N-body simulation rendering with Blender

Brian R. Kent, NRAO
http://www.cv.nrao.edu/~bkent/computing/

Described here are steps to load a "snapshot" file generated with GADGET-2 and render it with Blender. See *Springel, V. 2005, MNRAS, 364, 1105*

An example blend file created with these steps as well as a sample snapshot file can be found at: http://www.cv.nrao.edu/~bkent/computing/kentPASP.html

Generating a template object

1. Start Blender (Double click the icon or type ./blender from the command line).
2. Right click to select the default Cube object and press **'X'** to delete it.
3. SHIFT-A and select *Mesh->Plane*.
4. TAB key to switch the GUI to **"Edit Mode".**
5. Hold the Shift Key and Select three of the vertices on the plane. Press the **'X'** key to delete them. This vertex will be our template catalog object.
6. TAB key to switch the GUI back to **"Object Mode".**

Texturing the template object

1. Choose the Material Icon.
2. Click the **"+"** icon to start a new material.
3. Choose "Halo".
4. Change the size to "0.020".
5. Choose the Texture Icon.
6. Change the Type to **"Blend"**.
7. At the bottom of the dialog choose a color (Blue or light yellow works well in this case).

Loading a single snapshot

1. Open the Text editor to start/edit a Python script (one is provided in the blend file).
2. "***import bpy***" always needs to be at the top of the Python script.
3. There are multiple ways to read in ASCII files - we use the csv.DictReader method in this example.
4. Duplicate the template object (selected by default) with the method:
   `bpy.ops.object.duplicate()`
5. Move the object to the appropriate position with the method:

```
bpy.context.active_object.location.xyz=(float(row['xdisk'])/10.0,
float(row['ydisk'])/10.0, float(row['zdisk'])/10.0)
```

   where row is a Python dictionary and xdisk, ydisk, and zdisk are dictionary keys. Each

argument passed in the Python tuple needs to be a float data type. We divide each element by ten simply for scaling convenience.
6. Each snapshot can be read into the 3D viewspace in this manner.

Keyframing the camera

1. The Camera can be selected in the Outliner (upper right section of the GUI).
2. Keyframe the camera location with the **"I"** key and choose **"***LotRotScale***"**.
3. Reposition the Camera to a new location with the **"G"** and **"R"** keys for "*Grab to Move*" and "*Rotate*".
4. Keyframe the new camera location with the **"I"** key and choose "*LotRotScale*".

Animating and Rendering the Sequence

1. To see a preview of the animation, Choose *View->Camera* (last option) and then click the **Play** button at the bottom of the GUI. Press the square Stop button to halt the animation.
2. Choose the Render tab.
3. At the bottom of the Render dialog, Change the output to "*AVI JPEG*". The "**Stamp**" option is useful as it prints metadata about the animation over the video. Scroll back to the top of the dialog and Click Animate. A video file will be generated.

The final Python script looks as follows:

```
# -*- coding: iso-8859-1 -*-
#N-body simulation rendering with Blender
#Created with Blender
#http://www.blender.org
#Brian R. Kent, NRAO
#http://www.cv.nrao.edu/~bkent/computing/kentPASP.html

#Example 1

'''
Please note that this Python script should be run inside
the Blender environment.
'''

import math
import bpy
import pickle
import glob     #glob can be used to read a directory of snapshot
files
```

```
import csv

# MODIFY THIS TO THE DIRECTORY ON YOUR COMPUTER!
filepath = '/export/data_2/blender/threebody/text/0000b.txt'


'''
#This section can be used to remove all objects if so desired
for object in bpy.data.scenes['Scene'].objects:
     if 'point.' in object.name:
     #print(object.name)
     #bpy.context.scene.objects.active =
bpy.data.objects[object.name]
     bpy.data.objects[object.name].select = True
     bpy.ops.object.delete()

#bpy.data.scenes['Scene'].objects.unlink(bpy.context.active_object)
'''




######################################################################

#Set original file and positions for loading a single snapshot
bpy.ops.anim.change_frame(frame=1)
fields = ['xdisk', 'ydisk', 'zdisk']
reader = csv.DictReader(open(filepath), fields, delimiter=',')


for row in reader:
     bpy.ops.object.duplicate()

bpy.context.active_object.location.xyz=(float(row['xdisk'])/10.0,
float(row['ydisk'])/10.0, float(row['zdisk'])/10.0)

######################################################################
######################
```