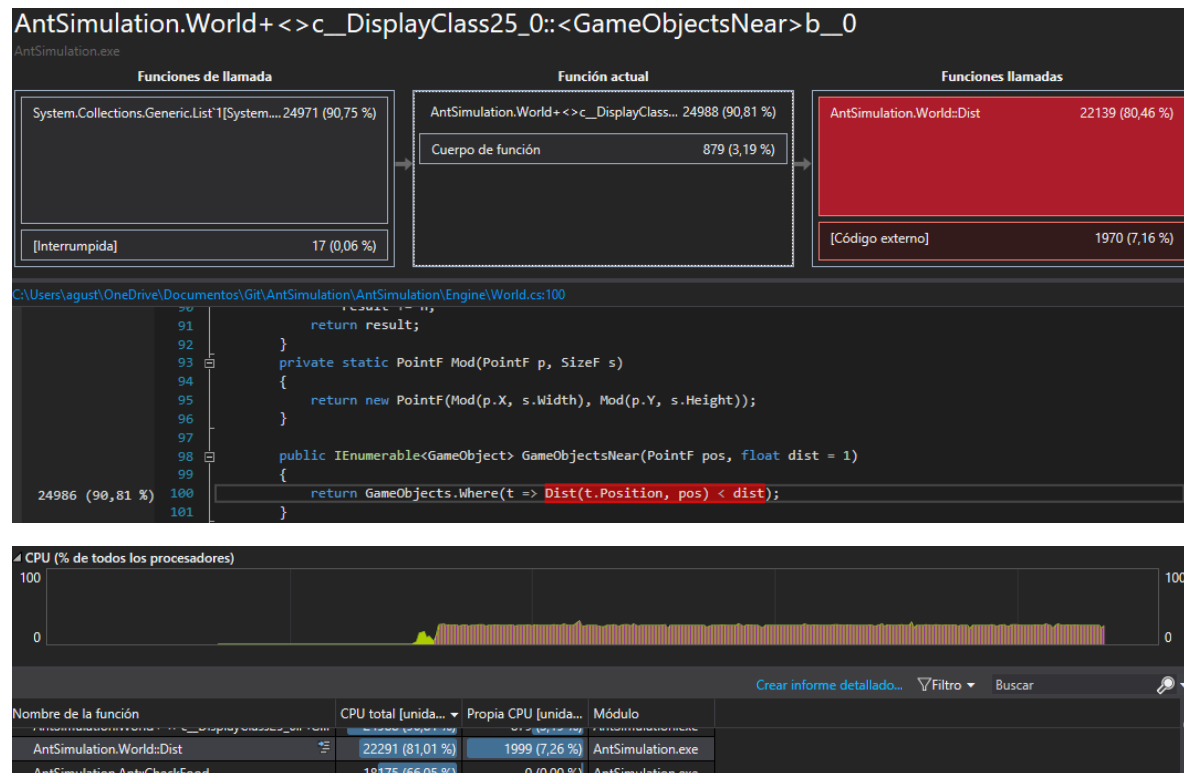


Nº 1

Problema identificado

Todo está en una única lista.

