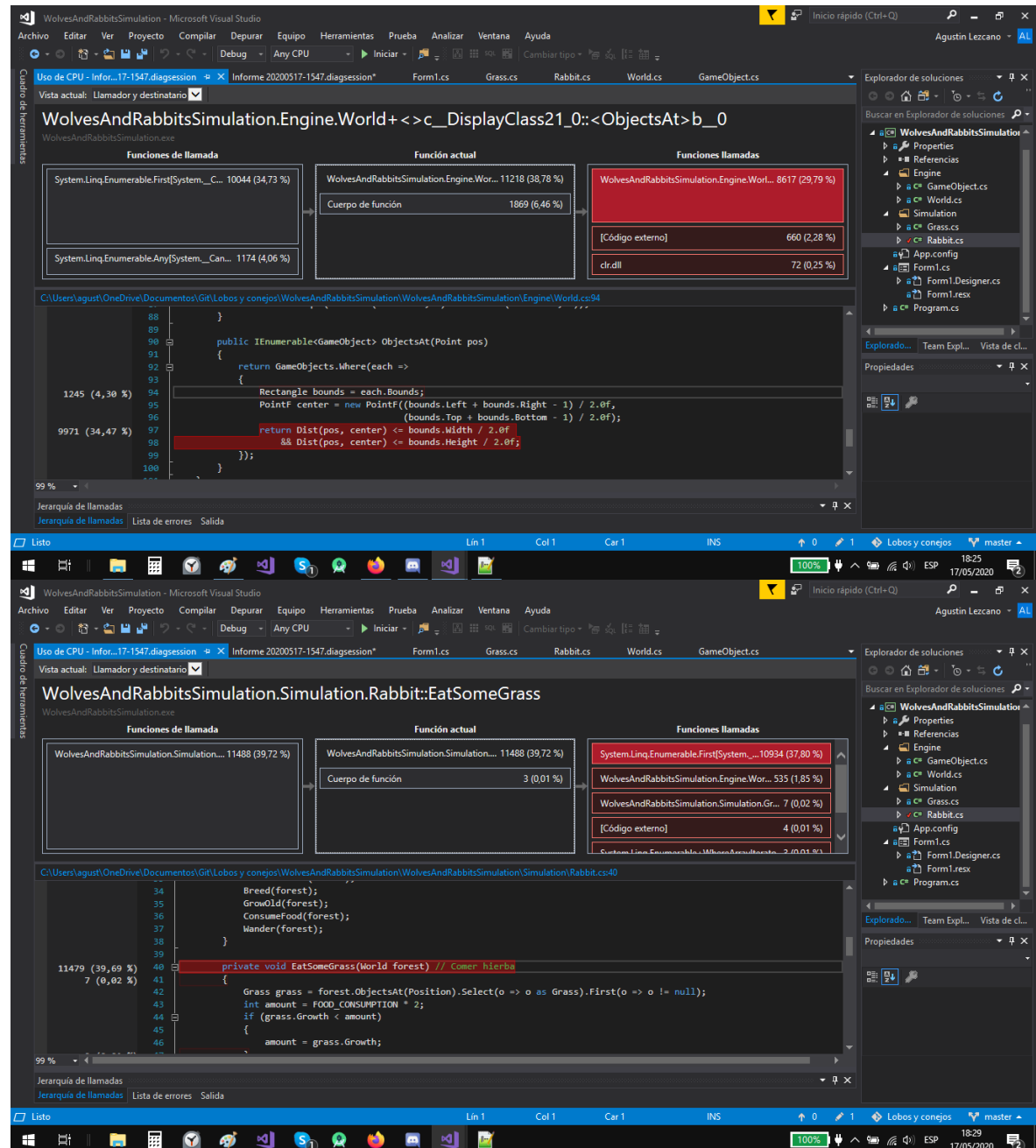


Nº 1 [FALLO]

Problema identificado

Todos los objetos se almacenan en el array "objects"

Medición



Hipótesis

Si en lugar de medir todos los objetos en la pantalla, medimos los que nos interesan (el pasto en este caso) vamos a tener menos elementos que medir.

