Updates: All camera positions. Fixed update #1 error about diameters. Added missile radius.

The Romulan Imperial War College has been quite a challenge. As a fourth year student your Computer Simulation course assignment looks like it will keep you up many a night. You have to write a visual simulation for the new "no-inertia" drive class of Warbird. Amazingly, this ship can stop in space immediately! Thankfully your assignment has three "phases". For the first phase you need to simulate the solar system and warbird. In the second phase you will add control of the warbird, target missile defense, and the intelligens-semita missiles. In the third phase you will add a lights and textures for the star field. You better start examine the specs and thinking of a design…

Ruber System. The simulation takes place in the remote red-drawf Ruber solar system. Besides the sun (Ruber) there are two planets, Unum and Duo, and Duo has a two moons. The planets orbit around the sun in the XZ plane (rotate on Y). The moons orbit Duo in XZ plane. The planets and moons do not revolve on their axes. There are missile installations (your targets) on Unum and Duo's moon Secundus. Your professor is kind, he only wants you to simulate gravity for Ruber (no gravity for the planets and moons). You will need to make models for the warbird, planets, moons, missile defense sites, and missiles. Your simulation should use the fixed interval timer's Time Quantum (TQ or interval) to call an update(...) function for each "simulation update frame". You should render (display) on an idle timer. A spreadsheet is available off the class page that describes all parameters and "codes" the simulation's parameters. Download and change values to see their effect.

				orbit		
Planet	**radius / side**	distance	"gravity vector"	Radians / frame	(sec)	(min)
Ruber	2000	0				
Unum	200	4,000 on x	5.63	0.004	63	1.0
Duo	400	9,000 on - x	1.11	0.002	126	2.1
Primus	100	8,100 on - x		0.004	63	1.0
Secundus	150	7,250 on - x		0.002	126	2.1
Warbird	100	(5000, 1000, 5000)	1.76			
missiles	25					

The orbits are specified for a fixed interval timer value (TQ time quantum) of 40 milliseconds ("trainee") which is 25 U/S (updates / seconds). For example Unum should orbit Ruber every 63

seconds at this speed. This is the "ace" setting. In the second phase your simulation will support selection of different animation speeds: ace, pilot, trainee, debug (as shown in the spreadsheet).

player	TQ
ace	40
pilot	100
trainee	250
debug	500

Warbird. Initially you can use the spaceShip-bs100.tri model in the triModels directory as a placeholder. You need to make your own ship model. Use AC3D (or another modeler) to model your warbird. Keep it simple. You can always replace the model as your simulation progresses through the review phases. The Warbird model should be contained within a bounding sphere with a radius of 100.

Cameras. There are two static cameras: "front" and "top" and three dynamic cameras ("ship", "Unum", and "Duo"). The cameras should be selected by pressing the 'v' key to toggle from the first to last view in a mod (wrap around) manner. The cameras are described in the table below. Both above planet cameras maintain position with the planet and look down on the planet as it orbits Ruber. The warbird camera will also maintains position and orientation with warbird as it moves (second phase).

cameras	looking at	distance from looking at
front	(0,0,0)	**(0, 10000, 20000)**
top	(0,0,0)	**(0, 20000, 0)**
ship	warbird	**eye (0, 300, 1000) up and behind at (0, 300, 0), over the ship**
Unum	Unum	**eye (0, 4000, 0) looking down**
Duo	Duo	**eye (0, 4000, 0) looking down**

Shaders. You should use the "simpleVertex.glsl" and the "simpleFragment.glsl" shader programs for the first phase. These are flat shaders with no lights. They aren't very interesting. Don't worry, the shaders you will write for phase 3 will make the simulation look better.

Missiles. The ship has 9 missiles and each missile defense site has 5 missiles. Missiles models must be contained within a bounding sphere radius of 25. Missiles are smart; they know and track their target. Missile firing and behavior is part of the second phase.

Information Display. You must display the missile counts, update rate (U/S), frame rate (F/S), and current view (camera) in the window title. "U/S" represents "updates per second". For example

```
Warbird ??   Unum ?   Secundus ?  U/S ??  F/S ????  View *
Warbird  7   Unum 5   Secundus 0  U/S 25  F/S  321  View Duo
```

Your assignment should be written in OpenGL and should be system independent. For classes use includes statements. If you use collections, use the Standard Template Library.

Phase 1. You should have the planets and moons orbiting Ruber and you should have the front, top, Unum, and Duo cameras. You should have the warbird and 1 missile at (4900, 1000, 4900). You can use the same model with different scale values for Ruber, planets, moons, and missile (see ManyCubes) if you change the color values after loading the model.

You should submit your source code and any documentation on moodle as a compressed (zip file). Do not include any projects that have or require version control. I will not connect to any remote version control site to build/run your project. If you are using Visual Studio you can "zip" your project directory and recursively its subdirectories. I will not install any other IDE. If you are using a different IDE please submit all source files in a single directory and if you have paths to model files in other directories; use relative paths. I will not modify my systems environment variables. In other words, do not hard code file path/names.

You should submit "electronic/soft" documentation as files included in your zip archive. You do not need to repeat information in the project specification and spreadsheets. Documentation should provide "meta information" – what you would like to read if you were looking at this project source code after not using (reading, editing) it for 1 year. What you would like to remind yourself about your design and implementation. Meta information includes diagrams, tables, and text narrative. You should include any information I may need to build/run/grade your project (beyond the obvious). If you did something neat – you should write to tell me about it so I can be sure to see it.

As before, ask questions if you have them. Lab is the most efficient way to discuss the project.