

UE4 모바일 게임 출시를 위해 궁금하실 바로 그것들!

신광섭

Developer Relations Lead/Programmer
Epic Games Korea

UNREAL
ENGINE

UNREAL SUMMIT 2016

목차

- 왜 이 콘텐츠가 쿠킹되고 포함되지? 앱사이즈 줄여보기
- 4.11 기준으로 다시 살펴보는 패치 시스템
- 알아두면 유용한 소소한 팁들

UNREAL
ENGINE

UNREAL SUMMIT 2016

왜 이 콘텐츠가 쿠킹되고 포함되지? 앱사이즈 줄여보기

UNREAL
ENGINE

UNREAL SUMMIT 2016

패키지 크기 줄이기

- “패키지 크기 줄이기” 문서로 시작하세요.
 - <https://docs.unrealengine.com/latest/KOR/Engine/Performance/ReducingPackageSize/index.html>
- “프로젝트 세팅”에 “패키징” 항목에
“Create compressed cooked packages” 옵션을 통해서 쿠킹 콘텐츠
압축하기

Create compressed cooked packages



UNREAL
ENGINE

UNREAL SUMMIT 2016

패키지 크기 줄이기

- 사용하지 않은 플러그인 제거
 - Paper2D
 - SpeedTree Importer
 - Script Generator Plugin
 - UObject Example Plugin
 - ArchVis Character
 - ChracterAI
 - Leap Motion Controller Plugin
 - Epic Survey
 - Experimental HTML5 Networking Plugin
 - Cable Component
 - Custom Mesh Component
 - Procedural Mesh Component
 - Gear VR
 - Oculus Input
 - Oculus Library
 - Oculus Rift
 - Steam VR

UNREAL
ENGINE

UNREAL SUMMIT 2016

패키지 크기 줄이기

- 사용하지 않는 모듈 제거
 - 일부 모듈들을 빌드에서 뺄 수 있는 옵션 있음
 - Unreal Match 3 예제로 참고 가능
- ```
IOSEngine.ini 에
[/Script/BuildSettings.BuildSettings]
bCompileAPEX=False
bCompileBox2D=False
bCompileICU=True
bCompileSimplygon=False
bCompileLeanAndMeanUE=True
bIncludeADO=True
bCompileRecast=True
bCompileSpeedTree=False
bCompilePhysXVehicle=False
bCompileFreeType=True
bCompileForSize=False
bCompileCEF3=False
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 패키지 크기 줄이기

- 패키지 블랙 리스트

- 플랫폼 별로 포함하지 않을 어셋/폴더 설정 가능
- 기본으로 포함되는 원치 않는/필요하지 않은 리소스들 제거
- 4.11에서는 이와 관련한 편리한 옵션 추가됨

Do not include editor content in this package may cause game to crash / error if you are using this content.

- 문서에 추가적으로 아래 항목들도 제거 가능

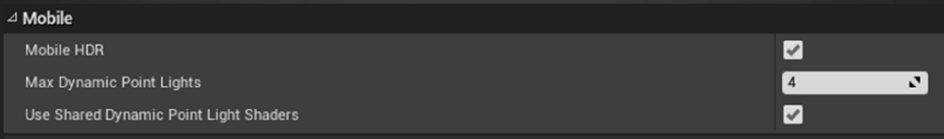
```
../..../Engine/Content/EditorBlueprintResources/
../..../Engine/Content/EditorLandscapeResources/
../..../Engine/Content/EditorMaterials/
../..../Engine/Content/EngineDebugMaterials/
../..../Engine/Content/EngineMaterials/AntiAliasedTextMaterialTranslucent.uasset
../..../Engine/Content/Tutorial/
../..../Engine/Content/EditorMeshes/
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 패키지 크기 줄이기

- 최대 다이내믹 포인트 라이트 개수



- "Used Shared Dynamic Point Light Shaders" 가 true일 경우는 Max Dynamic Point Lights가 큰 의미가 없음
- 퍼포먼스를 위해서 변수를 false로 두면 큰 의미가
- 전체 셰이더 개수에 따라 최종 패키지 사이즈 차이가 크게 날 수 있으니 최소한으로 하시는 것을 적극 추천

