Unity跨平台移植分享

Dafu Lei
Senior Solution Engineer, Unity
dafu.lei@unity3d.com

Agenda

- 跨平台问题及通用解决方法
- Unity目前已有的加速方案
- 单个平台移植的基本流程
- Unity引擎相关的优化
- 总结

Multi-platform development is critical

“Author once, deploy everywhere.”

The most successful games are multi-platform, and increasingly also cross-play.

It’s becoming the obvious choice for more studios.

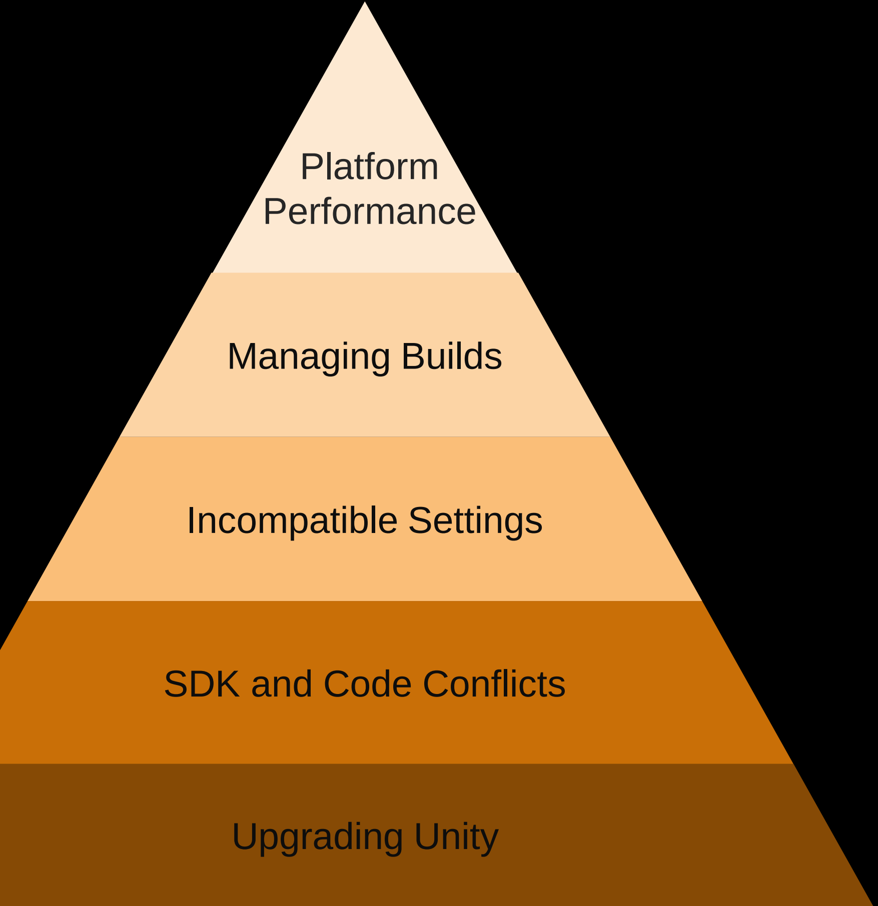
But there are major pain points for users.

跨平台问题及通用解决方法

The Multi-platform Journey

(aka “mountain of pain”)

Multi-platform Obstacles



- Platform Performance
- Managing Builds
- Incompatible Settings
- SDK and Code Conflicts
- Upgrading Unity

SDK and Code conflicts

Issues:

- *SDKs*
- *packages*
- *DLL problems*
- *third-party’s libraries and its dependencies for each platform*
- *third-party’s rule etc.*

SDK and Code conflicts

Solution:

- ✓ Fork the project
- ✓ Branch the project in VCS(Version Control System)
- ✓ Use Platform Define Symbols
- ✓ Exclude files from import
- ✓ Packages selectively applied for specific platforms
- ✓ Ability to use >1 version of a package

Incompatible Settings

Issues:

- *platform devices require lots of different libraries*
- *platform devices require lots of different graphics settings*
- *different project settings*

Incompatible Settings

Solutions:

- ✓ Write custom scripts to change static settings
- ✓ Create build steps that run pre/post build

Managing Builds

Issues:

- *developers represent the build data as scriptable objects, which represent each state that you can build to*
- *developers might need to share certain data between builds that are for the same environments (e.g. staging and production, etc)*

Managing Builds

Solutions:

- ✓ Manage config data outside Unity, apply during build steps
- ✓ Create a custom UI/Tool
- ✓ Download a tool from asset store

Platform Performance

Issues:

- *developers might have internal libraries that handles vast amount of platform stuff, e.g. how console users are different from mobile*
- *it's hard to understand why two platforms might have different build sizes. being able to diff/compre build reports would be useful*

Platform Performance

Solutions:

- ✓ Use a build report plugin or write your own
- ✓ Build reporting to reason about what goes into a build
- ✓ Platform resource budgeting
- ✓ Rich content and code profiling on device

❖ Platform Performance	✓ Platform Profiling + Build Reporting
❖ Managing Builds	✓ Build Configurations
❖ Incompatible Settings	✓ New Project Settings + Verified Platform Defaults
❖ SDK and Code Conflicts	✓ Exclude files from import

Unity目前已有的解决方案

- Features we have
- Long term: Unlocks Innovation in multiple areas

Features we have

- ✓ NetCode/Multiplayer
- ✓ Auto Streaming + Addressables
- ✓ 分布式导入
- ✓ 分布式构建Assetbundle
- ✓ 分布式光照贴图烘焙
- ✓ 分布式编译il2cpp
- ✓ XR SDK 集成
- ✓ 企业版China Cloud Build部署

