
2) Used as an ApplicationID of App Keys. Use a name other than _PSM_DEFAULT_ upon creating the master package. 31 characters or less [a-zA-Z0-9_-].

Development Build ☑
Autoconnect Profiler □

PlayStation®Mobile Development Assistant for Unity



PlayStation®Mobile Development Assistant for Unity

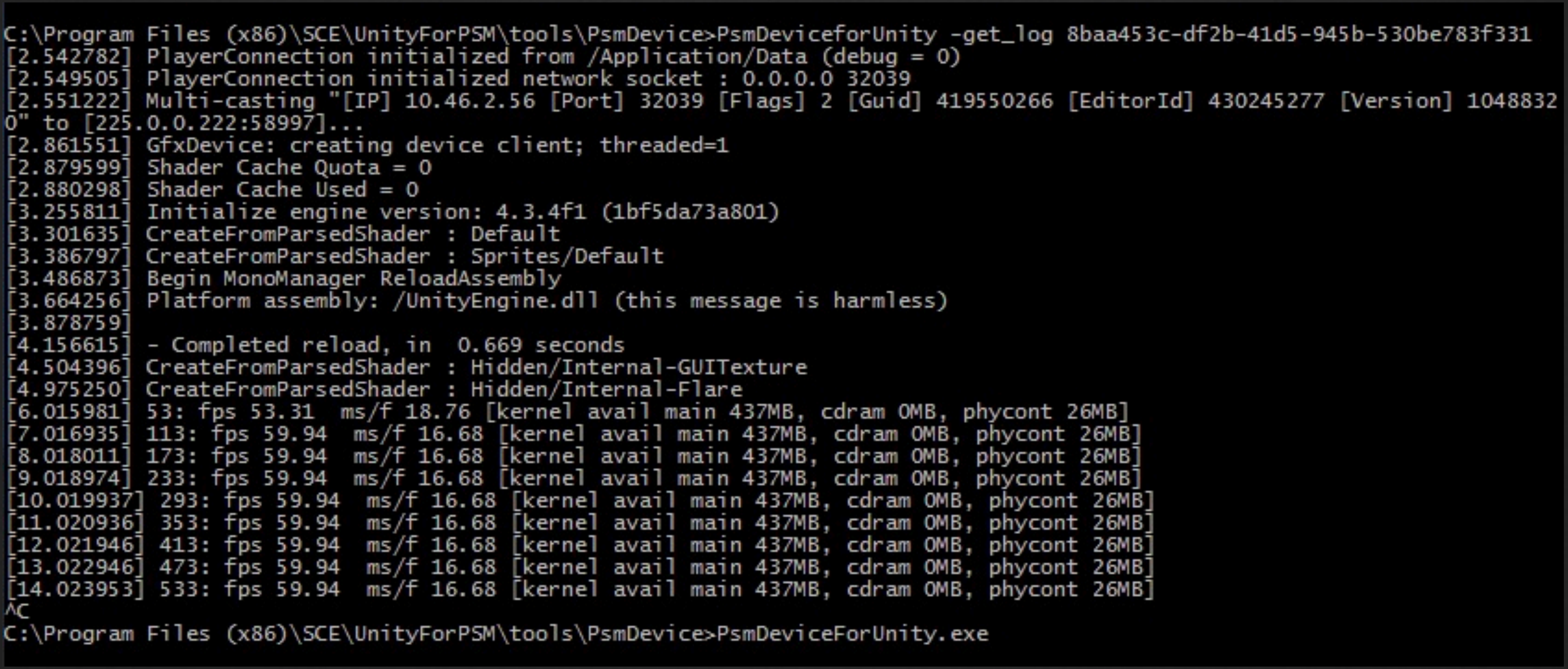
Version 1.00.00

- AngryBots
- Stealth
- RocketBunnies
- Samurai2
- TheChase
- ShadowGunSample

"PlayStation" is a registered trademark of Sony Computer Entertainment Inc.

