POWERED BY

UNREAL
TECHNOLOGY

Professional-Quality tool 개발
Unreal Engine 3 Customize
-KGC 2011-

이상우
엔진 프로그래머
Sangwoo.lee@epicgameskorea.com

개요

- Unreal Engine 3 Overview
- Unreal Editor Structure
- Engine features related to Editor Customization
- How to start new element in UE3
- More customization for Landscape
- WPF within C++
- Working with a tech artist to develop tools

Unreal Engine 3 Overview

- 최고의 렌더링 퀄리티와 범용성을 가진 게임 제작을 위한 **통합 엔진**
  - Industry 최고의 툴과 Engine features
  - 2003년 출시 이후 지속적인 혁신적 Upgrade
  - Unreal script : Game play programming
  - 멀티 플랫폼 지원
    - XBOX360, PS3, PC, iOS, Android, PS Vita, Mac 그리고 Flash!
  - Localized된 풍부한 Documentation (UDN)
- 100개가 넘는 타이틀이 사용한 검증된 엔진
- 범용성을 가진 엔진 - 모바일 부터 하이엔드까지
  - Infinity Blade 부터 DX 11를 사용한 사마리아인 데모까지

Unreal Engine 3 Tools

- Lightmass : GI Ray tracer integrated with engine
- Material Editor : Great GUI for Shader creation
- Kismet : Game logic programming tool
- Matinee : Cinematic programming tool
- Cascade : Particle Editor

언리얼 에디터 구조

- 에디터와 엔진 코드가 통합
  - Unique approach
  - 엔진과 에디터는 항상 같은 Code base
    - Feature 추가시 같이 반영
  - 엔진의 모든 부분을 Control 가능
    - 엔진과 툴이 같은 Data structure사용
      - Good Consistency
  - 빠른 Iteration
    - 엔진과 툴은 한몸, 人馬一體

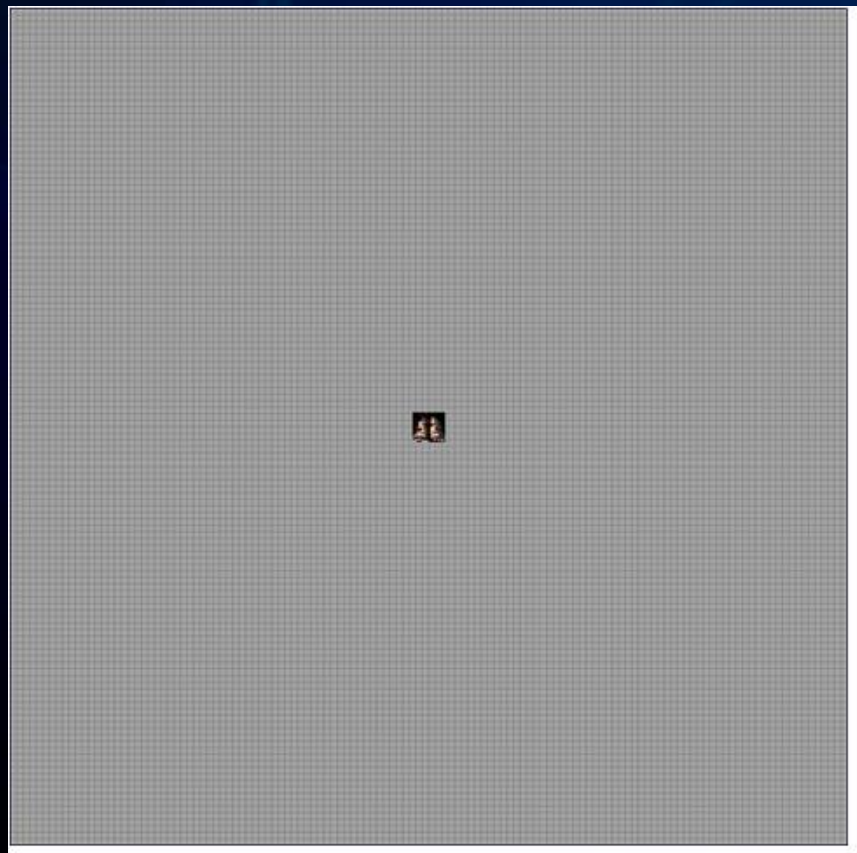
왜 Customize가 필요한가?