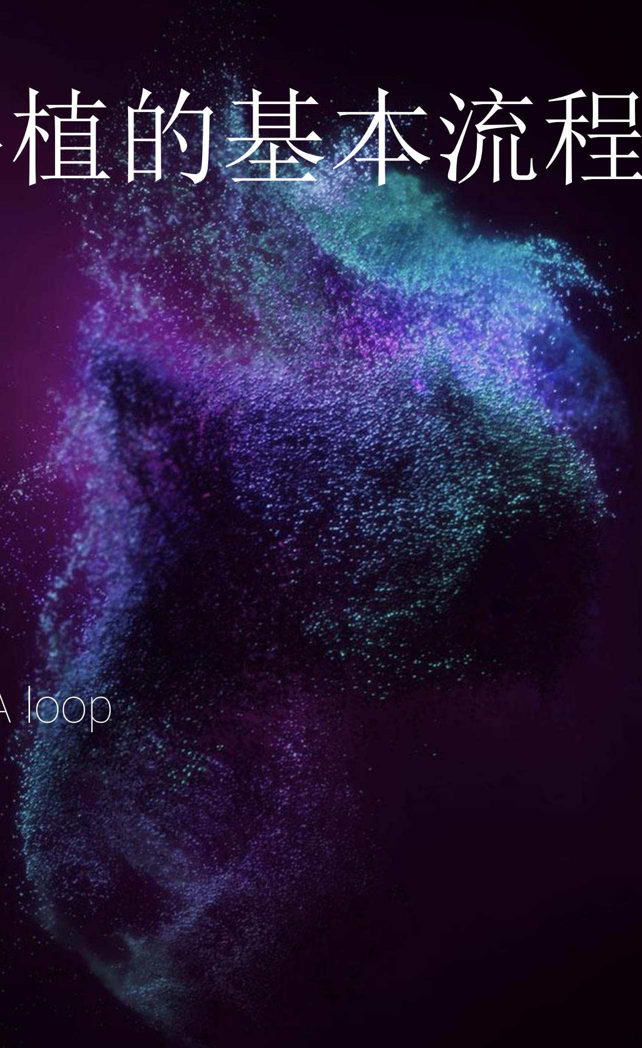
Long term: Unlocks Innovation in multiple areas

Multi-platform Development Foundation:

- Better XR device user experience/ support
- Smoother integration with Cloud Content Delivery (CCD) + Addressable
- Supports Cloud Build and other C.I. systems
- Smoother integration with Unity Distribution Portal
- More flexible configuration = better quality game MWU

单个平台移植的基本流程

- Pre-porting
- Porting
- work loop
    - Bug fix-QA loop
    - Optimization-QA loop
- Submission



Pre-porting

Preparation!

- develop device
- developer account
- develop environment (hardware & software) setup
  - unity version
  - people
  - SDKs
  - software
  - network
  - etc.

Pre-porting

Pick up on it!

- device characteristics: CPU, GPU, Memory, OS, File System etc.
- development documentation reading
- try to build and run with an empty project correctly on your device

Porting

The first step is always difficult!

- switch project to platform
- various error-fixing
- build
- run

Porting

Get through!

- stabilization: get rid of crash, freeze
- bugs: contents bugs, platform/rendering related (upside down, flickering, broken picture, black edges, and incorrect hollow, etc.)
- build
- run

Bug fixing-QA Loop

```
while (!buglist.empty)
{
        fix buglist

        QA-test
        {
                buglist++ or buglist–
        }
}
```

Optimization-QA Loop

```
while (true)
{
        profiling fps, memory, loading time, heat...

        coding...

        QA-test

        if (performance is ok)
                break;
}
```

Optimization-QA Loop

Profiling, Profiling, Profiling!

- Unity profilers

  Unity profiler, Memory profiler(package), Frame Debugger etc.

- platform's profiler

  iOS: instrument (timeline, allocation, etc.), GPU frame capture

  Android: system tracing, Arm Mobile Studio, Snapdragon Profiler, GAPID

  PC: Intel's VTune Amplifier (C++) , 3DMark

  console own its profiler

- UPR
- Unity official PR (Project Review)

Unity引擎相关的优化

- Use latest version
- Core and thread
- Unity source code optimization
- Take advantage of platform specific feature

Use Latest Unity

Profits:

- 2020.2 Editor performance big improvement
- Fix main thread wait preload thread issue
- Incremental GC
- Use DynamicAllocator
- GetShaderTagID
- Utif8
- etc.

Core and thread

Maximize!

- Control work thread numbers
- Do not migrate heave threads
- Migrate threads with low CPU usage
- Examine generally when any locks on the main thread actually does occur
- Use TempAlloc / JobTempAlloc

Unity source code optimization

Understand Source-code/use Unity with new version :

- SIMD improve FindeIndexOfValueInArray performance
- BucketAllocator for gfx thread to get rid of mutex
- etc.

总结

- 跨平台问题及通用解决方法
- Unity目前已有的加速方案
- 单个平台移植的基本流程
- Unity引擎相关的优化

References

- https://docs.unity3d.com/Manual/SpecialFolders.html
- https://docs.unity3d.com/Manual/SL-PlatformDifferences.html
- https://upr.unity.com
- https://developer.apple.com/documentation/metal/frame_capture_debugging_tools
- https://developer.android.com/topic/performance/tracing
- https://gapid.dev/about
- https://www.arm.com/products/development-tools/graphics/arm-mobile-studio
- https://developer.qualcomm.com/software/snapdragon-profiler

Thank you!