Descripción de los cambios

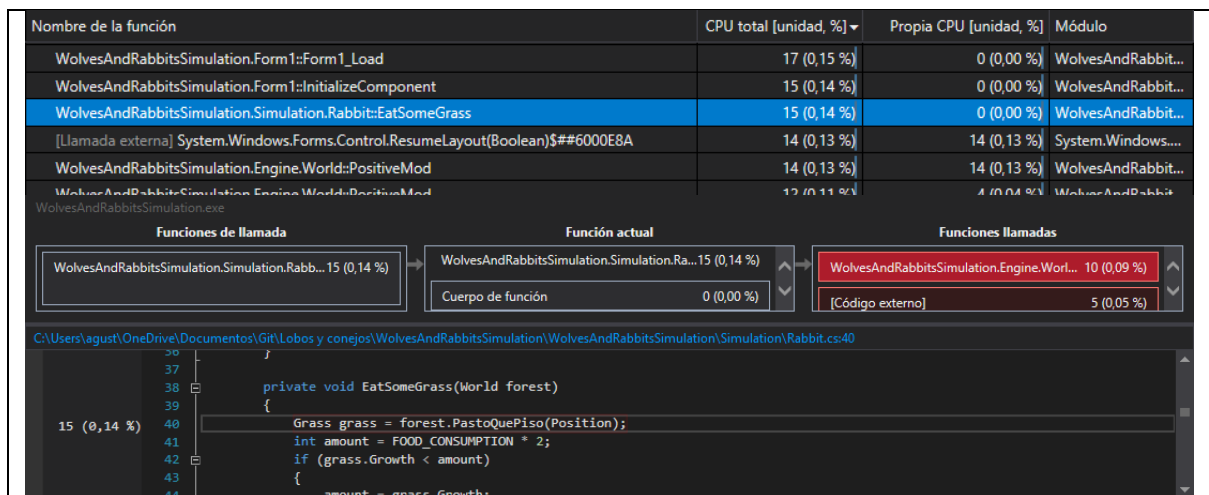
Reemplazo el arreglo "objects" por "conejo" y "pasto" para trabajar con cada uno por separado. Posteriormente cada método en el juego que agrega un elemento al "world" se especifica si se está agregando un "Rabbit" o un "Guess"

