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

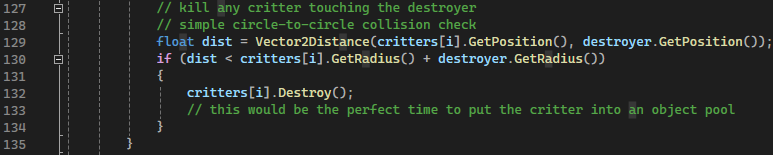
Description automatically generated

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 remove the UnloadTexture() function from the Destroy() function. As a result, texture will only unload textures when the program is closed.

I chose to make this change as I thought it would improve the speed of the program, and it has increased the framerate by about ten frames per second.

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.

“Use hash table to replace texture string with keys”

Object pool may be the same as dynamic array?