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 패키지 크기 줄이기

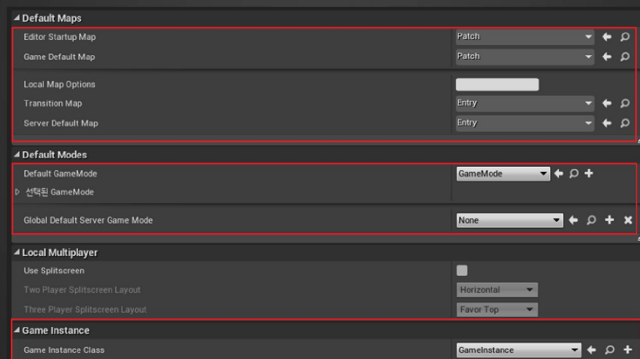
- 원하지 않는 콘텐츠가 쿠킹되어서 포함된다?
- 기본으로 참조/포함되는 콘텐츠들이 있음
- Slate를 고려한 기본으로 추가되는 콘텐츠 폴더들
  - BaseEditor.ini 에
  - [UI]
  - +ContentDirectories=/Game/UI
  - +ContentDirectories=/Game/Widget
  - +ContentDirectories=/Game/Widgets
  - +ContentDirectories=/Engine/MobileResources

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 패키지 크기 줄이기

- 기본으로 쿠킹되어서 포함되는 맵과 게임 오브젝트 확인
  - “프로젝트 세팅”에 “맵 & 모드”에 설정된 값들



UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 패키지 크기 줄이기

- 이런 과정들만 거쳐도
  - OBB 사이즈가 43.1 MB -> 13.4 MB
- 여기서 조금 더 하드코어 하게 줄여본다면?
  - 출시 버전에 엔진 디폴트 어셋들 교체도 고려 가능

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 조금 더 패키지 크기 줄이기

- 출시 버전에 엔진 디폴트 어셋들 교체하기
  - 엔진에서 기본적으로 사용할 수 있는 기본 어셋들이 존재
  - BaseEngine.ini에 설정되어 있음
  - void UEngine::InitializeObjectReferences() 에서 로딩해서 세팅을 함
  - 줄여볼 수 있는 것은?
    - 내부적으로 게임에서 사용하는 폰트가 있을 테니
      - TinyFontName=/Engine/EngineFonts/RobotoTiny.RobotoTiny
      - SmallFontName=/Engine/EngineFonts/Roboto.Roboto
      - MediumFontName=/Engine/EngineFonts/Roboto.Roboto
      - LargeFontName=/Engine/EngineFonts/Roboto.Roboto
      - SubtitleFontName=/Engine/EngineFonts/Roboto.Roboto

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 조금 더 패키지 크기 줄이기

---

- 출시 버전에 엔진 디폴트 어셋들 교체하기
  - 모든 머티리얼이 제대로 세팅되어 있다면
    - DefaultMaterialName=/Engine/EngineMaterials/WorldGridMaterial.WorldGridMaterial
    - 변경도 고려 가능
    - 특별한 material로 bUsedAsSpecialEngineMaterial가 true로 되어 있어서 모든 primitive type에 대한 셰이더를 전부 가지고 있기 때문에 크다
      - 쿨링하면 3.62MB 정도

UNREAL  
ENGINE

UNREAL SUMMIT 2016

---

## 4.11기준으로 다시 살펴보면 패치 시스템

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 모바일 게임 앱 출시 - 콘텐츠

- 3D 게임은 일반적으로 콘텐츠가 많고, 크기 때문에 구글의 최대 APK 사이즈 안에 다 넣기는 사실 불가능
  - 작년 9월쯤 구글이 APK 최대 사이즈 100 MB로 상향!
  - iOS는 사실 제한이 없지만 Wifi 없이 다운로드 가능하려면 100 MB 밑으로 해야함
- 100MB 이상의 앱을 위해 구글에서는 OBB 파일 지원
  - UE4도 기본으로 안드로이드 패키징시에 모든 게임 콘텐츠는 OBB로 빼서 자동으로 만들어 줌
- 하지만 한국의 모바일 게임은 빠른 콘텐츠 업데이트가 필요한 사실상 온라인 게임처럼 운영하기에 플랫폼 지원시스템을 쓰기는 어려움
- 그래서 개별 콘텐츠 서버를 두고, 서비스 중