Medición



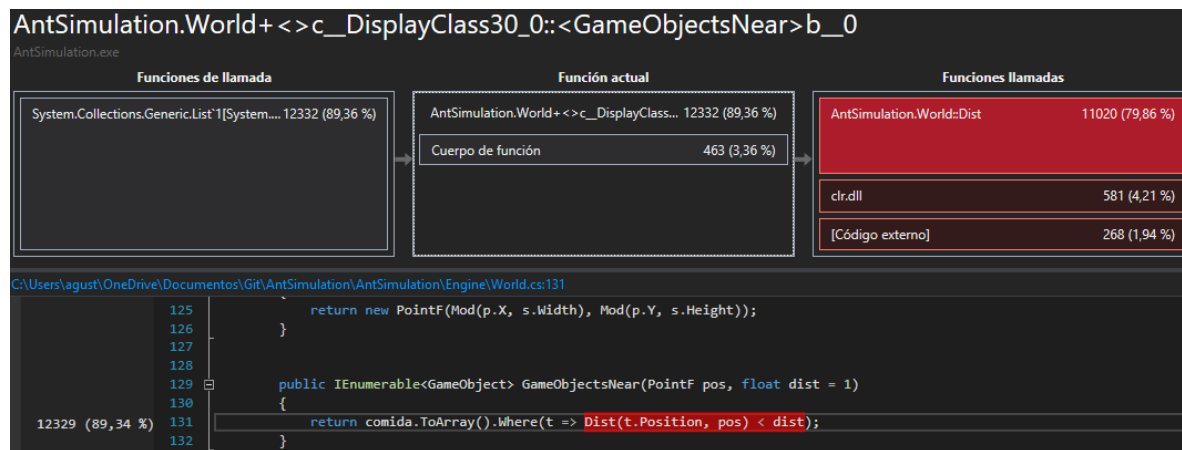
Hipótesis

Si creo una lista especial para la comida y las feromonas, evito perder tiempo descartando a las demás hormigas.

Descripción de los cambios

En la clase “World” reo una sobrecarga en el metodo add y remove para que si el objeto es de tipo “Pheromone” o “Food” sean ubicados en una lista privada nueva llamada “comida”. Y cambio el metodo “GameObjectsNear” haciendo que en lugar de retornar GameObjectsNear.Whe... retorne comida.ToArray().whe....

Nuevas mediciones



No se refleja el cambio en la medición pero en la practica la aplicación corre ligeramente más rápido, en todo caso, conservo los cambios debido a que me servirán para más adelante.

Nº 2

Problema identificado

El buscador por proximidad es muy ineficiente

Medición

World

```
128
129 public IEnumerable<GameObject> GameObjectsNear(PointF pos, float dist = 1)
130 {
12329 (89,34 %) 131     return comida.ToArray().Where(t => Dist(t.Position, pos) < dist);
132 }
```

Ant

```
103 private IEnumerable<T> FindNear<T>(World world, float radius) where T : GameObject
104 {
1   1 (0,01 %) 105     List<T> result = new List<T>();
106     for (float x = Position.X - radius; x <= Position.X + radius; x++)
107     {
594 (4,30 %) 108         for (float y = Position.Y - radius; y <= Position.Y + radius; y++)
109         {
12697 (92,01 %) 110             result.AddRange(world
111                 .GameObjectsNear(new PointF(x, y))
112                 .Select(t => t as T)
113                 .Where(t => t != null));
114         }
115     }
116     return result;
```

Medicion

```
Form1.cs Ant.cs World.cs Pheromone.cs
AntSimulation.Form1

private void updateTimer_Tick(object sender, EventArgs e)
{
    Text = world.GameObjects.ToString();
    ClientSize = new Size(world.Width * scale, world.Height * scale);
    var reloj = Stopwatch.StartNew();
    world.Update();
    reloj.Stop();
    using (StreamWriter stream = new StreamWriter("Medicion.txt", true))
    {
        stream.WriteLine(reloj.Elapsed.TotalSeconds);
    }
    Refresh();
}
```

https://drive.google.com/file/d/1D_Ma0DhEpkoWAI1lv-UerjYjsEWf9Dwd/view?usp=sharing

Hipótesis

Si le indico al GameObjectNear exactamente que buscar (en lugar de hacer un cálculo pesado por cada posible objetivo) agilizaría el proceso de búsqueda

Descripción de los cambios

Cambio la forma de rastrear el objetivo, haciendo que FindNear de la clase Pheromone y Ant reciban un arreglo del método GameObjectNear de la clase world. Y que estos solamente tengan que descartar a los elementos "null".

En el metodo FindNear de la clase Ant unicamente llame a FindNear y revise si hay elementos nulos

Creo un stopwatch para medir cuanto tarda el word.update (dejo un link a los resultados al final de las mediciones)

