# Custom Binary Formats.

Skeleton, Blend Shape, Model, Material, Light, Camera

Different filetypes, under same namespace

Convert fbx automatically.

If it finds an fbx file it runs it through the programme, that converts it to an actor file.

The actor file detects what is in it and creates the different objects. Under the name space actorfile

That way I can placec fbx files in the content directory and it converts it automatically. Unless a file with the same name already exists as an actor file. The framerate is 30 fps for animations.

GUI is not necessary.

Smoothing normal can be done inside the engine. The name should be game object not actor its too close to unreal. Short gofile.

# GUI using Python and a circular buffer.

Reimport file button.

Level editor. Transform objects. Move camera change camera speed fov.

# Physics Engine

Implemented as a library.

Sort and Sweep algorithm.

# Physics Experimentation

# Shaders

# 3D Techniques

Simple things first.

# Particle Systems

# DX11

# Structure

**Actor/Gameobject class.**

Abstract class. GO

Item//Actor

**Pawn DynamicItem**

An actor that can move around.

**Character**

An actor that can move around and have character logic.

**Game Mode**

Defines logic for the current scene.

Light

Camera

Mesh

SkeletonMesh

**RenderManager gRenderManager;**

Takes care of rendering.

**PhysicsManager gPhysicsManager;**

Physics

**AnimationManager gAnimationManager;**

All animations

**TextureManager gTextureManager;**

**MemoryManager gMemoryManager;**

Try to keep things in pools.

**FileSystemManager gFileSystemManager;**

**SceneManager gFileSystemManager;**

Handles game logic

# 2 Resolutions

# Fullscreen and Minimized and Level Editor

# Custom containers

Array

Map

Queue

Under a name space.

# Make small games

# Copy the same to a DX12 engine.