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 콘텐츠 서버 사용한 UE4 모바일 게임

- 초기 IPA/APK 만들기
- 게임에서 사용하는 콘텐츠들이 포함된 Pak 파일 만들기
- 만들어진 콘텐츠 파일을 다운로드에 용의하도록 나누기
- 다운로드를 위해서 나누어진 파일을 다운로드 하고 복구하기

UNREAL  
ENGINE

UNREAL SUMMIT 2016



## 기본 배포 버전 만들기

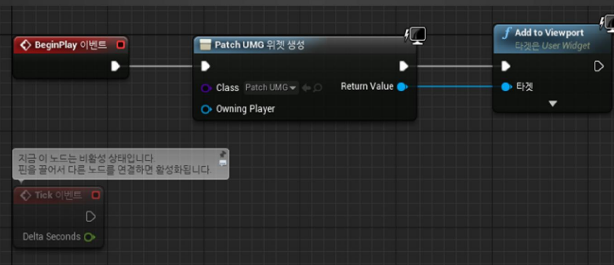
- UE4의 콘텐츠 다운로드 시스템은 UE4 네이티브 엔진 레벨에서 동작
- 즉, 엔진이 구동 되어야 다운로드 기능 사용 가능
- 엔진 구동을 위한 최소 콘텐츠  
+ 다운로드 관련 정보를 보여줄 최소 콘텐츠 필요

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 기본 배포 버전 만들기 - 안드로이드

- 배포를 위한 최소한의 콘텐츠들만 포함한 APK를 만드는 것 필요
1. 콘텐츠 다운로드를 위한 기본 맵 만들기
    - 콘텐츠 다운로드에 대한 메시지 보여주는 UMG
    - 맵 블루프린트에서는 그 UMG를 생성해서 추가만

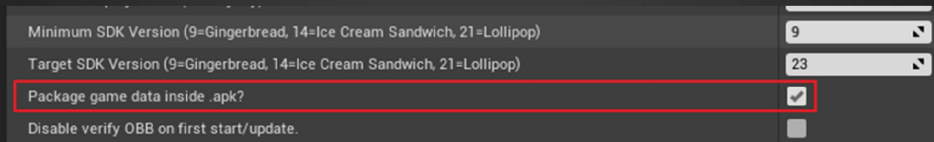


UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 기본 배포 버전 만들기 - 안드로이드

### 2. OBB 파일을 따로 만들지 않고, APK에 포함하기



### 3. 콘텐츠 다운로드 맵과 기본 콘텐츠만 쿠킹해서 포함한 APK 만들기

- RunUAT.bat 배치 파일을 이용해서 한번에 쿠킹에서 APK까지 만들면 편함

```
"C:\Program Files\Unreal Engine\4.11\Engine\Build\BatchFiles\RunUAT.bat" BuildCookRun
-project="D:\UnrealProjects\Match3\Match3.uproject" -clientconfig=Development -platform=Android_ETC1
-targetplatform=Android -cookflavor=ETC1 -cook -map=Patch -stage -package -cmdline=Patch -compressed
-pak -build
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 기본 배포 버전 만들기 - IOS

### 1. IOS의 경우 IPA에 기본으로 콘텐츠들 포함해서 만들기 때문에 콘텐츠 다운로드용 기본맵만 쿠킹 패키징 하면 됨 예)

```
"C:\Program Files\Unreal Engine\4.11\Engine\Build\BatchFiles\RunUAT.bat"
BuildCookRun
-project="D:\UnrealProjects\Match3\Match3.uproject" -clientconfig=Development
-platform=IOS -targetplatform=IOS -cook -map=Patch -stage -package -cmdline=Patch
-compressed -pak -build
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 다운로드 콘텐츠 만들기 - 안드로이드

1. 기본 배포 버전에 포함된 콘텐츠를 제외한 다운로드 콘텐츠 만들기 필요 - DLC 기능 사용
  - DLC 기능?
    - 배포 버전 설정해서 쿠킹하면 거기서 쿠킹된 콘텐츠들을 기억해두고, DLC 쿠킹시에 설정했던 배포 버전을 사용해서 쿠킹하면 배포버전에서 쿠킹된 콘텐츠들은 제외되는 기능

