Computational tracing of light rays in cavities.

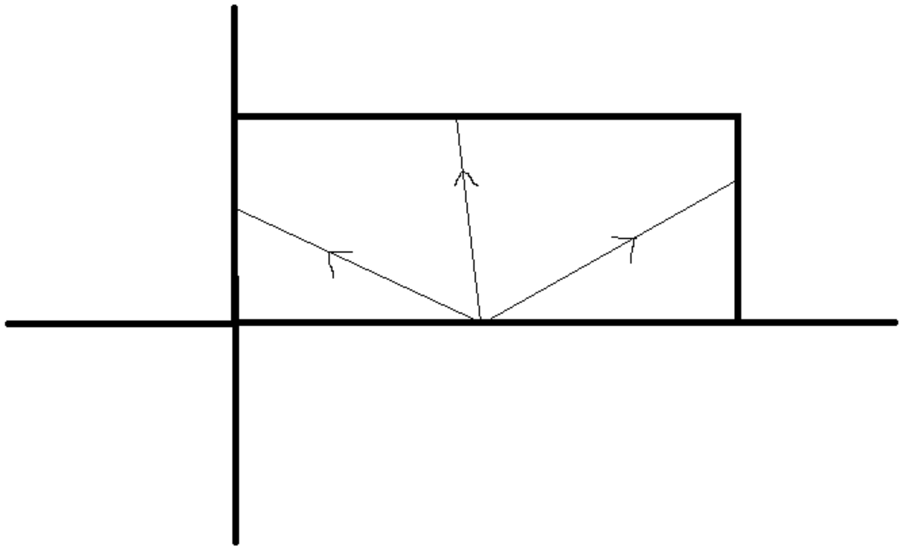
This project is basically is very much useful for re-searchers who re-search on the tracing of light path and awesome cool stuff.

The basic approach of the ellipse part is we found out some awesome shortcut formulae in our previous JEE books and implemented them, at least they came up to some use now!. Though it may be silly but it helped in reduction in computational speed by a good margin. We did try to derive the formula but failed quote lot of times but we are trying for level best to derive the formula during the presentation

We basically created a two .py files and imported them into a new file where it is user friendly to use the code or else it would be a big mess for a dumb user to access it.

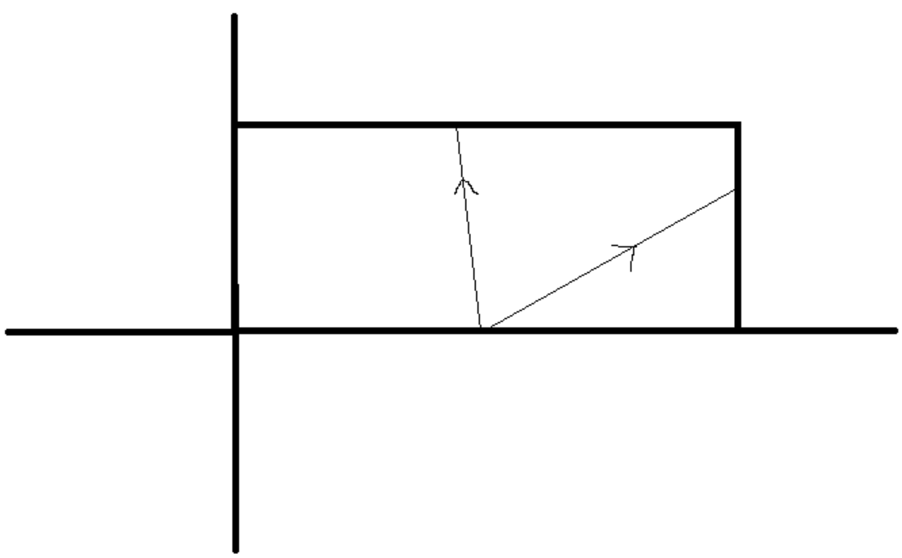
We planned user interface in the console where you need to select the desired choice to move forward. And it will ask you a bunch a lot of things which you need to give to it so that it computes and prints out the desired result and it even shows the points were reflections actually happened.

Rectangular cavity: The light ray can enter the cavity from any of the four sides. Four if conditions have been used-one for each side to know from which side the light ray enters.After entering the cavity, the light ray has three sides onto which it can hit.

