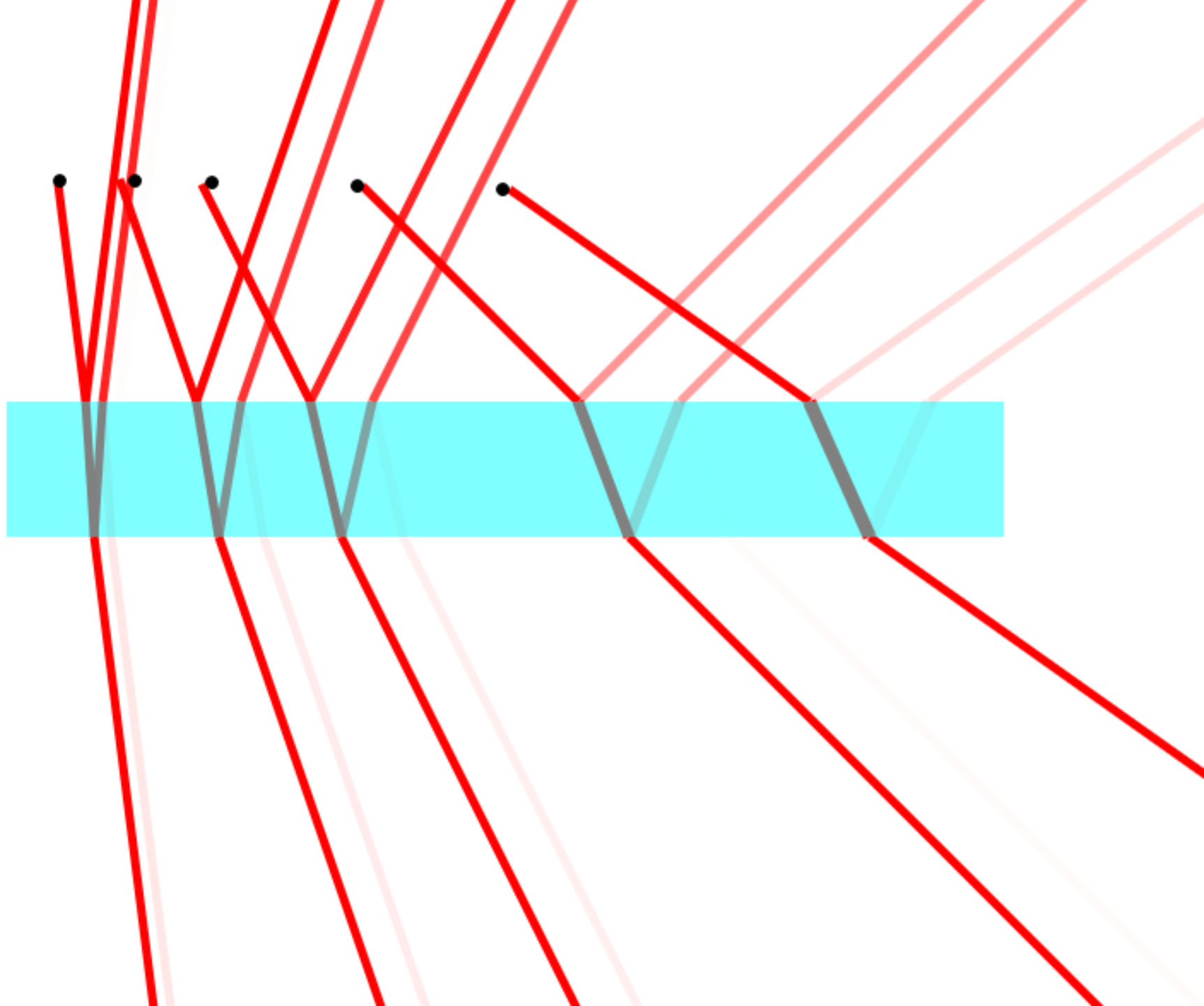


# light2D.2.0

加入光强

魏雨轩

17307110177



# Light和childlight对象的修改

## Light 属性

- 属性color修改为rgba=[r,g,b,a]，其中rgb为颜色值，而a为透明度用来形容光强。这里的a是指主光的初始光强。
- 加入了pole属性作为极化方向，只考虑横电和横磁两种模式的偏振。分别记做0,1

