The Mars Lander project was quite an adventure. This was my senior project and I was going to make it awesome. The first thing I learned in class was that this was a group project, the second thing was that we had to choose our project from a list of projects. I thought, *that doesn’t sound so bad, I can find a group of fellow programmers with my same gaming focus and we will knock this out of the park*. I looked thought the projects to find that most of the project were web developments for business sites, except for one which was a project to create a guidance control system for a mars lander. The mars lander was the only one for me. With that settled the professor created a group from others that were interested in the mars lander project. Thus, I was part of a group of five individuals.

The first thing I did was to inquire into everyone’s skills to see what role they could be most useful in. It turned out that not one of my teammates was a programmer! There were two project managers, an IT security specialist, web designer, and hardware engineer. All of them had taken programming classes years ago and were rusty in any programming, with the exception of the web designer who was comfortable with design documentation. My thoughts at that point were along the lines of, *great I get to do all the work myself.* The course was 8 weeks, so we had no time to waste, and the first week was just the group formation so that gave us 7 weeks to make a mars lander simulation. The first week was figuring out roles and completing the design of the program.

We went with C++ because that was my strongest programming language. The web designer and the IT security specialist were tasked with creating a user interface. The project managers were tasked with creation of a work breakdown structure and adherence to it, plus compiling all work for turn-in at the end of each week. The hardware engineer disappeared. That left me working on all the programming.

I was super excited about all the programming, working out the program to utilize object-oriented programming was a must. Setting about creating classes for all the different parts of a guidance control system went smoothly. The program seemed to be not as difficult as I thought it was going to be. Get some starting numbers from the user for velocity and angle and let the system do the rest, no problem. What I and my group had fail to consider was that we needed to create a simulation to provide input to the guidance control system. That is what turned out to be the hard part. Suddenly two weeks into the project I am researching the gravity on mars, and what terminal velocity is on the red planet. I had to consider weather conditions like wind on the surface of mars. With the thin atmosphere, what kind of drag is created and how will this effect the decent of the lander. It was clear to me that my knowledge of the 4th rock from the sun was insufficient for a simulation that I would deem acceptable.

I decided that the simulation would be as real as I could make it. I also chose to make the input from the simulation no more turbulent then the lander could handle, because it should be built to handle the conditions. I created random number generators to simulate turbulence, and a gravity function to pull on the lander. There were several phases to the landing like atmospheric entry, retrorocket braking maneuver, parachute deployment, and the landing. It all worked out in the end, except for the UI which never came together. That left the console window listing out statistical data on the landing at one second intervals. Not a super exciting UI but the code was great and worked just like I planned. We got an A on the project, so I would think that mission control in Huston would be proud.