

Nº 1

Identificar problemas

Me parece muy rara e ineficiente la forma de llamar el método.

```
public static Image[] Load(string fileName, Size size)
{
    return new Spritesheet(fileName).CutPieces(size);
}

public static Image[] Load(Image original, Size size)
{
    return new Spritesheet(original).CutPieces(size);
}
```

Mediciones

| Linea | Time (ms) | % |
|-------|-----------|-----------|
| 183 | | |
| 184 | | |
| 185 | 28700 | (41,76 %) |
| 186 | 1708 | (2,49 %) |
| 187 | | |
| 188 | 258 | (0,38 %) |
| 189 | | |
| 190 | | |
| 191 | 35 | (0,05 %) |
| 192 | 7 | (0,01 %) |
| 193 | | |

```
private Image LoadImage()
{
    Image[] ships = Spritesheet.Load(@"Resources\shipsheetparts.png", new Size(200, 200));
    foreach (Image img in ships)
    {
        img.RotateFlip(RotateFlipType.Rotate270FlipNone);
    }
    Image result = ships[shipIndex];
    result.RotateFlip(RotateFlipType.RotateNoneFlipX);
    Extent = new.SizeF(result.Size.Width / 2, result.Size.Height / 2);
    return result;
}
```

Hipotesis

Cambio el crear una nueva clase para llamar a un método por convertir al método en estático.

Descripcion de los cambios realizados

Cambio el “new Spritesheet(fileName).CutPieces(size)” por “CutPieces(size)” en los dos métodos “Load”

Elimino los constructores

Convierto el método “CutPieces” en estático y privado

Nuevas mediciones

| Linea | Time (ms) | % |
|-------|-----------|-----------|
| 183 | | |
| 184 | | |
| 185 | 40615 | (40,67 %) |
| 186 | 2719 | (2,72 %) |
| 187 | | |
| 188 | | |
| 189 | | |
| 190 | | |
| 191 | 36 | (0,04 %) |
| 192 | 1 | (0,00 %) |
| 193 | | |

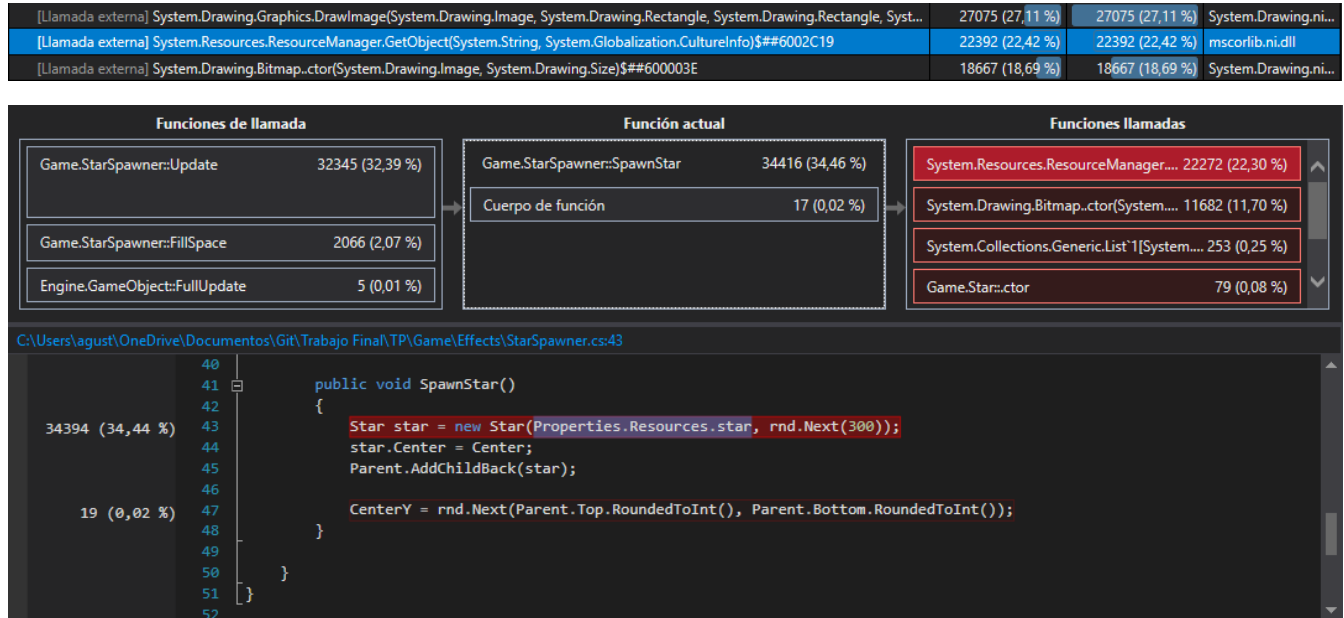
```
private Image LoadImage()
{
    Image[] ships = Spritesheet.Load(@"Resources\shipsheetparts.png", new Size(200, 200));
    foreach (Image img in ships)
    {
        img.RotateFlip(RotateFlipType.Rotate270FlipNone);
    }
    Image result = ships[shipIndex];
    result.RotateFlip(RotateFlipType.RotateNoneFlipX);
    Extent = new.SizeF(result.Size.Width / 2, result.Size.Height / 2);
    return result;
}
```

Nº 2

Identificar problemas

La clase “Start” tarda mucho en ser dibujada.

Mediciones



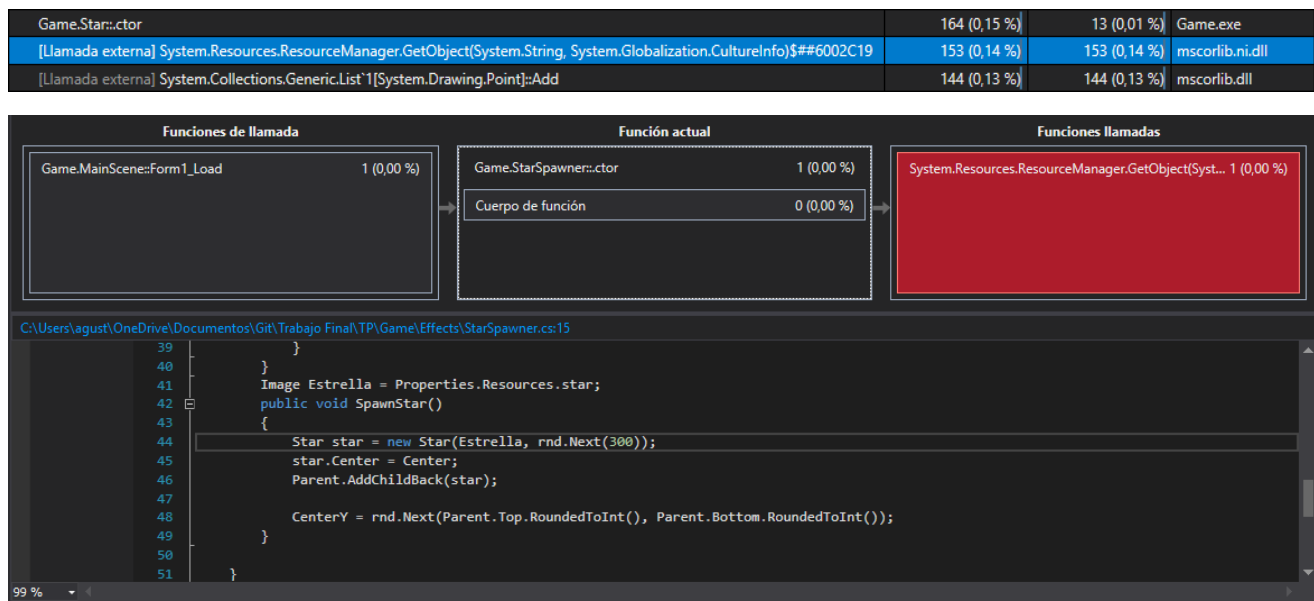
Hipotesis

Si hago que se guarde la imagen de la estrella, no tiene porque volver a buscarla cada vez que se crea una nueva estrella.