- UE3는 원하는 features의 대부분을 커버
- Nothing can be perfect for your product
  - Even Unreal Engine! ☺
- 개발자들의 간지러운 부분을 긁어주기
- Content creator과의 원만한 관계
- Iteration time 축소
- UE3/UnrealEd에 대한 좀더 깊은 이해

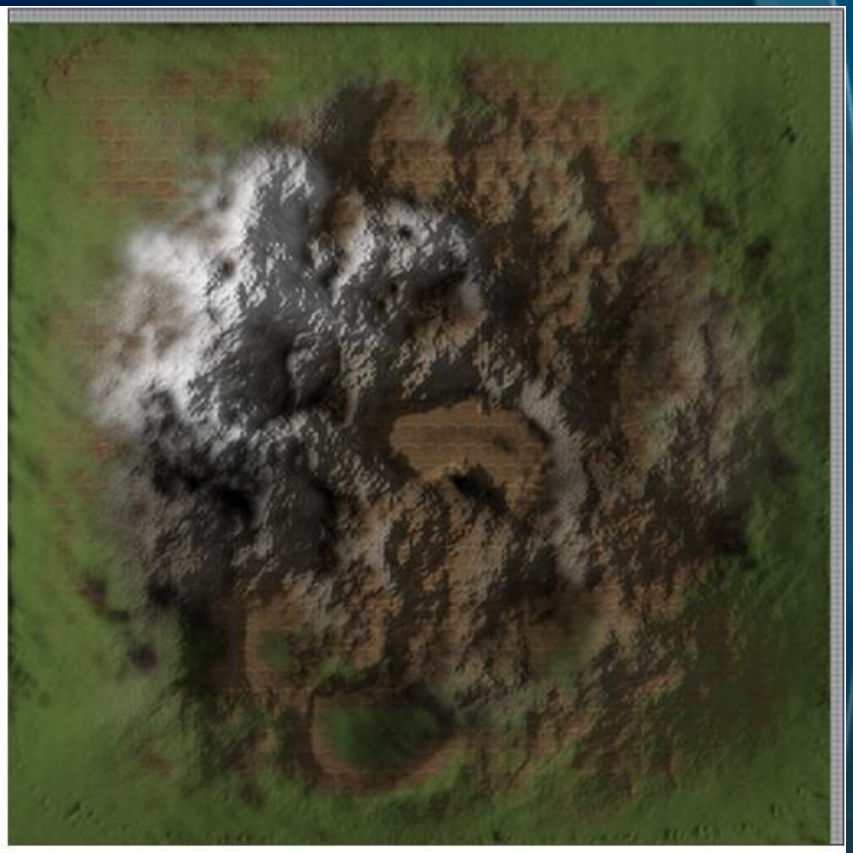
Landscape

- 한국 Licensee를 위한 새로운 Terrain 시스템
- KGC 2010에서 공개
- 크고 빠르고 아름다운 Terrain을 목표
- Jack Porter부장님과 함께 개발중
  - Local licensee의 feedback을 빠르게 수용

Landscape



Terrain in Unreal Tournament 3 level



Terrain in Landscape level

- Comparing typical old and new UE3 terrain sizes

Landscape

- One of the core engine change
- 관련되는 추가된 Engine 코드들이 매우 많음
- Landscape 개발 과정을 살펴보면서 UE3의 새로운 feature추가를 위한 과정을 알아봅시다.

Engine Feature related to Customization

- Landscape Actor
- Landscape Component
- Collision Component (PhysX)
- Render Proxy
- Decal support
- Vertex Factory

Engine Feature related to Customization

- Material system
- Material editor customize
- Lightmass support implementation

Unique customized features of Landscape

- Multiple streaming level support
- Large map scale related problems
- UI (WxWidget, WPF)
  - New usage of WPF binding to C++
- Multiplatform (DX11/XBOX/PS3) support
- Texture data editing features
  - Storing data in textures

How to start new element in UE3

- Actor
  - Container for Components
  - Base class for property presenter
  - Component Handling interfaces
    - AActor::UpdateComponentsInternal()
    - AActor::ClearComponents()
  - PhysX scene initialize
    - AActor::InitRBPhys
    - AActor::InitRBPhysEditor()

How to start new element in UE3

- Component
  - Base data structure for new element
  - Your specific data
  - Rendering data
    - Light/Shadow map
    - UPrimitiveComponent::CreateSceneProxy()
    - UPrimitiveComponent::GenerateDecalRenderData()
  - Collision data

How to start new element in UE3

- Collision Component/function
  - How to collide your new element in UE3
    - UPrimitiveComponent::PointCheck
    - UPrimitiveComponent::LineCheck
  - PhysX data handling
    - UPrimitiveComponent::InitComponentRBPhys
    - UPrimitiveComponent::SetComponentRBFixed
    - UPrimitiveComponent::TermComponentRBPhys
  - Landscape only use PhysX collision