## Childlight 属性

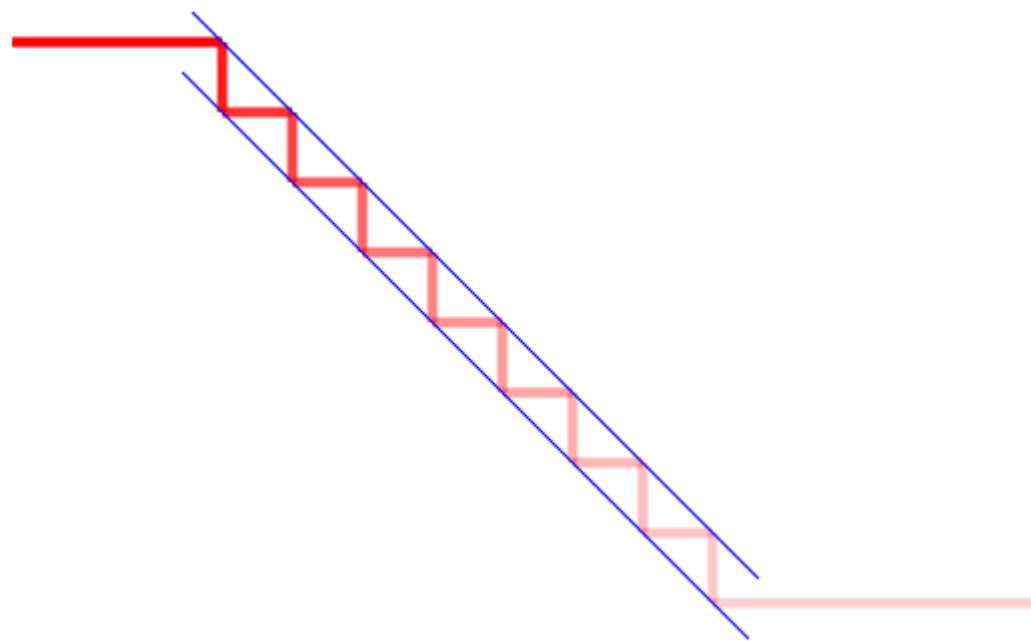
- Color属性修改成rgb=[r,g,b]单独储存，而加入intensityset属性用来储存path路径中对应的不同光强。加入endintensity用来储存光线末端的光强
- 由于考虑功率时要考虑光线在折射前后宽度的变化，所以加入widthset属性储存path路径中对应的不同宽度的光线。Endwidth作为光线末端的宽度。

# Light和childlight对象的修改

- 加入pole属性，表示偏振方向。
- Draw()方法，由于canvas不能绘制多色线条，所以只能一段一段绘制，根据widthset, intensityset, 来绘制光纤，但是在宽度比较大的时候，两段光线的连接处不契合。
- 更新reflect(normal,loc,R)方法，考虑反射率R，然后调节widthset, intensityset。同样更新refract(normal,n2,loc)方法，在反射后调节widthset, intensityset，其中通过我们熟知的菲尼尔公式计算光功率的变化，由于反射率和折射率是同时计算的，所以在折射光的同时，也要返回反射率和宽度变化关系，在之后的light\_on()中绘制多光线时，用到。

