
유니티

2084032
이재윤2184014
한승연



배재대학교
PAI CHAI UNIVERSITY

Index1

유니티란?

3D 계산 및 그림자 표시,사운드,메뉴 이동 등 게임을 개발할때 자주 쓰는 기능을 하나로 묶어 쉽게 사용할 수 있도록 만든 것.

기반언어: C,C++(런타임용)

스크립팅용: C#



Index2

Index3

Index4

Index5

패키지

우리가 요구사항에 적합한 기능들이 들어 있음.

패키지의 종류

검증된 패키지

프리뷰 패키지

코어 패키지

빌트인 패키지

키워드별 패키지

ex1

Index2

dex3

dex4

dex5

검증된 패키지

특정 Unity 버전에서 테스트를 마친 패키지이며 해당 버전에 대해
검증된 기타 모든 패키지를 포함함

Package Manager 창에서 verified 표시기에서 나타남.

- 2D 픽셀 퍼펙트: [com.unity.2d.pixel-perfect](https://www.unity3d.com/packages/2d/pixel-perfect)
- 2D PSD 임포터: [com.unity.2d.psdimporter](https://www.unity3d.com/packages/2d/psd-importer)
- 2D 스프라이트 셰이프: [com.unity.2d.spriteshape](https://www.unity3d.com/packages/2d/sprite-shape)
- 어댑티브 퍼포먼스: [com.unity.adaptiveperformance](https://www.unity3d.com/packages/2d/adaptive-performance)
- 어드레스블: [com.unity.addressables](https://www.unity3d.com/packages/2d/addressables)

ex1

ex2

Index3

ex4

ex5

프리뷰패키지

Package Manager 창에서 preview Packages in Use 표시기는 프로젝트에 현재 설치된 프리뷰 패키지가 있음을 알려줌.

- 2D IK: com.unity.2d.ik
- 2D 타일맵 엑스트라: com.unity.2d.tilemap.extras
- 기기 시뮬레이터: com.unity.device-simulator
- 게임 파운데이션: com.unity.game-foundation
- Unity용 Havok 피직스: com.havok.physics
- 잡: com.unity.jobs
- 키네마티카: com.unity.kinematica

ex1

ex2

ex3

ex4

Index5

코어 패키지

코어패키지는 다른 Unity 패키지와 비슷하지만 에디터 버전에 바인딩되어 있고 Unity 패키지 레지스트리에 나타나지 않는다는 점이 다르다.

- 2D 스프라이트: [com.unity.2d.sprite](#)
- 2D 타일맵 에디터: [com.unity.2d.tilemap](#)
- Unity UI: [com.unity.ugui](#)



ex1

ex2

ex3

ex4

Index5



빌트인 패키지

사용자들이 패키지 관리자를 통해 Unity 기능을 토글하도록 해줌
패키지를 활성화하거나 비활성화하면 런타임 빌드 크기가 감소

- AI: [com.unity.modules.ai](#)
- Android JNI: [com.unity.modules.androidjni](#)
- 애니메이션: [com.unity.modules.animation](#)
- 에셋 번들: [com.unity.modules.assetbundle](#)
- 오디오: [com.unity.modules.audio](#)
- 천: [com.unity.modules.cloth](#)
- 디렉터: [com.unity.modules.director](#)
- 이미지 전환: [com.unity.modules.imageconversion](#)

ex1

ex2

ex3

ex4

Index5

키워드별 패키지

키워드:	일치하는 패키지:
2d	<u>2D 스프라이트</u> <u>2D 타일맵 에디터</u>
2D	<u>2D 픽셀 퍼펙트</u>
3d	<u>ProBuilder</u>
abc	<u>Alembic</u>
조정	<u>원격 설정</u>
어댑티브	[어댑티브 퍼포먼스]
광고	<u>광고</u>
aec	[Unity 리플렉트]
조준	<u>시버머신</u>
alembic	<u>USD</u>
정렬	<u>ProGrids</u>
분석	<u>분석 라이브러리</u>
Android	<u>Android 로켓</u> <u>모바일 알림</u>
android	<u>Unity 퍼블리싱 포털</u>
애니메이션	<u>USD</u>
애니메이션	<u>애니메이션 리깅</u>
모든 것	<u>빠른 검색</u>
AOV	<u>레코더</u>
ap	[어댑티브 퍼포먼스]

ex1

ex2

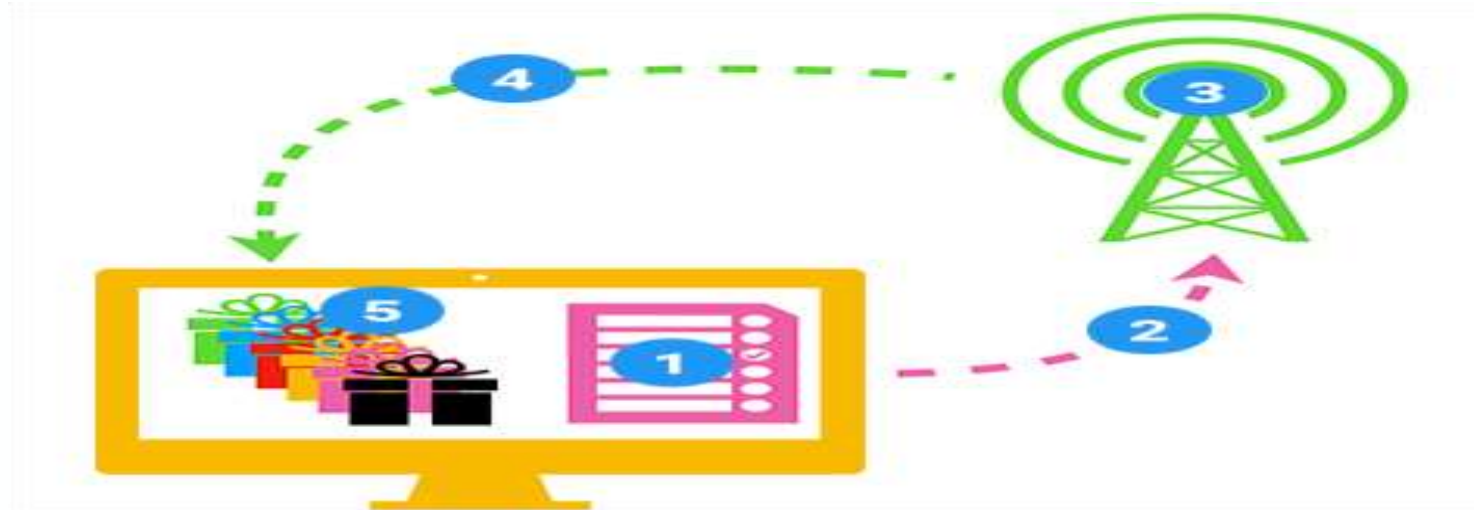
ex3

ex4

Index5

Unity의 패키지 관리자

- 1.Unity가 프로젝트를 열면 Unity 패키지 관리자가 프로젝트 매니페스트(1)를 읽은 후 프로젝트에 로드할 패키지를 알아냄
- 2.다음 요청(2)을 매니페스트에 종속성으로 나타나는 각 패키지의 패키지 레지스트리 서버(3)로 전송
- 3.패키지 레지스트리가 요청된 정보와 데이터를 패키지 관리자로 다시 보냄
- 4.패키지 관리자는 받은 패키지(5)를 프로젝트에 설치



ex1

ex2

ex3

ex4

Index5

패키지 관리자를 사용하는 방법

패키지 관리자는 세 가지 인터페이스를 사용하여 사용자, 매니페스트, 레지스트리와 소통합니다.

