

# GEOM3TR1C: A real-time rendering animation using DirectX

Thanik Sitthichoksakulchai  
ID: 1804397

## Introduction

GEOM3TR1C is a real-time rendering animation (Demoscene) utilizing DirectX API based on DirectX Tool Kit (DirectXTK).

## User Controls

- Press Space Bar to start playing the animation.
- Press Alt+Enter to go full screen.

## Essential Application Components

### Loading Objects and Models

The application supports two types of models, CMO file model and simple geometric model using GeometricPrimitive class in DirectXTK. When the application starts, it will read “Assets/ObjectList.csv” file to gather how many objects in the scene and type of them then initialize. After objects are initialized, the application will look for each object’s keyframe file, read and store data into a C++ vector.

### Format of an object list file

This file uses CSV (comma-separated values) format. Each data field is separated by a comma. The format is as follows.

Object ID, Type of object, Parameters to initialize the object

### Definition

**Object ID:** a number to identify the object.

**Type of object:** type of object which can be CAMERA, FILE, CUBE, CONE, DODECAHEDRON, GEOSPHERE, ICOSAHEDRON, OCTAHEDRON, SPHERE, TEAPOT, TETRAHEDRON, TORUS

**Parameters to initialize the object:** Each type needs different parameters to initialize the object, for example, FILE type needs a file name to load, CUBE type needs a size number of a cube.

## Hard-coded object

Directional light and camera have their own hard-coded object ID which are -1 and 0 respectively.

## Object Transformation

Every frame, all objects in the scene are transformed using this sequence: Scaling -> Rotation -> Position. All transformations utilizes static functions in SimpleMath::Matrix class. CreateScale function is used for scaling, CreateFromYawPitchRoll for rotation and CreateTranslation for position.

## Background Music

The application uses XAudio2 API to play background music. It initializes the audio engine (DirectX::AudioEngine) first then loads an audio file named “Assets/Audio/audio.wav” into a SoundEffect class. Play function on SoundEffect class will be called after the space bar key is pressed.

## Lighting

DirectX built-in BasicEffect is used for directional per-pixel lighting. This effect can be applied to any object by setting the effect in the draw function. The application uses SetDiffuseColor function for light reacting to the beat of background music, SetTexture for applying a texture to models and SetLightDirection for changing the direction of the light to pre-defined keyframes.

## Simple Animation System

The animation system will calculate the attributes of objects (position, rotation, scale) between keyframes based on the current application time. For Vector3 value, the class static Lerp method will be used but for float value, the calculation will be done manually using the following formula.

$$V_c = V_s + (V_e - V_s) \times P_c$$
$$P_c = \frac{(T_c - T_s)}{(T_e - T_s)}$$

$V_c$  is calculated current value.

$V_s$  is start keyframe's value.

$V_e$  is end keyframe's value.

$P_c$  is current percentage.

$T_c$  is current time.

$T_s$  is start keyframe's time.

$T_e$  is end keyframe's time.

Each object in the scene has its own keyframes file named after its object ID. For instance, if the object ID is 2, the keyframe file's name will be "Assets/Keyframes/2.csv".

## Format of a keyframes file

The keyframes file also uses CSV format. Each keyframes entry has the format as follows.

Start Time, End Time, Animation Curve Type, X value of start position, Y value of start position, Z value of start position, X value of end position, Y value of end position, Z value of end position, X value of start rotation, Y value of start rotation, Z value of start rotation, X value of end rotation, Y value of end rotation, Z value of end rotation, Start scale value, End scale value, Flag to in wireframe or not

## Animation Curve Type

This system has four animation curve types which are linear, ease-in, ease-out, ease-in-out.

### Linear

This animation curve type has a constant speed on the value changing. The current percentage will be used directly.

### Ease-in

This animation curve type has a slow starting speed on the value changing. The current percentage will square to form an exponential graph.

### Ease-out

This animation curve type has a slow ending speed on the value changing. The ease-out graph is similar to ease-in one but flipped. A new current percentage will be calculated using this formula.

$$P = 1 - (1 - P_c)^2$$

$P$  is new percentage.

$P_c$  is current percentage.

### Ease-in-out

This animation curve type has both slow starting and ending speed on the value changing. This graph combines both ease-in and ease-out by using linear interpolation between two graphs.

$$P = P_c^2 + (1 - (1 - P_c)^2)$$

$P$  is new combined percentage.

$P_c$  is current percentage.

## Further Improvements

- The animation system supports more graph curve types and spline movement.
- More types of light e.g. point light
- Better workflow for editing the keyframe file and previewing changes
- The music can be out-of-sync to the animation because SoundEffect class doesn't provide API to get the music time or jump to a specific time. This can be improved by using other API for sound.
- Object's Shadow

## References

Walbourn, C. (2019) *DirectXTK Wiki*. Available at:

<https://github.com/microsoft/DirectXTK/wiki> (Accessed: 10 December 2019).

Federico. (2018) *Easing Functions*. Available at:

<https://www.febucci.com/2018/08/easing-functions/> (Accessed: 10 December 2019).