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 다운로드 콘텐츠 만들기 - 안드로이드

2. 베이스 버전 쿠킹 하기  
예)  
"C:\Program Files\Unreal Engine\4.11\Engine\Build\BatchFiles\RunUAT.bat"  
BuildCookRun  
-project="D:\UnrealProjects\Match3\Match3.uproject" -clientconfig=Development  
-platform=Android\_ETC1 -targetplatform=Android  
-cookflavor=ETC1 -cook -map=Patch -cmdline=Patch  
-build -unversionedcookedcontent -createreleaseversion=ReleaseContent

쿠킹이 완료 되면  
게임프로젝트폴더\Releases\ReleaseContent\Android\_ETC1  
폴더 안에 쿠킹된 정보를 기억해둔 AssetRegistry.bin 생성됨

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 다운로드 콘텐츠 만들기 - 안드로이드

### 3. 다운로드 콘텐츠 만들기

- 배포 버전을 제외한 실제 게임에 필요한 콘텐츠들을 모두 포함한 다운로드 콘텐츠 제작 필요  
예)  
"C:\Program Files\Unreal Engine\4.11\Engine\Build\BatchFiles\RunUAT.bat"  
BuildCookRun -project="D:\UnrealProjects\Match3\Match3.uproject"  
-clientconfig=Development -platform=Android\_ETC1 -targetplatform=Android  
-cookflavor=ETC1 -build -cook -map=GameLevel+Main  
-unversionedcookedcontent -BasedOnReleaseVersion=ReleaseContent  
-DLCName=DownloadContent -stage -compressed -pak -cmdline=Main
- Map 부분에 포함되어야 하는 모든 맵을 넣어야 하는데 맵이 추가 될 때마다 comandline에 추가하는 것은 좋지 못함

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 다운로드 콘텐츠 만들기 - 안드로이드

### 3. 다운로드 콘텐츠 만들기(계속)

- "-MAPINISECTION=" 파라미터를 사용하면 좀 더 편함  
예)  
"C:\Program Files\Unreal Engine\4.11\Engine\Build\BatchFiles\RunUAT.bat"  
BuildCookRun -project="D:\UnrealProjects\Match3\Match3.uproject"  
-clientconfig=Development -platform=Android\_ETC1 -targetplatform=Android  
-cookflavor=ETC1 -build -cook -MAPINISECTION=DownloadContent  
-unversionedcookedcontent -BasedOnReleaseVersion=ReleaseContent  
-DLCName=DownloadContent -stage -compressed -pak -cmdline=Main
- DefaultEditor.ini 예]  
[DownloadContent]  
+Map=Main  
+Map=GameLevel  
이렇게 설정하면 DownloadContent에 설정된 맵들을 전부 쿠키

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 다운로드 콘텐츠 만들기 - 안드로이드

### 3. 다운로드 콘텐츠 만들기(계속)

- 이렇게 쿠킹이 완료되면 배포해야 하는 콘텐츠들이 전부 포함됨 하나의 큰 Pak 파일이 생성됨
  - 예)  
Match3\Plugins\DownloadContent\Saved\StagedBuilds\Android\_ETC1\UnrealMatch3\Content\Paks\UnrealMatch3-Android\_ETC1.pak

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 다운로드 콘텐츠 만들기 - IOS

1. 베이스 버전 쿠킹하기  
예)  
"C:\Program Files\Unreal Engine\4.11\Engine\Build\BatchFiles\RunUAT.bat"  
BuildCookRun  
-project="D:\UnrealProjects\Match3\Match3.uproject" -clientconfig=Development  
-platform=IOS -targetplatform=IOS  
-cook -map=Patch -cmdline=Patch  
-build -unversionedcookedcontent -createreleaseversion=ReleaseContent
2. 다운로드 콘텐츠 만들기  
예)  
"C:\Program Files\Unreal Engine\4.11\Engine\Build\BatchFiles\RunUAT.bat"  
BuildCookRun -project="D:\UnrealProjects\Match3\Match3.uproject"  
-clientconfig=Development -platform=IOS -targetplatform=IOS  
-build -cook -MAPINISECTION=DownloadContent  
-unversionedcookedcontent -BasedOnReleaseVersion=ReleaseContent  
-DLCName=DownloadContent -stage -compressed -pak -cmdline=Main