인터페이스 방식	사용 방법
사용자 인터페이스	<u>패키지 관리자 창</u> 을 사용하여 기능을 빠르게 탐색하고 검색할 수 있습니다. 또한 설치하거나 업데이트할 패키지를 빠르게 선택하고, 패키지 종속성 충돌을 해결할 수 있습니다. 패키지 관리자는 <u>프로젝트 매니페스트</u> 에 변경 사항을 직접 적용할 수 있도록 해주는 사용자 인터페이스를 제공합니다.
패키지 매니페스트	패키지 관리자는 Unity에서 <u>전용 인스펙터</u> 를 제공합니다. 이를 통해 패키지의 <u>매니페스트</u> 를 확인할 수 있으며, 그와 동시에 해당 매니페스트는 프로젝트 뷰에서 메인 포커스를 받습니다. 이 <u>통합</u> 으로 인해 Unity에서 <u>내장</u> 또는 <u>로컬</u> 패키지에 대한 패키지 매니페스트를 직접 편집할 수 있게 되었습니다.
스크립팅 API	<u>패키지 관리자 스크립팅 API</u> 를 통해 사용자들이 패키지 관리자와 프로그래밍 방식으로 상호작용할 수 있습니다. 또한 이 API를 사용하여 패키지 레지스트리를 쿼리하고, 패키지를 설치, 포함 및 제거하고, 다양한 기준으로 패키지를 나열할 수 있습니다.

ex1

ex2

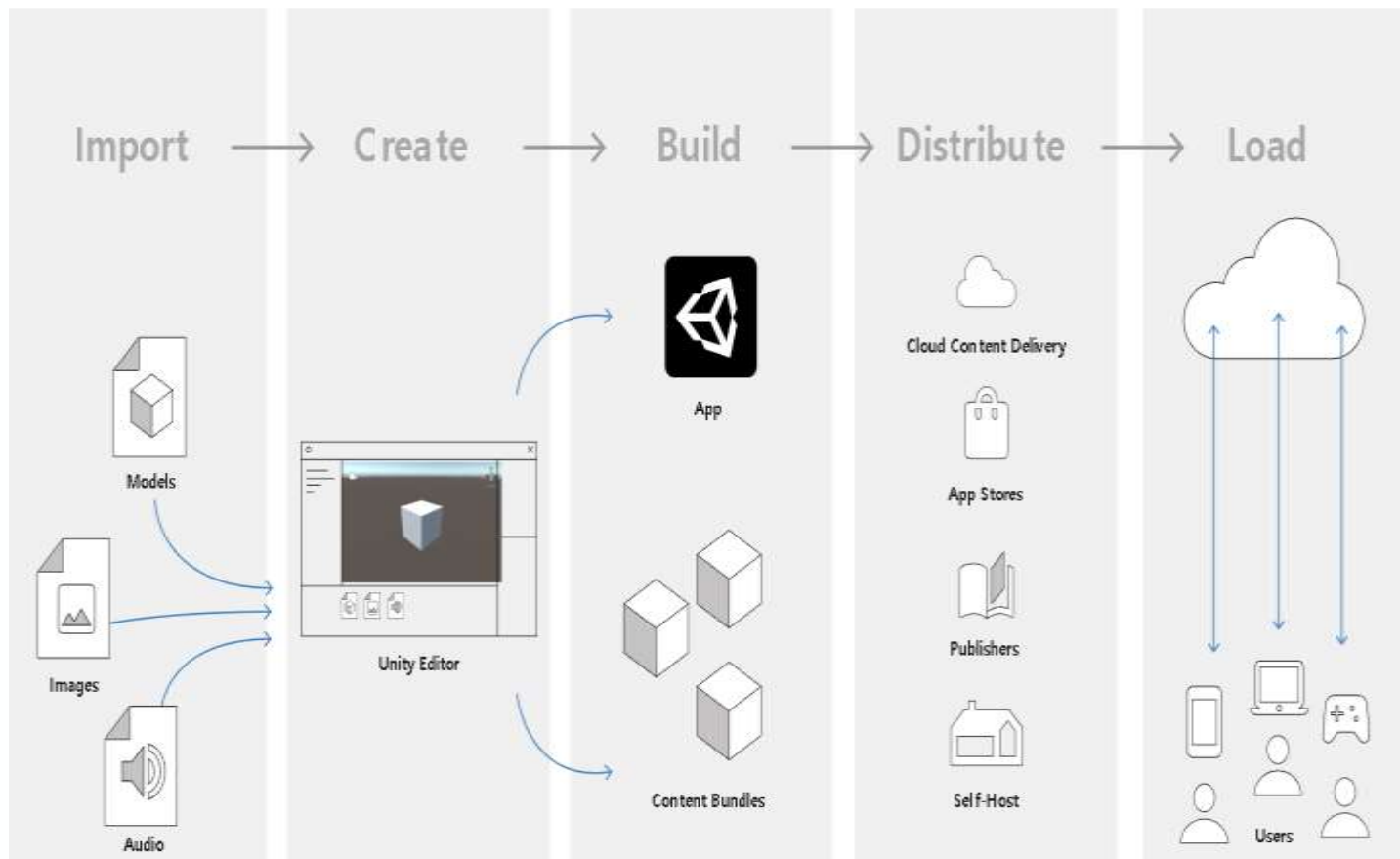
ex3

ex4

Index5

에셋 워크플로

Unity 프로젝트에서 게임이나 앱을 만드는 데 사용하는 모든
아이템



ex1

ex2

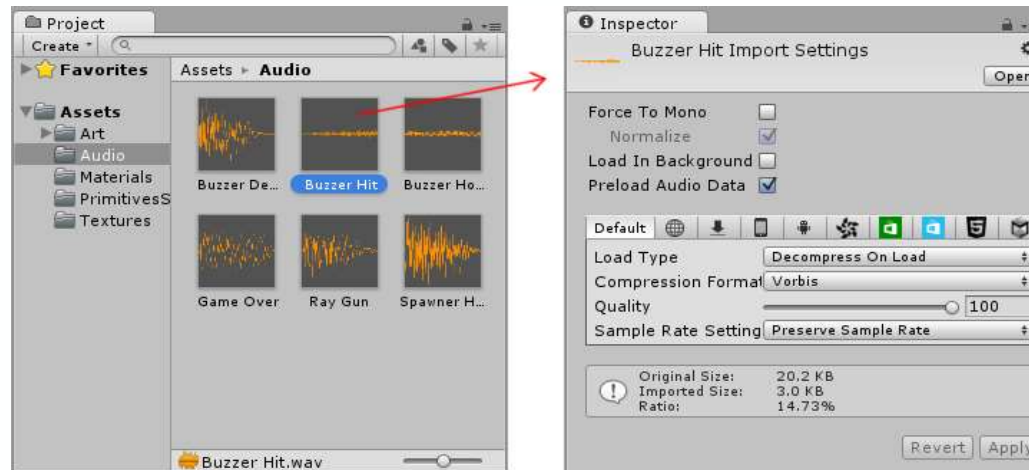
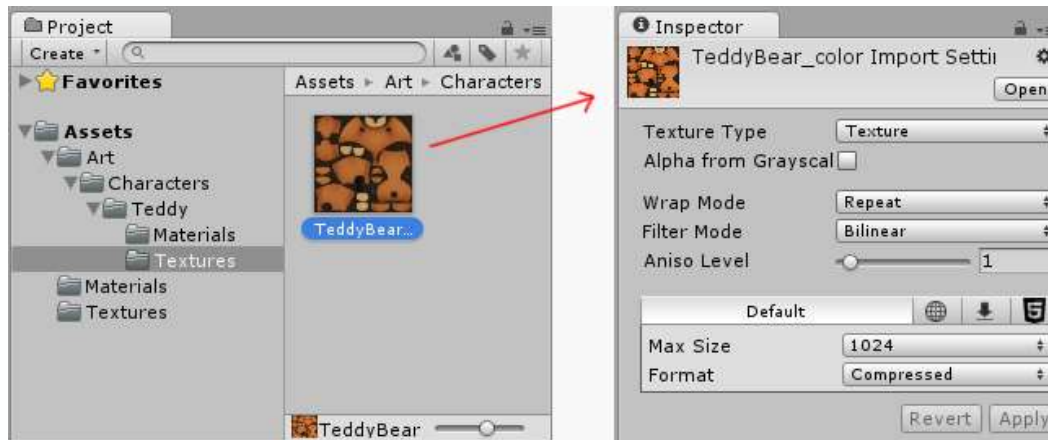
ex3

