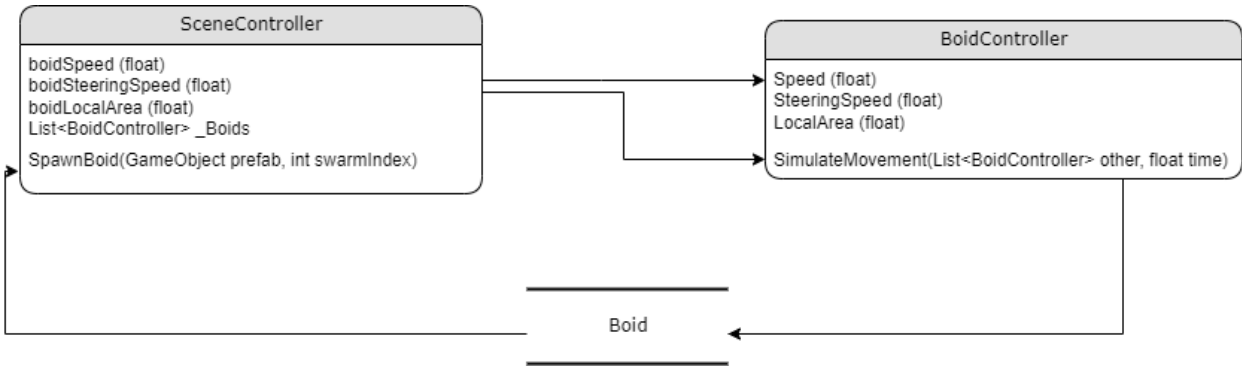
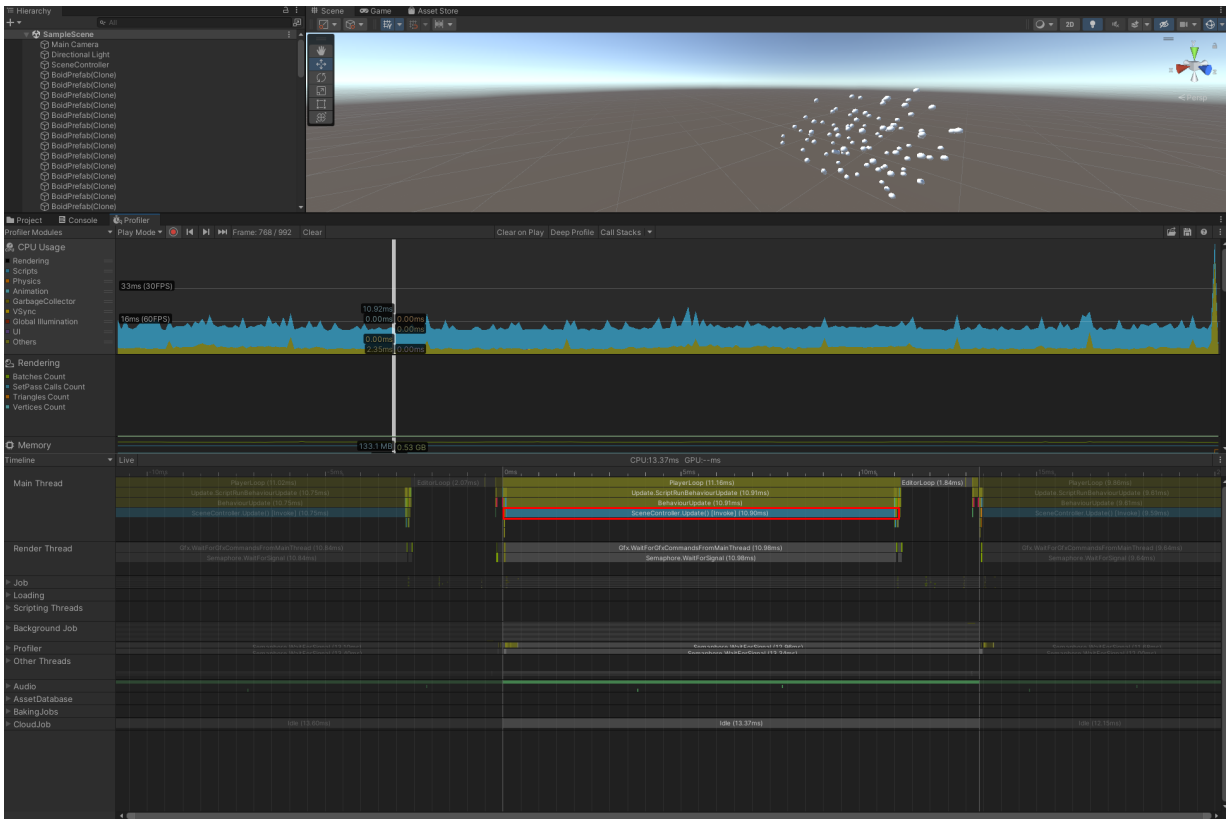
Boids

