

## 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	Medida	Porcentaje
183	private Image LoadImage()	
184	{	
185	Image[] ships = Spritesheet.Load(@"Resources\shipsheetparts.png", new Size(200, 200));	28700 (41,76 %)
186	foreach (Image img in ships)	1708 (2,49 %)
187	{	
188	img.RotateFlip(RotateFlipType.Rotate270FlipNone);	258 (0,38 %)
189	}	
190	Image result = ships[shipIndex];	
191	result.RotateFlip(RotateFlipType.RotateNoneFlipX);	35 (0,05 %)
192	Extent = new.SizeF(result.Size.Width / 2, result.Size.Height / 2);	7 (0,01 %)
193	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	Medida	Porcentaje
183	private Image LoadImage()	
184	{	
185	Image[] ships = Spritesheet.Load(@"Resources\shipsheetparts.png", new Size(200, 200));	40615 (40,67 %)
186	foreach (Image img in ships)	2719 (2,72 %)
187	{	
188	img.RotateFlip(RotateFlipType.Rotate270FlipNone);	
189	}	
190	Image result = ships[shipIndex];	
191	result.RotateFlip(RotateFlipType.RotateNoneFlipX);	36 (0,04 %)
192	Extent = new.SizeF(result.Size.Width / 2, result.Size.Height / 2);	1 (0,00 %)
193	return result;	

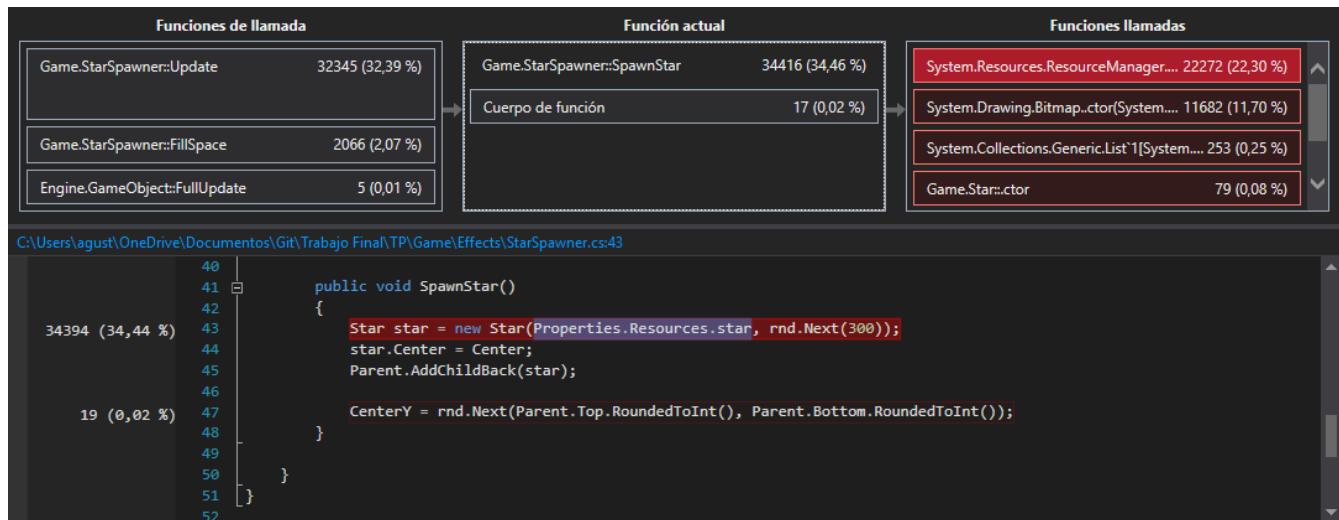
## Nº 2

### Identificar problemas

La clase “Start” tarda mucho en ser dibujada.