# Mirros对象修改

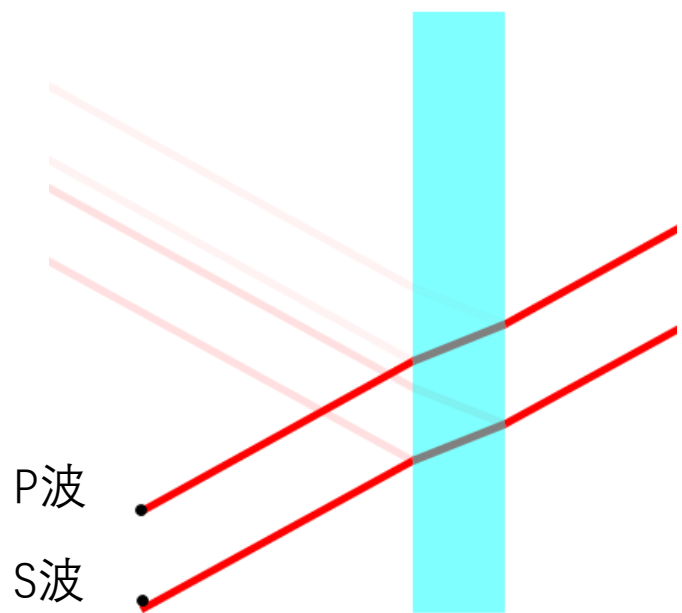
- 加入反射率属性，这个是一个绝对值，而不是和lens的反射是一个计算出来的相对值。（如右图：两个反射率为0.9的平面镜）
- Show\_info()函数修改，体现折射率信息。（前面的light对象show\_info()也要修改，略）



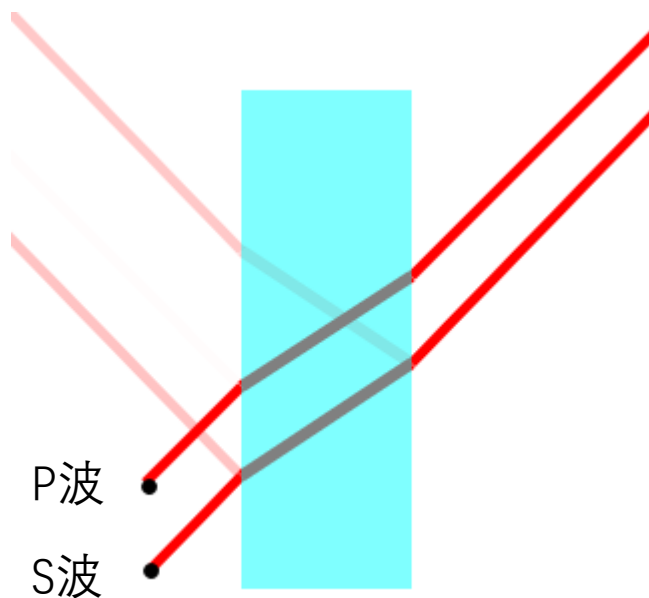
# Light\_on()修改

- 在next\_action = “mirro”的reflect过程引入mirro的反射率
- 在next\_action = “lens”的refract过程不变，因为折射率是自动计算的，但是反射率是同时生成的，所以，必须在light.refract()调用的时候返回反射率，而且光线宽度也是同时生成的，所以要引入alphaset,widthset两个列表来记录一次light\_on()循环中，要产生的新的子光对象的光强和宽度，并且在refract\_only为false时，把这些子光对象添加到light的childlight列表中。

