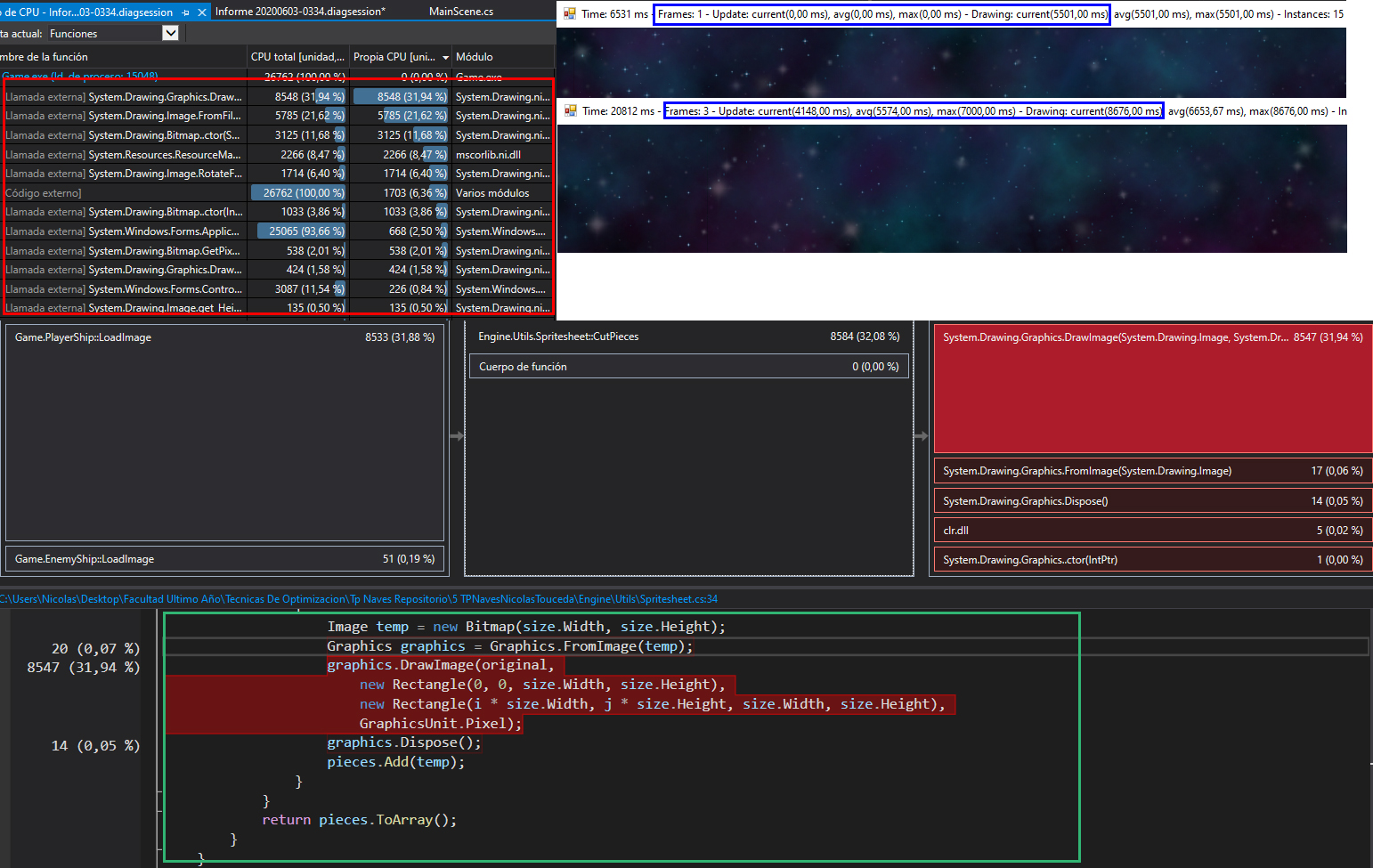
TP NAVES

Nicolas Touceda

HIPOTESIS 1

Mediciones:

Hipotesis:

Mis principales problemas gracias a las mediciones son:

Cuadro Rojo de la imagen: Encontramos el Primer HotSpot, donde el método de dibujo (System.Drawing.Graphis.Draw…) Utiliza muchos recursos y ocupa el 32% del procesador, Esto también genera mucho retraso en el programa.

Cuadro Verde de la imagen: Encontramos un bloque de código (graphics.DrawImage() que se encuneta en el método public Image[] CutPieces(Size size) manda a dibujar la lista de imágenes), donde ocupa el 32% del tiempo de la CPU.

Cuadros Azules de la imagen: Vemos que en cada Frame que se ejecuta el juego el Drawing tarda 5 a 8 segundos en dibujar. Esto es un problema porque no deja avanzar el programa.

Teniendo estas 3 mediciones en cuenta sabemos que este método draw es un problema.

Revisando el código llegue a la conclusión de que tanto el PlayerShip como EnemyShips, al momento de ser dibujadas, llaman 2 veces al método LoadImage. Lo que este método hace es desencadenar una serie de pasos, en el que, se recortan y dibujan todas las imágenes y son almacenadas en un array , finalmente este array es retornado, para ser dibujado (Esto ocurre en en CutPieces() dentro de la clase Spritesheet).

El problema de esto es que, cada ves que quiero cargar una imagen de una nave, el programa manda a recortar y crear todas las imágenes de vuelta.

Solución Propuesta:

Modificar 2 métodos para un mejor funcionamiento del programa, primero el método CutPieces voy a implementar un IF para que cree las imágenes una sola vez y no tenga que crearlas cada vez que se le llama el método LoadImage()

Voy a modificar el Método LoadImage() para acceder a mi imagen directamente por el índice de array y voy a guardar esta imagen en una variable.

Resolución:



Comparando esta imagen con la de medición, llegue a la conclusión que el programa mejoro mucho, ya que el tiempo de dibujado paso de 5-8 segundos a 1-2 segundos y por ende el programa avanza mucho mas rápido, en el mismo tiempo de prueba avanzo 9 frames en vez de 3.

La Implementación y los cambios Fueron Positivos.

Hipotesis 2