The Final Project

The class proposal was definitely my weakest part of the project. I was a little overwhelmed at the idea of creating a game and in the first assignment I missed a few key parts of the proposal like a class diagram for instance. When I first started creating my game, I decided on a space game and I also decided on doing on my own. The decision to create my own project wasn’t done out of any need for seclusion but because I live on the other side of the country from most of my fellow students and my time online is limited to the evenings, even on the weekends. I didn’t really keep notes of my development except for those found within the program itself. I did however keep a number of video and website links. I used a lot of this information to create my game:

<https://msdn.microsoft.com/enus/library/seyhszts.aspx>

<http://www.entityframeworktutorial.net/stored-procedure-in-entity-framework.aspx>

<https://www.tutorialspoint.com/linq/linq_quick_guide.htm>

<https://www.youtube.com/watch?v=NLs44hxV514>

<https://www.youtube.com/watch?v=V2A8tcb_YyY>

<https://www.lassiemarlowe.com/tutorials/c-and-mysql-tutorial/>

<https://www.youtube.com/user/sentdex>

<http://stackoverflow.com/questions/28680675/moving-a-picture-box-in-timer-c-sharp-winforms>

I also did a lot of bug testing. Most of the bugs I found had to do with the movement of the picture boxes. So basically I took picture boxes, customized their shape and filled them with pictures. The little spaceship I created myself and, as is pretty evident, I have never done that before. The asteroids are shaped randomly. There weren’t many sources that told me how to draw picture boxes in random shapes. That caused a lot of issues and it took me about two weeks after starting on them, to get them to work. Basically just constant system crashes, the asteroids would all be exactly the same shape, ect. Those issues are a thing of the past, there is a small bug remaining from the randomly generated picture box issue however.

When the asteroid picture boxes run over each other, they reroll their shape. I never bothered to correct the issue because I actually like it. To me, it looks like they are sent into a spin when they crash into each other. So I guess that “bug” is now a feature?

I had another issue that caused me a great deal of pain and it was a horrible newbie mistake. In my database I named my table…”table.” It took me forever to figure out that I had to use [table] when I referred to it. So the database part of the assignment was a complete nightmare to figure out and completely my own fault. Yes the table is still named [Table] but I was reluctant to change anything. When I figured out that I simply had to put brackets on the table name, everything worked great. When I tried to change it or create a new table, I ran into problems over and over again.

The game itself is fairly simple to play. In order to move the space ship around you use the standard w,a,s,d keys. The ship has a small range laser you can fire using the space bar key. Use the laser sparingly because it does have a recharge time. The ship will warp from one side of the screen to the other of you leave the paying area. You are docked points for movement so excessive movement is not recommended. You have 3 lives. When you collide with an asteroid you lose a life (there is a brief cooldown after a collision) when you lose three lives the game is over. After the game is over a “Save Your Score” button appears on the bottom right hand side of the screen. Click on that and you are sent to the scoreboard screen where you can save your current score, update your score in the database, view scores (using the LINQ stored procedures), or even delete scores form the database. Click on the appropriate radio buttons at the top of the screen and the associated buttons, text boxes and labels will be available to you. I wanted to prevent confusion so I hid the functions that aren’t needed