PsmDeviceForUnity

- PsmDeviceForUnity.exe -get_log <GUID>



```
C:\Program Files (x86)\SCE\UnityForPSM\tools\PsmDevice>PsmDeviceforUnity -get_log 8baa453c-df2b-41d5-945b-530be783f331
[2.542782] PlayerConnection initialized from /Application/Data (debug = 0)
[2.549505] PlayerConnection initialized network socket : 0.0.0.0 32039
[2.551222] Multi-casting "[IP] 10.46.2.56 [Port] 32039 [Flags] 2 [Guid] 419550266 [EditorId] 430245277 [Version] 1048832
0" to [225.0.0.222:58997]...
[2.861551] GfxDevice: creating device client; threaded=1
[2.879599] Shader Cache Quota = 0
[2.880298] Shader Cache Used = 0
[3.255811] Initialize engine version: 4.3.4f1 (1bf5da73a801)
[3.301635] CreateFromParsedShader : Default
[3.386797] CreateFromParsedShader : Sprites/Default
[3.486873] Begin MonoManager ReloadAssembly
[3.664256] Platform assembly: /UnityEngine.dll (this message is harmless)
[3.878759]
[4.156615] - Completed reload, in  0.669 seconds
[4.504396] CreateFromParsedShader : Hidden/Internal-GUITexture
[4.975250] CreateFromParsedShader : Hidden/Internal-Flare
[6.015981] 53: fps 53.31  ms/f 18.76 [kernel avail main 437MB, cdram 0MB, phycont 26MB]
[7.016935] 113: fps 59.94  ms/f 16.68 [kernel avail main 437MB, cdram 0MB, phycont 26MB]
[8.018011] 173: fps 59.94  ms/f 16.68 [kernel avail main 437MB, cdram 0MB, phycont 26MB]
[9.018974] 233: fps 59.94  ms/f 16.68 [kernel avail main 437MB, cdram 0MB, phycont 26MB]
[10.019937] 293: fps 59.94  ms/f 16.68 [kernel avail main 437MB, cdram 0MB, phycont 26MB]
[11.020936] 353: fps 59.94  ms/f 16.68 [kernel avail main 437MB, cdram 0MB, phycont 26MB]
[12.021946] 413: fps 59.94  ms/f 16.68 [kernel avail main 437MB, cdram 0MB, phycont 26MB]
[13.022946] 473: fps 59.94  ms/f 16.68 [kernel avail main 437MB, cdram 0MB, phycont 26MB]
[14.023953] 533: fps 59.94  ms/f 16.68 [kernel avail main 437MB, cdram 0MB, phycont 26MB]
^C
C:\Program Files (x86)\SCE\UnityForPSM\tools\PsmDevice>PsmDeviceForUnity.exe
```