### Mediciones

[Llamada externa] System.Drawing.Graphics.DrawImage(System.Drawing.Image, System.Drawing.Rectangle, System.Drawing.Rectangle, Syst...	27075 (27,11 %)	27075 (27,11 %)	System.Drawing.ni...
[Llamada externa] System.Resources.ResourceManager.GetObject(System.String, System.Globalization.CultureInfo)##6002C19	22392 (22,42 %)	22392 (22,42 %)	mscorlib.ni.dll
[Llamada externa] System.Drawing.Bitmap..ctor(System.Drawing.Image, System.Drawing.Size)##600003E	18667 (18,69 %)	18667 (18,69 %)	System.Drawing.ni...



### Hipotesis

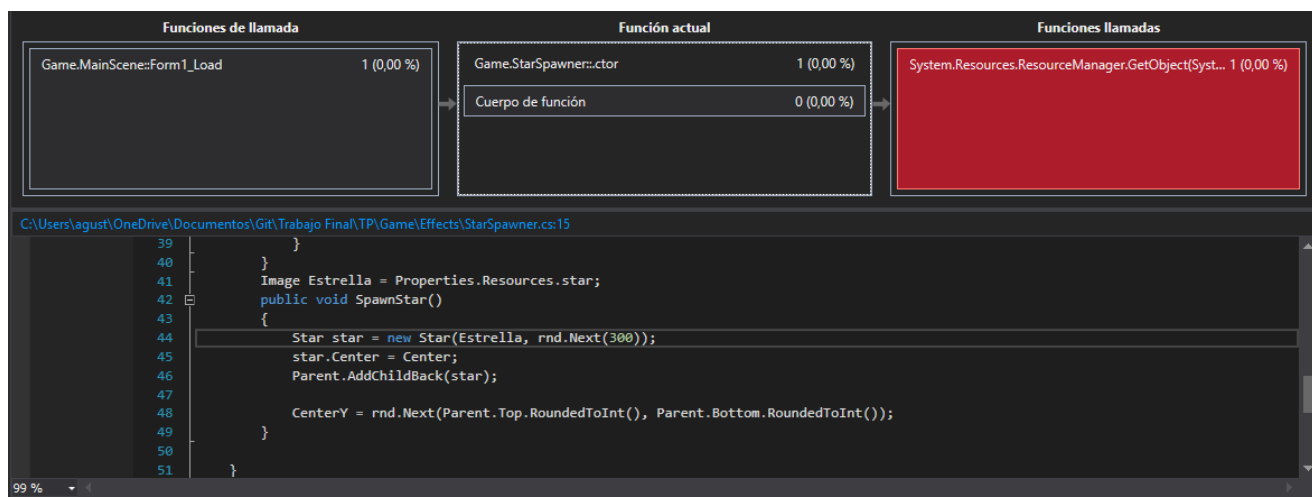
Si hago que se guarde la imagen de la estrella, no tiene porque volver a buscarla cada vez que se crea una nueva estrella.

### Descripcion de los cambios realizados

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

### Nuevas mediciones

Game.Star..ctor	164 (0,15 %)	13 (0,01 %)	Game.exe
[Llamada externa] System.Resources.ResourceManager.GetObject(System.String, System.Globalization.CultureInfo)##6002C19	153 (0,14 %)	153 (0,14 %)	mscorlib.ni.dll
[Llamada externa] System.Collections.Generic.List`1[System.Drawing.Point].Add	144 (0,13 %)	144 (0,13 %)	mscorlib.dll