Nuevas mediciones

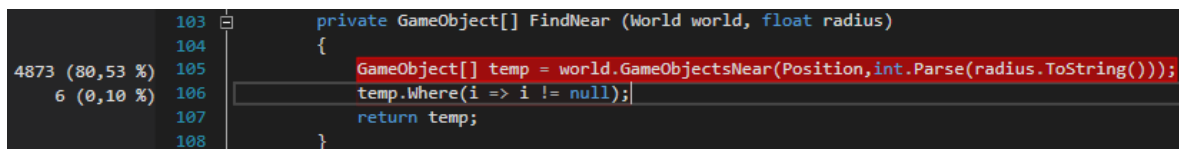
World



```
AntSimulation.exe
C:\Users\agust\OneDrive\Documents\Git\AntSimulation\AntSimulation\Engine\World.cs:130

128 public GameObject[] GameObjectsNear(PointF pos,int radio)
129 {
130     List<GameObject> temp = new List<GameObject>();
131     Point ubicacion = new Point(0, 0);
132     ubicacion.X = int.Parse(pos.X.ToString());
133     ubicacion.Y = int.Parse(pos.Y.ToString());
134     byte contador = 0;
135     GameObject[] miComida = comida.ToArray();
136
137     for (int x = ubicacion.X - radio; x < ubicacion.X + radio; x++)
138     {
139         for (int y = ubicacion.Y - radio; y < ubicacion.Y + radio; y++)
140         {
141             foreach (var item in miComida)
142             {
143                 if (pos == item.Position)
144                 {
145                     temp.Add(item);
146                 }
147             }
148             contador++;
149         }
150     }
151     return temp.ToArray();
```

Ant



```
103 private GameObject[] FindNear (World world, float radius)
104 {
105     GameObject[] temp = world.GameObjectsNear(Position,int.Parse(radius.ToString()));
106     temp.Where(i => i != null);
107     return temp;
108 }
```

(velocidad del update en el forms1)

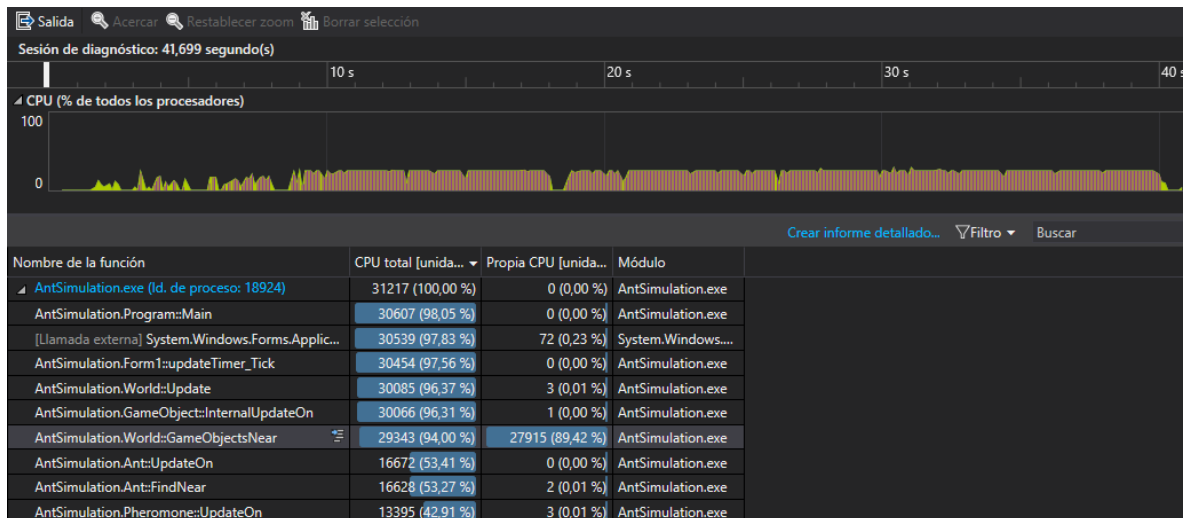
<https://drive.google.com/file/d/176do2lfwDjTujoh6Fm6twYfOWyhPzjB0/view?usp=sharing>

Nº 3

Problema identificado

Optimizar el GameObjectsNear parte 2

Medición



AntSimulation.World::GameObjectsNear

AntSimulation.exe

C:\Users\agust\OneDrive\Documentos\Git\AntSimulation\AntSimulation\Engine\World.cs:130

```

128 public GameObject[] GameObjectsNear(PointF pos,int radio)
129 {
130     List<GameObject> temp = new List<GameObject>();
131     Point ubicacion = new Point(0, 0);
132     ubicacion.X = int.Parse(pos.X.ToString());
133     ubicacion.Y = int.Parse(pos.Y.ToString());
134     byte contador = 0;
135     GameObject[] miComida = comida.ToArray();
136
137     for (int x = ubicacion.X - radio; x < ubicacion.X + radio; x++)
138     {
139         for (int y = ubicacion.Y - radio; y < ubicacion.Y + radio; y++)
140         {
141             foreach (var item in miComida)
142             {
143                 if (pos == item.Position)
144                 {
145                     temp.Add(item);
146                 }
147             }
148             contador++;
149         }
150     }
151     return temp.ToArray();

