Computer Graphics Lab #2
Splines, Animation and 3D

Prof. Ricardo Fabbri, Ph.D.

Polytechnic Institute (IPRJ) at the Rio de Janeiro State University
http://wiki.nosdigitais.teia.org.br/CG

November 8, 2013

Abstract

Now that you finished Lab#1 which consisted of rotating a camera around an arbitrary axis using the Blender Python API, its time to move on to further concepts and consolidate your learning. This should help you with the bigger goal of the final project of generating a complex animation.

Any material need for the lab can be found from the course website. Please render all your videos in full HD 1080p quality. **All extra work will be considered for bonus grade points.**

1 Construct the IPRJ Logo in 2D

Your goal is to generate the IPRJ logo of Figure 1 (or any official variant) in Blender [1], using two Python API's: Blender OpenGL `bgl` and standard Blender `bpy`.

Steps:

Figure 1: IPRJ logo, to be programmatically drawn with splines and animated, in both Blender and OpenGL.

1. Construct the IPRJ logo in 2D using the standard Blender API (`bpy`), without any data file. The standard blender API is used to help you construct models that can be further manipulated by other Blender scripts or GUI. Use spline curves and surfaces for the sine wave and curved letters, possibly for everything.

2. Construct the IPRJ logo in 2D using the Blender OpenGL API (`bgl`), without any data file. The OpenGL API is best for making blender tools that display interactive graphics *prior* to a full render. Use spline curves and surfaces for the sine wave and curved letters, possibly for everything.

2 Animate the IPRJ Logo

1. Create an animation of the IPRJ logo. Make the sine wave oscilate in 2D or 3D (off the plane). Do this with Blender OpenGL and standard Blender API

2. Make the ball bounce or roll in a physically pleasing way. You may choose to do it in `bgl` or `bpy` API (and similarly for the next items)

3. Make an intense zoom-in effect in some part and move the zoom around – you will realize an advantage of using splines instead of polylines: *scalability*!

4. Devise your own effects. Make the sine wave oscilate in crazy ways, *e.g.*, modulate it a real sound signal. Add a motion blurring effect, brushed metal textures, gradients, multiple illumination, futuristic glass texture, lens flare, make it look foggy or more mountain-like, make it look more 3D and animate more in 3D, make it look more abstract, add a 3D background ambiance, animate a 3D cartoon mascot with the logo and toon-shade it, or even write your own shader in GLSL or Cg language. Be creative!

5. **If done with stunning, professional high quality, or contains a great many different ways of doing things, or lasts more than 5min, the last item can also pass as a final project.** You may use the blender GUI for some or most of this item, you don't have to do everything programmatically.

References

[1] B. Foundation, "Blender: an open source 3D computer graphics modeling application, nonlinear video editor and game engine," www.blender.org.