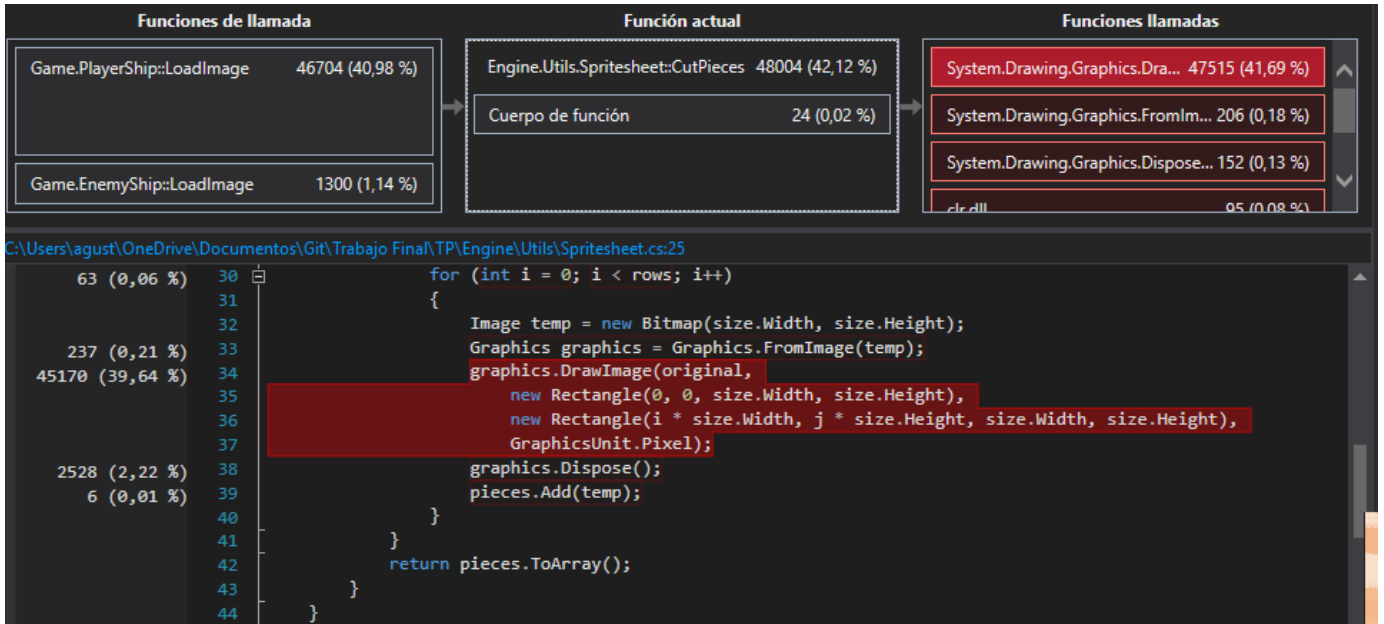
### Nº 3

#### 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... *E	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.

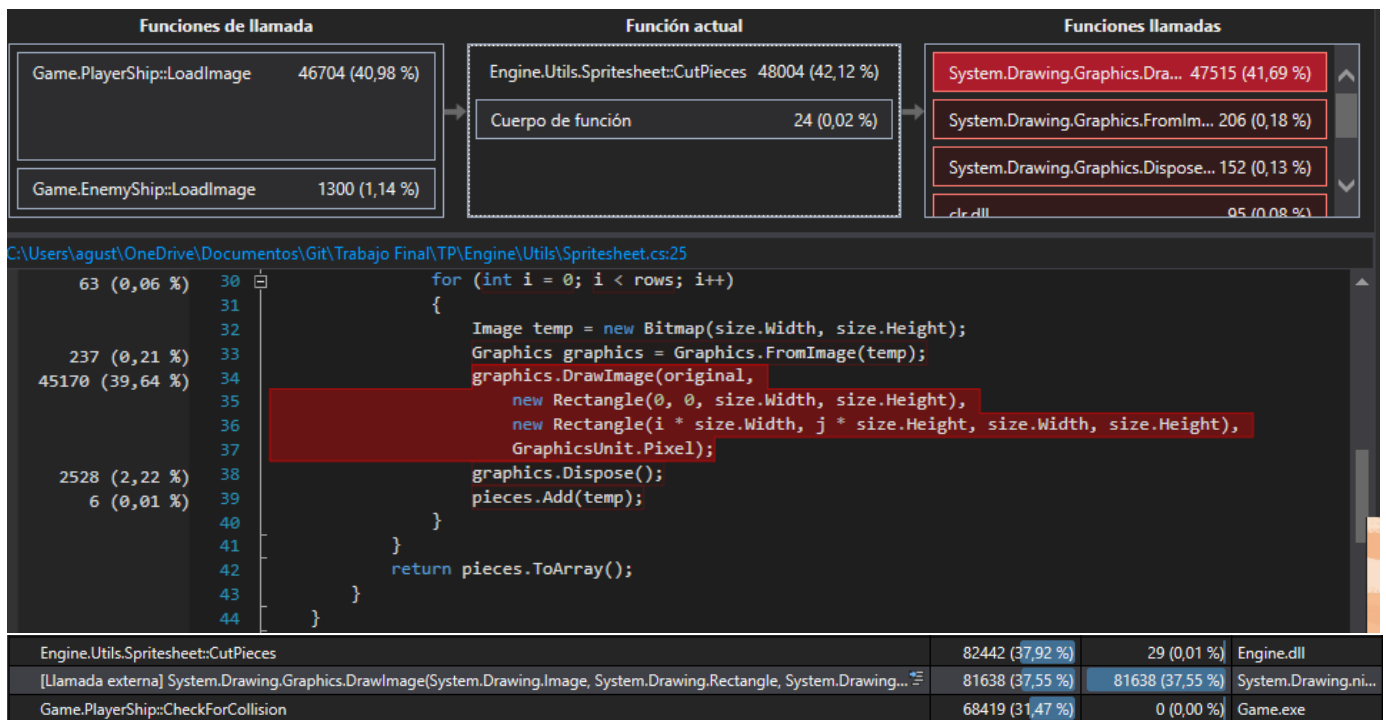
#### Nuevas mediciones

### Nº 3

#### Identificar problemas

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

