CSC 575 Game Engine 2
PA5 Report
Zeying Yu

How my converter works

1. Reading data from fbx files and tga files.
2. Hijacking all vertices and polygons from fbx files and saving in FBX containers.
3. Fat then reduce all data then saving in vertices chunk, trilist chunk, and texture chunk.
4. Saves the bone data header, then the bone data, then saves all of the animation data header info, then the animation data.
    a. Uses DisplayAnimation function to get all animation data from the gbx file
    b. Saves to custom float struct to hold the bone data
    c. Written as one large buffer
5. Running the batch file
    a. Using Arguments “ImportScene.exe inputfbxfile.fbx outputfile.azul”
6. Outputting azul files with vertices trilist and animation data combined for each model in specified path.

My format of the runtime

Package header containing the name, version, and total size. Following is each of the chunks, containing the chunk type, name, and size. After each chunk is the data corresponding to that chunk type. There is a chunk type for the animation data, then the animation data is written as a collection of floats following that header.

The work I have done

- **Engine**

  Created an animation manager that can accept data from files instead of preloaded clip and skeleton data (AnimMan.cpp). Edited Skeleton and Clip objects to remove hardcoded data values.

  Loaded data from azul file then parsed into proper bone and animation structures allowing for full animation. Reads large buffer then separates into individual frame buffers to send to the animation engine.

  Created new animations with more frames than the initially given animations. My animation has 20 total frames over 20 time periods instead of the supplied 3 frames over 11 time periods. This makes the animation smoother and longer.

Created support for looping and playing animations loaded from the file in the scene. Can load any azul converted animation with the expected data format.

- **Converter**

Created custom data objects to hold the data. Parsed the skeleton bone and animation data into my custom format, then created the corresponding headers. Wrote all data to the azul output file.

Created batch file to support converting multiple gbx files to azul files at once.

Difficulties

I had trouble dealing with the memory layout of my converted files. I sometimes wrote the wrong data, and it was difficult to notice until much later. I also found it hard to support multiple models on the screen at the same time, as they seem to collide with each other in memory. I found the setup up working on two halves of a project to be difficult, but a good experience to learn from.