Build & Run

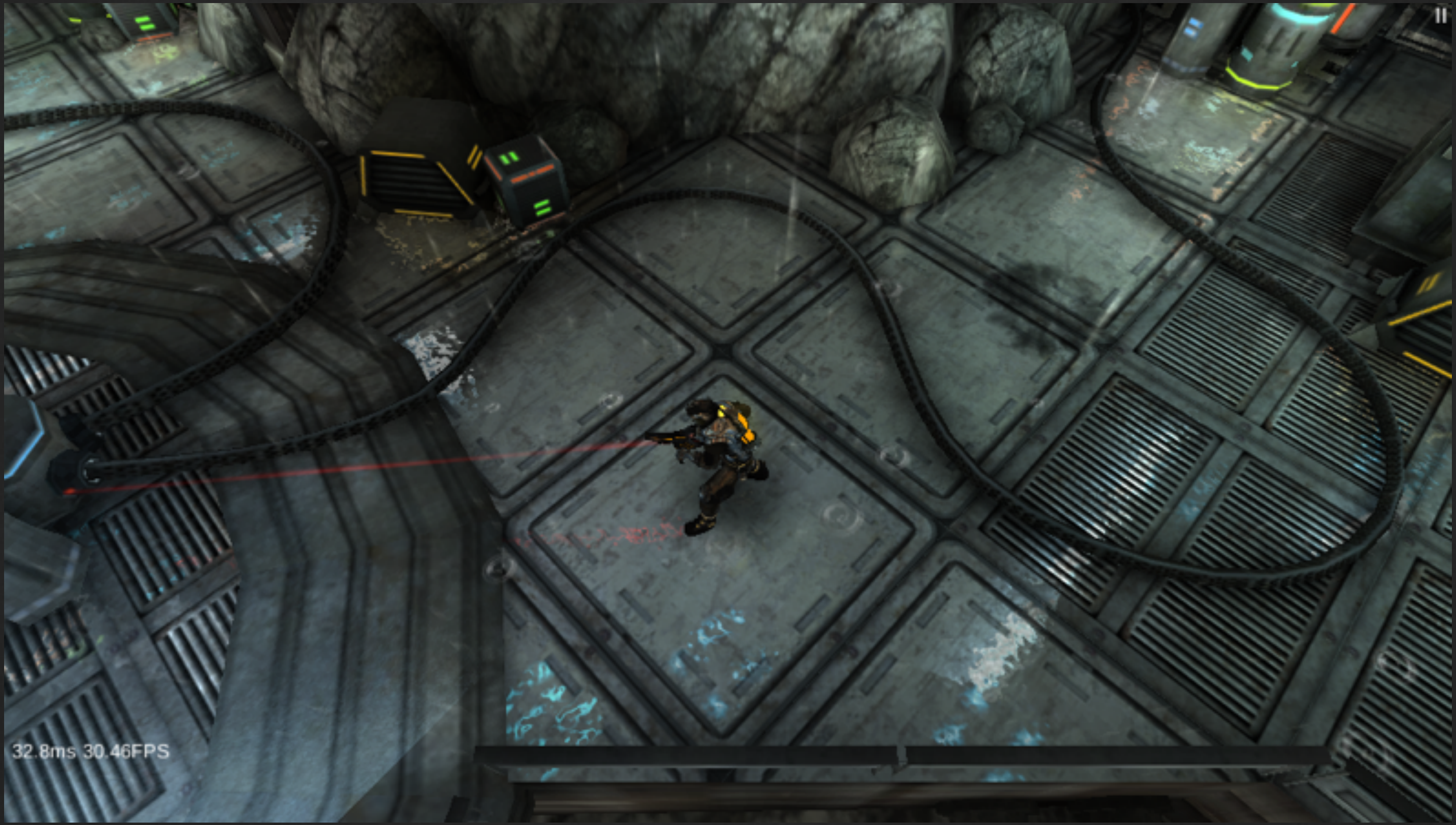
- Unity Editor will
  - Build all your level assets
  - Compile your game code (C#) into an ‘intermediate language’
    - ‘managed’ assembly .dll
  - Package it to a .psdp file (‘development package’)
  - Transfer the .psdp to the ‘Development Assistant’ (PS Vita)
- Development Assistant ‘plays back’ the content
  - Uses the Unity runtime (native) combined with User scripts (managed)

Performance of Unity for PSM

- PSM === PS Vita, except
  - Just-In-Time compilation of scripts to native code, “on demand”
    - When a method is accessed
  - Runtime compiled shaders
    - When loading a level (currently)
      - But cached - subsequent runs will be (much) faster
- Any optimizations done for PS Vita will benefit PSM automatically

