Optimisation Design Brief

Texture Loading:

A computer screen with text and images

Description automatically generated

A computer screen with text

Description automatically generated

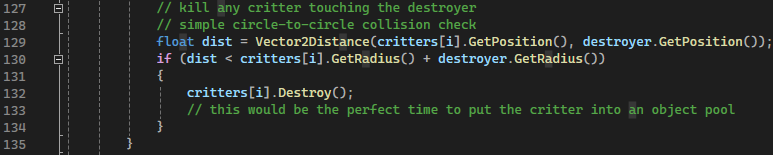
A screen shot of a computer code

Description automatically generated

The texture loading needs optimisation because as the program is now, whenever a critter gets made, it loads a texture, and when one gets destroyed, it unloads a texture. Because this happens so often, the program is slowed down.

To fix this, I will load the texture once at the start of the program, and then have the critters point to that texture.

Object Pool:



This part of the code needs optimising, as without an object pool, there is a risk of memory fragmentation, where space that should be available to store a critter is unavailable. Adding an object pool would allow the program to take critters from a list of critters, instead of frequently making new ones.

To do this I made a new class for object pools, which acts as an array of critters.