Report:

Giray Eryılmaz

I have implemented a simple raytracer with C++. It uses classic ray tarcing method.
It sends a ray (actually 100 rays) from each pixel of the frame from the eye point and determines if that ray intersects with any spheres. If it does so than makes tha pixel the same color of the closest of intersecting spheres. Note that there maybe more than one spheres intersecting with one ray but we accept the closest one to get color from.

Also makes shading. This too works with rays. From the intersaction point a ray is shut to the light source and if any intersaction occur then that point is shaded. Self shading is also done.

The program is coded on Ubuntu and tested on Ubuntu. Compled using g++.

Reads the input from <input file> and prints an output to <input file>_output.ppm .
A bash script is provided so that it compiles and runs the program .

name of the script is 'compile_run'
Note dont forget to give the file executaale rights first , i sometimes do :

```
$ chmod +x compile_run
```

Program is usage example :

```
$ ./compile_run input_1.txt
```

the output is an image file named ' input_1_output.ppm '

if no input file is specified the default input file name is "input.txt"

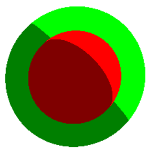
Also the executable is give too. It's named rtp. Can run it as

```
$ ./rtp <input file>
```

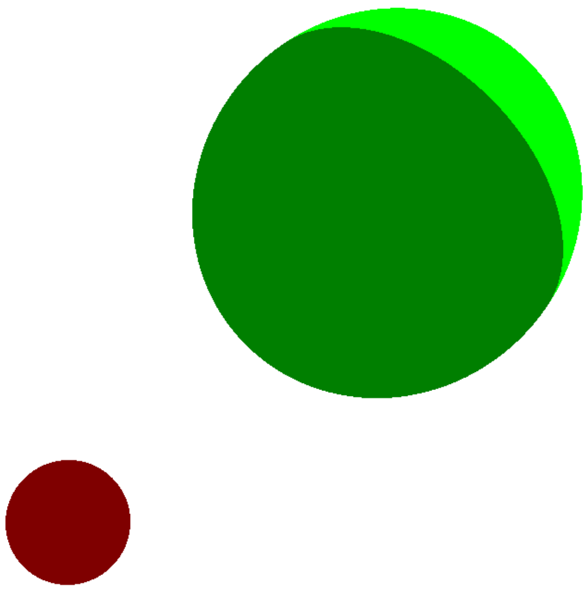
Again this is not an .exe file. It is for ubuntu -
linux

Some examples:

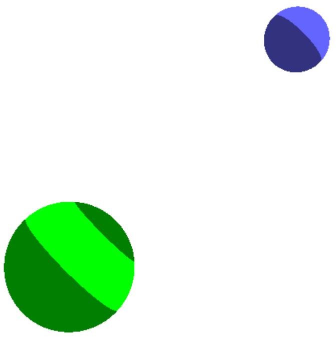
number of spheres 2
color of sphere 1: (R,G,B)=(255,0,0);
color of sphere 2=(0,255,0);
Position (x,y,z) of sphere 1: (50,50,300)
Radius of sphere 1 = 20
Position (x,y,z) of sphere 2: (100,100,600)
Radius of sphere 2 = 60



number of spheres 2
color of sphere 1: (R,G,B)=(255,0,0);
color of sphere 2=(0,255,0);
Position (x,y,z) of sphere 1: (-30,-30,300)
Radius of sphere 1 = 20
Position (x,y,z) of sphere 2: (70,70,300)
Radius of sphere 2 = 60



number of spheres 2
color of sphere 1: (R,G,B)=(100,100,255);
color of sphere 2=(0,255,0);
Position (x,y,z) of sphere 1: (90,90,500)
Radius of sphere 1 = 20
Position (x,y,z) of sphere 2: (-50,-50,500)
Radius of sphere 2 = 40



Note: i resized the images so that they would fit in the page