Descripcion de los cambios realizados

```
Image Estrella = Properties.Resources.star;
public void SpawnStar()
{
    Star star = new Star(Estrella, rnd.Next(300));
}
```

Nuevas mediciones



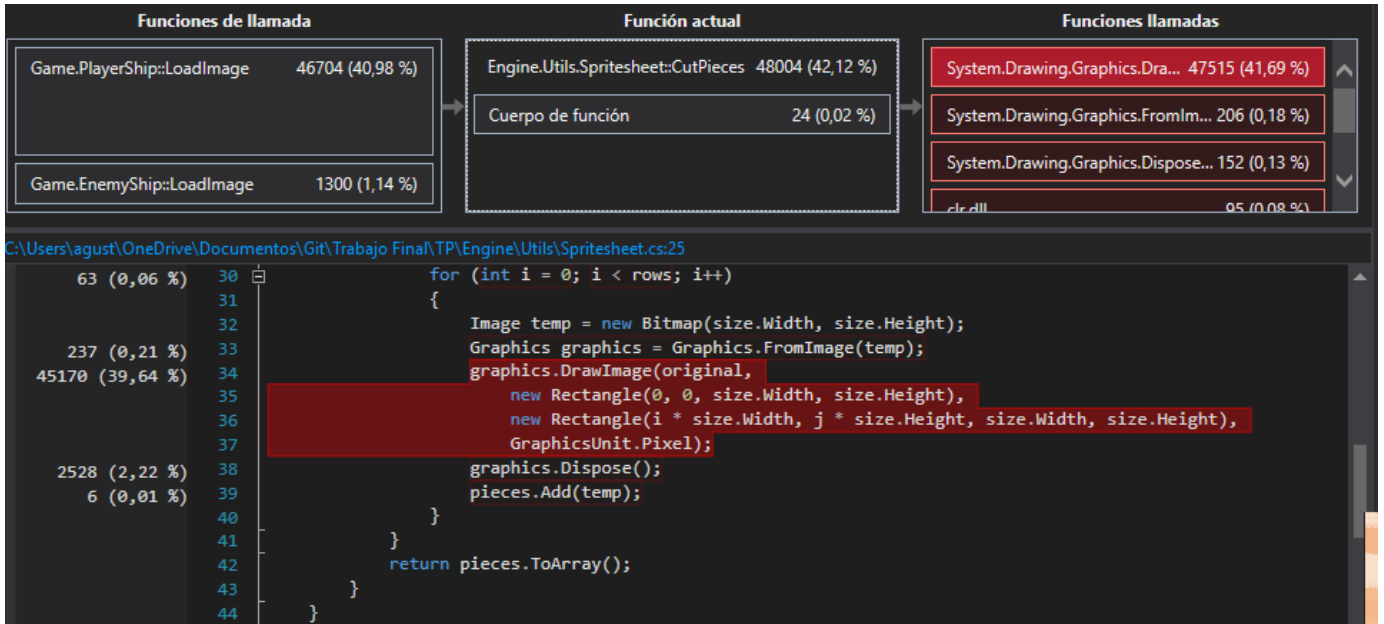
Nº 3

Identificar problemas

Lentitud en el corte

Mediciones

| | | | |
|---|-----------------|-----------------|----------------------|
| Engine.Utils.SpriteSheet::CutPieces | 48004 (42,12 %) | 24 (0,02 %) | Engine.dll |
| [Llamada externa] System.Drawing.Graphics.DrawImage(System.Drawing.Image, System.Drawing.Rectangle, System.Drawing... | 47516 (41,69 %) | 47516 (41,69 %) | System.Drawing.ni... |
| Game.PlayerShip::CheckForPowerUps | 38422 (33,72 %) | 0 (0,00 %) | Game.exe |



Hipótesis

Es una forma ineficiente de recortar una imagen

Descripción de los cambios realizados

Moví la instrucción de la línea 32 afuera de las iteraciones for.

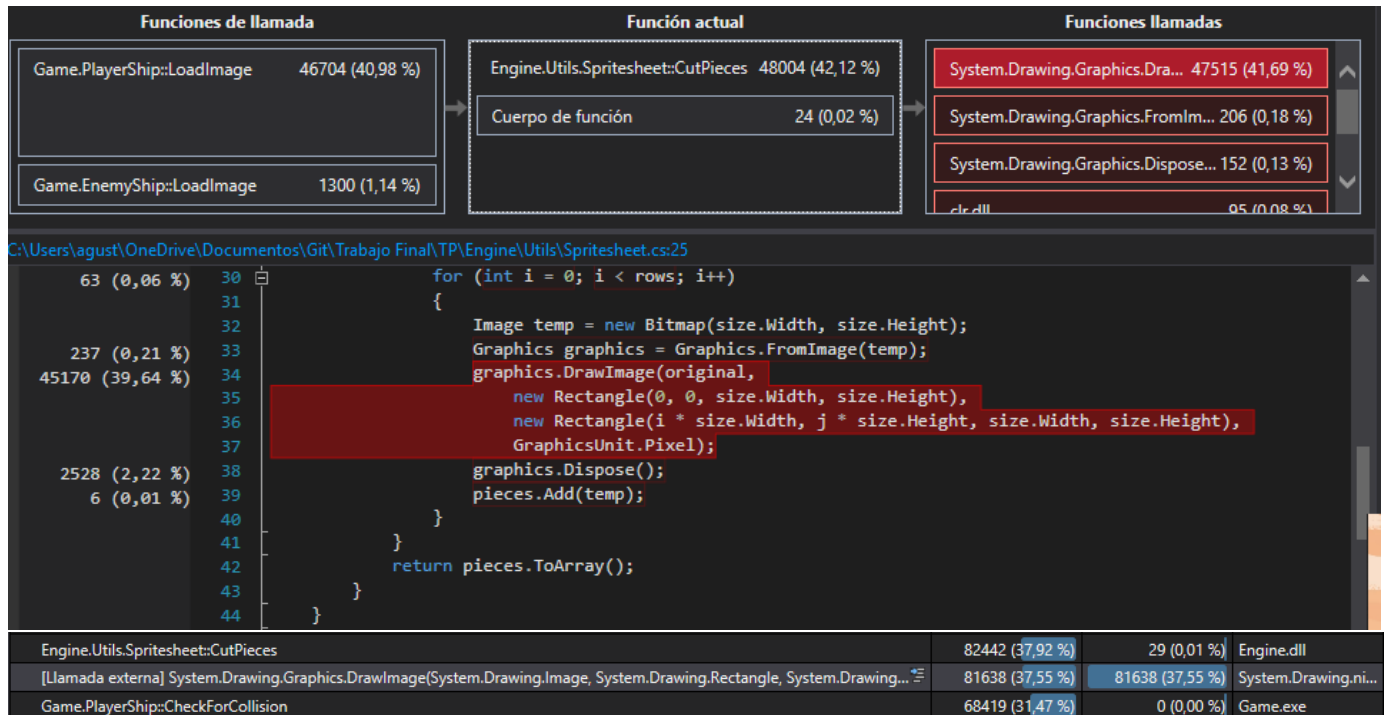
Nuevas mediciones

Nº 3

Identificar problemas

“EnemyShip” y “PlayerShip” provocan que “Spritesheet” funcione lento.

Mediciones



Hipótesis

Si hago que “EnemyShip” y “PlayerShip” guarden sus imágenes y no las vuelvan a pedir, eso liberaría al “Spritesheet” de una buena parte de la carga.

Descripción de los cambios realizados

En las dos clases (“EnemyShip” y “PlayerShip”) moví las instrucciones en donde conseguían su Sprite.

