The Romulan Imperial War College is 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, missile sites, and the new intelligens-semita missiles. In the third phase you will add a Point light and textures for the star field. You better start examine the specs and thinking of a design … prof Renzo is demanding…

**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. In the first phase there is no gravity. You will need to make models for the warbird, planets, moons, missile 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" to “update” the planet and moon orbits (rotations). You should render (display) on an idle timer. The WarbirdSimulation.xls spreadsheet file is available off the class page that describes all parameters and "codes" the simulation's parameters. Download and change values to see their effect. The spreadsheet is also shown at the end of these specifications.

**Animation.** The orbits are specified for a fixed interval timer value (TQ or time quantum) of 5 (“fast”) or 40 (“ace”) milliseconds. Start with the “ace” setting) which is 25 U/S (updates / seconds). For example Unum should orbit Ruber every 8 seconds at the “ace” speed. In the second phase your simulation will support selection of different animation speeds; the spreadsheet will be updated.

**Models.** Use AC3D (or another modeler) to model your scene objects. Keep the models simple. Each model should be contained within its bounding sphere. Each scene objects bounding sphere radius is specified in the spreadsheet. For example, the Warbird model’s is 100. Using AC3D create the model and then export as a triangle file. The includes465/triModel465.hpp file supports loading AC3D’s triangle files (a simple colored model file format). Initially you can use models from the triModels directory as temporary placeholders (for example spaceShip-bs100.tri. Replace the placeholders with your own models as you make them.

**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’ (next) and 'x' (previous) keys to toggle from the first to last view in a mod (wrap around) manner. The cameras are described below and in the spreadsheet. Both planet cameras maintain position with the planet and look at the planet as it orbits Ruber towards the camera. The warbird camera also maintains position and orientation with warbird when it moves (in the second phase).

**Shaders.** You should use the "simpleVertex.glsl" and the "simpleFragment.glsl" shader programs for the first phase (in ViewTriModel directory). 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. Missile models must be contained within a bounding sphere radius of 25. Missile models can't be spherical, they should have a visible front and back. Missiles are smart; they know and track their target. Missile firing and behavior is part of the second phase. You only have to have a model for the missile in phase 1.

**Information Display.** You must display the missile counts, update rate (U/S), frame rate (F/S), and current view (camera) in the window title. Not all of these values will update in the first phase. "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
```

There is posted spreadsheet “warbirdP1.xls” that has values for the simulation. For example you can change the TQ and see how it affects orbit times.

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, 4850). You can use the same model with different scale values for Ruber, planets, and moons (see examples ManyCubes and ManyModelsStatic) although the color will be the same as a place holder until you make your own models.

You should submit your project (source code) and any documentation on canvas 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 (solution) 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 that contains all subdirectories and files. If you have paths to model files in other sub-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.

Ask questions if you have them. Lab is the most efficient way to discuss the project. Where else can you talk one-on-one with your professor (or the person that is going to be grading your submission) ?

| 9/5/2017 | WarbirdSimulation.xls worksheet | | | Simulation | msec. |
|---|---|---|---|---|---|
| | | | | ace | 40 |
| Time Quantum | 40 | updates / sec | 25 | fast | 5 |

| | | center | | radians / | orbit | |
|---|---|---|---|---|---|---|
| Planet | radius | distance on +x | (+x, 0, 0) | update | (sec) | (min) |
| Ruber | 2000 | 0 | (0,0,0) | | | |
| Unum | 200 | 4,000 | (4000,0,0) | 0.004 | 8 | 0.1 |
| Duo | 400 | 9,000 | (9000,0,0) | 0.002 | 16 | 0.3 |
| Primus | 100 | 11,000 | (11000,0,0) | 0.010 | 3 | 0.1 |
| Secundus | 150 | 13,000 | (12000,0,0) | 0.030 | 1 | 0.0 |
| WarBird | 100 | 15,000 | (13000,0,0) | | | |
| missiles | 25 | 14,500 | (13500,0,0) | | | |

| Camera | looking at | translation from object camera us 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 |
| | (0, 300, 0) | looking at (0, 300, 0) see above and past the ship |
| Unum | Unum | eye (-8000, 0, -8000) looking at Unum as Unum approaches view |
| Duo | Duo | eye (-8000, 0, -8000) looking at Duo as Duo approaches view |