ex4

Index5

임포트

소스 파일을 Unity 에디터로 가져와 작업하는
프로세스



Index1

Index2

Index3

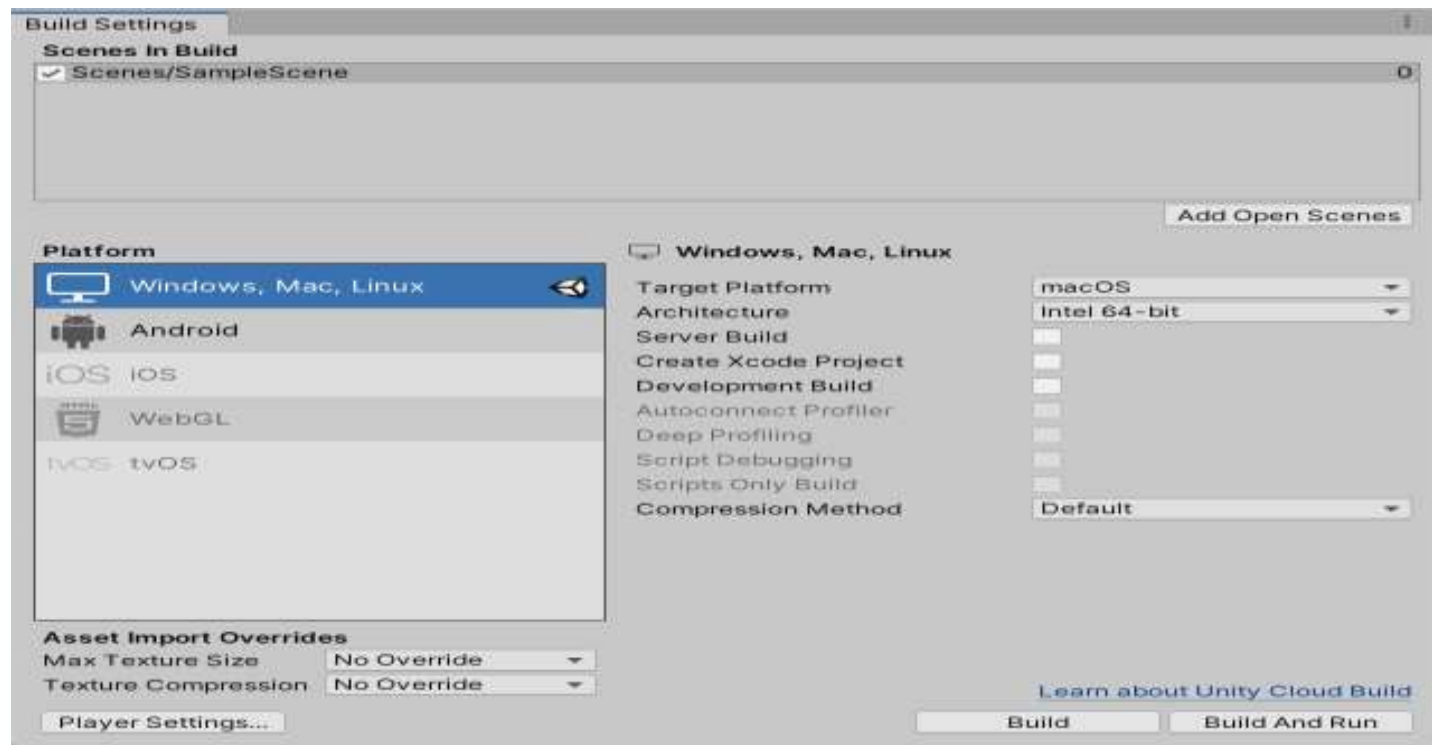
Index4

Index5

빌드

생성: 에셋을 임포트하면 게임이나 앱을 만들기 시작 가능

빌드: 완료된 프로젝트를 바이너리 파일로 익스포트하고
선택한 플랫폼에서 배포하고 실행할 수 있는 프로세서 말함



Index1

Index2

Index3

Index4

Index5

배포

게임 또는 앱을 시장에 제출하는 것

Unity는 자체 클라우드 콘텐츠 전달서비스를 제공하여 게임이나 앱, 해당 콘텐츠를 호스트하고 사용자에게 전달할 수 있음.

Unity 개발 플랫폼에 완벽히 통합되어 있음.



ex1

ex2

ex3

ex4

Index5

게임

- 1.유니티를 이용해 다양한 게임의 장르 제작가능
- 2.C#이나 C++로 구성



ex1

ex2

ex3

ex4

ex5

그래픽



고해상도 렌더 파이프라인을 이용한 그래픽

Index1

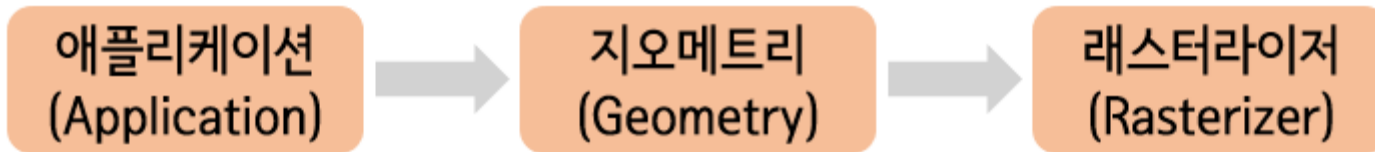
Index2

Index3

Index4

Index5

렌더링 파이프라인



3D 데이터들이 2D이미지로 구성되어 그려지는 것이
이 순서로 데이터를 처리함..

Index1

Index2

Index3

Index4

Index5



애플리케이션 스테이지

애플리케이션 상에서 처리하는 단계를 의미함
본격적인 렌더링 파이프라인 진입전에 CPU에서 필요한 연산들을 처리함..



Index1

Index2

Index3

Index4

Index5

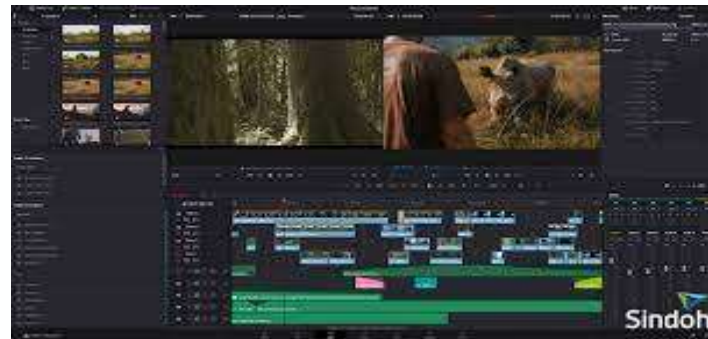
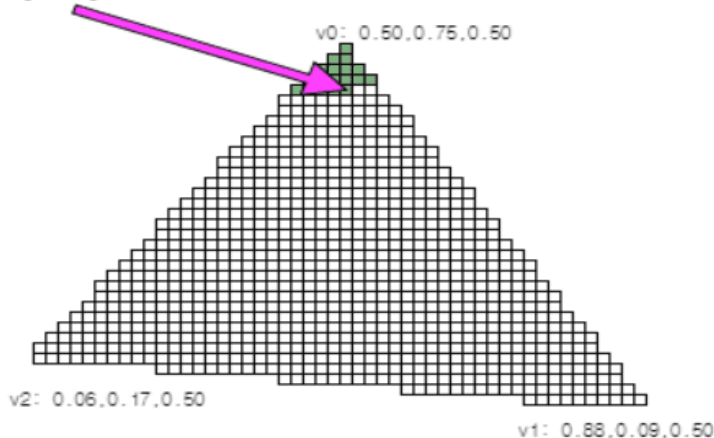
지오메트리

지오메트리의 구성요소인 버텍스와 폴리곤 처리를 담당함.



월드 뷰 프로젝션
트랜스폼

`MODELVIEW = 0.51, 0.67, 0.50`
`gl_FragColor = v_color`



클리핑

버텍스 셰이더

Index1

Index2

Index3

Index4

Index5

래스터 라이저

GPU pipeline에서 맨 처음 단계인 vertex shader의 output을 input으로 받는 두 번째 단계이다. 이 단계는 하드웨어로 고정되고 하드웨어가 수행해 준다.

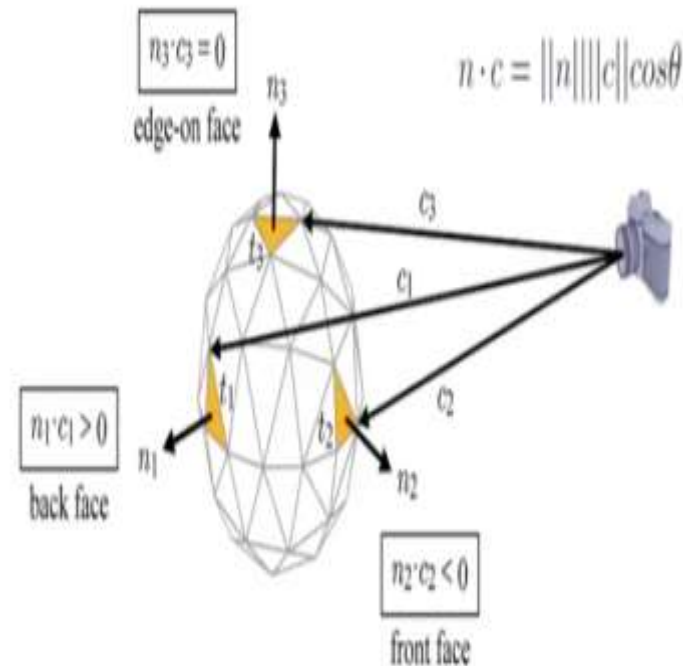
원근 나눗셈(Perspective Division)

- 아핀 변환과 달리, 투영 행렬 M_{proj} 의 네 번째 행은 (0 0 0 1)이 아니라 (0 0 -1 0). M_{proj} 을 (x,y,z,1)에 적용하면, 변환된 w-좌표는 -z임.

$$M_{\text{proj}} = \begin{pmatrix} \cot(\frac{\text{fov}}{2}) & 0 & 0 & 0 \\ 0 & \cot(\frac{\text{fov}}{2}) & 0 & 0 \\ 0 & 0 & \frac{f}{f-a} & \frac{af}{f-a} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} m_{11} & 0 & 0 & 0 \\ 0 & m_{22} & 0 & 0 \\ 0 & 0 & m_{33} & m_{34} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11}x \\ m_{22}y \\ m_{33}z + m_{34} \\ -z \end{pmatrix} \rightarrow \begin{pmatrix} -\frac{m_{11}x}{z} \\ -\frac{m_{22}y}{z} \\ -m_{33} - \frac{m_{34}}{z} \\ 1 \end{pmatrix}$$

- 동차 클립 공간에서 카테시안 공간으로 변환하려면, 각 정점을 w-좌표(-z)로 나누어야 함.