Nuevas mediciones

| | | | |
|--|---------------|---------------|--------------------|
| Engine.Utils.SpriteSheet::CutPieces | 5408 (7,74 %) | 6 (0,01 %) | Engine.dll |
| [Llamada externa] System.Drawing.Graphics.DrawImage(System.Drawing.Image, System.Drawing.Rectangle, System.Drawing.Point)##600027F | 5350 (7,66 %) | 5350 (7,66 %) | System.Drawing.dll |
| [Llamada externa] System.Drawing.Graphics.DrawImage(System.Drawing.Image, System.Drawing.Point)##600027F | 5226 (7,48 %) | 5226 (7,48 %) | System.Drawing.dll |

Engine.Utils.SpriteSheet::CutPieces

Engine.dll

| Funciones de llamada | Función actual | Funciones llamadas |
|---|--|---|
| <div>Game.EnemyShip::ctor5246 (7,51 %)</div> <div>Game.PlayerShip::ctor162 (0,23 %)</div> | <div>Engine.Utils.SpriteSheet::CutPieces5408 (7,74 %)</div> <div>Cuerpo de función6 (0,01 %)</div> | <div>System.Drawing.Graphics.DrawImage(System.Drawing.Image, System.Drawing.Rectangle, System.Drawing.Point)##600027F5350 (7,66 %)</div> <div>System.Drawing.Graphics.FromImage(System.Drawing.Image)25 (0,04 %)</div> <div>clr.dll15 (0,02 %)</div> <div>System.Drawing.Graphics.Dispose()##600027F11 (0,02 %)</div> <div>System.Drawing.Graphics.CheckErrorStatus(System.Drawing.GraphicsException)1 (0,00 %)</div> |

C:\Users\agust\OneDrive\Documentos\Git\Trabajo Final\TP\Engine\Utils\SpriteSheet.cs:25

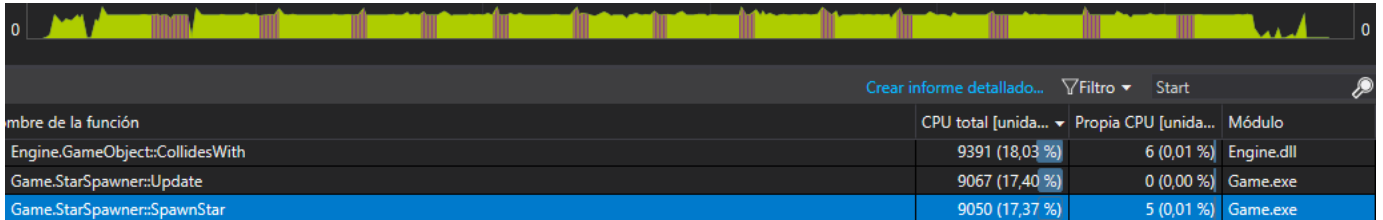
| | | |
|---------------|----|--|
| 9 (0,01 %) | 29 | { |
| | 30 | for (int i = 0; i < rows; i++) |
| | 31 | { |
| 31 (0,04 %) | 32 | Image temp = new Bitmap(size.Width, size.Height); |
| 5342 (7,65 %) | 33 | Graphics graphics = Graphics.FromImage(temp); |
| | 34 | graphics.DrawImage(original, |
| | 35 | new Rectangle(0, 0, size.Width, size.Height), |
| | 36 | new Rectangle(i * size.Width, j * size.Height, size.Width, size.Height), |
| | 37 | GraphicsUnit.Pixel); |
| 26 (0,04 %) | 38 | graphics.Dispose(); |
| | 39 | pieces.Add(temp); |
| | 40 | } |
| | 41 | } |
| | 42 | return pieces.ToArray(); |

Nº 4

Identificar problemas

El spawn de la estrella ralentiza el programa

Mediciones



Hipótesis

Si creo una estrella generica afuera del metodo, y hago que la velocidad se calcule dentro de la estrella aceleraria significativamente el dibujado.

Descripción de los cambios realizados

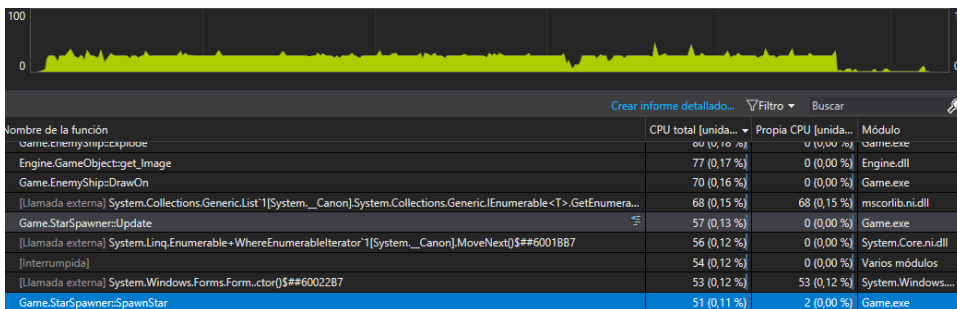
StarSpawner:

```
Star star = new Star(Properties.Resources.star);  
public void SpawnStar()  
{  
    star.Center = Center;  
    Parent.AddChildBack(star);  
}
```

Star:

Elimino el parámetro “float speed” y hago que el valor de la variable “speed” sea aleatorio.

Nuevas mediciones

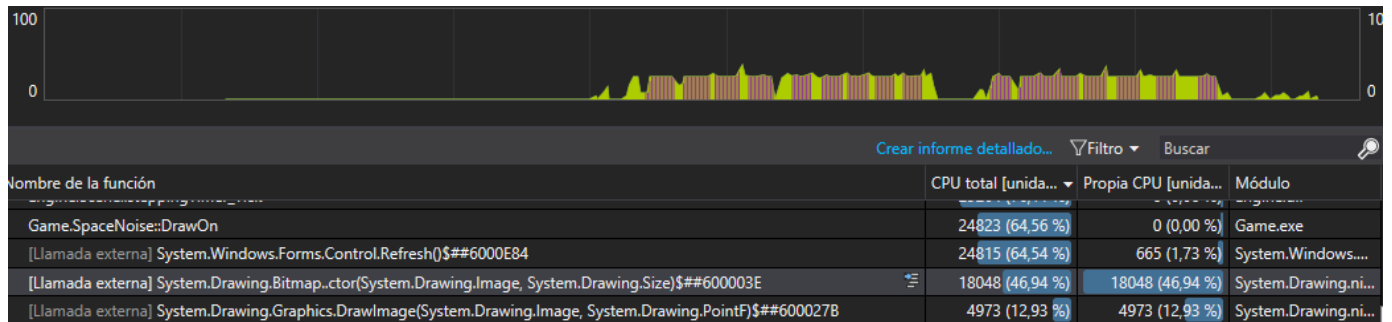


Nº 5

Identificar problemas

El metodo "FillScreenTiled" responde muy lentamente

Mediciones



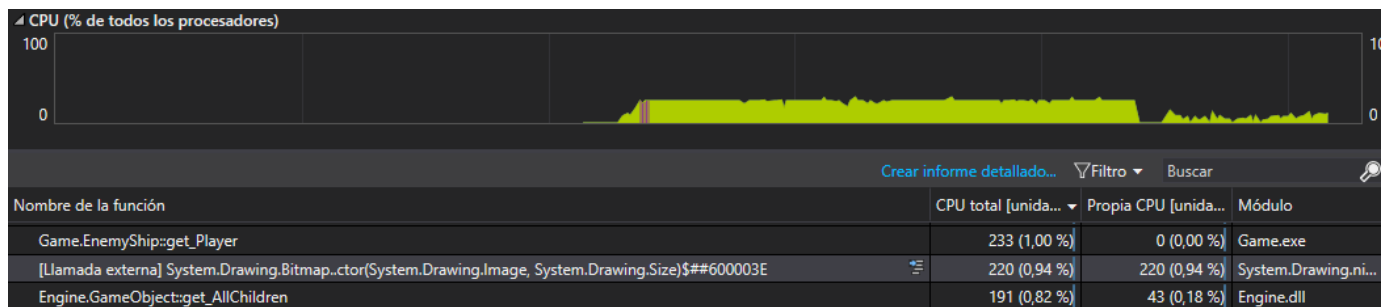
Hipótesis

Si la imagen en lugar de crearse reiteradas veces en el ciclo for, se crea en el constructor, y una vez creados, se llama todas las veces que sea necesarios, se ahorraría mucha performance.

