**INFO6019 – Physics 1**

**Midterm Exam – Monday, October 30th, 2023**

Instructor: Michael Feeney

## The exam format:

* You may use any resources you feel are necessary to complete the exam, but you are to answer the questions **on your own**. I will be looking for plagiarism (i.e. copying) very carefully. There is *no possible way* that the specific code to answer these questions, or the output to the screen, would be very similar to the look of another student’s code. Remember, this is a test and there are very clear policies about cheating on tests.   
  + <http://www.fanshawec.ca/admissions/registrars-office/policies/cheating-policy>
  + <http://www.fanshawec.ca/sites/default/files/assets/Ombuds/cheating_flowchart.pdf>
* It is an “open book” exam. You have access to anything you book or internet resource you’d like
* The questions are ***NOT*** of equal weight. The exam has **six (6)** questions and **five (5)** pages. The questions involve submitting a working Visual Studio solution.

|  |
| --- |
| You have until **11:59 PM** on **Monday, October 30th** to submit all your files to the appropriate drop box on Fanshawe Online.   **NOTE:** Although this may “look and feel” like a project, it isn’t, it’s an **exam**, so there is **no concept of “late marks**”; if you don’t submit your files the time the drop box closes, you don’t get any marks at all.  *Please don’t be late submitting.*  (Also be **SURE** that you are actually submitting the correct files) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * The questions build on each other, to make a complete scene. However, you may decide that a different camera angle will better show the scene, so:   + You are to submit a single solution (not multiple solutions/projects)   + Use keyboard controls to best show the appropriate question, so pressing “1” will place the camera to best show question 1, “2” to show question 2, etc.   + NOTE: you do not *need* to do this, but it’s very likely that it’s to your advantage.   **Do not** comment out your code, or expect me to alter it in any way at all. It should be ready to run as is with the **default setting** in Visual Studio **2022**, as an **x64** executable with “Release” libraries:   |  |  | | --- | --- | |  | 🡨 This type of executable | |  |  | |

* For applications: if it doesn’t build and run, *it’s like you didn’t answer it*. I’ll correct trivial, obvious problems (like you clearly missed a semicolon, etc.), but you need to be sure that it compiles and/or runs.

(Also be **SURE** that you are actually submitting the correct files)

When ready to submit, PLEASE for the love of all that is good on this Earth, delete all the “extra” Visual Studio files before zipping it up, like the “Debug” and “Release” folders with the “obj” and log files, as well as the “vs” (intellisense) folder – this will save a tremendous amount of space and shorten your upload time..

* + **But**, give me the **ENTIRE SOLUTION**, not just the source files.

**If the solution does not build (and run), I will not mark it** (so you will receive zero on questions that can't be built and/or won't run). When I say "run", I'm not speaking about some, random, unforeseen bug, but rather something that you should have obviously dealt with, like memory exceptions, etc.

**“Nucleonic Particles Attack!”**

In the TV show “Star Trek: Enterprise” season 3, episode 10, the Enterprise flies through a nebula only to discover that these small “Nucleonic Particles” are plugging up the engines and magnetically attaching themselves to the outside of ship: <https://memory-alpha.fandom.com/wiki/Similitude_(episode)>

You can see this happening at 7:05, 11:19, and 15:13. They end up using the phasers (or whatever) to blast them off the ship (seen at 27:34 and later).

These “nucleonic particles” are basically small magnetic rocks that stick to the outside of the ship.

I’m going to use “nucleonic particles”, “particles”, and “asteroids” interchangeably through this exam.

You’re going to simulate the basics of this, with some small simplifications.

The episode video is on FOL, as it’s too big for github. You don’t have to watch it, but it’s there if you’re interested.

You’re going to use this ship model instead of the Enterprise:



Note that this model (SM\_Ship\_Massive\_Transport\_01), and the asteroids (Asteroid\_011\_x10 and Asteroid\_015\_x10) are huge. You’ll have to adjust your projection transformation near and far plane to show all of it and to avoid z-fighting.

The front of the space ship is to the left in the picture. The bridge/cockpit is the small light blue windows at the top left of the image. You can see the large engines in the back.

It’s aligned along the z-axis, facing “forward” along the positive (+ve) z axis.

1. (10 marks) Load and display the entire space station and the two (2) types of asteroids, in your application. Place the camera so the entire space ship is taking up about ½ to 1/3 or the screen, and the space ship is in the centre.   
     
   While you aren’t getting marks for lighting, it should be clearly visible (i.e. if you can’t see the space station, you will *lose* marks). My suggestion is to place one, or more, point light(s) far away from the station, and drop the attenuation, so that it looks like a directional light.
2. (40 marks) Generate an “asteroid field”. Instead of moving the ship through the asteroid field, you are to move the asteroid field past the ship. In the episode, the ship is stopped.   
     
   At no *less* than approximately once a second:

* Place five (5) of either type of asteroid in a random location “just off screen” from the ship. The ideas is that the ship is completely surrounded by a huge asteroid field.
* Pick a random direction for the asteroids to travel. They need to travel “towards” the ship. They don’t have to necessarily collide with the ship, but we want them to *not* appear off-screen and fly *further* off-screen.
* When the asteroid has passed the station and is “far away” enough to not been seen, you can stop drawing it.
* At this point the asteroids can pass through the space ship (we’ll deal with collision later).

1. (50 marks) Simulate the asteroids colliding and sticking to the outside of the space ship.   
     
   As the asteroids fly past the space station, some will intersect (collide).  
     
   Test for the collision of the asteroids at each frame.   
     
   If an asteroid collides with the space ship, simulate the collision in this manner:   
   * Draw an “explosion” by placing a small red sphere at the point of collision.
   * Over the next few seconds, change the scale of the sphere so that it grows to about 5x its original size, then disappears.
   * Remove the asteroid from the simulation.
   * Place a small (smaller than the original red sphere, but large enough that it can be seen), dark grey sphere at the place where the impact occurred.   
       
     NOTE: This dark grey sphere will stay in place, and *not* disappear or move. This is to mimic the nucleonic particles sticking to the hull.
2. (30 marks) Some of the asteroids hit each other.

If the asteroids touch each other, they should obliterate each other.

If they get close enough to touch (you can “eye ball” this – i.e. does it *look* like they collided?) they are to explode. Do the same “expanding red sphere” technique as in question 3, but after it’s gone, *both* asteroids are gone.

1. (40 marks) Instead of a laser/phaser/whatever to shoot at the asteroids, your crew has come up with some kind of shield/repulsion technology to assist them.   
     
   By pressing the space bar enable this shield thing and blast away all the asteroids.  
     
   The asteroids should fly away from the centre of the space ship. You set the direction of these asteroids to be *away* from the centre of the ship (possibly the origin if you didn’t move the space ship). They fly away and off screen.
2. (BONUS: 30 marks:) Change the asteroid movement so they are “attracted” to the space ship.  
     
   Instead of travelling in a straight line and colliding, they are gently drawn to the space ship.   
     
   You can do this by slightly adjusting their acceleration towards the centre of the ship.   
     
   They would collide just like before.

That’s it.