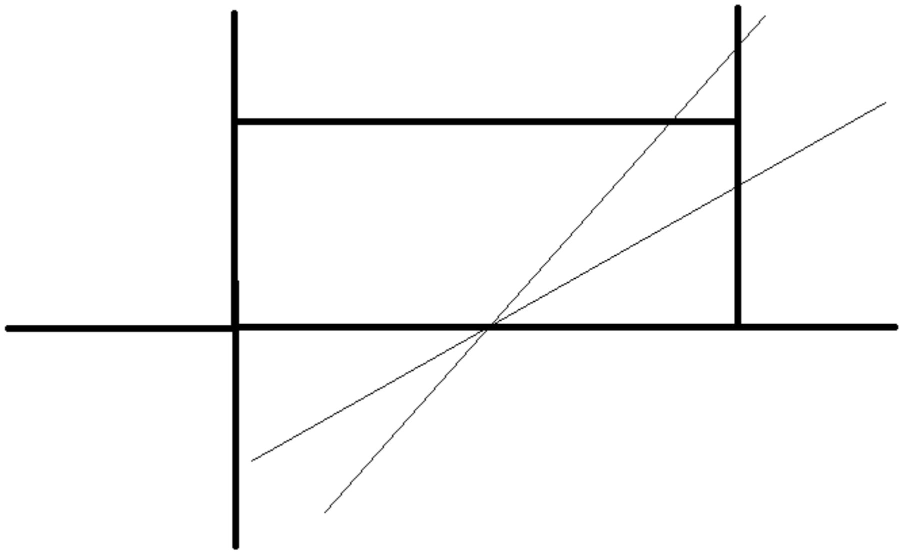


We introduced another factor slope which eliminates one side onto which the light ray can hit.

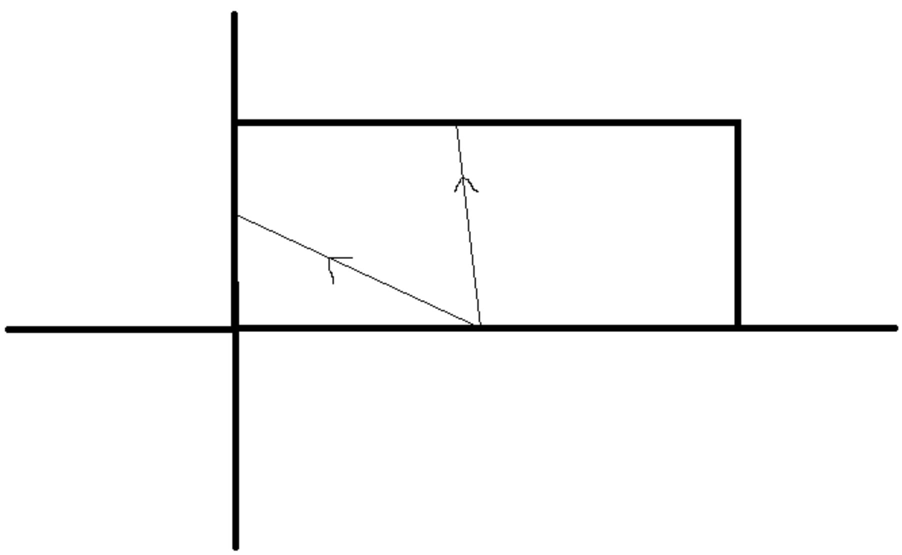
Case 1: Slope>0



Now, we know one point and slope of the line. From this we can find out the line equation i.e., y=mx+{p[1]-m*p[0]}. The y=co ordinate can be obtained by substituting x=length in the above line equation.We know, y-co ordinate is the perpendicular distance from the x-axis.Therefore, if the y-co ordinate is lesser than the breadth, the light ray will not reach the top side. This leaves only one side for the light ray to hit.



Case 2: Slope<0



We can use the same conditions as above for this case as well but since one of the sides of the reactangle is the y-axis itself, it is easier to use the incercept condition.(when x=0 is substituted in the line equation, we get the y-intercept i.e. c=p[1]-m*p[0]. If the intercept is lesser than the breadth, the light ray will not reach the top side. This leaves only one side for the light ray to hit.

Elliptical cavity:

There not much calculations included we used shortcut formulae as mentioned above to solve this task.

We found this project interesting cause it requires both math and science and our request is to make only one improvement delete all this stuff and keep only cube for next year that would be great to work with.

Modules used:

Pylab

Math

Sources used:

www.wolframalpha.com

http://www.purplemath.com/modules/ellipse.htm

http://matplotlib.org/examples/shapes_and_collections/artist_reference.html

My old math notes for shortcut formulae!