Problem 1

Set the color matrix(pixels) to constants to create stripes of red, green and blue

Problem 2

Created a sphere at the origin and moved the light around the sphere to test for hits. To do this I need to create a sphere class, Vector class, ray class, light class

Then created a plane for the sphere to be on and tested which object was closer to the camera and therefore be viewed by the camera. To do this I created a plane class and then a class in main to check if the sphere is closer or the plane

Problem 3

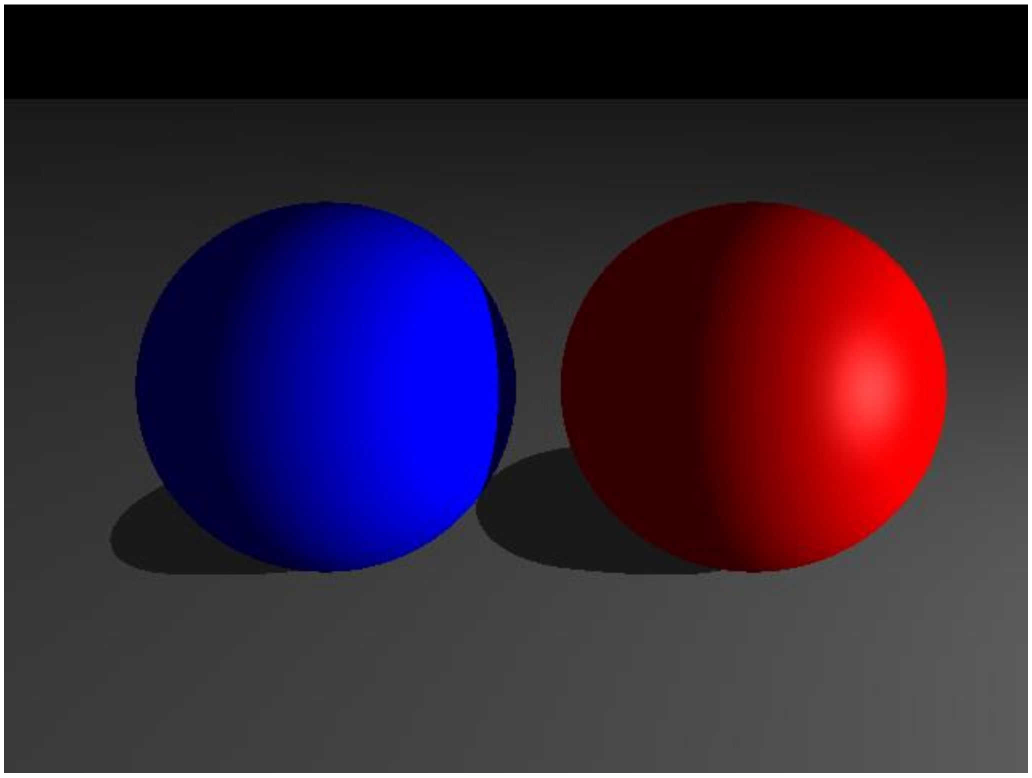
Ambient light is a constant that I choose to scale the color

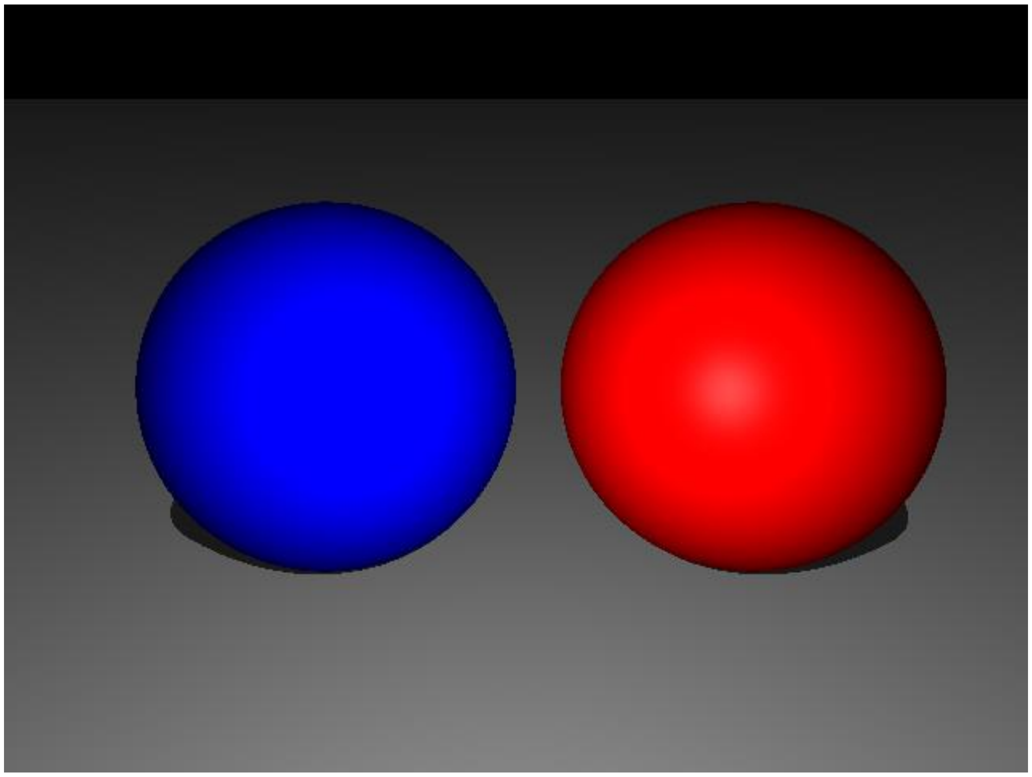
Diffuse shading I used the light ray as well as a shadow ray to determine the brightness of the pixel