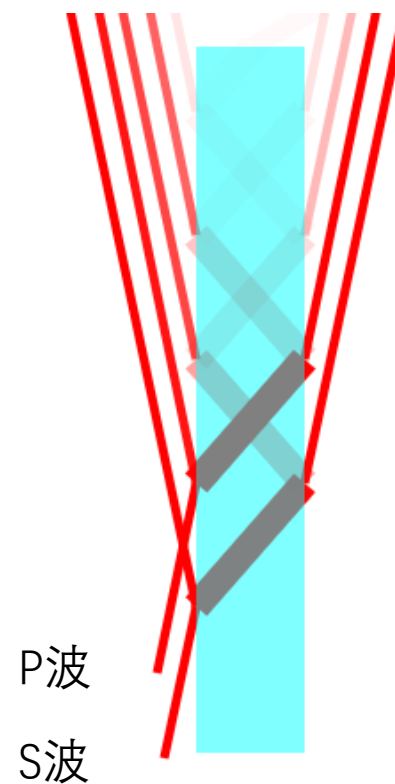
# 一些光路实例——布儒斯特角



小角度入射

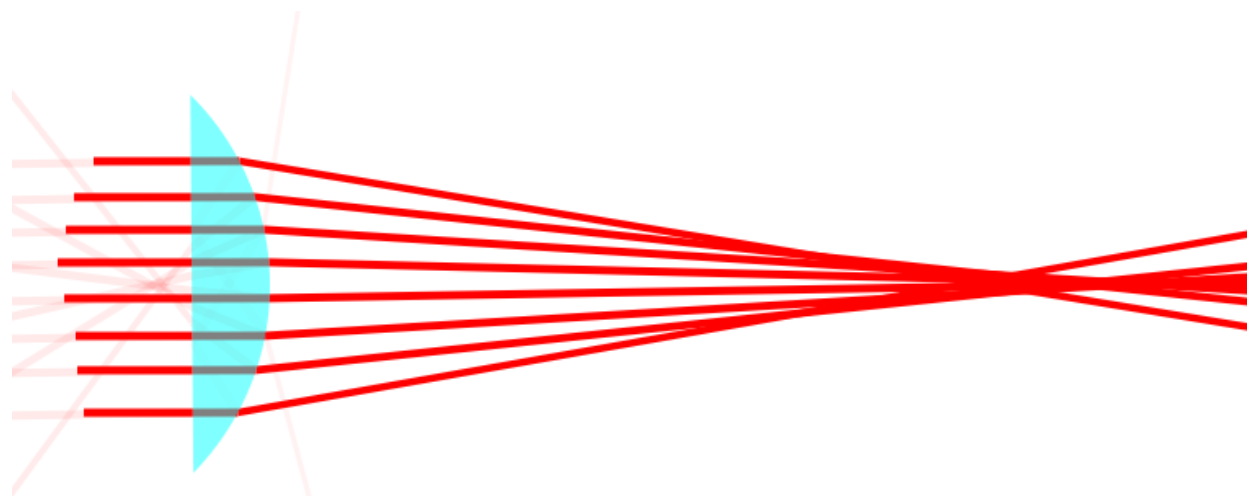


布儒斯特角

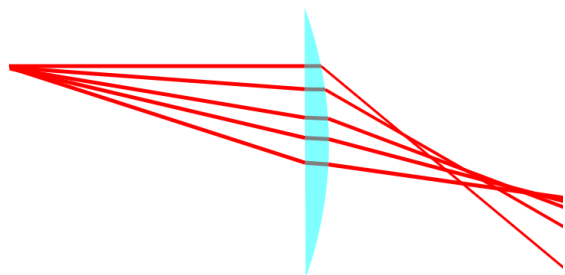


大角度掠射

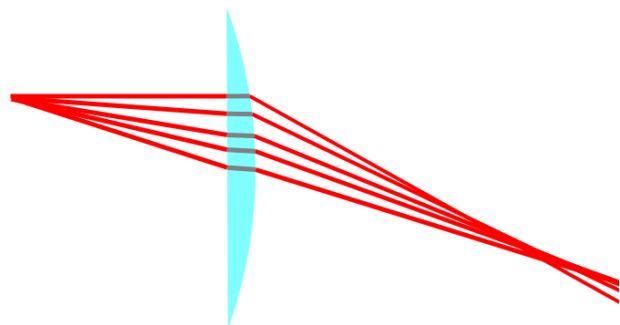
# 一些光路实例——透镜成像



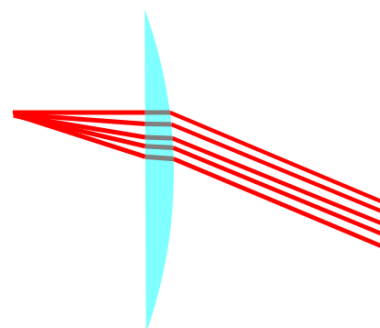
球面透镜的成像畸变



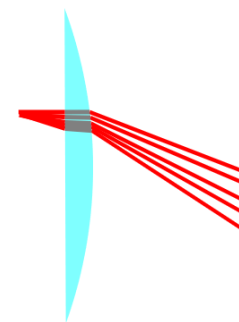
$$u_w > 2f$$



$$2f > u_w > f$$



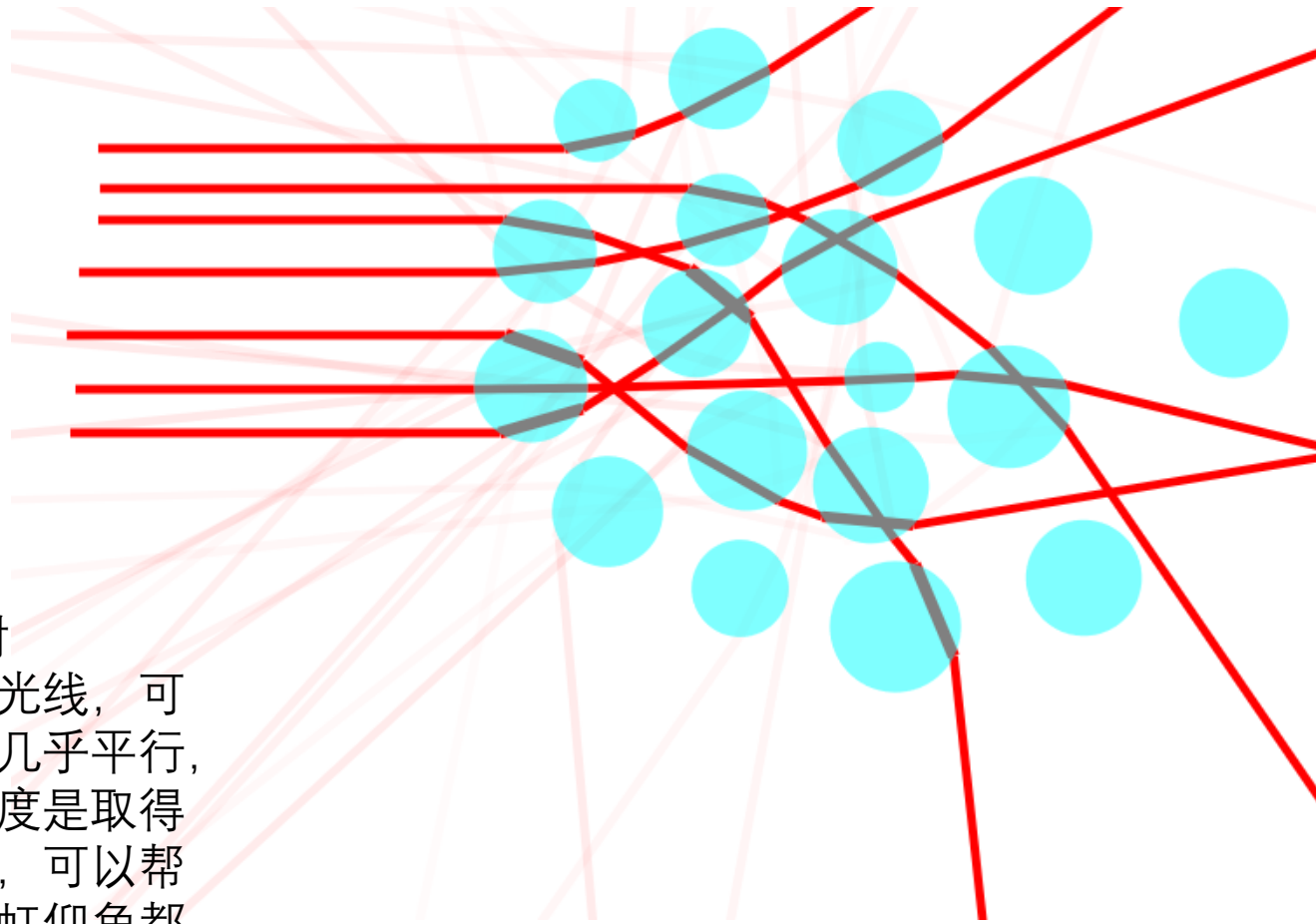
$$u_w = f$$



$$u_w < f$$

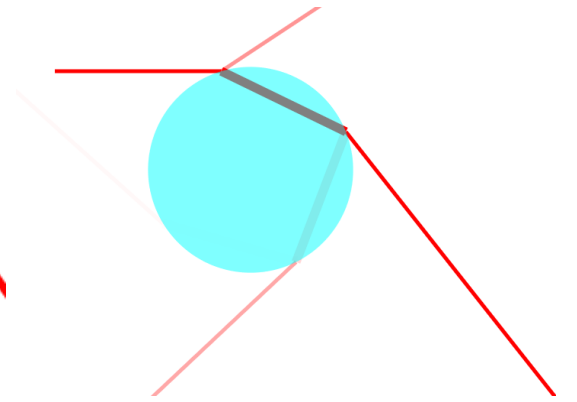
# 一些光路实例——彩虹的形成

直接反射光线比较杂乱



主光的散射，两次折射，  
这个散射有些夸张化了，  
水滴不可能这么密集，  
所以主光束大体的方向  
是不变的。

主光的散射，两次折射