UNREAL  
ENGINE

UNREAL SUMMIT 2016

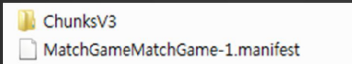
## 다운로드 콘텐츠 만들기

- 이제 안드로이드/IOS 별로 만들어진 Pak 파일을 UE4 콘텐츠 배포 시스템에서 사용할 수 있도록 작업 필요
- 사용되는 툴과 모듈은 BuildPatchTool 과 BuildPatchServices
  - BuildPatchTool
    - 파일을 다운로드에 편하도록 청크로 나누어 주고, 복구를 위한 manifest 파일을 만드는 툴
  - BuildPatchServices
    - BuildPatchTool로 만들어진 manifest를 이용해서 청크들을 다운로드 하고, 원본 파일을 복구 해주시는 모듈

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 다운로드 콘텐츠 만들기

- BuildPatchTool 사용법
  - 위치 Engine\Binaries\Win64\BuildPatchTool.exe  
예)  
Engine\Binaries\Win64\BuildPatchTool.exe -BuildRoot="D:\UnrealProjects\Match3\Plugins\DownloadContent\Saved\StagedBuilds\Android\_ETC1" -CloudDir="D:\UnrealProjects\Match3\CloudDir" -AppID=0 -AppName="MatchGame" -BuildVersion="MatchGame-1" -AppLaunch=". \Engine\Binaries\Win64\UE4Editor.exe" -AppArgs="" -customint="ChunkID=1"
  - BuildVersion 을 비교해서 새로운 콘텐츠 인지 비교 하기 때문에 BuildVersion은 중요
  - 생성되는 것은 

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 컨텐츠 다운로드 하기

- BuildPatchServices 사용법 – 안드로이드 환경
  - 사용을 위해 필요한 작업들 살펴보면
    1. BuildPatchTool로 만들어진 manifest 파일과 체크 폴더를 웹호스팅 서버에 올려둔다.
    2. BuildPatchServices에서 사용하기 위해 Manifest 파일을 다운로드한다.
    3. BuildPatchServices에 체크 폴더를 다운로드 할 웹서버 주소를 세팅
    4. 다운로드된 체크 및 임시 파일을 저장할 폴더와 복구된 원본 파일이 놓여질 경로를 BuildPatchServices에 세팅
    5. BuildPatchServices에 다운로드 요청

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 컨텐츠 다운로드 하기

- BuildPatchServices에서 사용하기 위해 Manifest 파일을 다운로드한다.
  - HTTP 모듈을 사용해서 Manifest 파일 다운로드 필요
  - 단, 추후에 컨텐츠가 변경되어 패치가 있을 수도 있고, 그렇게 되면 다운로드 받아야 하는 manifest 파일의 이름이 변경 될 수 있음
  - 그래서 간단한 manifest 파일 정보를 가지고 있는 파일을 올리거나 알맞은 manifest 파일을 리턴해주는 웹프로그램 필요

UNREAL  
ENGINE

UNREAL SUMMIT 2016



## 컨텐츠 다운로드 하기

- 예) manifest 파일 정보를 가지고 있는 간단한 json 파일을 사용한다면?
  - 그럼 실제 BuildPatchServices을 사용하기 위해서 Build.cs 에 필요한 모듈들은

```
PublicDependencyModuleNames.AddRange(
 new string[]
 {
 "HTTP",
 "Json",
 "OnlineSubsystem",
 "BuildPatchServices",
 "PakFile",
 }
);
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 컨텐츠 다운로드 하기

- 예) manifest 파일 정보를 가지고 있는 간단한 json 파일을 사용한다면? (계속)

```
TSharedRef<class IHttpRequest> HttpRequest = FHttpModule::Get().CreateRequest();
HttpRequest->OnProcessRequestComplete().BindRaw(this, &FPatchManager::GetManifestFileU
RL_HttpRequestComplete);
HttpRequest->SetURL(TEXT(http://patch.myserver.com/downloadcontent.json));
HttpRequest->SetVerb(TEXT("GET"));
HttpRequest->ProcessRequest();
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016