#### Mediciones



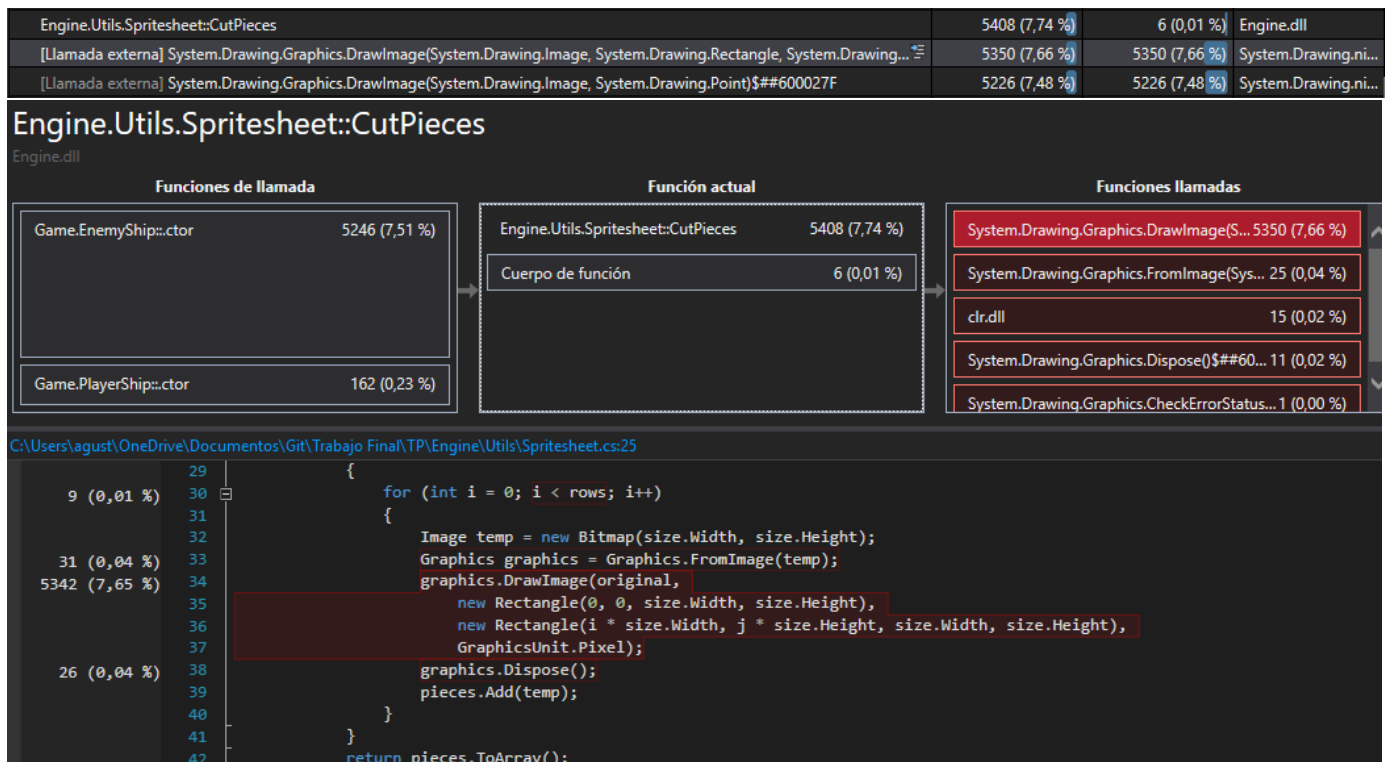
## Hipótesis

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

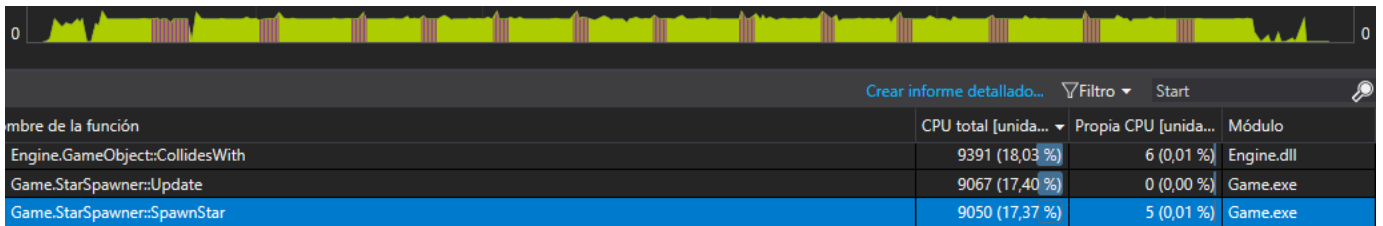


Nº 4

## 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

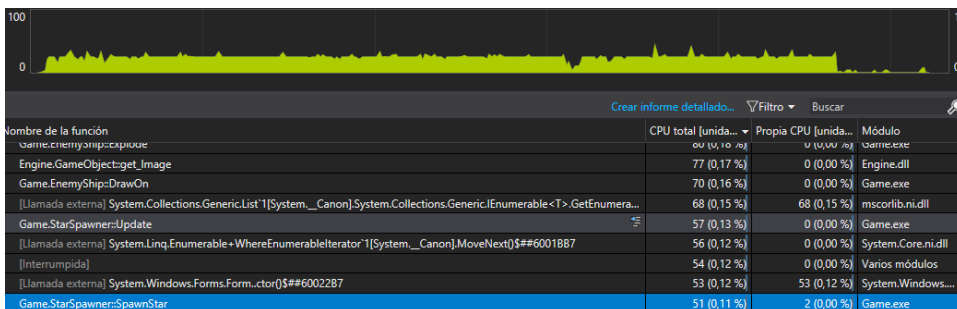
### StarSpawner:

```
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



**N° 5**

**Identificar problemas**

**Mediciones**

**Hipótesis**

**Descripción de los cambios realizados**

**Nuevas mediciones**

**N° 5**

**Identificar problemas**

**Mediciones**

**Hipótesis**

**Descripción de los cambios realizados**

**Nuevas mediciones**