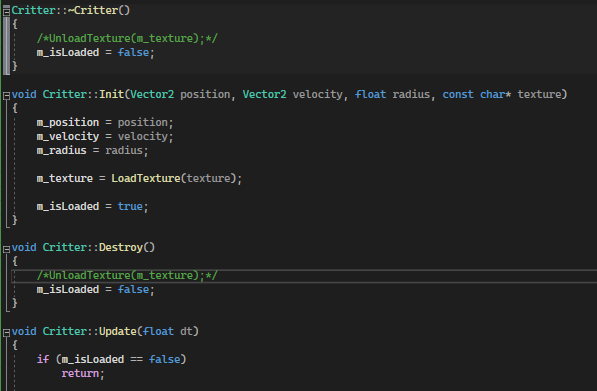
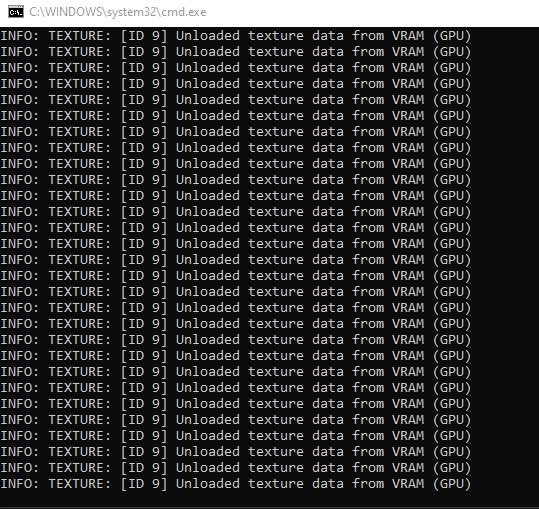
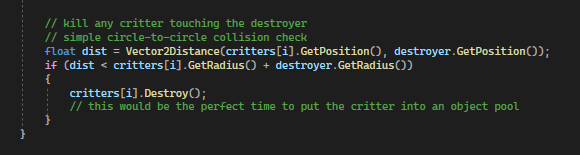
**Task 1: Design Game Optimisations**

Unloading Textures

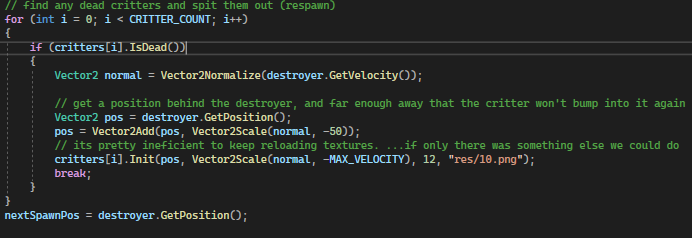
Every time a critter collides with the destroyer it unloads the texture a bunch of times in the command prompt window. Commenting out these “UnloadTexture” functions stops the program from unloading the texture from the GPU heaps of times which makes the program more efficient.



Respawning Critters

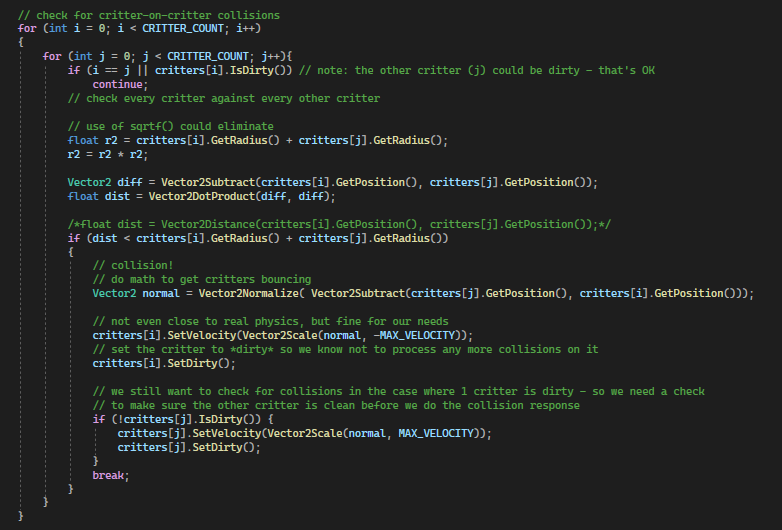


In the code above it checks if the critters are colliding with the destroyer by checking the distance between the radius of the destroyer and every critter. Instead of destroying the critter it is collided with you could create an object pool to store the destroyed critter then return the critter back to the game after the respawn timer has finished. In the respawning code(below) it inefficiently reloads the texture again while spitting it out behind the destroyer. If we had an object pool set up this could be avoided. You could also set up the program to only load in the image once instead of re loading the image for each individual critter.



Critter-On-Critter Collisions

There is for loop inside a for loop checking for critter-on-critter collisions constantly while the program is running. This is operating at a (O)n^2 (quadratic) order which means that the more critters in the game the slower this function will run. This is considered a very slow algorithm.



To improve this, we could use a technique called spatial hashing which gets objects from a 2D space and project them into a 1D hash table which allows for very fast queries on the objects. Objects are hashed every frame for real-time applications. The spatial hashing technique works in linear time O(n) which will search through critter collision much quicker than the current quadratic solution that is implemented. If we add more critters to the game, then the query time will still increase but not exponentially like it is currently. A diagram with spatial hashing shown below to show how it stores data in the table.