Descripción de los cambios realizados

Muevo las lineas de las posiciones 67, 68 y 69 al constructor.

Nuevas mediciones

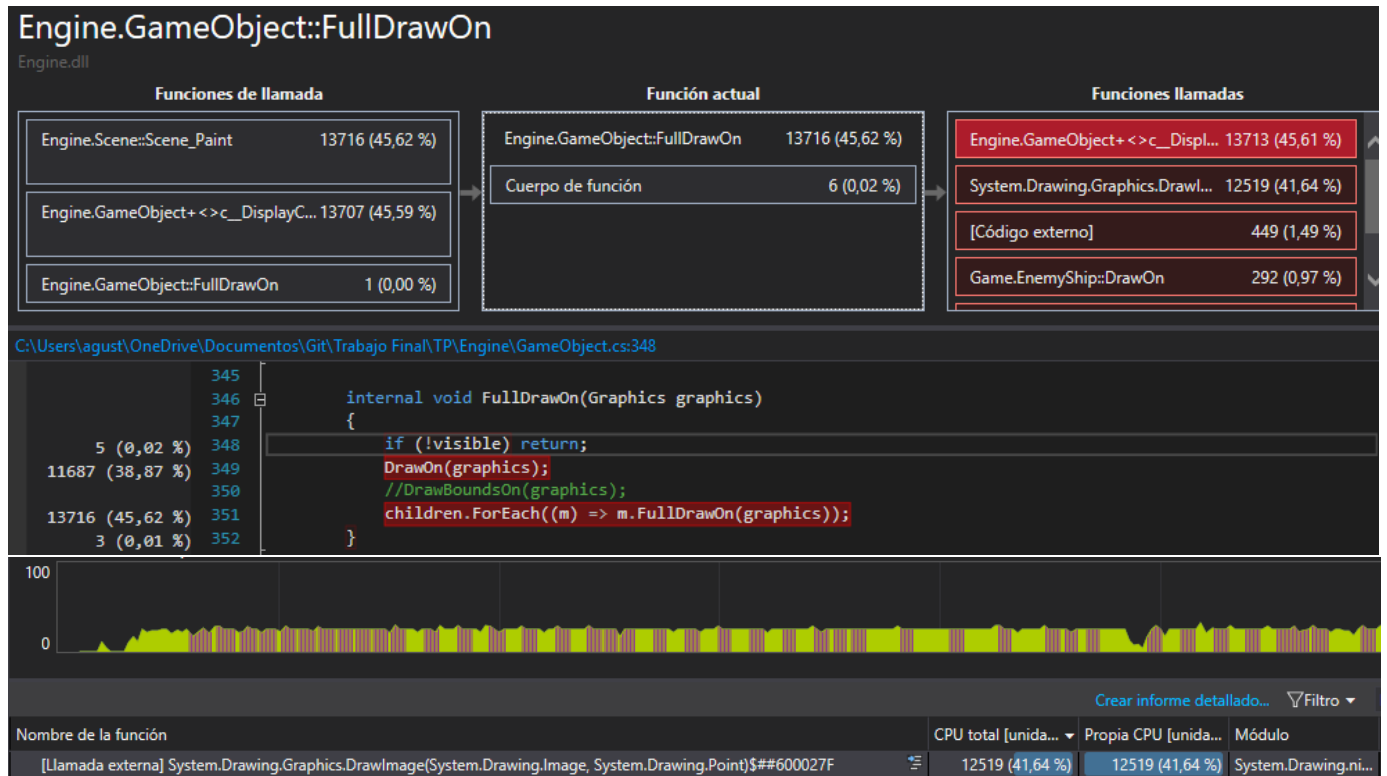


Nº 6

Identificar problemas

Lentitud en el dibujado.

Mediciones



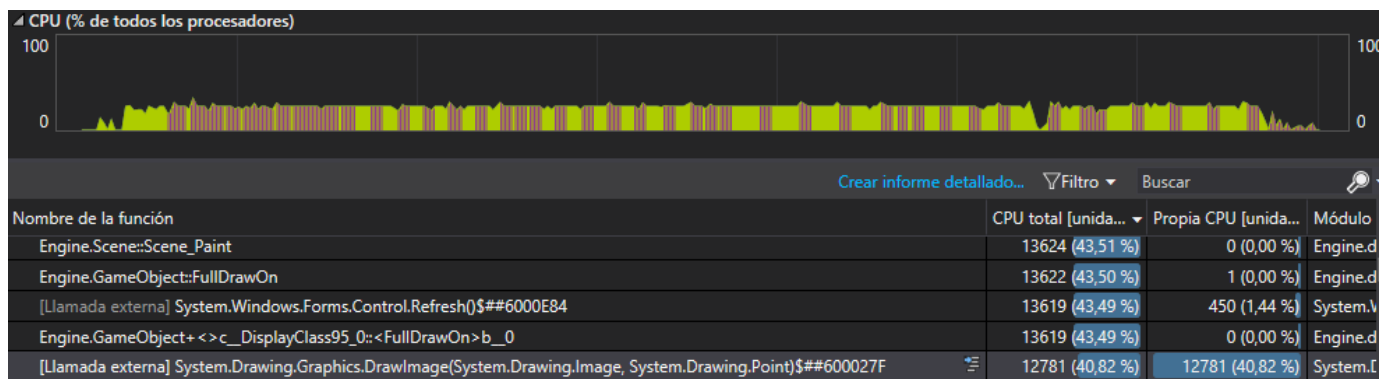
Hipótesis

Tengo la teoría de que estan dibujando a todos los objetos (incluidos los que no estan en la pantalla)

Descripción de los cambios realizados

Si pongo un condicional antes de dibujar y llamar a los hijos de que deban estar en el rango de la pantalla solamente dibujaria lo que el jugador vea y no gastaria recursos en los que estan fuera de pantalla