Index1

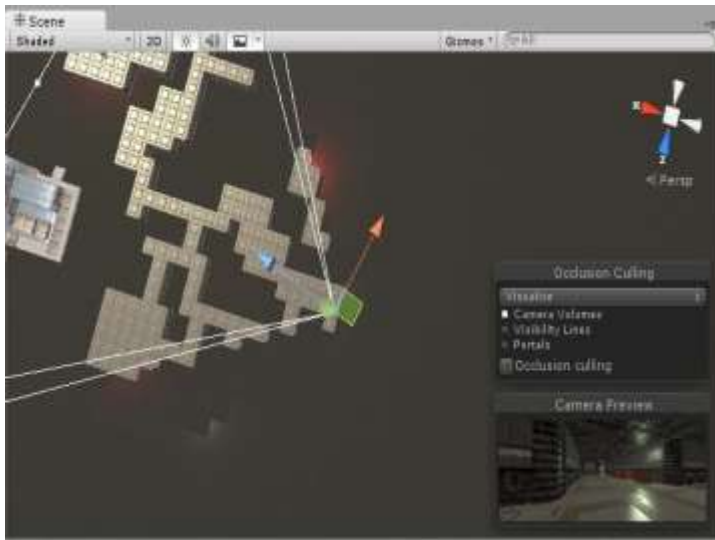
Index2

Index3

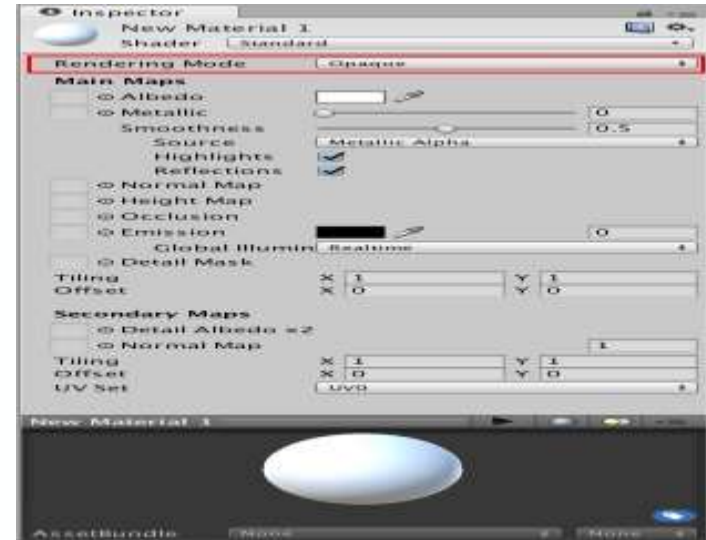
Index4

Index5

랜더 파이프라인 예시



<컬링>



<렌더링 모드>

-> 포스트 프로세싱

ex1

ex2

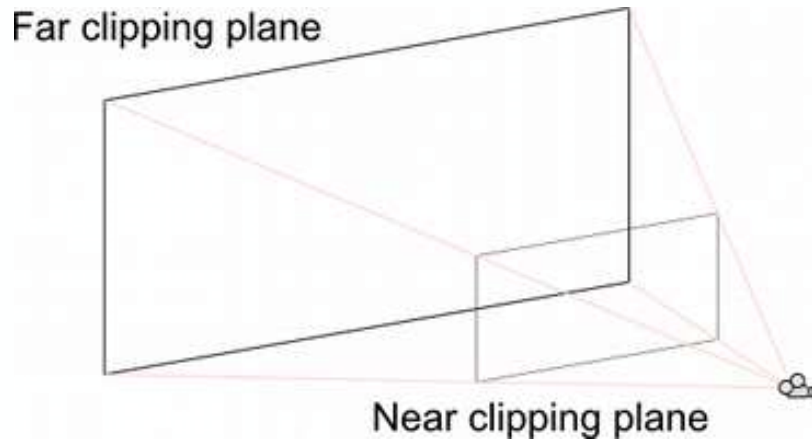
ex3

ex4

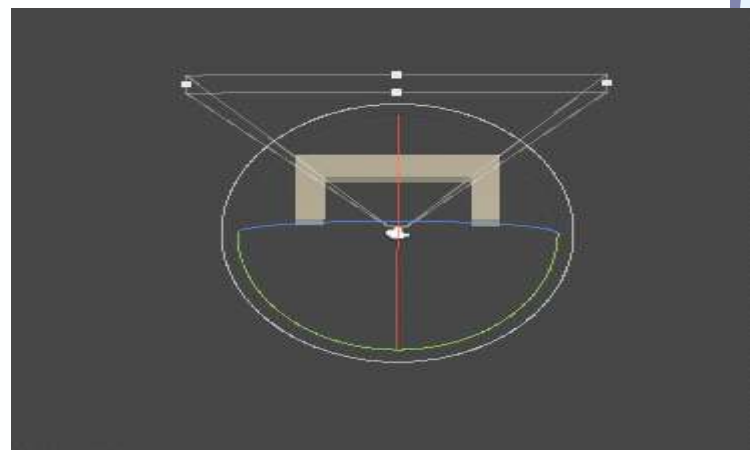
Index5

컬링

이 때 눈에 보이지 않는 오브젝트를 제외함으로써 렌더링할 오브젝트의 수를 줄일 수 있는데, 이처럼 렌더링이 필요하지 않은 오브젝트들을 추려 내는 것을 컬링이라함.