折射+反射+折射  
也就是形成彩虹光线，可  
以看到这些光线几乎平行，  
因为在们某个角度是取得  
光强的极大值的，可以帮  
助理解为什么彩虹仰角都  
是近似42度。



# 可以改进的地方

- 光线在很粗的时候由于分段绘制，canvas不能绘制多色线条使得图形很丑。改进手段：必须把绘图程序做非常大的改动，比如专门写一个绘制光线的程序，用三角形和矩形拼合出光线。
- 有必要加入集合体对象，就是可以把某些对象组合在一起，可以同时拖动，旋转。而这个集合体对象可以方便我们引入各种各样的光源类型，数据结构变为，一个系统有多个光源，一个光源有多条主光线，每条主光线又衍生多个子光线。这个光源可以是点光源，可以是平行光源，这样就可以方便处理一些成像问题，也方便我们处理多光线的排布问题。生成集合体的操作是通过绘制选择框，或者control选择来设置。这个又是一个大改动。
- 考虑是否要加入理想凸透镜，理想凹透镜对象，有时候我们不需要看相散等畸变。考虑把mirror对象改成像容许曲面镜存在的镜子对象，mirror对象设定过于简单。

# 可以改进的地方

- 无法实现波动光学。引入波动光学虽然只是要计算光程差，但是这个带来的直接难度是绘图的困难，如何绘制两束光的干涉图样。