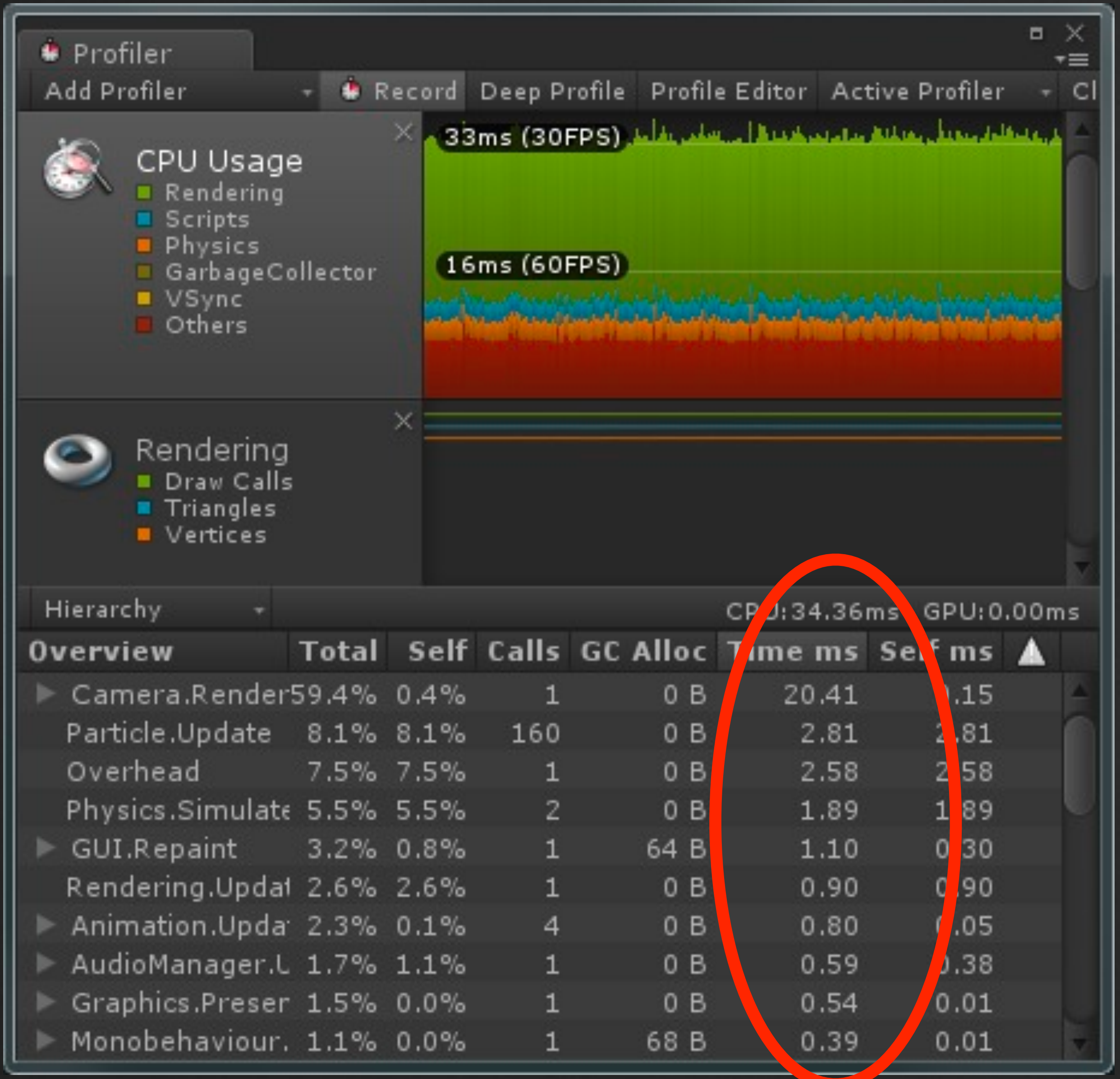
Performance of Unity for PSM

- Angry Bots - Unity example project (available on the AssetStore)



