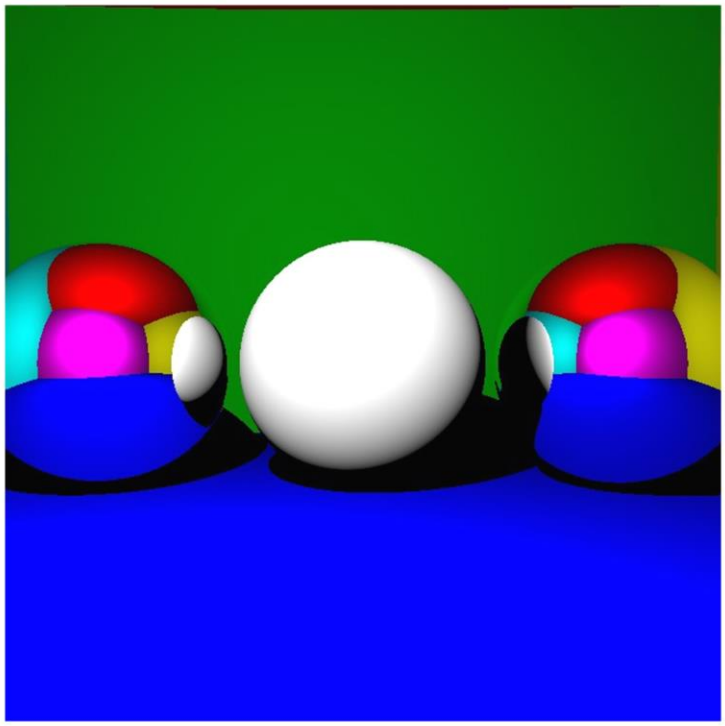
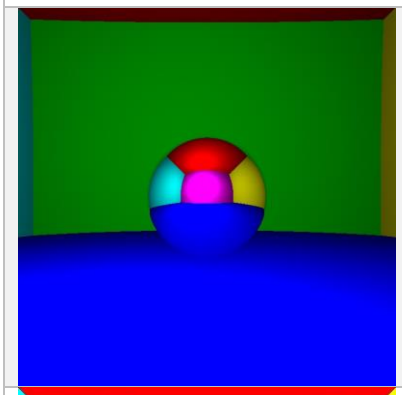
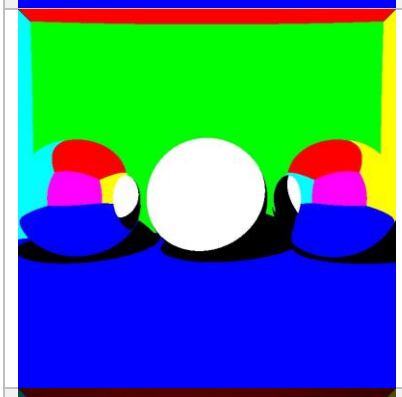
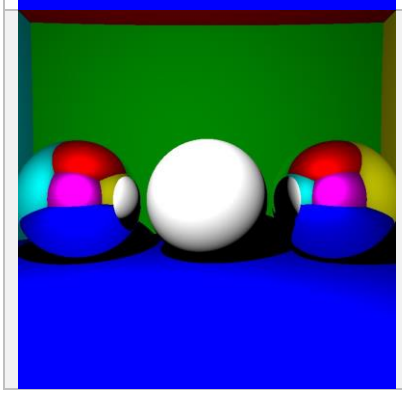


Ray-tracer

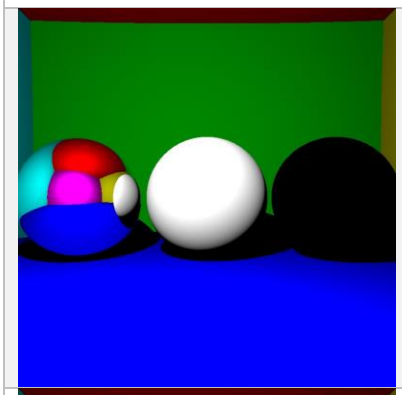
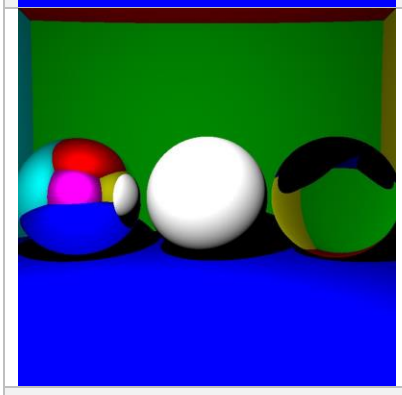
CSE306

Disclaimer: Basis of code up to and including reflection were copied from Guillaume Loranchet. This was in order to aid me since I did not have a strong knowledge of C++ and was struggling to begin. All work from refraction onwards is my own. All of the code is well annotated to show my understanding.

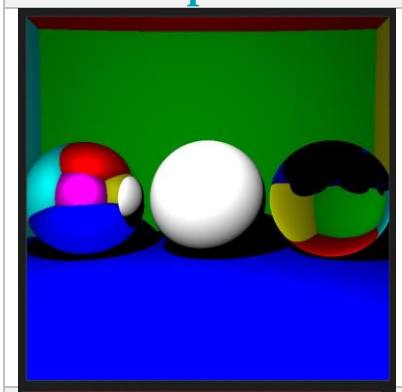
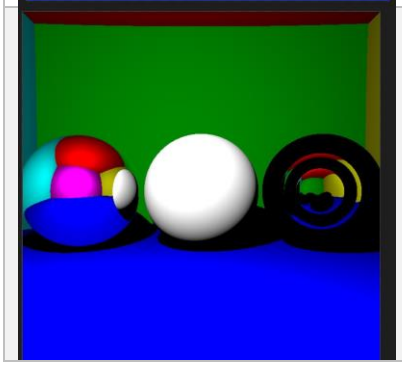
Mirror – reflection