这个时候就必须在绘图的时候要计算每一个点的干涉后的光强，这个是很困难的，canvas绘图首先就必须需要一个像素点一个像素点的绘制。所以从程序实现上，必须把光的路径储存的一个一维的列表改写成一个二维的形式才能更方便的编写，从而导致要引入波动光学必须从根本上改变程序的内容。几何光学可用的光学元件很少，没有波动光学丰富的内容。
- 绘图窗口大小的限制，可以采用比例尺的做法，实现整体的放大和缩小，而可以用canvas平移的做法，实现整体的移动，这样可以实现绘制窗口的无限大。
- 光路算法上的优化，在计算“彩虹”光路的时候，可以感受到卡顿，因为边界的点越多（比如圆形），计算量越大。可以再checkcross函数上做优化，比如使用一个矩形框标定元件的大概位置和大概大小，先粗略的判断时候有交点，然后再细化计算，可以节省很多运算量。

# 可以改进的地方

- 旋转和平移操作不能精确的控制，可以加入输入框，精确控制旋转的角度和平移的长度。还可以学习solidworks的配合操作，对垂直关系，平行关系，重合关系，对齐关系等对精确构建光学系统提供帮助。
- 加入measure工具箱，可以再canvas中起到一定的测量作用（和solidwork功能一样），比如实现焦距测量，物距测量。
- 辅助线的添加，除了绘图的辅助线，有时候光轴，反向延长线，发现等等都需要标定，对二维系统的构建和虚像的构建有帮助。
- .....