Performance of Unity for PSM

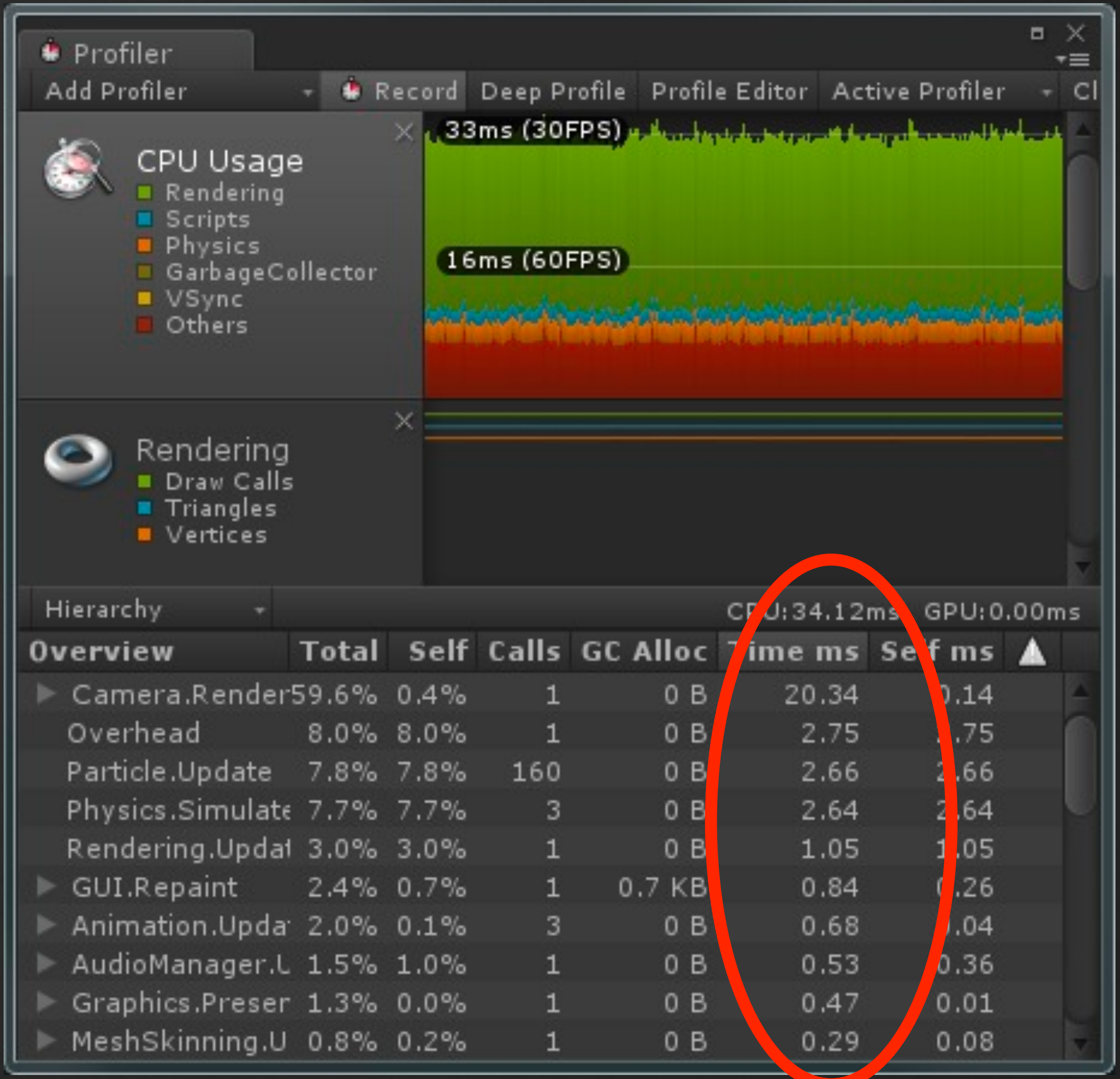
PS Vita (native)



CPU:34.36ms GPU:0.00ms

Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
▶ Camera.Render	59.4%	0.4%	1	0 B	20.41	0.15
Particle.Update	8.1%	8.1%	160	0 B	2.81	2.81
Overhead	7.5%	7.5%	1	0 B	2.58	2.58
Physics.Simulate	5.5%	5.5%	2	0 B	1.89	1.89
▶ GUI.Repaint	3.2%	0.8%	1	64 B	1.10	0.30
Rendering.Updat	2.6%	2.6%	1	0 B	0.90	0.90
▶ Animation.Upda	2.3%	0.1%	4	0 B	0.80	0.05
▶ AudioManager.U	1.7%	1.1%	1	0 B	0.59	0.38
▶ Graphics.Preser	1.5%	0.0%	1	0 B	0.54	0.01
▶ Monobehaviour.	1.1%	0.0%	1	68 B	0.39	0.01