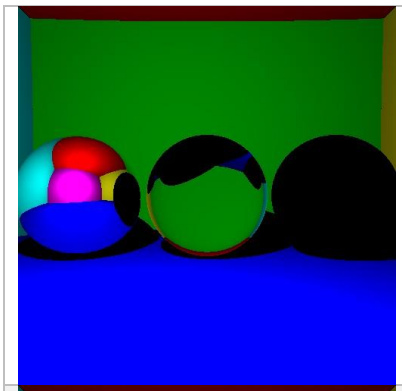
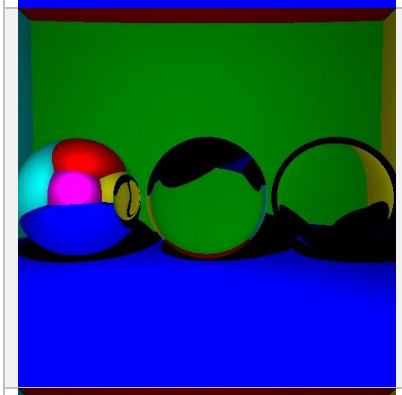
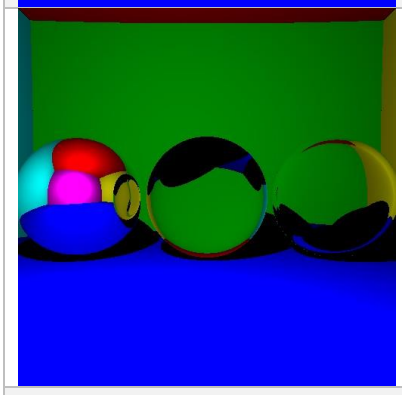
	Image created by Guillaume Loranchet
	Image created by Guillaume Loranchet Light Intensity 10^5
	Light intensity seemed far too strong so lowered it. Light intensity 2.10^{10}

Transparency – refraction

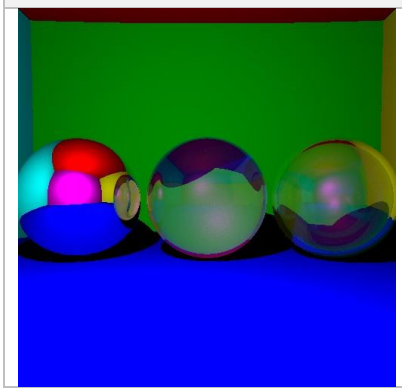
	Transparency without code to impede rays from entering the sphere
	Transparency

Hollow Spheres

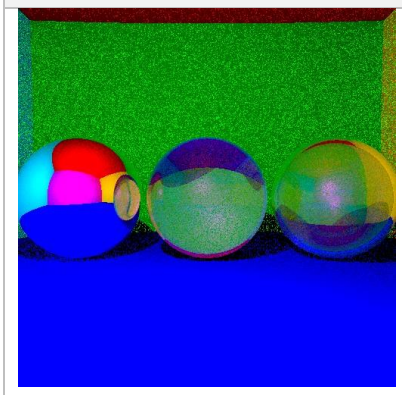
	Add a hollow circle inside, no flip of normal
	Sphere_right and sphere_right_hollow both set to hollow

	Max depth length = 2
	Max depth length = 10 Has total internal reflection
	Hollow sphere (RHS) Fixed total internal reflection Time to render: 197 milliseconds

Fresnel Law

	Fresnel Law implemented Time to render: *88925 milliseconds*

Indirect Lighting



Max depth length = 10

Rays per pixel = 50

Time to render = 7908 milliseconds

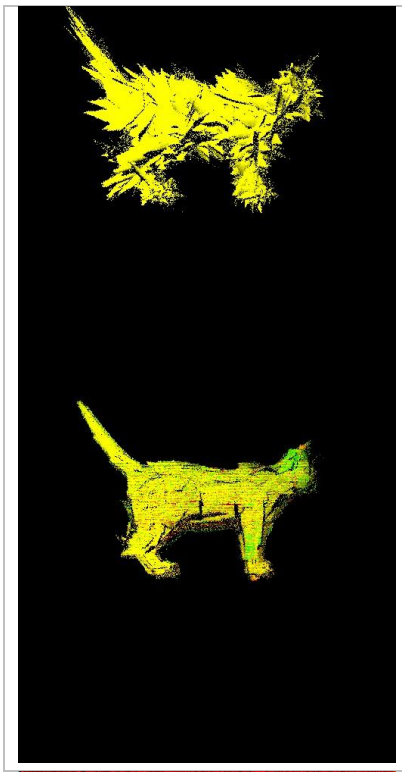
Antialiasing

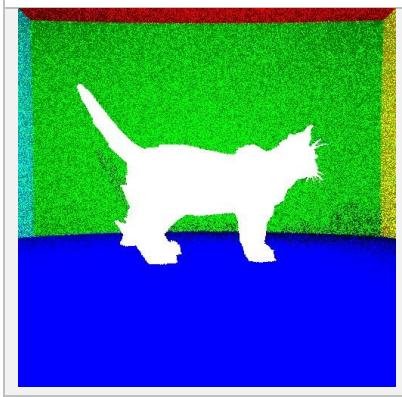
Max depth length = 10

Rays per pixel = 100

Time to render = 55 seconds

Cat Triangle Mesh





Max depth length = 5
Rays per pixel = 32
Time taken = 47 minutes

Beginning of code for the albedo of triangle mesh is commented out in the code.