Nuevas mediciones

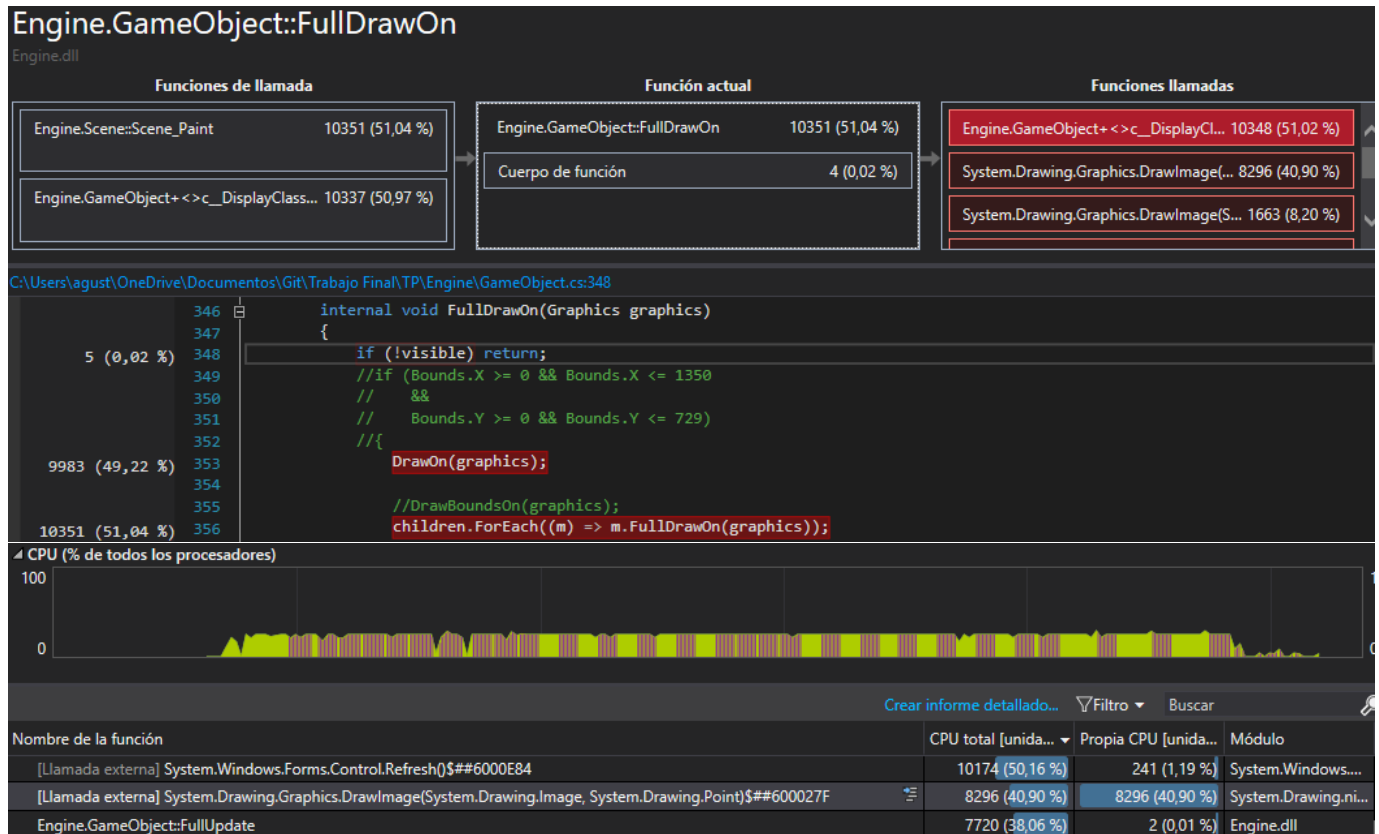


Nº 6

Identificar problemas

Lentitud en el dibujado

Mediciones



Hipótesis

El "Space noise" consume muchos recursos y como no interactúa, ni interfiere con el jugador, ni con los enemigos, si creo un thread en donde se dibuje de fondo, se agilizaría el cálculo del dibujado principal

Descripción de los cambios realizados

El graphics no se comparte

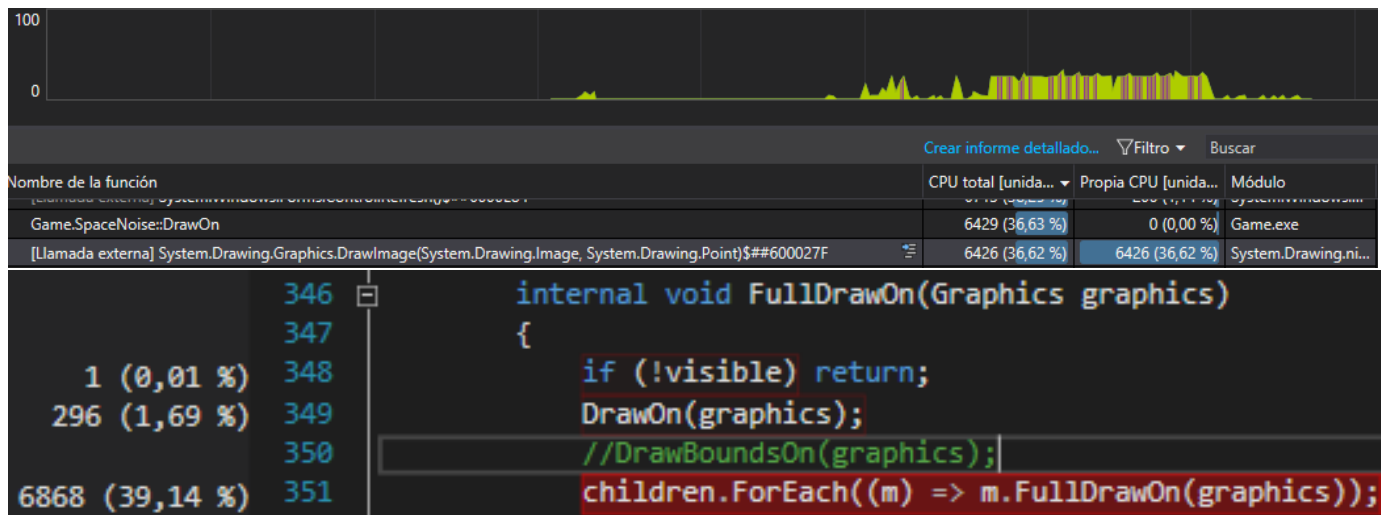
Nuevas mediciones

Nº 6

Identificar problemas

El space noise consume mucho proceso

Mediciones



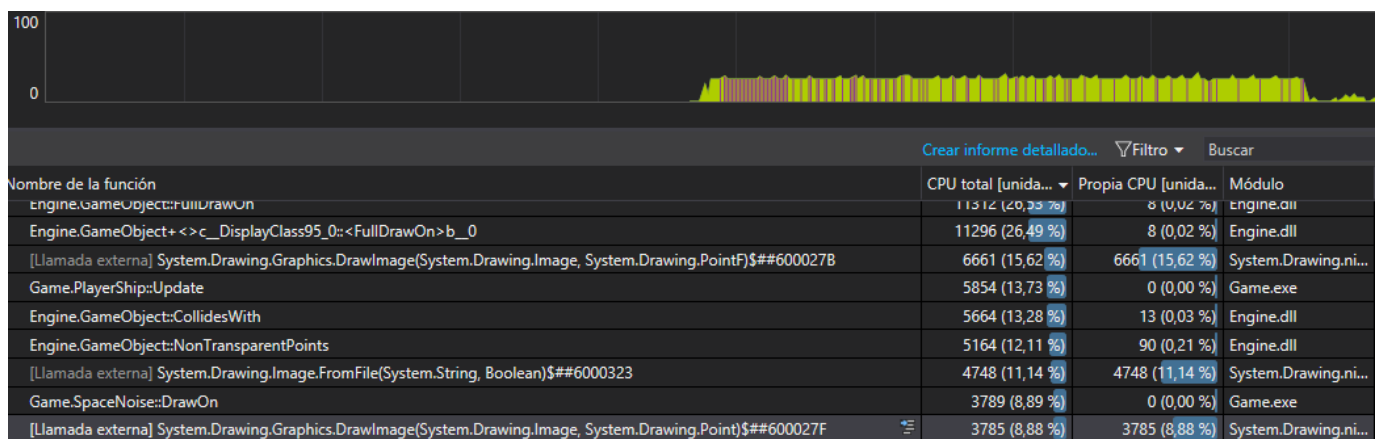
Hipótesis

El space noise esta dibujandose 4 veces, una forma de agilizar el dibujado es juntar los dibujos en uno, y reproducir un solo space noise.

Descripción de los cambios realizados

Convierte las 3 imágenes en una sola imagen, para únicamente tener que dibujar un sólo space noise.