Unity-for-PSM



CPU:34.12ms GPU:0.00ms

Overview	Total	Self	Calls	GC Alloc	Time ms	Self ms
▶ Camera.Render	59.6%	0.4%	1	0 B	20.34	0.14
Overhead	8.0%	8.0%	1	0 B	2.75	2.75
Particle.Update	7.8%	7.8%	160	0 B	2.66	2.66
Physics.Simulate	7.7%	7.7%	3	0 B	2.64	2.64
Rendering.Updat	3.0%	3.0%	1	0 B	1.05	1.05
▶ GUI.Repaint	2.4%	0.7%	1	0.7 KB	0.84	0.26
▶ Animation.Upda	2.0%	0.1%	3	0 B	0.68	0.04
▶ AudioManager.U	1.5%	1.0%	1	0 B	0.53	0.36
▶ Graphics.Preser	1.3%	0.0%	1	0 B	0.47	0.01
▶ MeshSkinning.U	0.8%	0.2%	1	0 B	0.29	0.08

Rendering

- Cg - not GLSL
- Runtime compiled shaders
- Fixed function pipe available
- Fixed hardware
  - Easier to optimize shaders
  - Knowledge obtained from developing on iOS (or Androids with PVR) applies.
- DXTn texture compression

Input

- Axis mapping
  - Joystick X/Y Axis = Left thumbstick
  - Joystick 4th/5th Axis = Right thumbstick
- Button mapping
  - Input.GetKey("joystick 1 button <N>")
- Touch input
  - Input.GetTouch(n) et al.
  - Currently no support for back touch
- Gyroscope / accelerometer also available

Script API specifics

- UnityEngine.Handheld
  - PlayFullScreenMovie from StreamingAssets/
  - Start-/StopActivityIndicator
  - No vibrate support
- UnityEngine.TouchScreenKeyboard
- UnityEngine.WebCamTexture
- UnityEngine.Microphone
- New Platform Define
  - #if UNITY_PSM

Plugins

- Only managed plugins
  - C# / UnityScript / Boo
  - Pre-built managed assembly .dlls
  - Not native code
- Existing AssetStore plugins
  - PlayMaker
  - Photon
  - SmartFox
  - …

Publishing

- Not available yet
- Similar to ‘Build & Run’ workflow
- Will create a ‘master package’ (.psmp)
- Submit package
  - Process equal to that of regular PSM SDK
- Available on the PS Store
  - No different from publishing using the PSM SDK
  - Stand-alone application - no separate player installation needed