For specular shading I used the normal to calculate which rays were reflected to the camera by giving the balls a “shiny” surface

Problem 4

Made some tries at this but no tetrahedron was formed





Run and Complie

Compile using make then main

Used the compile instructions from the CImg documentation

- Main.cpp
  - Void savebmp
    - Used to assist in saving a .bmp file
  - Int closestObjectIndex
    - Used to find which object are closer to the camera in case of overlap
  - Color getColorAt
    - Used to get the color for each pixel
    - Covers all forms of shading as well
  - Main
    - Brings it all together
    - Set colors

        - Set cameras
        - Place objects and lights in the scene

  Add pixels to the bmp file Int closestObjectIndex

        -

- Vector.h
    - Vector
        - Constructors
    - Double getVectorX
        - Returns x value
    - Double getVectorY
        - Return y value;
    - Double getVectorZ
        - Return z value;
    - Double magnitude
        - Returns vector magnitude
    - Vector normalize
        - Returns the normal vector
    - Vector invert
        - Reverses vector
    - Double dotProduct
        - Returns vector dot product
    - Vector crossProduct
        - Returns vector cross product
    - Vector vectorAddition
        - Returns two vectors added
    - Vector vectorScalar
        - Returns vector times a scalar
- Souce.h(used so light can go in vectors)
    - Vector getLightPosition
        - Return light positions
    - Color getLightColor
        - Return light color
- Plane.h
    - Plane
        - Constructors
    - Vector getPlaneNormal
        - Returns vector normal to the plain
    - Double getPlaneDistance
        - Distance of plane to source
    - Color getColor
        - Return plane color
    - Vector getNormal
        - Returns normal to plane

  - Double intersection
    - Returns value to indicate intersection happens
- Object.h(used so plane, sphere can go into vectors)
  - Color getColor
    - Return color
  - Vector getNormal
    - Return zero vector
  - Double intersection
    - Return intersection value
- Ray.h
  - Ray
    - Constructors
  - Vector getRayOrigin
    - Return origin of ray
  - Vector getRayDirection
    - Ray direction
- Camera.h
  - Camera
    - Constructor
  - Vector getcameraposition
    - Returns camera position
  - Vector getCameraDirection
    - Return camera facing
  - Vector getCamraRight
    - Return camera rotation
  - Vector getCAmeraDown
    - Return camera level
- Color.h
  - Color
    - Constructors
  - getColorRed
    - returns red
  - getColorGreen
    - returns green
  - getColorBlue
    - returns blue
  - getColorSepcial
    - return spectral light
  - double setred
    - set red ratio
  - double setGreen
    - set green ratio
  - double setBlue
    - set blue ratio

    - double setSpecial
        - set spectral ratio
    - double brightness
        - color brightness
    - color colorScalar
        - multiply color by a scalar
    - Color colorAddition
        - Add two colors
    - Color colorMultiply
        - Color times a color
    - Color colorMean
        - Mean of two colors
    - Color cutoff
        - If RGB goes over 1 or below 0 corrects it
- Light.h
    - Light
        - Constructors
    - Vector getlightPosition
        - Return light position
    - Color getLightColor
        - Get the light color
- Sphere.h
    - Sphere
        - Constructors
    - Double getSphereRadius
        - Get the radius
    - Double getColor
        - Get color of sphere
    - Vector getNormal
        - Get normal to the sphere
    - Double intersection
        - Returns positive value if ray intersects with the sphere