Sample Code README:

| Class Name | Description |
|---|---|
| Animation.h<br>Animation.cpp | Animation files loader and store all the channels. |
| AnimationPlayer.h<br>AnimationPlayer.cpp | Stores pointer to an animation and a skeleton. Track the time, get pose data from animation and pose the skeleton. |
| BMPImage.h<br>BMPImage.cpp | A simple object class that loads a bmp file and stores it inside. |
| camera.h<br>camera.cpp | Stores values of a camera. |
| Channel.h<br>Channel.cpp | Stores types of channel and a vector of Keyframes. |
| core.h<br>core.cpp | A helper class that uses to draw wire box and axis using glut. |
| cube.h<br>cube.cpp | An example of a basic animating object. |
| DOF.h<br>DOF.cpp | Degree of freedom class, it contains the value of current DOF and min/max in this DOF. |
| Joint.h<br>Joint.cpp | A joint object stores an array of DOFs, world/local matrices and its tree data. It loads data and update in each frame. |

| | |
|---|---|
| Keyframe.h<br>Keyframe.cpp | Stores time, value, tangent type and cubic function to next frame. |
| matrix34.h<br>matrix34.cpp | The basic component of graphic system, uses to describes location, rotation, scale and etc. |
| morph.h<br>morph.cpp | A class that loads modified skin data and stores it. |
| Pose.h<br>Pose.cpp | An object stores all pose value in every joint. |
| Skeleton.h<br>Skeleton.cpp | Skeleton files loader, stores a pointer to a root joint. Update each joint’s local and world matrices in traverses tree. |
| skin.h<br>skin.cpp | Skin files loader, stores all vertexes and triangles. Update and draw skin in each frame. |
| tester.h<br>tester.cpp | It contains main function, stores all skeleton, skin and animation data, and runs this project using openGL. |
| token.h<br>token.cpp | A helper class that stores files data, reads files data and output different kinds of value. |
| triangle.h<br>triangle.cpp | It is an object that store three vertexes’ pointer. Each triangle is used to render a real triangle in openGL. |
| vector3.h<br>vector3.cpp | The basic component of graphic system, uses to describes location, direction and etc. |
| vertex.h<br>vertex.cpp | An object stores its local and world location and normal. It is used to render model in openGL. |

There are some related files in the folder.

The .skel file is a indented hierarchy of joints. Each joint lists some relevant data about its configuration like location, rotation and etc.

This is what is looks like:

```
balljoint root {
    [data for root]
    balljoint head {
        [data for head]
        [children of head]
    }
    balljoint leg_l{
        [data for leg]
        [children of leg]
    }
    [more children of root]
}
```

The .skin file contains arrays of vertex data, an array of triangle data, and an array of binding matrices.

This is what is looks like:

```
positions [numverts] {
   [x]  [y]  [z]
   …
}
normals [numverts] {
   [x]  [y]  [z]
   ….
}

skinweights [numverts]
    [numattachments]  [joint0]  [weight0] … [jointN] [weightN]
    …
}

triangles [numtriangles] {
    [vertex0]  [vertex1]  [vertex2]
    …
}

bindings [numjoints]
    matrix {
        [ax]  [ay]  [az]
        [bx]  [by]  [bz]
        [cx]  [cy]  [cz]
        [dx]  [dy]  [dz]
    }
    …
}
```

The .morph file represents a modified version of some base skin, and only contains the data that is different from the base.

This is what is looks like:

```
positions [numverts] {
    [index] [x]  [y]  [z]
    …
}

normals [numverts] {
    [index] [x]  [y]  [z]
   ….
}
```

The .anim file contains an array of channels, each channel containing an array of keyframes.

The structure of the anim file is as follows:

```
animation {
  range [time_start] [time_end]
  numchannels [num]
  channel {
    extrapolate [extrap_in] [extrap_out]
    keys [numkeys] {
      [time] [value] [tangent_in] [tangent_out]
      …
    }
  }
  channel {
    …
  }
  …
}
```