Nuevas mediciones

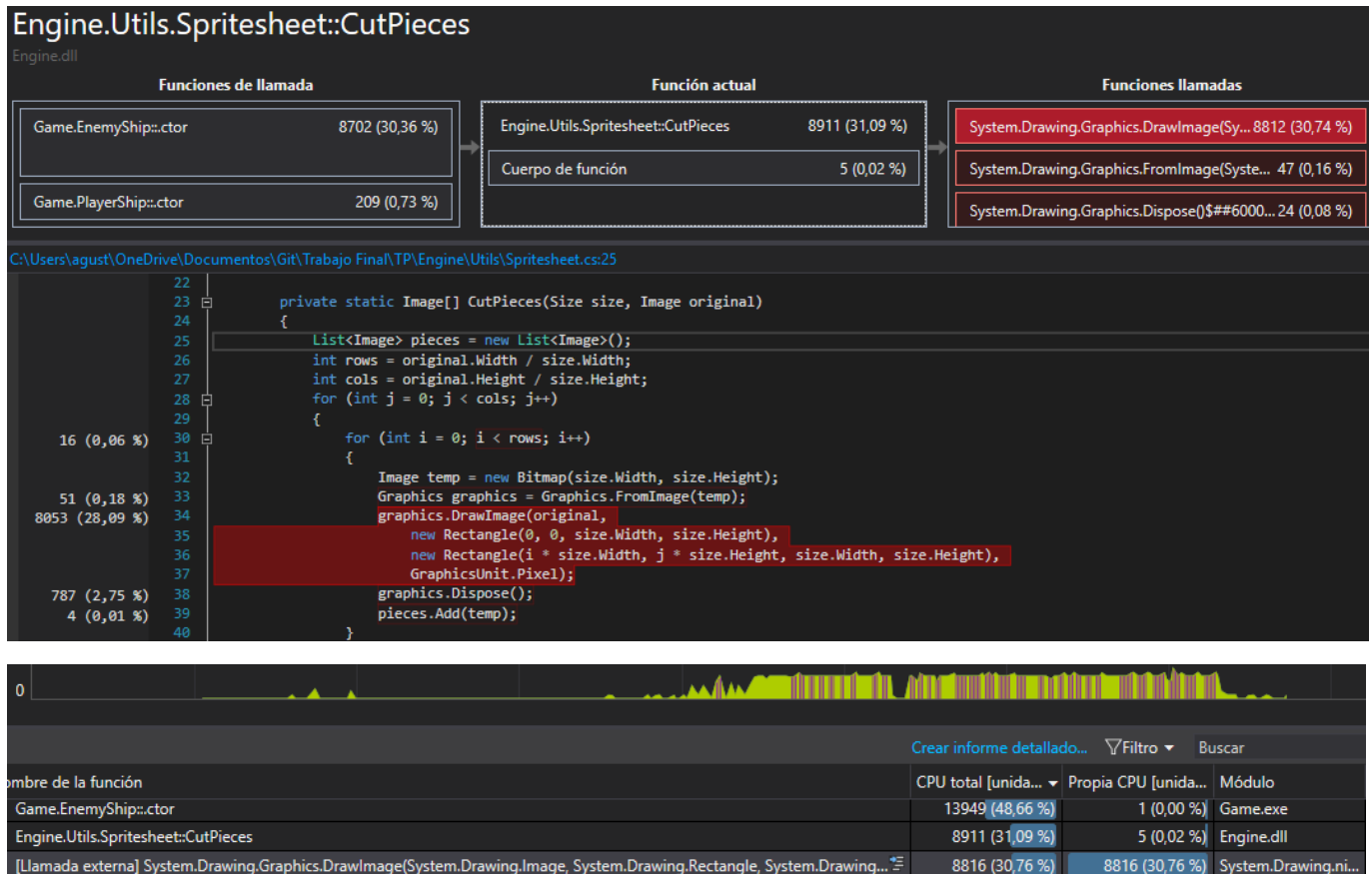


Nº 7

Identificar problemas

El "Cut pieces" esta enlenteciendo el dibujado

Mediciones



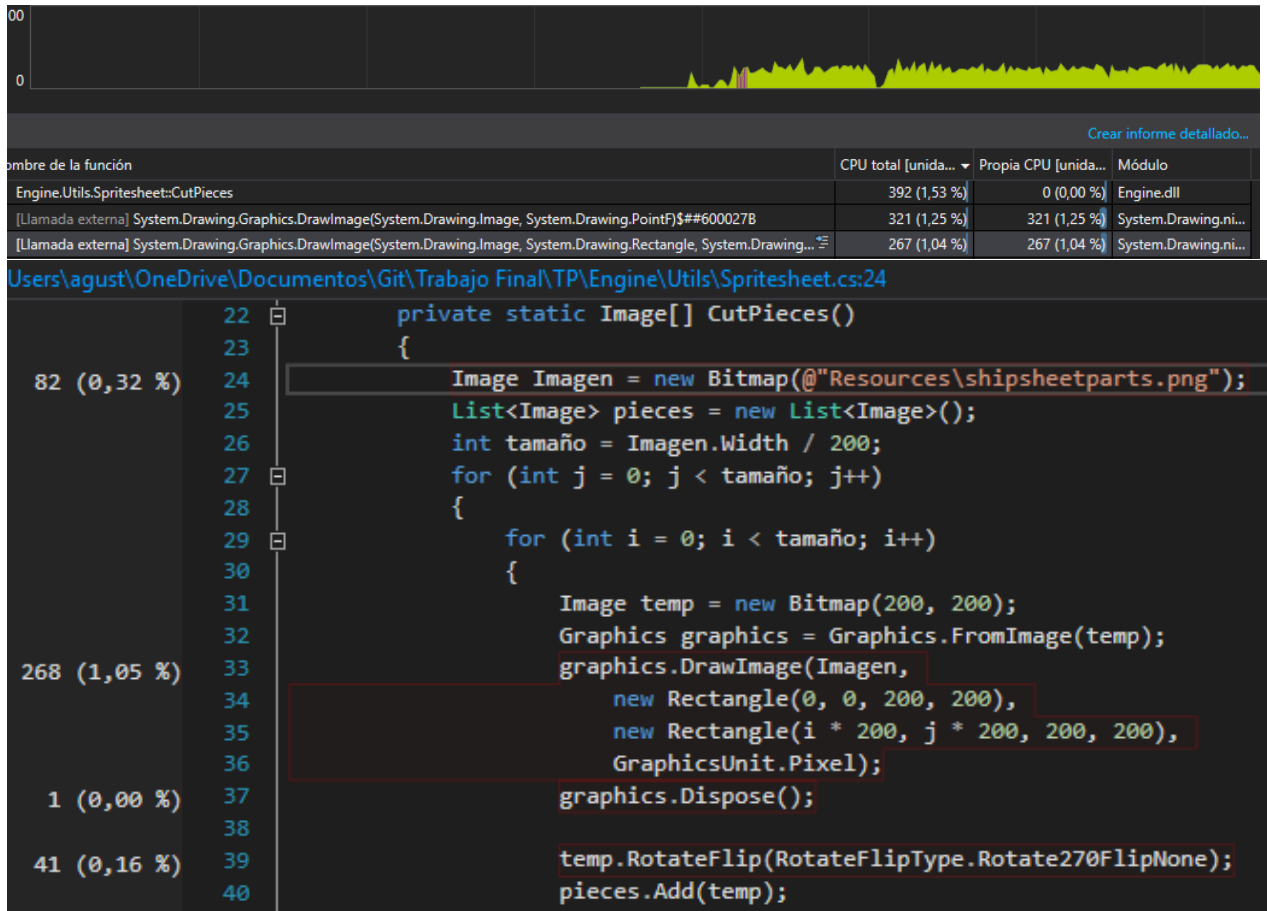
Hipótesis

El problema es que spritesheet tiene que recorrer la misma instrucción Para devolver él mismo a array constantemente, con la mismas medidas e imágenes a recortar. En lugar de pedirle las medidas y la imagen que se tiene que recortar (ya que siempre piden las mismas) la clase les pide a las naves el índice que buscan, y solamente hace el proceso de crear la matriz una vez, agilizaría mucho el proceso.