## 컨텐츠 다운로드 하기

- 예) manifest 파일 정보를 가지고 있는 간단한 json 파일을 사용한다면?(계속)

`downloadcontent.json` 는 아주 단순하게

```
{
 "manifest": "http://patch.myserver.com/MatchGameMatchGame-1.manifest"
}
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 컨텐츠 다운로드 하기

- 예) manifest 파일 정보를 가지고 있는 간단한 json 파일을 사용한다면?(계속)

`FPatchManager::GetManifestFileURL_HttpRequestComplete`  
(`FHttpRequestPtr` HttpRequest, `FHttpResponsePtr` HttpResponse, `bool` bSucceeded) 함수에서는

```
ResponseStr = HttpResponse->GetContentAsString();
if (EHttpResponseCodes::IsOk(HttpResponse->GetResponseCode()))
{
 TSharedPtr<FJsonObject> JSONManifestObject;
 TSharedRef<TJsonReader<TCHAR>> Reader = TJsonReaderFactory<TCHAR>::Create(ResponseStr);

 // Attempt to deserialize JSON
 if (FJsonSerializer::Deserialize(Reader, JSONManifestObject) || !JSONManifestObject.IsValid())
 {
 ...
 }
}
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 컨텐츠 다운로드 하기

```
// Attempt to deserialize JSON
if (FJsonSerializer::Deserialize(Reader, JSONManifestObject) || !JSONManifestObject.IsValid())
{
 // Get the values map
 TMap<FString, TSharedPtr<FJsonValue>> &JsonValueMap = JSONManifestObject->Values;

 TSharedPtr< FJsonValue > JsonAppNameString = JsonValueMap.FindRef(TEXT("manifest"));
 if (JsonAppNameString.IsValid())
 {
 FString ManifestFile = JsonAppNameString->AsString();

 // Create the Http request and add to pending request list
 TSharedPtr<class IHttpRequest> HttpRequest = FHttpModule::Get().CreateRequest();
 HttpRequest->OnProcessRequestComplete().BindRaw(this, &FPatchManager::CompareManifestFiles_HttpRequestComplete);
 HttpRequest->SetURL(ManifestFile);
 HttpRequest->SetVerb(TEXT("GET"));
 HttpRequest->ProcessRequest();
 }
}
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 컨텐츠 다운로드 하기

CompareManifestFiles\_HttpRequestComplete 함수에서는

```
IBuildPatchServicesModule* BuildPatchServices =
&FModuleManager::LoadModuleChecked<IBuildPatchServicesModule>(TEXT("BuildPatchServices"));
if (bSucceeded && HttpResponse.IsValid())
{
 ResponseStr = HttpResponse->GetContentAsString();
 if (EHttpResponseCodes::IsOk(HttpResponse->GetResponseCode()))
 {
 InstallManifest = BuildPatchServices->MakeManifestFromJSON(ResponseStr);
 FString DownloadedManifestFileName =
FPaths::Combine(*GExternalFilePath, TEXT("Patches"), TEXT("DownloadedManifest.manifest"));
 CurrentManifest = BuildPatchServices-
>LoadManifestFromFile(DownloadedManifestFileName);
 if (!CurrentManifest.IsValid() || (InstallManifest->GetVersionString() != CurrentManifest-
>GetVersionString()))
 {
 // download new content
 }
 }
}
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 컨텐츠 다운로드 하기

```
// download new content
IBuildPatchServicesModule* BuildPatchServices =
&FModuleManager::LoadModuleChecked<IBuildPatchServicesModule>(TEXT("BuildPatchServices"));

IPlatformFile& PlatformFile = FPlatformFileManager::Get().GetPlatformFile();

StageDir = FPaths::Combine(*GExternalFilePath, TEXT("Patches"), TEXT("Staged"));
InstallDir = FPaths::Combine(*GExternalFilePath, TEXT("Patches"), TEXT("Installed"));

BuildPatchServices->SetCloudDirectory(CloudURL);
BuildPatchServices->SetStagingDirectory(StageDir);

if (CurrentManifest.IsValid())
{
 BuildInstaller = BuildPatchServices->StartBuildInstall(CurrentManifest, InstallManifest, InstallDir,
FBuildPatchBoolManifestDelegate::CreateRaw(this, &FPatchManager::OnDownloadCompleted));
}
else
{
 BuildInstaller = BuildPatchServices->StartBuildInstall(nullptr, InstallManifest, InstallDir,
FBuildPatchBoolManifestDelegate::CreateRaw(this, &FPatchManager::OnDownloadCompleted));
}
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 컨텐츠 다운로드 하기

