Reagan's Renderers
Astrophysical Phenomena Renderer
System Unit Testing
Revision 2
This Document Covers all the Testing done by every team member for CMPS115.

Sprint 1 Tests: N/A

Sprint 2 Tests: N/A

Sprint 3 Tests: All user stories other than documentation were not testable until sprint 3.

User Story: As an astrophysicist I would like to be able to view volumes generated in yt so that I can interact with my data.

1. Volume Render Test -- In the python interpreter input the following code:

```
import h5py

from yt.config import ytcfg
#data_dir = ytcfg.get('yt','test_data_dir')
f = h5py.File("/home/bogert/test.hdf", "r") # Read-only access to the file

print f.keys()
from yt.mods import *
import numpy as np

from yt.utilities.physical_constants import cm_per_kpc, cm_per_mpc

import cuda_module as cm
data = {k:v for k,v in f.items()}

lcut = 0.2
dcut = 10

oo = np.zeros(data["Density"].size, dtype="uint16")
oovse1 = np.zeros(data["Density"].size, dtype="uint16")
oovse2 = np.zeros(data["Density"].size, dtype="uint16")
oovse3 = np.zeros(data["Density"].size, dtype="uint16")

#Commented out functions are used to reshape data to visualize in yt's volume renderer
#oo = np.reshape(oo, (256,256,256))
#oovse1 = np.reshape(oovse1, (256,256,256))
#oovse2 = np.reshape(oovse2, (256,256,256))
```

```
#oovse3 = np.reshape(oovse3, (256,256,256))

#data["Density"] = np.reshape(data["Density"], (256,256,256))
#data["Te1"] = np.reshape(data["Te1"], (256,256,256))
#data["Te2"] = np.reshape(data["Te2"], (256,256,256))
#data["Te3"] = np.reshape(data["Te3"], (256,256,256))
#data["Vse1"] = np.reshape(data["Vse1"], (256,256,256))
#data["Vse2"] = np.reshape(data["Vse2"], (256,256,256))
#data["Vse3"] = np.reshape(data["Vse3"], (256,256,256))

oovse1[data["Vse1"] > lcut] = 1
oovse2[data["Vse2"] > lcut] = 1
oovse3[data["Vse3"] > lcut] = 1

#for k in oovse1
#       if k != 0
#       print k
        #oovse1[data["Density"] < dcut] = 0
#oovse2[data["Density"] < dcut] = 0
#oovse3[data["Density"] < dcut] = 0


#oo[data["Density"] < 10] = 0
#oo[(data["Te1"] > -10000.0)] = 0
oo = np.copy(data["Density"])
#oo *= 1000
data["Density"] = np.copy(oo)
oo = data["Density"].astype(dtype="uint16")
ii = (oovse1 + oovse2 + oovse3 == 1)
#oo[ii] = 40000
cm.Render(oo)
#for k in data["Density"] :

  # print data["Density"][k]
#bbox = np.array([[-0.5, 0.5], [-0.5, 0.5], [-0.5, 0.5]])
#pf = load_uniform_grid(data, data["Density"].shape, 250.0*cm_per_mpc, bbox=bbox,
nprocs=8, periodicity=(False,False,False))

import time

while True:
        for i in range (0,10000):
        cm.SetTransferScale(1000 - i%1000)
```

```
cm.SetBrightness((i%1000)/300.0)
cm.SetRotation(i/10.0, i/8.0, i/20.0)
time.sleep(0.01)
```

With this code entered you should get a volume render.

2. Transfer Scale Test -- To change the sampling range of colors from the transfer function type the following in the python interpreter.

```
cm.SetTransferScale(x)
```

Equivalence Class 1: If x is 0 the render will appear in only one color
Equivalence Class 2: As x goes from 1 to infinity, the sample range of colors from the transfer function will increase. At some point the bounds of the transfer function is reached and you will not get any more colors.

3. Brightness Test -- To change the brightness of the render in the python interpreter type.

```
cm.Brightness(x)
```

Equivalence Class 1: If x is 0 or negative the brightness of the rendered image will become very dim.
Equivalence Class 2: As x goes positive and increases the brightness of the image will become more intense.

4. Rotation Test -- To change the rotation speeds of the rendered image in the python interpreter type the following.

```
cm.SetRotation(x,y,z)
```

Equivalence Class 1: If x,y,and z are positive the rotation speed on the x,y and z axes will increase
Equivalence Class 2: If x and y are positive and z is negative, the rotation speed on the x and y axes will increase while the rotation speed of the z axis will decrease
Equivalence Class 3: If x is positive, y and z are negative, the rotation speed on the x axis will increase, while the rotation speed of the z axis will decrease
Equivalence Class 4: If x,y, and z are negative, the rotation speed of all axes will decrease
Equivalence Class 5: If x is negative, y and z are positive, the rotation speed of the x axis will decrease while y and z axis rotation speed will increase
Equivalence Class 6: If x, y are negative and z is positive, the rotation speed of the x and y axes will decrease while the rotation speed of the z axis will increase.