Consignas

1.Analizar el comportamiento de la aplicación. ¿Qué problemas observa?

2. Realizar mediciones que permitan diagnosticar el problema.

3. Identificar el origen del problema. Explicar en detalle.

4. Proponer por lo menos una solución. Explicar en detalle.

5. Implementar la solución. Demostrar que la misma resuelve el problema. En caso contrario, explicar por qué no funciona y proponer una alternativa.

1)

Al comenzar la aplicación inmediatamente el programa va extremadamente lento y continua de esa manera de forma indeterminada. Se puede observar a simple vista como los frames por segundo de la aplicación son muy bajos. El problema se vuelve aun peor cuando se instancian mas objetos.

2)

Captura de pantalla de computadora

Descripción generada automáticamenteCaptura de pantalla de computadora

Descripción generada automáticamenteCaptura de pantalla de computadora

Descripción generada automáticamente

3)

Se esta llamando muchas veces a la funcion Dist, esta funcion calcula la distancia entre un objeto y otro. El problema con esto es que esta funcion esta siendo llamada muchisimas veces por el metodo GameObjectsNear de el world. Esta método a su vez esta siendo llamado por todas las hormigas y todas las feromonas. Esto en teoría esta bien, el problema es que esta muy mal implementado.

public IEnumerable<GameObject> GameObjectsNear(PointF pos, float dist = 1)

{

return GameObjects.Where(t => Dist(t.Position, pos) < dist);

}

Como se puede observar el array GameObjects busca por cada objeto, todos los objetos que tiene cerca, el problema es que este array contiene a todas las hormigas y todas las feromonas también, por lo cual es llamada muchísimas veces sin ningún sentido ya que las hormigas no tienen porque calcular a las otras hormigas cercanas y la feromonas no tienen porque calcular a las hormigas tampoco.

4)

Lo que se puede hacer es ya que lo único que importa son las feromonas cercanas, hacer que el linq solamente busque a los objectos cercanos si son feromonas, de esta manera se tendría que reducir mucho la cantidad de veces que se calcula la distancia.

5)

El código dentro de gameobjectsnear se cambio al siguiente:

return GameObjects.Where(t => t is Pheromone).Where(t => Dist(t.Position, pos) < dist);

La aplicación ahora si anda rápido, el problema es que no me di cuenta que las hormigas también buscaban pasto, por lo cual ahora ya no lo encuentran entonces las feromonas nunca se instancian y el juego se rompe. A continuación intentare arreglar esto al hacer métodos distintos, uno para las feromonas y uno para las hormigas así el de las hormigas busca pasto y feromonas y el de las feromonas solo busca feromonas.

Arreglando el código me di cuenta que la hormiga nunca tiene que buscar a la feromona sino que a la comida, por ende modifique el código para que la hormiga solo busque la comida. Ahora hay un método para la hormiga que busca comida cercana y uno para las feromonas para que busquen feromonas cercanas. El programa volvió a su estado de ir muy lento. Captura de pantalla de computadora

Descripción generada automáticamenteCaptura de pantalla de computadora

Descripción generada automáticamente

La aplicación esta un poco mejor que antes pero el dist sigue siendo calculado demasiadas veces, creo que lo que tendria que haber hecho es optimizar tambien la funcion dist en si o hacer algo para que calcule menos veces dist. Como se puede observar en la siguiente imagen no es el calculo de encontrar el objeto entre los objetos lo que mas “pesa” sino el calculo de la distancia en si.

Captura de pantalla de computadora

Descripción generada automáticamente