<프러스텀 컬링>



<오클루전 컬링>

Index5

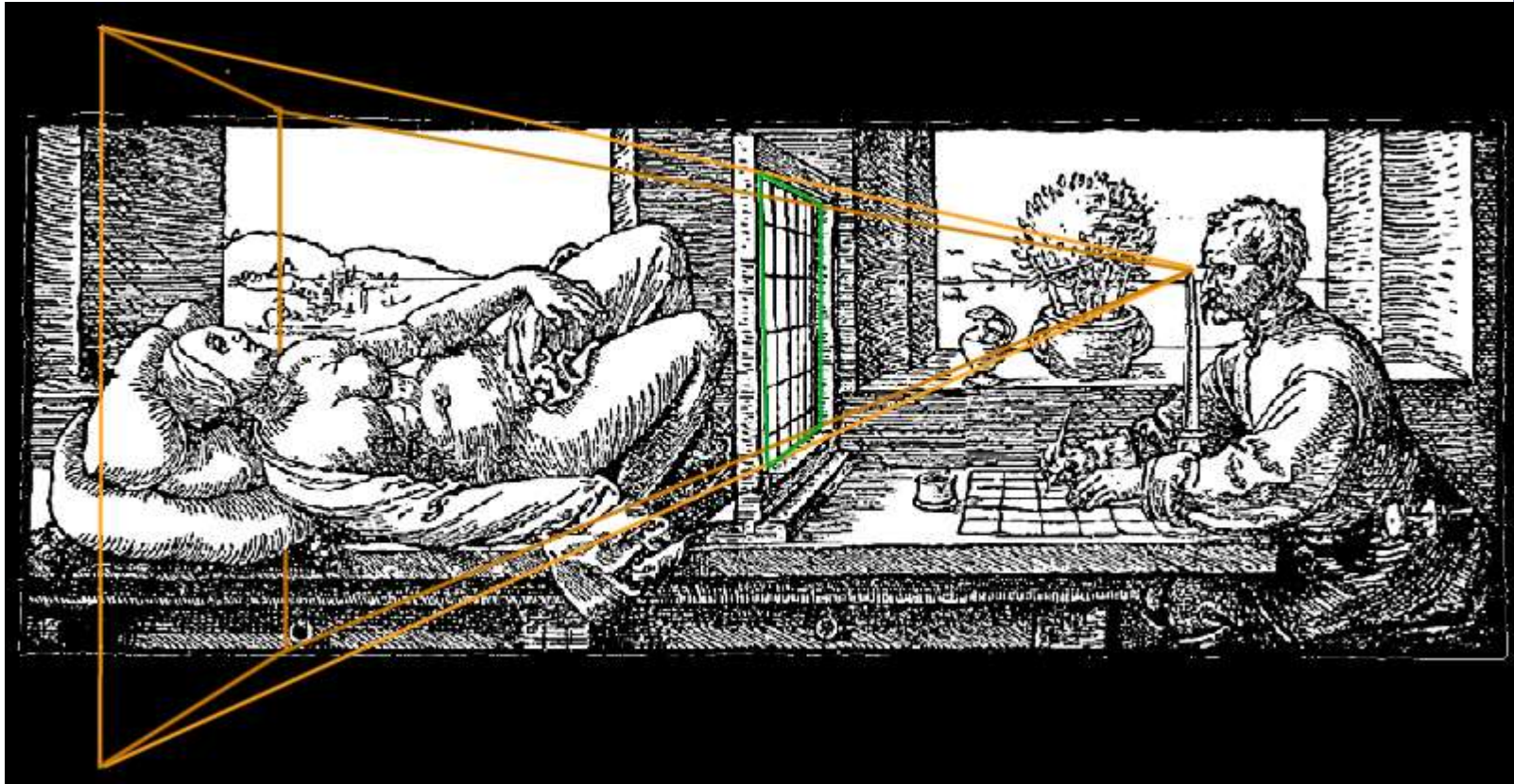
ex1

ex2

ex3

ex4

프러스텀 컬링



우리가 보지 못하는 것을 그려주지 않아 컴퓨터의
계산량을 줄여주는 방법

ex1

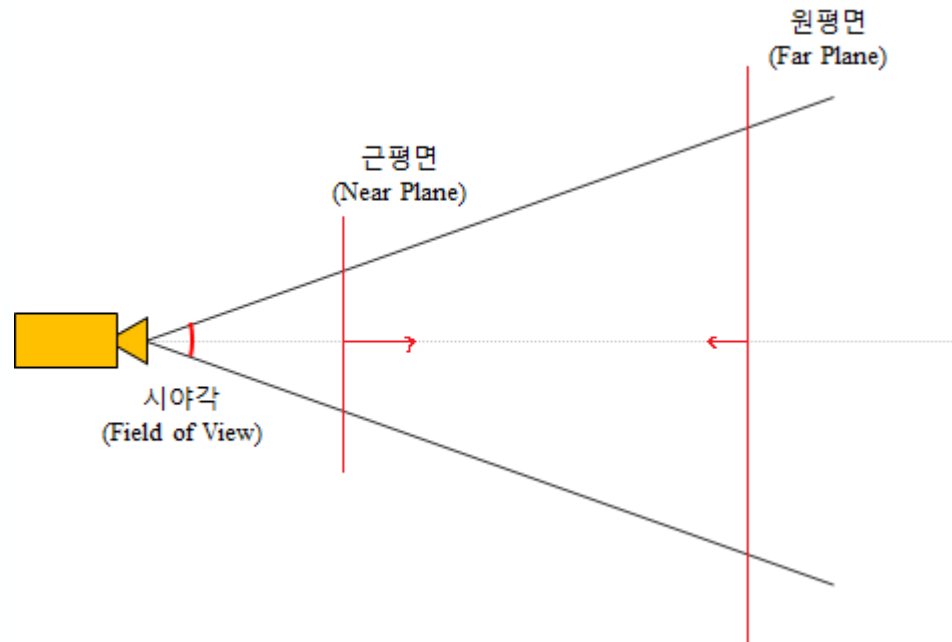
ex2

ex3

ex4

Index5

프러스텀 컬링



상하좌우 근원 평면 6개를 구해야 함.

1. 근평면과 원평면 구하기
2. 상하좌우평면 구하기
3. 뷰 좌표계로 옮기기

ex1

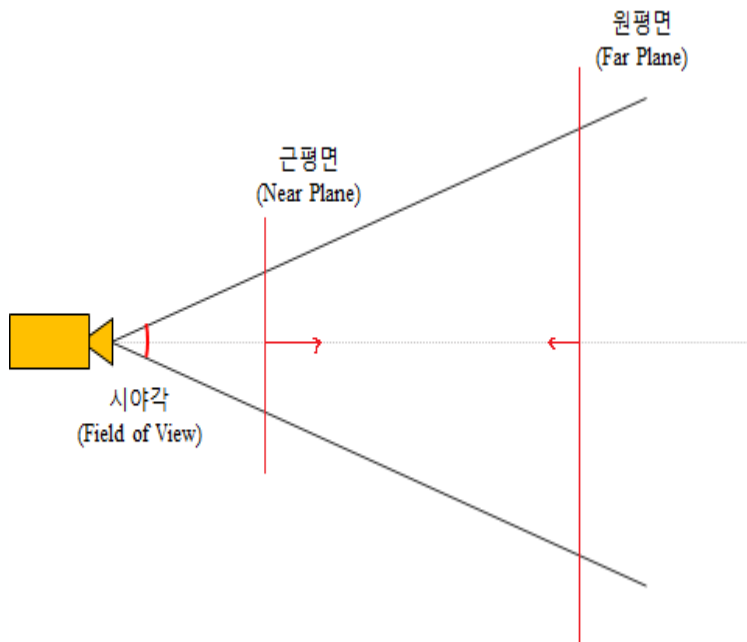
ex2

ex3

ex4

Index5

근평면과 원평면



우리가 알 수 있는 값: Near Z(시야각과 근평면 사이의 거리), Far Z(원평면과 시야각의 사이 거리)

근평면: forward 벡터(0,0,1) x Near Z = 근평면의 점 법선벡터(0,0,1)을 사용

원평면: forward 벡터 x FarZ = 원평면의 점 법선벡터는(0,0,-1)을 사용

ex1

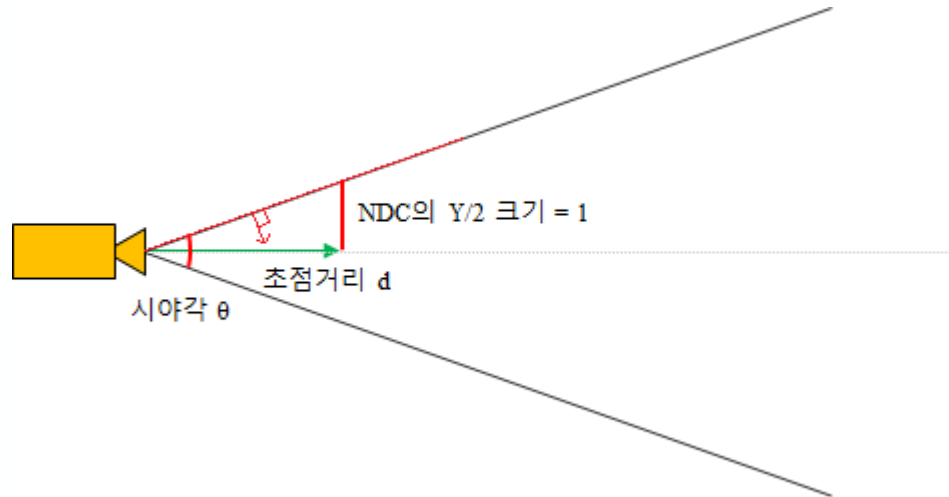
ex2

ex3

ex4

Index5

상하좌우평면



상:(0,-invTanHalfFOV,1)

하:(0,invTanHalfFOV,1)

좌:(-invTanHalfFOV,0,1)

우:(invTanHalfFOV,0,1)

근평면

원평면

이거를 뷰 좌표계에 곱해서 뷰좌표계로 옮겨준다.

x1

ex2

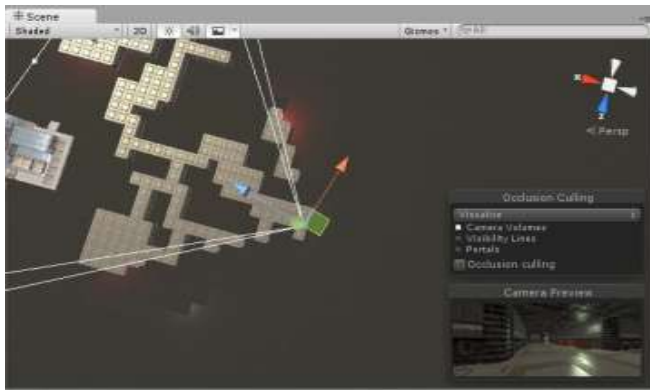
ex3

ex4

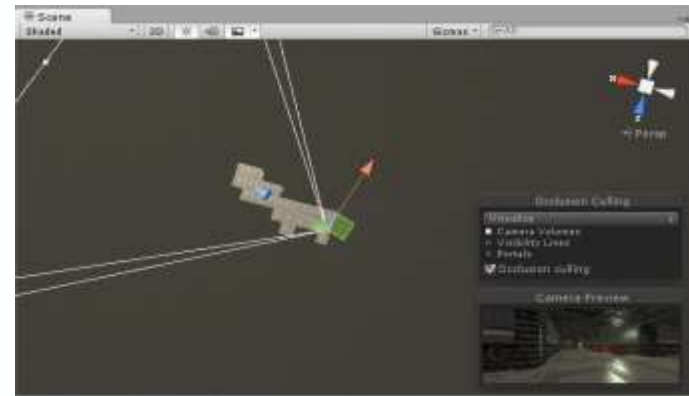
Index5



오클루전 컬링



일반 절두체 컬링:카메라
뷰 내의 모든 렌더러를
렌더링 한다.