Nuevas mediciones

FALLO

Nº 1

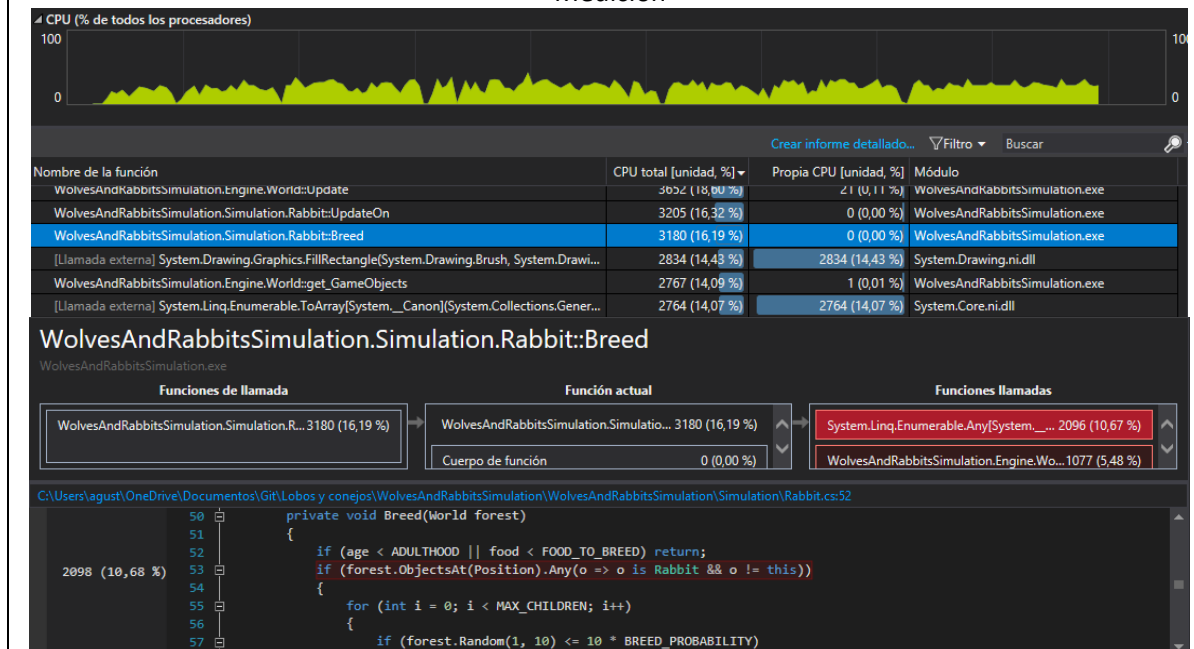
Problema identificado																												
El arreglo es una forma ineficiente de guardar una cantidad no estatica de elementos																												
Medición																												
Se realizo la medicion en este metodo de la clase Form1																												
<pre>private void InitializeWorld() { var reloj = Stopwatch.StartNew(); FillWithGrass(); SpawnSomeRabbits(); reloj.Stop(); using (StreamWriter stream = new StreamWriter("Medicion.txt", true)) { stream.WriteLine(reloj.Elapsed.TotalSeconds); } }</pre>																												
De cuatro mediciones dio un promedio de = 26,326958125 seg.																												
Hipótesis																												
Si cambio el arreglo por una lista se almacenaria de una forma mas agil.																												
Descripción de los cambios																												
Cambio el arreglo por una lista.																												
Hago que los metodos que añaden o sacan cosas de la lista se almacenen en																												
Nuevas mediciones																												
De cuatro mediciones dio un promedio de = 0,212928625 seg																												
Nº 2																												
Problema identificado																												
El conejo tiene que consultar a toda la lista para recibir el pasto en donde esta parado																												
Medición																												
<table><tr><th>Nombre de la función</th><th>CPU total [unidad, %]</th><th>Propia CPU [unidad, %]</th><th>Módulo</th></tr><tr><td>WolvesAndRabbitsSimulation.Form1::Step</td><td>6061 (83,23 %)</td><td>0 (0,00 %)</td><td>WolvesAndRabbit...</td></tr><tr><td>WolvesAndRabbitsSimulation.Engine.World::Update</td><td>4413 (60,60 %)</td><td>2 (0,03 %)</td><td>WolvesAndRabbit...</td></tr><tr><td>WolvesAndRabbitsSimulation.Simulation.Rabbit::UpdateOn</td><td>4357 (59,83 %)</td><td>1 (0,01 %)</td><td>WolvesAndRabbit...</td></tr><tr><td>WolvesAndRabbitsSimulation.Simulation.Rabbit::EatSomeGrass</td><td>4350 (59,74 %)</td><td>1 (0,01 %)</td><td>WolvesAndRabbit...</td></tr><tr><td>WolvesAndRabbitsSimulation.Engine.World::get_GameObjects</td><td>2609 (35,83 %)</td><td>0 (0,00 %)</td><td>WolvesAndRabbit...</td></tr><tr><td>[Llamada externa] System.Linq.Enumerable.ToArray[System._Canon](System.Collections.Gener...</td><td>2608 (35,81 %)</td><td>2608 (35,81 %)</td><td>System.Core.ni.dll</td></tr></table> <div><div>WolvesAndRabbitsSimulation.Simulation.Rabbit::EatSomeGrass</div><div>WolvesAndRabbitsSimulation.exe</div><div><div>Funciones de llamada</div><div>WolvesAndRabbitsSimulation.Simulation.R... 4350 (59,74 %)</div><div>Función actual</div><div>WolvesAndRabbitsSimulation.Simulation... 4350 (59,74 %)</div><div>Cuerpo de función</div><div>1 (0,01 %)</div><div>Funciones llamadas</div><div>WolvesAndRabbitsSimulation.Engine.W... 2382 (32,71 %)</div><div>System.Linq.Enumerable.First[System.... 1962 (26,94 %)</div></div><div>C:\Users\agust\OneDrive\Documentos\Git\Lobos y conejos\WolvesAndRabbitsSimulation\WolvesAndRabbitsSimulation\Simulation\Rabbit.cs:40</div><div><pre>36 } 37 } 38 private void EatSomeGrass(World forest) 39 { 40 Grass grass = forest.ObjectsAt(Position).Select(o => o as Grass).First(o => o != null); 41 int amount = FOOD_CONSUMPTION * 2; 42 if (grass.Growth < amount) 43 { 44 amount = grass.Growth;</pre></div></div>	Nombre de la función	CPU total [unidad, %]	Propia CPU [unidad, %]	Módulo	WolvesAndRabbitsSimulation.Form1::Step	6061 (83,23 %)	0 (0,00 %)	WolvesAndRabbit...	WolvesAndRabbitsSimulation.Engine.World::Update	4413 (60,60 %)	2 (0,03 %)	WolvesAndRabbit...	WolvesAndRabbitsSimulation.Simulation.Rabbit::UpdateOn	4357 (59,83 %)	1 (0,01 %)	WolvesAndRabbit...	WolvesAndRabbitsSimulation.Simulation.Rabbit::EatSomeGrass	4350 (59,74 %)	1 (0,01 %)	WolvesAndRabbit...	WolvesAndRabbitsSimulation.Engine.World::get_GameObjects	2609 (35,83 %)	0 (0,00 %)	WolvesAndRabbit...	[Llamada externa] System.Linq.Enumerable.ToArray[System._Canon](System.Collections.Gener...	2608 (35,81 %)	2608 (35,81 %)	System.Core.ni.dll
Nombre de la función	CPU total [unidad, %]	Propia CPU [unidad, %]	Módulo																									
WolvesAndRabbitsSimulation.Form1::Step	6061 (83,23 %)	0 (0,00 %)	WolvesAndRabbit...																									
WolvesAndRabbitsSimulation.Engine.World::Update	4413 (60,60 %)	2 (0,03 %)	WolvesAndRabbit...																									
WolvesAndRabbitsSimulation.Simulation.Rabbit::UpdateOn	4357 (59,83 %)	1 (0,01 %)	WolvesAndRabbit...																									
WolvesAndRabbitsSimulation.Simulation.Rabbit::EatSomeGrass	4350 (59,74 %)	1 (0,01 %)	WolvesAndRabbit...																									
WolvesAndRabbitsSimulation.Engine.World::get_GameObjects	2609 (35,83 %)	0 (0,00 %)	WolvesAndRabbit...																									
[Llamada externa] System.Linq.Enumerable.ToArray[System._Canon](System.Collections.Gener...	2608 (35,81 %)	2608 (35,81 %)	System.Core.ni.dll																									
Hipótesis																												
Si usando solamente su ubicación consultamos a un rango muy pequeño de “pastos” la busqueda seria mas agil.																												
Descripción de los cambios																												
Cree un nuevo metodo llamado “PastoQuePiso” que devuelve un “grass”																												
Nuevas mediciones																												



