

USER'S MANUAL

Program Description

The program performs 2 operations: Ray Tracing and Ray Casting.

In Ray Tracing, the program emits rays from each pixel to the scene. Each ray then intersects with an object in the scene and we check if a light ray cast from the light source can reach that intersection point without being obstructed by any other object in the scene.

In Ray Casting, emit rays from the light source through the window into the scene. We then use a method called ray marching where we take points non this emitted ray at equal intervals if the ray. We then check if the point is within the view frame. If it is , them we map it to a pixel point and scale the color of the pixel with the energy factor of that ray point. We then repeat this step for every point on every ray emitted from the window to the floor.

We then further build on the Ray marching method by implementing a randomizer function to randomly select points on the ray instead of uniformly selecting the points. This will allows us to get a more realistic image of the medium of the light ray.

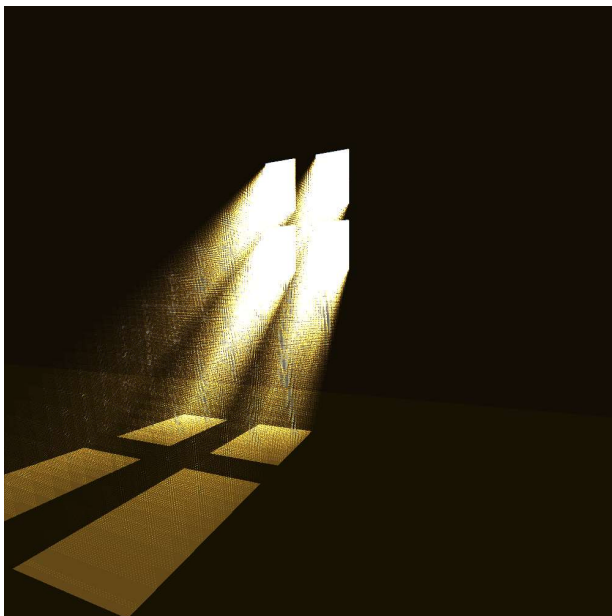
Input

All the inputs are hardcoded into the code, but there are 5 variables that can be changed to control the god rays:

1. Step: this variable stores the number of points taken per ray. This variable can be edited in the list of global variables in **Tracer2.java** file
2. Delta: this variable holds the minimum distance between each point on the ray, this value gets increased whenever we move on to the next point in the ray. This variable can be edited in the list of global variables in **Tracer2.java** file
3. Enhance: this variable stores the factor value that is used to scale the pixel color. The value of this variable decreases at each step. (lower limit = 1.0). This variable can be edited in the list of global variables in **Tracer2.java** file

4. Method: this variable holds either a 0 or 1 as a value. It controls the ray marching method that the program will implement. 0 => equidistant Ray Marching and 1 => randomized Ray Marching. This variable can be edited in the list of global variables in **Tracer2.java** file
5. Amb: This variable essentially controls the ambient color of the room. This variable can be edited in the list of global variables in **Scene.java** file

Normal Output



Ray marching without randomness



Ray marching with randomness

The output will be a PNG file named **IMAGE.png** in the src folder

Exception Reports

The most common exception you are likely to encounter is:

“ Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 1000 out of bounds for length 1000 ”

The reason for this is either:

1. The Step value is too high
2. The delta value is too small

Program Limitations

When using the randomizer variation of the Ray Marching method, do not go over Steps = 999 as it will cause an out of bounds error.

This bound does not apply to the equidistant variation of the Ray Marching method. Although this method can handle more Step sizes, the computational power increases drastically with each increase in step size, followed by a decrease in delta. It is advised to not exceed the value of Steps by 1000000 and to not reduce the value of delta any lower than 0.000001.

The enhance variable must not go lower than 1.0, because scaling the pixel color by a value lower than 1.0 would reduce the brightness of the color which is an unrealistic property of a light ray.

Command Sequence

1. Open the directory containing the Master.java file in terminal. (It will be in the src folder)
2. Then enter the following lines to compile the code: `>> "javac Master.java"`
3. Once it has compiled, enter the following lines to run the code: `>> "java Master"`

4. The code will then run and display " IMAGE COMPLETED :) " when finished
5. It will generate an IMAGE.png file in the src directory which will contain the rendered image

Author Information

Name: Atharva Sawe

Email: as6791@g.rit.edu