오클루전 컬링:더 가까운
렌더러에 의해 완전히 가려진
렌더러 제거

ex1

ex2

ex3

ex4

Index5

오클루전 컬링 작동방식

1. 유니티에서 씬에 대한 데이터를 생성한 후 런타임 시점에서 해당 데이터를 사용하여 카메라가 볼 수 있는 요소를 결정 (베이크)
2. 오클루전 컬링 데이터를 베이크하면 유니티는 씬을 셀로 나누고 셀 내 지오메트리와 인접 셀 간의 가시성을 설명하는 데이터를 생성한 뒤, 생성된 데이터의 크기를 줄이기 위해 가능한 경우 셀을 병합
3. 런타임 시점에서 유니티가 베이크된 데이터를 메모리에 로드하고, 오클루전 컬링 프로퍼티가 활성화된 각 카메라에 대해 해당 데이터에 대한 쿼리를 수행하여 카메라가 볼 수 있는 요소를 결정한다.
4. 오클루전 컬링이 활성화 되면 카메라는 절두체 컬링과 오클루전 컬리를 둘 다 수행하게 된다.

ex1

ex2

ex3

ex4

Index5

스크립팅

플레이어 입력에 반응하고 게임플레이의 이벤트가 적시에 발생하도록 올바르게 준비하도록 하는 것



프로그래밍



Unity 아키텍처

ex1

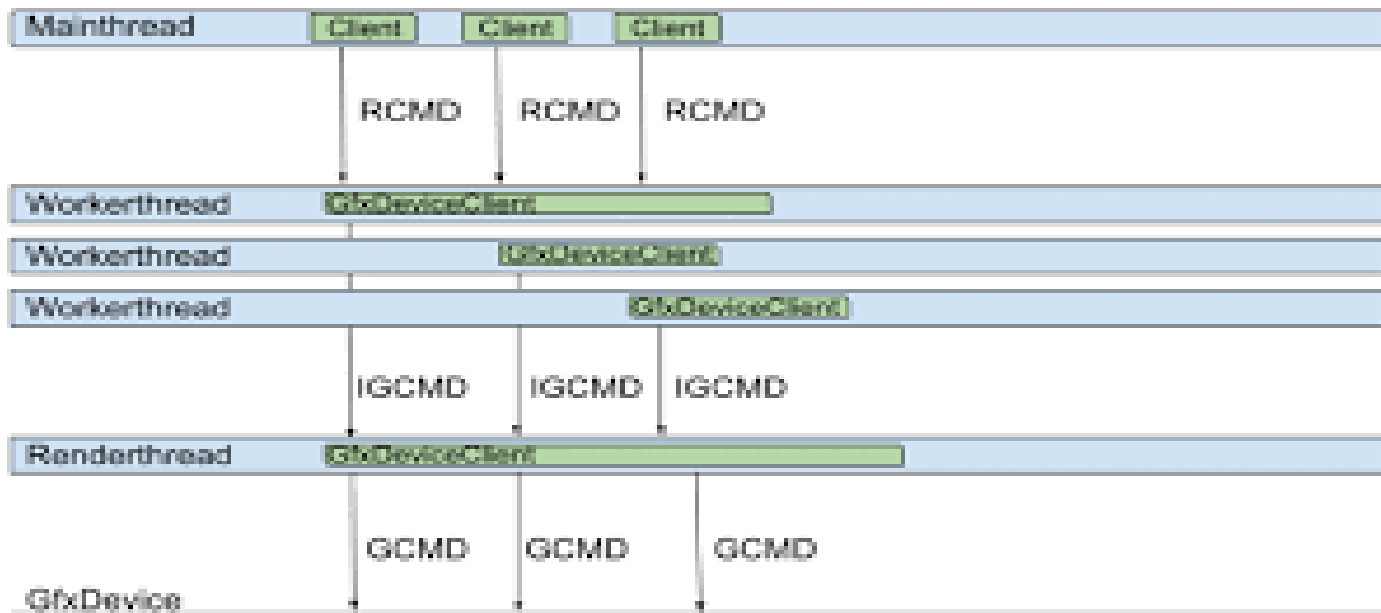
ex2

ex3

ex4

ex5

C# 잡 시스템



Unity 엔진과 상호작용하는 간단하고 안전한 멀티스레드 코드를 작성하여 게임 성능을 개선가능
워커 스레드(렌더링 할 준비를 해주는 것)에 작업을 지시할 수 있게 해주는 시스템을 job system.이라 한다.

ex1

ex2

ex3

ex4

Index5

C# 잡 시스템의 장점

1. 빠르다.
2. 메모리 절약
3. 각종 스레드 안전장치 사용 가능
4. 유니티의 로직 맞춰서 돌아가므로 비교적 안정적인 프레임 기대 가능
5. 다수의 작은 작업을 처리하기에 용이함.

ex1

ex2

ex3

ex4

Index5

오디오



Unity의 오디오 기능에는 완전한 3D 입체 사운드, 실시간 믹싱 및 마스터링, 믹서 계층 구조, 스냅샷 및 미리 정의된 효과 등이 있다.

ex1

ex2

ex3

ex4

Index5

AudioListener



<오디오를 재생하는 방법>

1. GUI를 사용한 재생
2. 스크립트를 사용한 재생

Index1

Index2

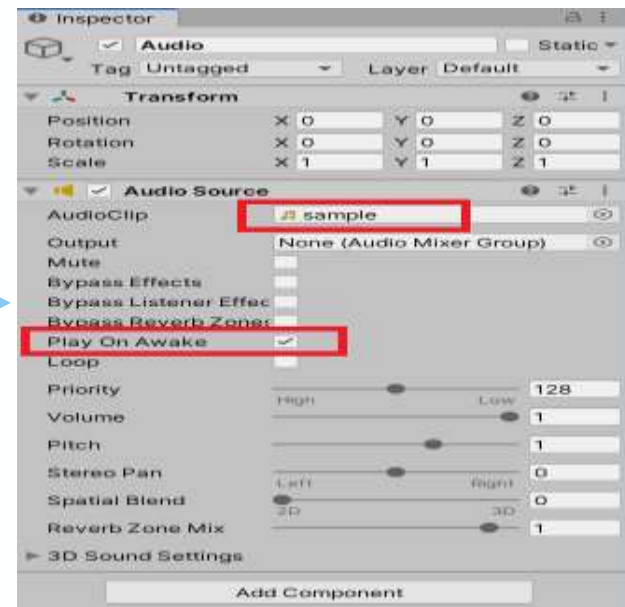
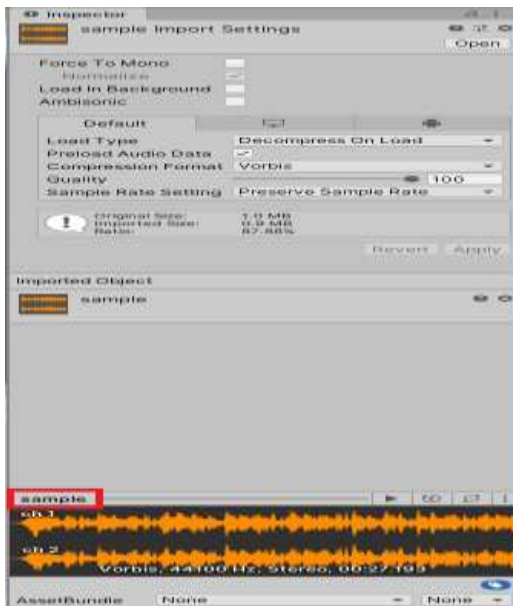
Index3

Index4

Index5

GUI를 사용한 재생

1. mp3등 오디오 파일을 유니티의 Assets 폴더 밑에 복사
2. 유니티 GUI로 들어가면 유니티가 알아서 해당 오디오 파일을 Vorbis로 변환해 리소스로 추가해 준다.
3. 하는 게임 오브젝트에 Add Component 버튼을 통해 Audio->AudioSource를 추가한다. 그리고 해당 Audio Source 컴포넌트의 AudioClip에 위에서 임포트한 sample을 드래그 드랍으로 지정한다.
4. Play On Awake 체크하면 끝



ex1

ex2

ex3

ex4

Index5

스크립트를 이용한 재생

```
// 오디오 소스 생성해서 추가
AudioSource audioSource = gameObject.AddComponent<AudioSource>();

// 뮤트: true일 경우 소리가 나지 않음
audioSource.mute = false;

// 루핑: true일 경우 반복 재생
audioSource.loop = false;

// 자동 재생: true일 경우 자동 재생
audioSource.playOnAwake = false;
```

audio.Source.Play(); -> 소리를 플레이한다.
audio.Source.Stop(); -> 소리를 멈춘다.