Nº 3

Problema identificado
El conejo tarda mucho en encontrar otro conejo

Medición



Hipótesis

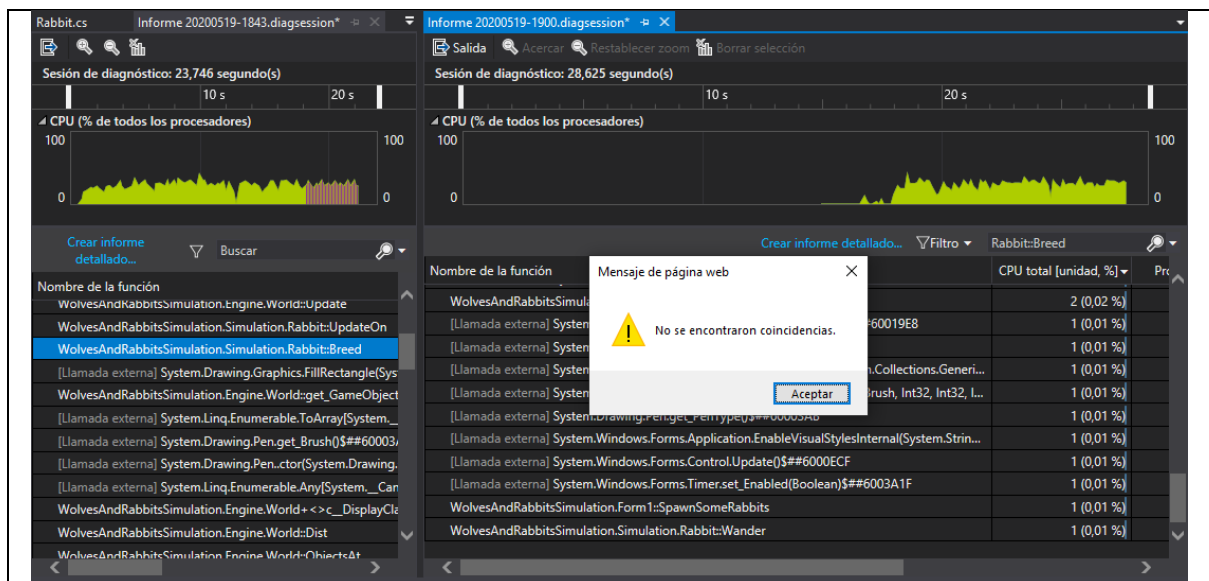
Si en lugar de recorrer una lista con todos los elementos, recorremos la lista de los conejos, tendrán menos elementos que recorrer

