

# DirectX 11 PF FrameWork

## 외부라이브러리 관련 링크

### Assimp

assimp/assimp

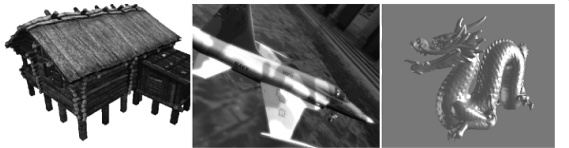
A library to import and export various 3d-model-formats including scene-post-processing to generate missing render data. APIs are provided for C and C++. There are various bindings to other languages (C#, Java, Python, Delphi, D). Assimp also runs on Android and iOS. Check the latest

 <https://github.com/assimp/assimp>



Asset-Importer-Lib

 <https://assimp-docs.readthedocs.io/en/latest/>



## Project Setting

- 미리컴파일된 헤더 사용 - D3DUtil.h
- 유니코드 문자 집합 사용
- 외부라이브러리 사용 - Assimp\_5.0.1
- 프로젝트속성(Properties) > C/C++ > 언어 > 준수모드 > 아니오

## How To Use FrameWork

- 텍스쳐파일과 모델파일이 같은 폴더에 있으면 자동으로 텍스쳐를 불러오지만 다른폴더에 있거나 Assimp가 텍스쳐 파일을 불러오지 못하는경우 수동으로 설정해줘야 합니다.

## Issue

- ☒ FBX 모델을 Assimp로 불러올때 계층구조가 부정확하게 불러와지는 문제
- ☐ Animation 실행시 프레임 저하되는 문제.

## List To Add

- ☐ Add Image Component
- ☐ Add Event System

## Description

아직 완성되지 않은 프레임워크입니다. 이해안가는부분, 구조상 문제, 메모리누수, 버그 발견시 말씀해주세요. 혹은 더 낫은 구조가 생각하면 말씀해주세요.

# Reference

## Class

▼ GameObject

▼ Function

```
void AddComponent(Component* _component);
컴포넌트를 추가합니다.
```

```
T* GetComponent();
컴포넌트를 가져옵니다.
```

```
template<typename T>
vector<T*> GetComponentsInChildren();
해당 오브젝트의 자식 오브젝트를 탐색해 컴포넌트를 vector에 담아 반환합니다.
```

```
template<typename T>
vector<T*> GetComponentsInAllChildren();
해당 오브젝트의 모든 자식 오브젝트를 탐색해 컴포넌트를 vector에 담아 반환합니다.
```

```
GameObject* GetRootObject();
최상위 오브젝트를 반환합니다.
```

```
void AddChild(GameObject* _gameObject);
오브젝트를 자식으로 추가합니다.
```

▼ variable

```
GameObject* parent;
부모 오브젝트의 인스턴스를 가리키는 포인터 변수
```

```
Transform* transform;
오브젝트의 Transform 컴포넌트의 인스턴스를 가리키는 포인터 변수
```

```
RectTransform* rectTransform;
오브젝트의 RectTransform 컴포넌트의 인스턴스를 가리키는 포인터 변수
오브젝트가 2D, UI 일때만 추가된다.
```

```
bool active;
오브젝트의 활성화 여부.
```

```
string directory;
불러온 모델의 경로
```