Descripción de los cambios realizados

Dentro de la clase SpriteSheet añado una variable estática llamada imágenes que es un arreglo de imágenes. Elimino la sobrecarga del load, y a ese método le cambio la firma por un int para que las naves enemigas y la nave del jugador le pase el índice de la nave que buscan. Y para finalizar hago que el CutPieces asuma que la imagen que se quiere cortar sea shipsheetparts y que El tamaño sea de 200.

Nuevas mediciones

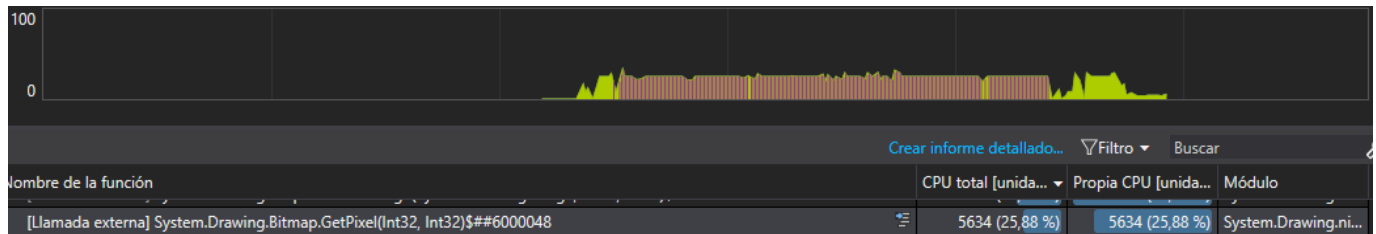


Nº 8

Identificar problemas

Deteccion de colisiones

Mediciones



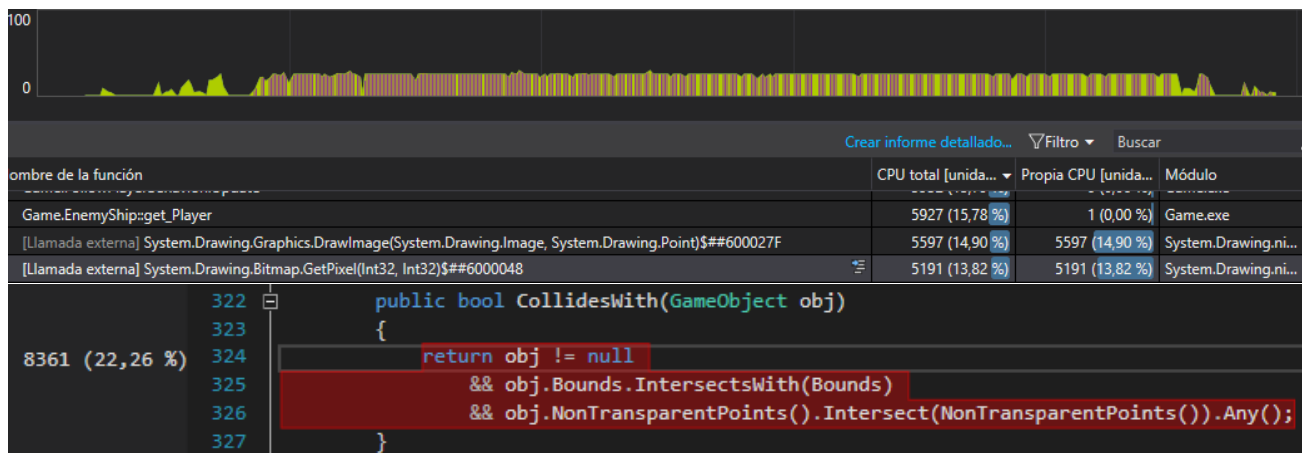
Hipótesis

Creo que las naves y los proyectiles pierden el tiempo teniendo al fondo como parte de los objetos que investigan si estan colisionando, si se dibujan en otra lista donde esten los objetos que no nesecitan deteccion de colisiones (como las estrellas y el espacio) solamente se fijarian en las otras naves

Descripción de los cambios realizados

Creo una nueva clase llamada fondo en la clase "Scene" y "MainScene", en el mainscene le cargo los objetos "space noise" y "star" y en la scene llamo a que se actualizen y se dibujen.