How to start new element in UE3

- Render Proxy
  - How to render your new element
    - Transform Matrix
    - FPrimitiveSceneProxy::GetViewRelevance()
    - FPrimitiveSceneProxy::DrawStaticElements()
    - FPrimitiveSceneProxy::DrawDynamicElements()

How to start new element in UE3

- Render Proxy
  - How to interact with Lighting/Shadowing
    - FPrimitiveSceneProxy::GetLightRelevance()
    - FPrimitiveSceneProxy::GetLightMapType()
  - How to interact with Editor picking
    - FPrimitiveSceneProxy::CreateHitProxies()

How to start new element in UE3

- Vertex Factory
  - Define how your vertex buffer generated
  - Landscape need special vertex/index buffer
  - Shader Parameter setting
    - FVertexFactoryShaderParameters::Bind()
    - FVertexFactoryShaderParameters::Set()
      - Shared Mesh resources
    - FVertexFactoryShaderParameters::SetMesh()
      - Individual Mesh resource

How to start new element in UE3

- Many editor specified interfaces
  - UObject::PostEdit~() interfaces
- Handling other processing
  - UObject::PostLoad()
  - UObject::BeginDestroy()

How to start new element in UE3

- Data storing / Garbage collecting
  - UObject::Serialize()
  - UObject::AddReferencedObjects()
- How to import/export
  - UObject::ExportCustomProperties()
  - UObject::ImportCustomProperties()

How to start new element in UE3

- Decal Support
  - Only if you use some special decal rendering handling for your new element
  - Need to update to support Rendering part
    - If new element has a unique vertex factory
    - UPrimitiveComponent::GenerateDecalRenderData()
    - FPrimitiveSceneProxy::DrawStaticDecalElements()
    - FPrimitiveSceneProxy::DrawDynamicDecalElements()

How to start new element in UE3

- Lightmass support implementation
  - Only if you use special mesh generation for your new element
  - Lightmass should simulate all your rendering data in ray-tracing processing
  - Need data structure to send Lightmass
    - FStaticLightingMesh::GetTriangle()
    - FStaticLightingMesh::GetTriangleIndices()
    - FStaticLightingTextureMapping

How to start new element in UE3

- Non-lightmass Lightmap render support
  - Legacy fallback support
  - Should be same for Lightmass
  - Good for prototyping for Lightmass
  - FStaticLightingMesh::GetStaticLightingVertex()

More customization for Landscape

- Lightmass support implementation
  - Light/Shadow map seam problem
  - Light/Shadow map size problem
    - Shadow map compression
  - Large memory usage handling
  - Too many Lighting SH sample
    - Increased SH probe distance based on Texel density
  - Need to change material rendering if you need special material handling

More customization for Landscape

- Material system
  - Landscape Material Instances
    - Two layers for Material instances
      - 1st Layer : Based on Layer combination
        - » Layer Name and Texture Index
      - 2nd Layer : Set proper textures
    - Added TerrainLayerWeightParameters in FStaticParameterSet
  - Compile time optimization

More customization for Landscape

- Material editor customize
  - Add a new material node for material editor
    - UMaterialExpressionTerrainLayerWeight
    - UMaterialExpressionTerrainLayerSwitch
    - UMaterialExpressionLandscapeLayerBlend
    - UMaterialExpressionTerrainLayerCoords
  - UMaterialExpression Interfaces
    - Compile
    - SetStaticParameterOverrides
    - GetAllParameterNames

More customization for Landscape

- Multiple streaming level support
  - Landscape in MMO is usually divided to multiple streaming levels
  - Need to be edited consistently with any subset of streaming levels
  - Korean licensee request/feedbacks

