Fbx Artist Instructions

↓

AssetPipeline

(Creates the runtime assets while

performing validity checks to make

sure source assets fit the rules)

↓

ozz Runtime

↑ ↑

The Forge Gladiator

Artist Instructions

* Animated Objects
  + Must be in FBX
  + Create folder with the name of the animated object
  + In this folder a single file entitled “riggedMesh.fbx”
    - This file must have all the skeletal data required with the mesh it is bound to included as well
    - No additional information/nodes that do not relate to the skin or skeleton
    - No animation data, skeleton and mesh must be in bind pose. Before exporting, select the mesh and go to Rigging->Skin->Go to Bind Pose
  + In this folder a subfolder entitled “animations”
    - Each animation must be in its own separate fbx file
    - No mesh/skin data or other extraneous data besides skeleton and animation data should be present

Programming documentation

**Library interface:**

struct ProcessAssetsSettings  
{  
 bool quiet; // Only output warnings.  
 bool force; // Force all assets to be processed.  
 uint minLastModifiedTime; // Force all assets older than this to be processed.  
};  
   
class AssetPipeline  
{  
public:  
 static bool ProcessAnimations(const char\* animationDirectory, const char\* outputDirectory,

ProcessAssetsSettings\* settings);

static bool CreateRuntimeSkeleton(const char\* skeletonAsset, const char\* skeletonName,

const char\* skeletonOutput, ozz::animation::Skeleton\* skeleton,

ProcessAssetsSettings\* settings);

static bool CreateRuntimeAnimation(const char\* animationAsset,

ozz::animation::Skeleton\* skeleton, const char\* skeletonName,

const char\* animationName, const char\* animationOutput,

ProcessAssetsSettings\* settings);

};

class AssetLoader  
{  
public:  
 static bool LoadSkeleton(const char\* skeletonFile, FSRoot root,

ozz::animation::Skeleton\* skeleton);

static bool LoadAnimation(const char\* animationFile, FSRoot root,

ozz::animation::Animation\* animation);

};

! AssetLoader currently only used by AssetPipeline !

**Errors and warnings:**

|  |  |
| --- | --- |
| Errors: | Warnings: |
| * Invalid usage of the application. | * No assets found. |
| * Failure to read, write or create files. | * Assets don’t follow artist instructions. |
| * Invalid skeleton. | * Invalid animation. |

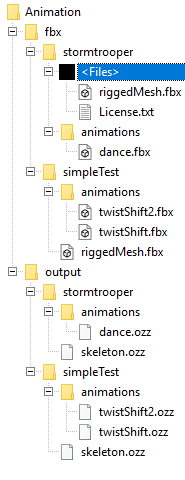
! Invalid skeletons throw errors because they’re required to draw skeletal mesh !

**Command line tool:**

/> AssetPipelineCmd.exe processanimations “<path to fbx assets>” “<output path>” [--quiet] [--force]

Example:

/> AssetPipelineCmd.exe processanimations “<Animation/fbx>” “<Animation/output>”



**Code:**

The code should be self explanatory. The comments explain the basic flow of the program.

**References:**

Assimp: <http://assimp.sourceforge.net/lib_html/index.html>

<http://assimp.sourceforge.net/lib_html/data.html>

OZZ: <http://guillaumeblanc.github.io/ozz-animation/>

**Known problems:**

* Skeletons and animations scaled from centimeters to meters (\* 0.01). This is only required for FBX files. If we want to support other formats this has to be changed.
  + See: AssetPipeline.cpp - line: 392 & 537 & 546
* Boris (Unreal Paragon asset) rotated 90 degrees on x axis after putting it through the animation pipeline. Blender also doesn’t import model correctly. Rotating the model in either Maya or Blender causes severe problems in the animation. Makes it look like his nose, fingers and ankle are all broken.