Descripción de los cambios

Hago que en lugar de llamar al método "ObjectsAt" llame a un nuevo método que devuelva un booleano dependiendo si hay algún conejo en su coordenada.

Nuevas mediciones

Aparentemente la optimización fue tan efectiva que desapareció de los registros



N° 4

Problema identificado	Lentitud en el dibujado
-----------------------	-------------------------

Medición

De cuatro mediciones dio un promedio de = 1,3362159 seg.

Hipótesis

Psupongo que el problema es que se tienen que crear un nuevo rectangle, brush y pen por cada elemento en la pantalla.

Descripción de los cambios

Cambie el contenido del método "DrawOn"

```

Brush RabbitBrush = new Pen(Color.White).Brush;
Brush[] Degradado = new Brush[256];
Rectangle RabbitRectangle = new Rectangle(0,0,1,1);
Rectangle GrassRectangle = new Rectangle(0, 0, Grass.PATCH_SIZE, Grass.PATCH_SIZE);

public virtual void DrawOn(Graphics graphics)
{
    for (int i = 0; i < Conejera.Count - 1; i++)
    {
        RabbitRectangle.Location = Conejera[i].Position;
        graphics.FillRectangle(RabbitBrush, RabbitRectangle);
    }
    for (int i = 0; i < Garden.Length - 1; i++)
    {
        GrassRectangle.Location = Garden[i].Position;
        graphics.FillRectangle(Degradado[Garden[i].Growth], GrassRectangle);
    }
}

```

Nuevas mediciones

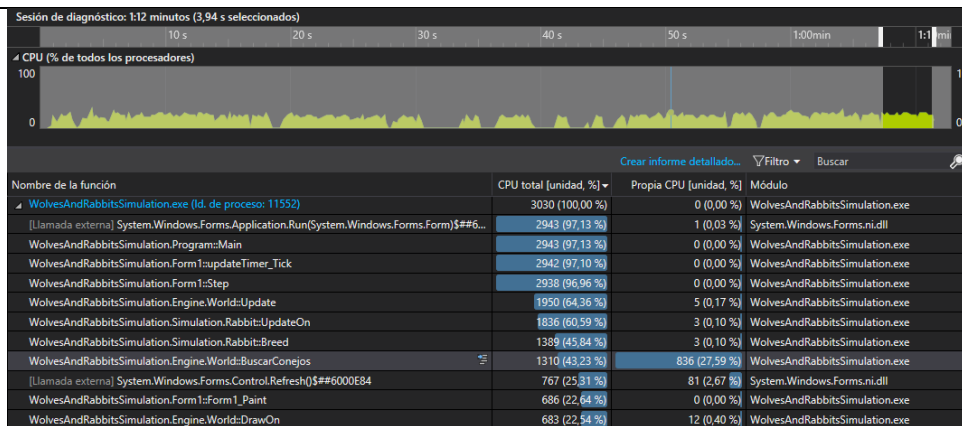
De cuatro mediciones dio un promedio de = 0,0322854 seg.

N° 5

Problema identificado

Los conejos se reproducen mucho

Medición



Hipótesis

El programa busca a un conejo, pero no considera su edad, para solucionarlo la edad de los conejos debe ser publica, y el programa debe tomar en cuenta la edad del localizado.

Descripción de los cambios

En rabbit:

```
private int age = 0;
private int food = 500;

public int Edad
{
    get { return age; }
}
```

En World:

```
internal bool BuscarConejos(Point position, int ADULTHOOD)
{
    byte contador = 0;
    foreach (Rabbit item in Conejera)
    {
        if (item.Position == position && item.Edad > ADULTHOOD)
            contador++;
    }
    return contador > 1;
}
```

Nuevas mediciones