```

Hipotesis

Me doy cuenta de que con un solo foreach le ahorro una parte importante del trabajo al gameObjectsNear

Descripción de los cambios

Cambio el buscador por este:

```

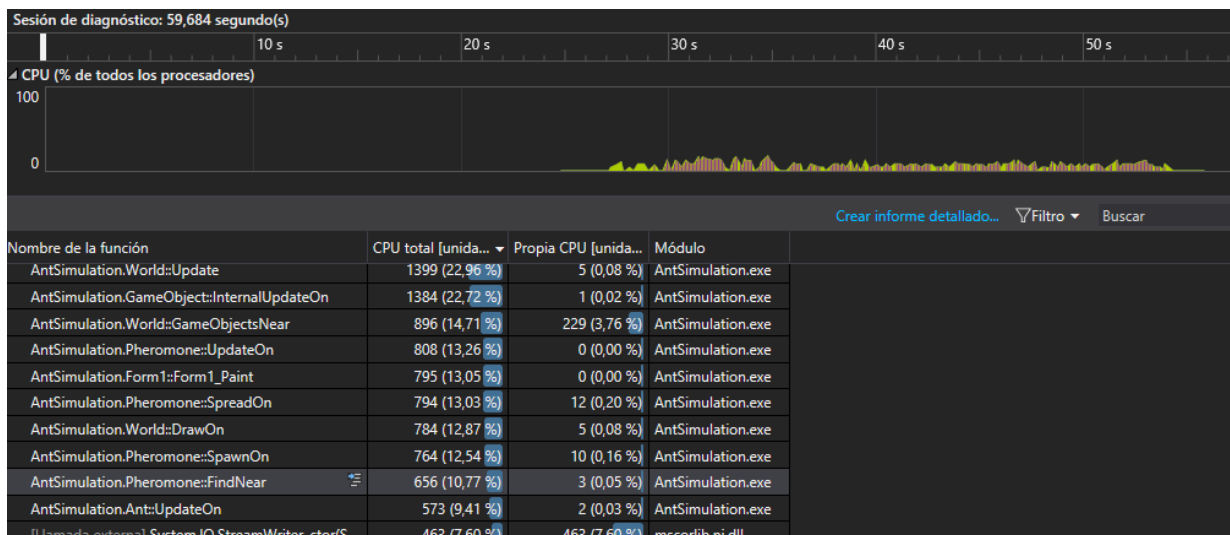
public GameObject[] GameObjectsNear(PointF pos,int radio)
{
    tempNear.Clear();
    int Cx = int.Parse(pos.X.ToString()) - radio;
    int Fx = int.Parse(pos.X.ToString()) + radio;
    int Cy = int.Parse(pos.Y.ToString()) - radio;
    int Fy = int.Parse(pos.Y.ToString()) + radio;

    foreach (var item in comida)
    {
        if (item.Position.X >= Cx && item.Position.X <= Fx)
        {
            if (item.Position.Y >= Cy && item.Position.Y <= Fy)
            {
                tempNear.Add(item);
            }
        }
    }
    return tempNear.ToArray();
}

```

Y en el metodo CheckFood de la clase Ant hago que a la hora de eliminar la comida o feromona se envíe con su respectivo tipo.

Nuevas mediciones



AntSimulation.World::GameObjectsNear

AntSimulation.exe

C:\Users\agust\OneDrive\Documentos\Git\AntSimulation\Engine\World.cs:132

```

130 public GameObject[] GameObjectsNear(PointF pos,int radio)
131 {
132     tempNear.Clear();
133     int Cx = int.Parse(pos.X.ToString()) - radio;
134     int Fx = int.Parse(pos.X.ToString()) + radio;
135     int Cy = int.Parse(pos.Y.ToString()) - radio;
136     int Fy = int.Parse(pos.Y.ToString()) + radio;
137
138     foreach (var item in comida)
139     {
140         if (item.Position.X >= Cx && item.Position.X <= Fx)
141         {
142             if (item.Position.Y >= Cy && item.Position.Y <= Fy)
143             {
144                 tempNear.Add(item);
145             }
146         }
147     }
148     return tempNear.ToArray();

```

https://drive.google.com/file/d/1VAaCsPhQ-nvr6vyPf5jqSeLg5xv3tL_i/view?usp=sharing