< is problem  
>>> is solution to all inserted problems  
  
Intention  
I want to update the quality of using the application.   
  
< Poll events during rendering. Right now, when the scene is being rendered, the whole program is stuck until the scene is done the rendering. You also can't move or resize the window while this is happening. Nor can you close the window because the window event are only being polled when the ray tracer is finished. I want to do this while still having real-time like updates so that the user can see how tracing progresses.  
  
  
< Break dependency that ray tracer has with window. The ray tracer should not know anything about the window. It only has to concern itself with the actual ray tracing. Not with updating the information in the window title bar.  
  
  
< Support resizing of the window. Right now, the window screws up the rendered image when it is resized. I want to scale (preferably in a constant aspect ratio) the rendered image so that users with a larger screen can benefit from their screen.  
  
  
>>> The ray tracer receives a time value. This time value represents how long it is allowed to run. It returns a bool if the whole tracing operation is completed. If it isn't it returns false. This way. Internally, the ray tracer remembers where it was at the last time. This way, the ray tracer is only concerned with the image. Not the window. The class that calls the window can modify the window at will. The window class will also poll the events after the specified time.  
  
  
< Wrap all surface, light and camera information into a class of its own. Right now the main function creates all these variables. I want to make a scene-like class that manages this for me. It's also more sensible to pass a scene into the ray tracer instead of containers.  
  
  
< I want to support on the fly surface modifications. I want to load the surface metadata from text files instead of it being embedded in the code. This way, I don't have to rebuild the project every time I want to move something around.  
  
  
>>> I will create a scene class that holds all the surface data. I will also support parsing the data files.   
  
---------------------------------------------------------  
  
\_\_Flow of Program\_\_  
create window  
create image  
create scene  
create event  
create ray tracer  
  
load scene data  
  
bool isRendering is true  
  
while window is open  
 poll events  
 if event toggleRenderState  
 isRendering is !isRendering  
   
 if event resetRender  
 clear image  
 clear ray tracer  
 isRendering is true  
  
 if event reloadScene  
 load new data into scene   
  
 if event saveRender  
 save rendered file to image  
  
  
 Renderer.render(timeLimit,scene,image)  
  
  
 display image in window