ex1

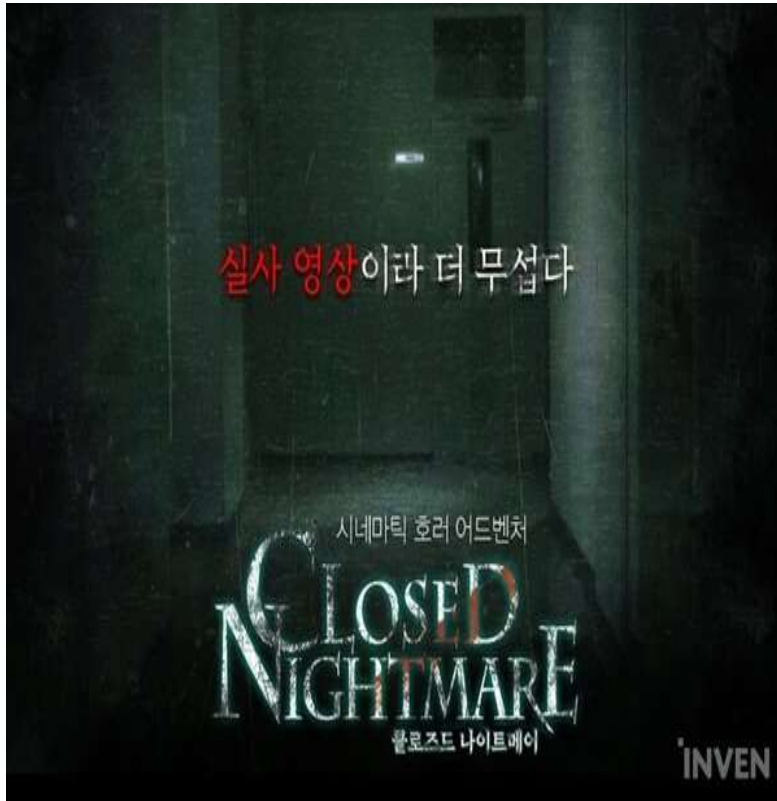
ex2

ex3

ex4

Index5

동영상



현실감을 더하거나, 렌더링
복잡성을 줄이거나, 외부에
존재하는 콘텐츠를 통합 가능함.

동영상을 재생하는 방법
Vidio Player 컴포넌트를 이용해
동영상 틀기

ex1

dex2

dex3

dex4

Index5

Video play 컴포넌트를 이용한 동영상 틀기



Source: 비디오 타입 선택
Video Clip: 비디오 클립을
비디오 플레이어에 할당
Play on Awake: 비디오 재생
Wait for First Farme: 첫 프레임
표시 될 때 가지 기다림
Loop: 비디오 반복
Playback speed: 플레이 속도
Render Mode: 드롭다운을
사용하여 비디오 렌더링 방법
정의
Material property: Video Player
컴포넌트 이미지를
수신하는 머티리얼 텍스처
프로퍼티
Audio out put mode: 소스의
오디오 트랙 출력방법 정의

ex1

ex2

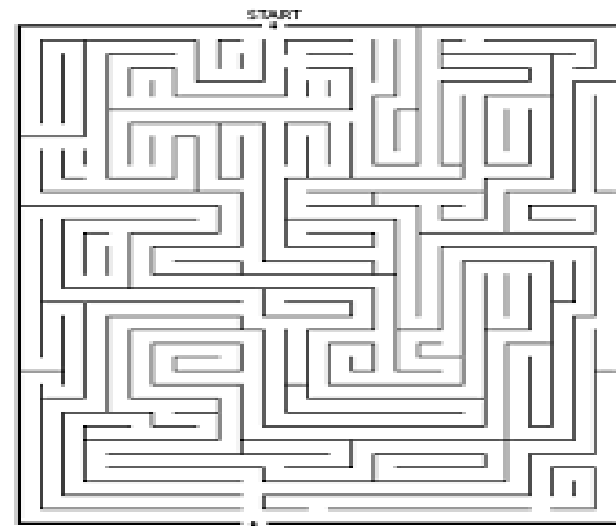
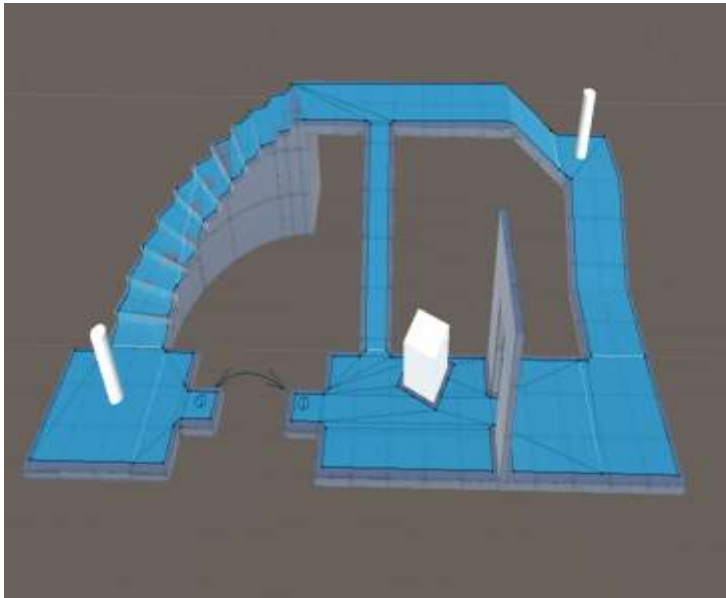
ex3

ex4

Index5

내비게이션과 경로탐색

썬 지오메트리에서 자동으로 생성되는 내비게이션 메시를 사용하여 게임 월드에서 지능을 갖고 움직일 수 있는 캐릭터를 생성하는데 도움을 줌.



© 2015 All Rights Reserved. All rights reserved.