```
void FPatchManager::OnDownloadCompleted(bool bSuccess, IBuildManifestRef Manifest)
{
 FString DownloadedManifestFileName = FPaths::Combine(*GExternalFilePath, TEXT("Patches"),
TEXT("DownloadedManifest.manifest"));
 BuildPatchServices->SaveManifestToFile(DownloadedManifestFileName, Manifest);

 // Pak 파일 찾기
 ...

 // Pak 파일 찾고, Mount 하기
 ...
}
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

# 컨텐츠 다운로드 하기

```
void FPatchManager::OnDownloadCompleted(bool bSuccess, IBuildManifestRef Manifest)
{
 // Pak 파일 찾기
 class FPakSearchVisitor : public IPlatformFile::FDirectoryVisitor
 {
 public:
 TArray<FString>& FoundPakFiles;
 FPakSearchVisitor(TArray<FString>& InFoundPakFiles)
 : FoundPakFiles(InFoundPakFiles)
 {}
 virtual bool Visit(const TCHAR* FilenameOrDirectory, bool bIsDirectory)
 {
 if (bIsDirectory == false)
 {
 FString Filename(FilenameOrDirectory);
 if (FPaths::GetExtension(Filename) == TEXT("pak"))
 {
 FoundPakFiles.Add(Filename);
 // Add Filename to Manifest
 IFileManager::Get().SetTimeStamp(*Filename, FDateTime::UtcNow());
 }
 }
 return true;
 }
 };
}
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

# 컨텐츠 다운로드 하기

```
void FPatchManager::OnDownloadCompleted(bool bSuccess, IBuildManifestRef Manifest)
{
 // Pak 파일 찾고, Mount 하기
 TArray<FString> FoundPaks;
 IPlatformFile& PlatformFile = FPlatformFileManager::Get().GetPlatformFile();
 FPakSearchVisitor Visitor(FoundPaks);
 PlatformFile.IterateDirectoryRecursively(*InstallDir, Visitor);

 for (const auto& PakPath : FoundPaks)
 {
 FPakPlatformFile* PakPlatformFile = static_cast<FPakPlatformFile*>(FPlatformFileManager::Get().FindPlatformFile(TEXT("PakFile")));
 PakPlatformFile->Mount(*PakPath, 1, TEXT("../Match3/"));
 }