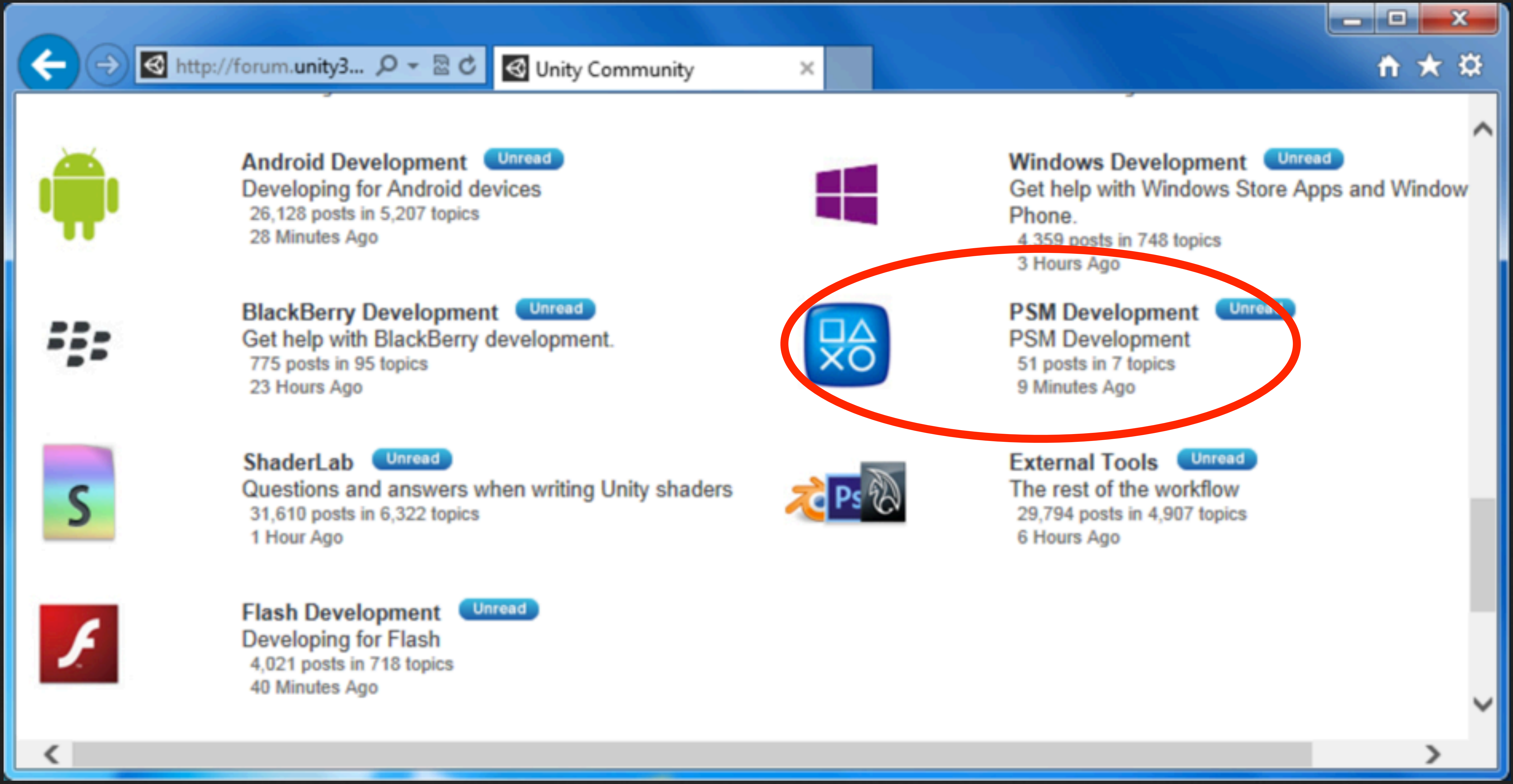
In App Purchases

- Not available yet
- Process similar to regular PSM SDK
  - IAP Goods declared through the Publishing Utility
  - Supports Free-to-Play model.
  - Purchasing is available through the PlayStation®Store.
- Integrated inside the editor
  - Testing can be done in the editor
  - No need to deploy
  - Saves iteration time

License cost

- In essence free
  - Follows Unity Free / Pro licensing
- Unity Free → PSM Basic
- Unity Pro → PSM Pro
- Main differences:
  - Profiler
  - Network socket APIs
  - Splash screen
  - …

Unity Forums - for PSM



Android Development Unread
Developing for Android devices
26,128 posts in 5,207 topics
28 Minutes Ago

Windows Development Unread
Get help with Windows Store Apps and Window Phone.
4,359 posts in 748 topics
3 Hours Ago

BlackBerry Development Unread
Get help with BlackBerry development.
775 posts in 95 topics
23 Hours Ago

PSM Development Unread
PSM Development
51 posts in 7 topics
9 Minutes Ago

ShaderLab Unread
Questions and answers when writing Unity shaders
31,610 posts in 6,322 topics
1 Hour Ago

External Tools Unread
The rest of the workflow
29,794 posts in 4,907 topics
6 Hours Ago

Flash Development Unread
Developing for Flash
4,021 posts in 718 topics
40 Minutes Ago

Roadmap

- Public Preview
  - Available now
  - Based on Unity 4.3
- Official Release
  - Summer 2014
  - Based on Unity 4.3
- Unity 5
  - Full integration with regular editor
  - Summer / Fall 2014