ex1

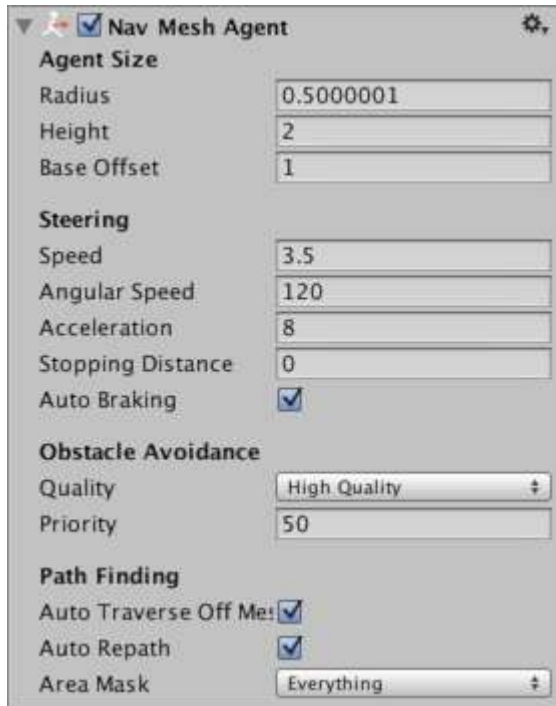
ex2

ex3

ex4

Index5

내비메시 에이전트



Radius:에이전트의 반경은 장애물과 다른 에이전트 간의 충돌 계산하기 위해 사용

Height:높이간격

Speed:최대이동속도

Angular Speed:최대 회전속도

Acceleration:최대 가속

Stopping Distance:에이전트는 목표 위치에 가까워졌을 시 정지

Auto Braking:목적지에 다다를때 속도 줄임

Quality:장애물 회피

Priority: 낮은 우선 순위의 에이전트는 회피대상에서 제외

ex1

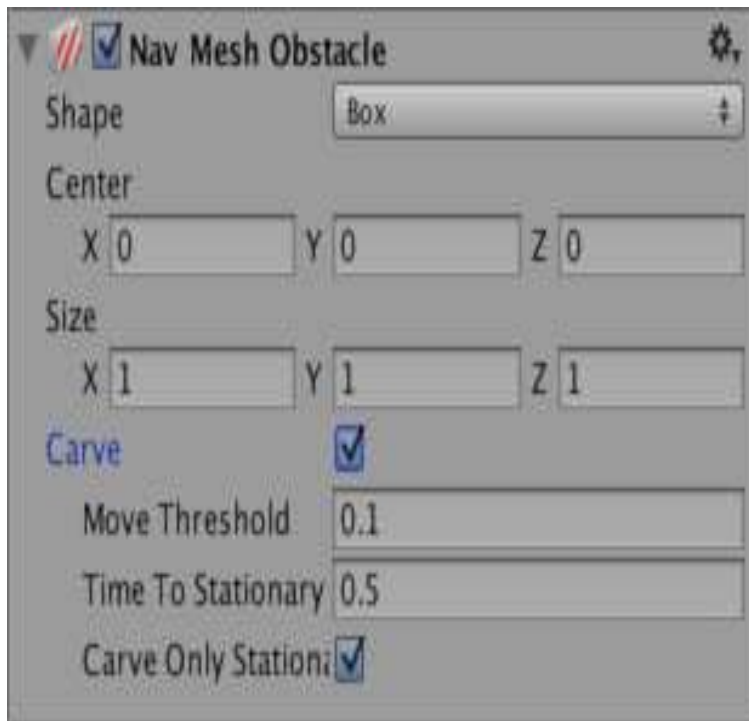
ex2

ex3

ex4

Index5

내비메시 장애물



Shape:장애물 지오메트리의 모양
Center:변환 포지션에 대한 박스의 상대적인 중심

Size: 상자의 크기

Crave:내비메시 장애물이 내비메시에 구멍을 만듦

Move Threshold: 움직이는 파인 구멍을 업데이트하는 임계 거리를 설정하는 데 사용

Time to Stationary:장애물이 정지했다고 간주할 때 까지 기다리는 시간

Crave Only Sation:체크시 장애물이 정지되어 있을때 구멍을 팜

ex1

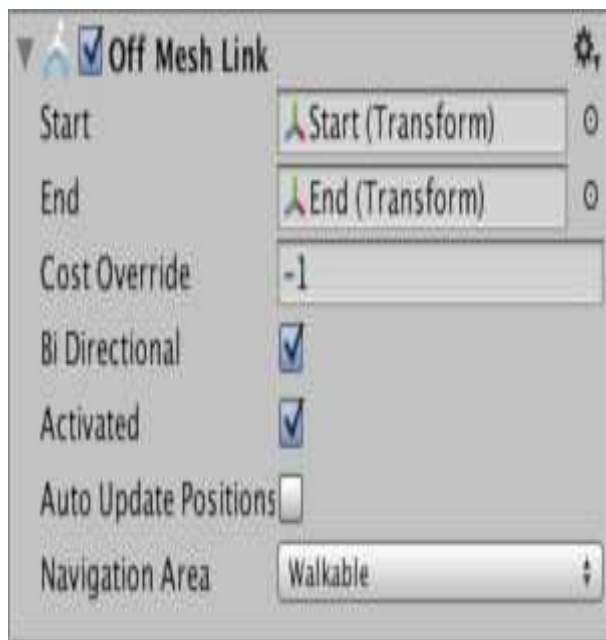
ex2

ex3

ex4

Index5

오프 메시 링크



Start: 오프 메시 링크의 시작 지점

End: 오프 메시 링크의 시작 지점

Cost Override: 값이 양수이면 패스 요청을 처리하는 데 드는 패스 비용 산출에 사용(ex)같은 거리를 이동하는데 비용이 세배가 됨

Bi Directional: 양방향으로 이동 가능

Activated: 링크를 경로 탐색에 사용할 것인지 여부를 설정합니다(거짓인 경우 무시됨).

Auto Update Positions: 오프 메시 링크의 끝 지점이 이동할 때 내비메시에 재연결

Navigation Area: 링크의 네비게이션 영역 타입

ex1

ex2

ex3

ex4

Index5

유니티 C++ 문법정리

형태	예약어	메모리 크기	디폴트 값	표현 범위
논리값	boolean	1 비트	false	true 혹은 false
문자	char	2 바이트	'\0'	0~65535
정수	byte	1 바이트	0	-128 ~ 127
	short	2 바이트	0	-32768 ~ 32767
	int	4 바이트	0	-2의 31승 ~ 2의 31승 -1
	long	8 바이트	0	-2의 63승 ~ 2의 63승 -1
실수	float	4 바이트	0.0	-3.4E38 ~ 3.4E38
	double	8 바이트	0.0	-1.7E308 ~ 1.7E308

우선순위	연산자	내용
1	() , []	괄호 / 대괄호
2	! , ~, ++, --	부정 / 증감 연산자
3	*, /, %	곱셈 / 나눗셈 연산자
4	*, =	덧셈 / 뺄셈 연산자
5	<<, >>, >>>	비트단위의 쉬프트 연산자
6	<, <=, >, >=	관계 연산자
7	==, !=	
8	&	비트단위의 논리연산자
9	^	
10		
11	&&	논리곱 연산자
12		논리합 연산자
13	?:	조건 연산자
14	=, +=, -=, *=, /=, %%, <<=, >>=, &%=, -=, ~=	대입 / 할당 연산자

연산자의 모습	설명
++식	단항 연산자, += 1과 동일하게 작동함. 먼저 피연산자의 값을 1증가시키고 다른 연산에 사용. l-value에만 사용이 가능하며 결과값도 l-value. 후위 증감 연산자보다 우선순위가 낮음
--식	단항 연산자, -= 1과 동일하게 작동함. 먼저 피연산자의 값을 1감소시키고 다른 연산에 사용. l-value에만 사용이 가능하며 결과값도 l-value. 후위 증감 연산자보다 우선순위가 낮음
식++	단항 연산자. 먼저 피연산자의 값을 다른 연산에 사용하고 값을 1증가시킴. l-value에만 사용이 가능하며 결과값도 l-value. 전위 증감 연산자보다 우선순위가 높음
식--	단항 연산자. 먼저 피연산자의 값을 다른 연산에 사용하고 값을 1감소시킴. l-value에만 사용이 가능하며 결과값도 l-value. 전위 증감 연산자보다 우선순위가 높음

ex1

ex2

ex3

ex4

Index5

조건문

성적 출력 프로그램

```
#include <iostream>
using namespace std;
int main() {
    int score;
    cin >> score;
    if (score >= 90 && score <= 100) cout << "A";
    else if (score >= 80 && score < 90) cout << "B";
    else if (score >= 70 && score < 80) cout << "C";
    else if (score >= 60 && score < 70) cout << "D";
    else cout << "F";
    return 0;
}
```

ex1

ex2

ex3

ex4

Index5

반복문과 배열

```
2023 03 30.cpp * x
2023 03 30 (전역 범위) main()

1 #include <iostream>
2 using namespace std;
3 int main() {
4     int array[100] = {};
5     int temp = 0;
6     for (int i = 0; i < 10; i++)
7     {
8         cin >> array[i];
9         for (int j = 0; j < i; j++)
10        {
11            if (array[j] > array[j + 1])
12            {
13                temp = array[j];
14                array[j] = array[j + 1];
15                array[j + 1] = temp;
16            }
17        }
18    }
19    for (int i = 0; i < 10; i++)
20    {
21        cout << array[i] << " ";
22    }
23    return 0;
24 }
```

버블정렬

ex1

ex2

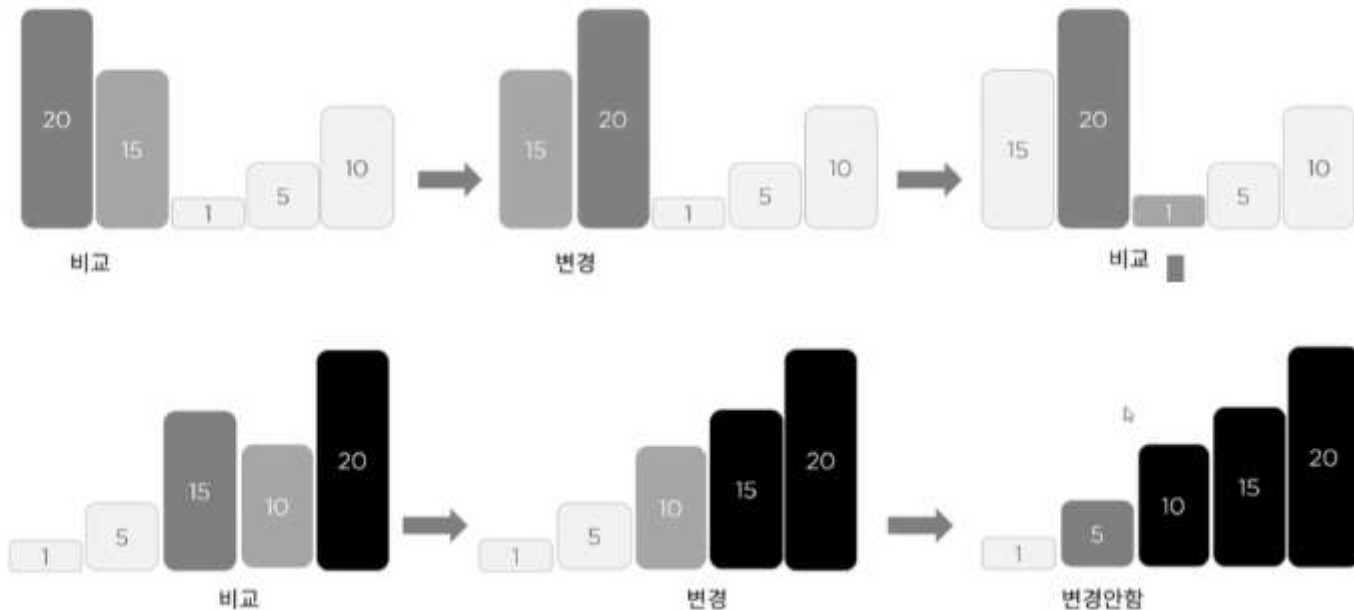
ex3

ex4

Index5

버블정렬 (Bubble sort)이란?

- 버블정렬(Bubble sort)이란?



지금까지 발표를 들어주셔서
감사합니다.