Nuevas mediciones

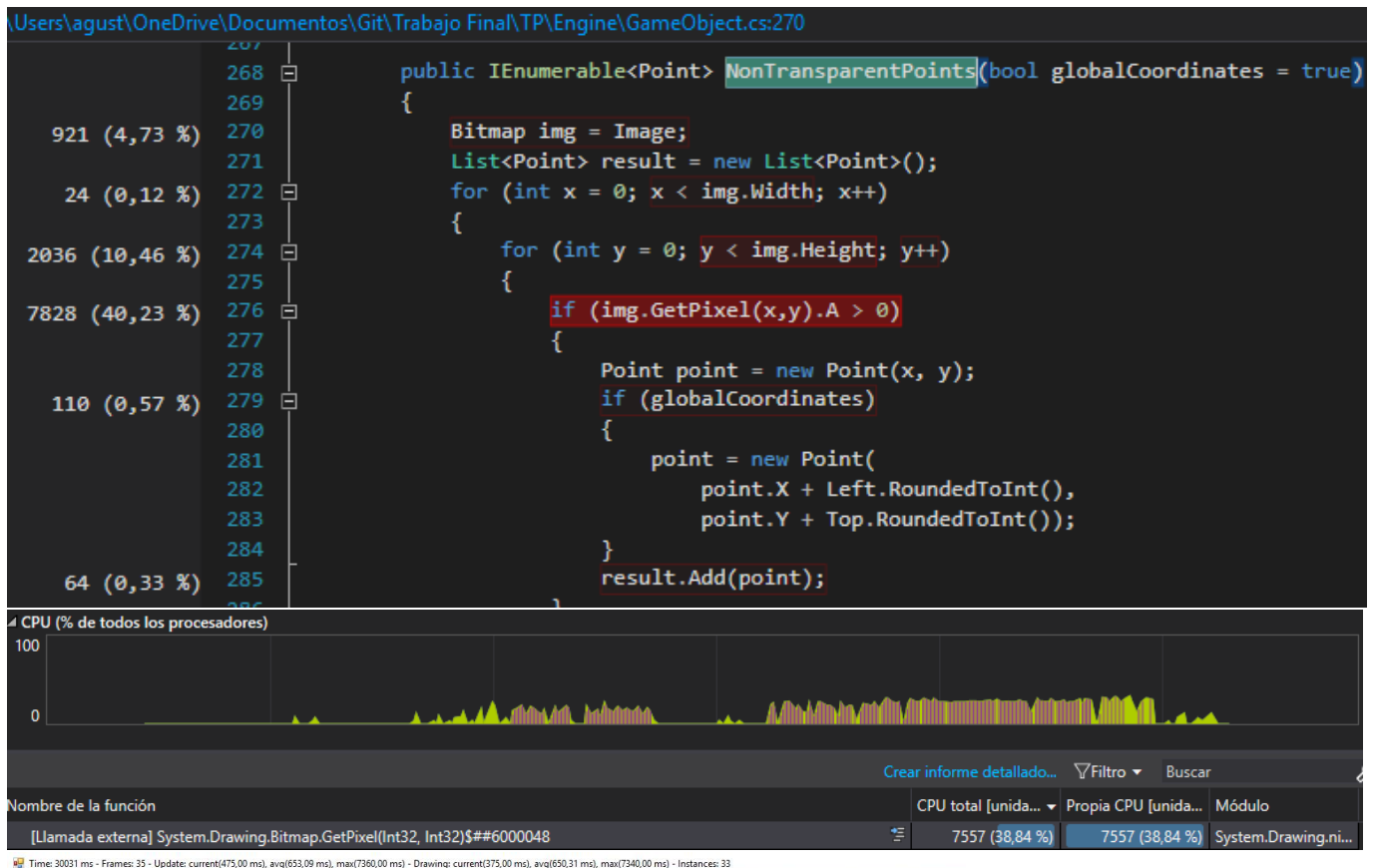


Nº 9

Identificar problemas

El NonTransparentPoints es un poco inútil

Mediciones



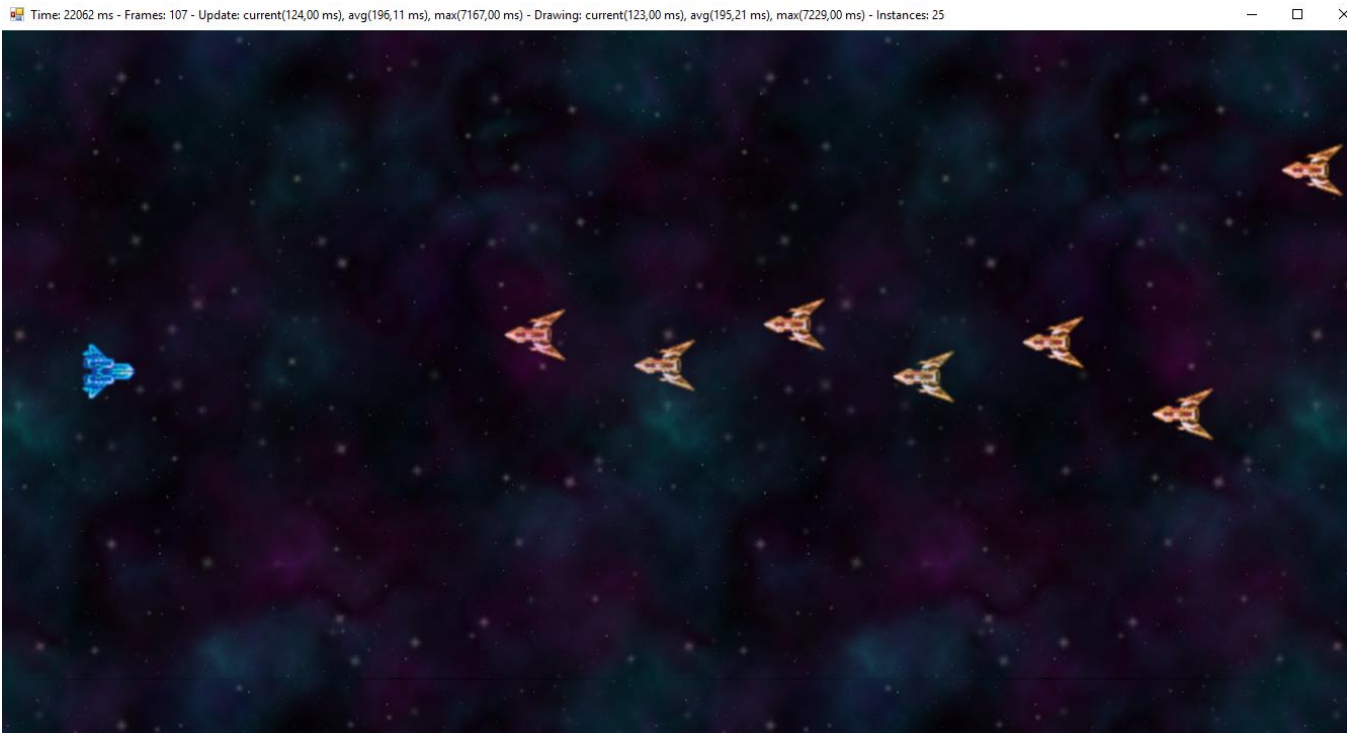
Hipótesis

Si lo elimino estaria acelerando considerablemente el juego

Descripción de los cambios realizados

Elimino la clase NonTransparentPoints y todas las clases que la llaman.

Nuevas mediciones

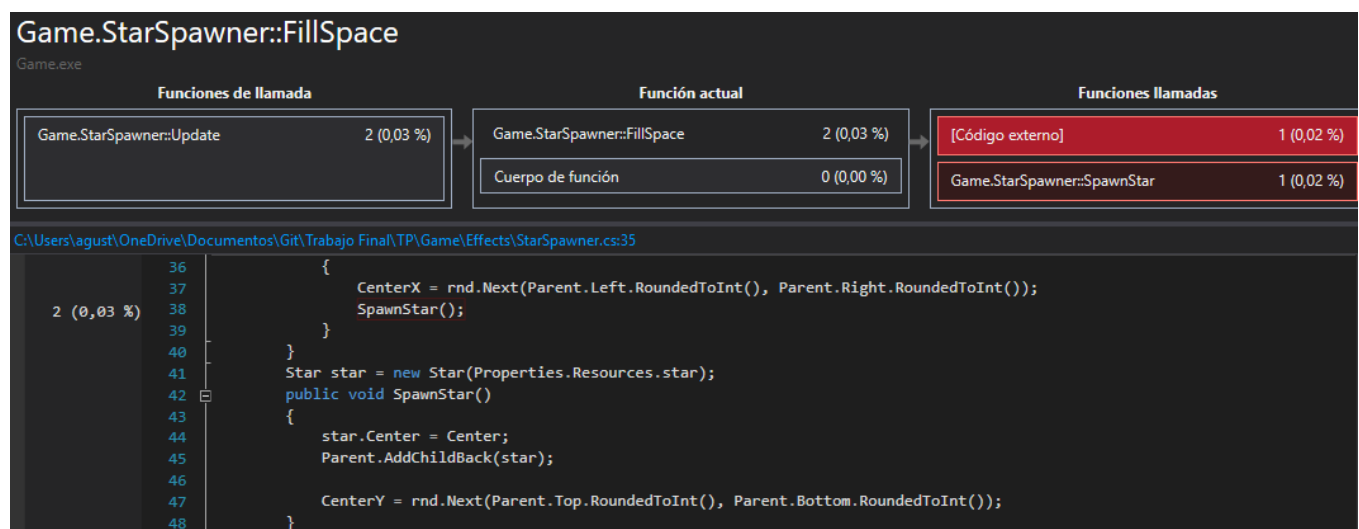


N° 10

Identificar problemas

Bug de las estrellas

Mediciones



Hipótesis

Me di cuenta un poco tarde de que las estrellas no se dibujaban.

Descripción de los cambios realizados

Creo las estrellas en el SpawnStar porque si la creo fuera del metodo no se dibuja, y hago que en lugar de pedir siempre al padre por su tamaño, creo una variable interna con el tamaño del padre.

Nuevas mediciones

Game.StarSpawner::FillSpace

Game.exe

| Funciones de llamada | Función actual | Funciones llamadas |
|------------------------------------|--|---|
| Game.StarSpawner::Update2 (0,03 %) | Game.StarSpawner::FillSpace2 (0,03 %) Cuerpo de función0 (0,00 %) | [Código externo]1 (0,02 %) Game.StarSpawner::SpawnStar1 (0,02 %) |

C:\Users\agust\OneDrive\Documentos\Git\Trabajo Final\TP\Game\Effects\StarSpawner.cs:35

| | |
|------------|---|
| 2 (0,03 %) | 36 { 37 CenterX = rnd.Next(Parent.Left.RoundedToInt(), Parent.Right.RoundedToInt()); 38 SpawnStar(); 39 } 40 } 41 Star star = new Star(Properties.Resources.star); 42 public void SpawnStar() 43 { 44 star.Center = Center; 45 Parent.AddChildBack(star); 46 47 CenterY = rnd.Next(Parent.Top.RoundedToInt(), Parent.Bottom.RoundedToInt()); 48 } |
|------------|---|