 // 메인 게임 레벨 로딩
 ...
}
```

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 패치 시스템 정리

- 최소한의 콘텐츠 다운로드를 위한 배포용 APK/IPA 만들기
- 실제 게임 콘텐츠를 모두 포함한 pak 파일 만들기
  - DLC 기능 사용으로 초기 배포 콘텐츠 제거해서 생성
- BuildPatchTool을 사용해서 pak파일을 다운로드 할 수 있게 가공
- BuildPatchServices를 사용해서 BuildPatchTool로 만들어진 manifest와 청크 파일 다운로드 하고, pak 파일 복구해서 mount!

## 패치 시스템 정리

- 생각 해볼 이슈들
  - 4.11 전의 엔진 버전 사용하시는 경우 같은 리소스 재 쿠킹시에 쿠킹된 콘텐츠 바이너리가 매번 바뀌는 이슈가 많을 수 있음
    - 4.11 업데이트 추천드리고, 경우에 따라서는 어셋 다시 저장해야 해결됨
  - Compressed 옵션으로 Pak 파일을 압축 할 경우 실제 어셋이 많이 바뀌지 않았어도 만들어진 Pak 파일은 전에 Pak 많은 바이너리 차이가 날 수 있고, 이로 인해 다운로드 받는 청크가 많아 질 수 있음
    - Pak 파일을 여러개로 나누어 만들어서 다운로드 받고, Mount를 여러개 하는 것도 현재로서는 이 이슈를 우회할 방법
    - 내부적으로 개선할 수 있는 점이 있을지 고민중
- 마지막으로, 설명드린 초기 배포 버전과 다운로드 콘텐츠 제작법을 GUI 기반으로 쉽게 할 수 있는 기능 개발 예정!

# 알아두면 유용한 소소한 팁들

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## Android 6.0 runtime permission 처리 예제

- 안드로이드에서 Target SDK를 23(6.0)으로 할 경우 runtime permission 사용이 필요
- UE4에 지원 추가 방법은?
  1. Engine\Build\Android\Java\src\com\epicgames\ue4\GameActivity.java에 runtime permission 관련 처리 추가 필요
    - <http://stackoverflow.com/questions/36098366/android-marshmallow-permissions/36098423#36098423> 참조
    - 첨부된 GameActivity.java.txt 파일 참조.
  2. Engine\Build\Android\Java\src\com\google\android\vendor\expansion\downloader\impl\DownloadNotification.java 수정 필요
    - 현재 구글에서 제공한 APK Expansion Zip Library를 사용하고 있는데 아직 6.0 호환 버전을 지원하고 있지 않음
    - 그래서 어차피 콘텐츠 다운로드를 사용하는 거면 setLatestEventInfo를 코멘트 처리해서 임시로 우회
  3. 최신 android.support.v4로 교체
    - C:\NVPACK\android-sdk-windows\SDK Manager.exe 실행 후에 Extras 항목에 Android Support Library를 최신 버전 받기
    - C:\NVPACK\android-sdk-windows\extras\android\support\v4\android-support-v4.jar 파일을 Engine/Build/Android/Java/libs/android-support-v4.jar 덮어쓰기
  4. 안드로이드 6.0은 org.apache.http가 제거되었기에 따로 추가 필요
    - C:\NVPACK\android-sdk-windows\platforms\android-23\optional\org.apache.http.legacy.jar을 Engine/Build/Android/Java/libs/org.apache.http.legacy.jar 복사 추가

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 안드로이드 비디오 녹화

- 디버깅이나 기타 목적을 위해서 안드로이드 화면을 녹화 가능
- PC에 디바이스 연결 후 콘솔 창에 아래 명령어 입력
  - adb shell screenrecord /sdcard/파일이름.mp4 -bit-rate 8000000
  - 디바이스에서 벌어지는 내용을 180 초까지 녹화할 수 있음
  - 언제든지 콘솔 창에서 **Ctrl + C** 를 누르면 비디오 녹화 중지됨
- 비디오는 일반적으로 폰의 루트 디렉토리에서 찾을 수 있음

UNREAL  
ENGINE

UNREAL SUMMIT 2016

## 감사합니다!

- Q/A
  - Q: 다운로드 시스템을 썼을 때 스트리밍 레벨의 경우 제대로 로딩이 안된다.
    - 테스트 해본 결과 스트리밍 레벨도 로딩이 잘 되는 것으로 확인이 됩니다. 대신에 Engine에 있는 콘텐츠를 사용할 경우 이 콘텐츠들은 현재 추천 드린 방식의 다운로드 받아서 프로젝트 경로로 Mount하는 경우에는 포함이 되지 않기 때문에 그 리소스들이 나오지 않아서 제대로 로딩이 안된다고 보일 수 있을 것 같습니다. [예를들어 대표적인 것이 /Engine/EngineSky/BP\_Sky\_Sphere.uasset] 그래서 그런 콘텐츠를 복사해서 게임의 콘텐츠 폴더에 옮겨서 사용하시면 될 것 같습니다.
  - Q: 4.10에서는 패치 관련한 실행파일이나 모듈이(BuildPatchTool.exe와 BuildPatchServices 모듈) 런처 버전에는 없다.
    - 네. 4.10에 런처 버전에서는 해당 툴이나 모듈이 없습니다. 그래서 github에서 풀 소스 코드 다운로드 후에 빌드해서 만드셔야 하는데요. 대신에 4.11에는 런처 버전에도 포함이 되어 있고, 콘텐츠 다시 쿠킹시에 변경되지 않은 콘텐츠가 변경되는 이슈들이 많이 해결 되었으므로 4.11로 업그레이드해서 사용하시기를 추천드립니다.

UNREAL  
ENGINE

UNREAL SUMMIT 2016