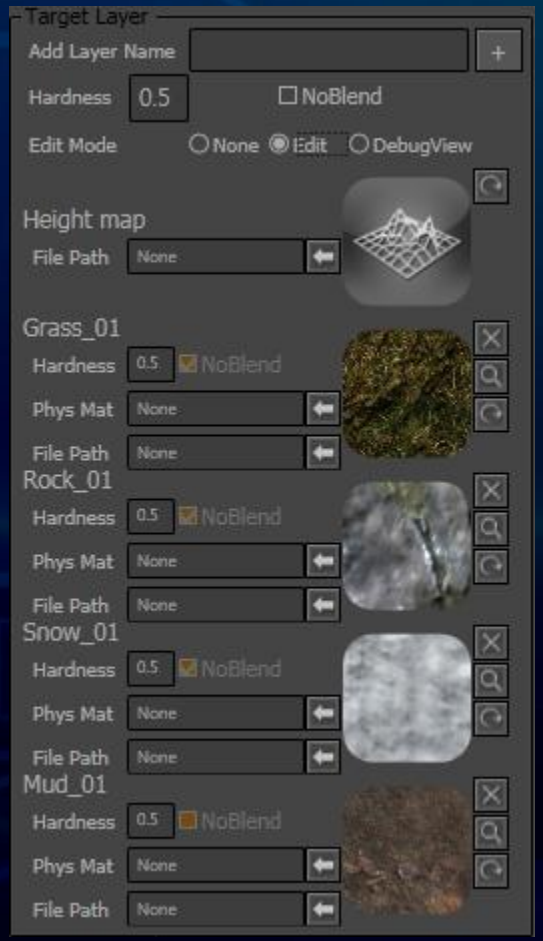
WPF within C++

- UnrealEd based on wxWidget
  - New UI windows are based on WPF
    - Content Browser, Vertex Paint, etc
- Using WPF for rapid and beautiful UI development
- First approach
  - Data declaration in C# (.cs)
  - UI logic in C#/C++
  - Data inconsistency problem

WPF within C++

- Second approach
  - Data declaration in C++
  - UI logic in C++/xaml
    - XAML에게 최대한 많은 logic을!
  - Using ref class declaration in C++
    - ref class Aclass : INotifyCollectionChanged
  - No duplicated data representation
  - No duplicated UI logic call in C++/xaml

Landscape WPF UI



More customization for Landscape

- Multiplatform support
  - DX11
    - Tessellation support
    - Implemented functions to send data for DomainShader
      - FVertexFactoryInterpolantsVSToDS
      - GetMaterialTessellationParameters
      - VertexFactoryGetTextureCoordinateDS
      - VertexFactoryGetWorldNormalDS
      - VertexFactoryGetWorldTangentDS
    - Texture format difference
      - Need color swizzle

More customization for Landscape

- Multiplatform support
  - PS3 : Cannot use vertex shader texture sampling
    - GPU is too slow
    - SPU processing to generate height data stream
  - XBOX : Easy!!
    - Only some Shader trick

Communication for development

- Tech Artist – Mr. Kang
- Jack Porter – The BOSS
- HQ – The BOSS's BOSS
- QA (Testers)
  - Requests for testing new features
  - TTP

Working with a tech artist to develop tools

- 새로운 Editor tool이 등장
  - Programmer 생각대로 만들면 Parameter만 잔뜩
    - UI Control의 향연

Working with a tech artist to develop tools

- 새로운 Editor tool이 등장
  - 유저가 사용법을 모르면 사용하지 않는다.
  - 사용하지 않으면 Bug report/feedback이 없다.
    - 악순환의 시작
  - 유저가 편하지 않으면 쓰기 어려운 툴
  - Artist와 Programmer는 다른 종족
    - 요구사항은 많지만 항상 Communication이 엇갈린다.

Working with a tech artist to develop tools

- Tech artist는 혼혈 (Infested Terran?)
  - 원하는 바를 Programmer가 알아듣기 편하게 표현해준다.
  - Terminology가 같아서 대화가 편하다.
  - 하지만 우리편은 아님
  - 저쪽 편하고도 취향이 다름 (왕따?)

Working with a tech artist to develop tools

- Case study – Landscape Gizmo
  - 부분 Import/export가 필요하다는 Licensee 요청
  - Mr. Kang inspired by 3DS Max Gizmo
  - Extend the idea with direct communication
    - Rotate, Scale(non-Uniform, negative), Translate
    - Copy/Paste, Import/Export
    - Navigation history
    - Height map data visualization

Working with a tech artist to develop tools

- Landscape Gizmo Screen shots

Working with a tech artist to develop tools

- Erosion, Smooth(FFT) Screen shots

Working with a tech artist to develop tools

- 반전 – Mr. Kang도 완성된 기능 중 모르는 부분이 있었다.
  - Programmer는 만들고 사용하도록 열심히 설득
  - 사용하지 않으면 Idea 제안자도 잊는다
- 결론
  - TA가 있으면 좋다
  - TA한테 자주 물어보자
  - Idea 교류를 자주 하자
  - 새 툴을 사용하게 전도하자

HQ vs EGK

- 공유하는 부분
  - Codebase via Perforce
  - Development progress
    - Daily/Weekly status, other discussions
    - Via email
  - QA build cycle
    - Based on UDK QA build!
    - Feedback from world-wide licensees
- EGK가 하는 부분
  - Landscape development
  - Feedbacks from Korean licensees

에픽 부스에 방문해 주세요!

- 이미 다 부수고 있습니다.
- 부스의 최후를 함께 해 주세요

- Questions?

감사합니다.