Computer graphics are used in simple front end applications all the way to video games. From my experience I have a good take on 2d and 3d graphics from the little game development I have done. I have no knowledge of the difference between Direct3X and OpenGL or its interface, although I do have some experience in C++. On top of this I am curious on how shaders are actually compiled. From my knowledge they are compiled the first time that a graphic is seen unless forcefully compiled from viewing these (a lot of applications just secretly do this as a loading screen). Linking shader however I have no knowledge of but can guess that it is the connection between different graphics and how they meld together. I know the concept on how linear algebra is used to get the users view from the 3d objects and from there display what it should like if they were there. Taking the normal vector from the object to the users point of view and rendering whatever it touches. I am curious to know at what point shaders are compiled as well as if there is a more efficient way to compile them before a use ever interacts with an application to prevent live shader compilation and stuttering.