# Design Brief – Critters

# Introduction

This application features a simulation which loads a set amount of creatures and a destroyer. Whenever the critter collides with the destroyer, the critter will be destroyed. Each second a new critter will be spawned behind the destroyer until the set amount of critters is reached. This system features many superfluous systems which could be removed or altered to improve performance, possibly allowing for more critters or other added features to add to the simulation.

# Problem 1 – IsDirty()

This system appears to be a check which stops the critters from bouncing off of each other multiple times or sticking inside of each other through the collision check. This system is superfluous as the system will immediately check the collision of a new pair of critters. As this system is ran during every check which happens every frame between all critters, it would make it much faster to remove this system.

# Problem 2 – Destroy()

Each time a critter is destroyed, the texture is unloaded. When a new critter is spawned in a new texture is loaded. Instead of unloading and reloading, we can move the critter offscreen, set m\_isLoaded = false and use an if(!critter[i].IsDead()) at line 105 to stop the “dead” critter from moving. Then in the “respawn” loop, we can remove the critters[i].Init() and instead just change the position and direction and set m\_isLoaded = true.