Diagrama de clases



Desde el SceneController se instancian todos los BOIDS guardandolos en una lista y se asignan los parámetros compartidos por todos los objetos al BoidController. En el BoidController está definido el comportamiento que deben seguir los objetos, sin embargo, los parámetros de estos comportamientos son asignados desde el SceneController. El SimulateMovement define el comportamiento que deben seguir los Boids pero esté es utilizado únicamente en el SceneController, el cual se encarga de gestionar el comportamiento de todos los Boids dentro de la lista.

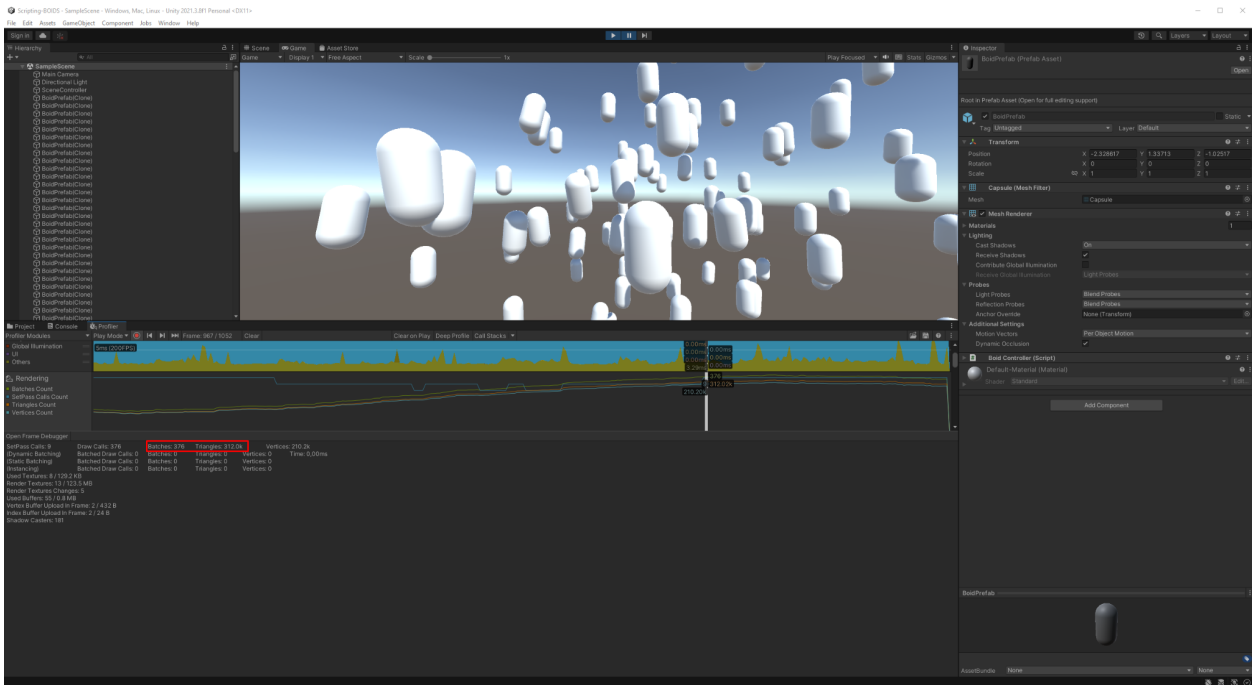
Impacto Flyweight



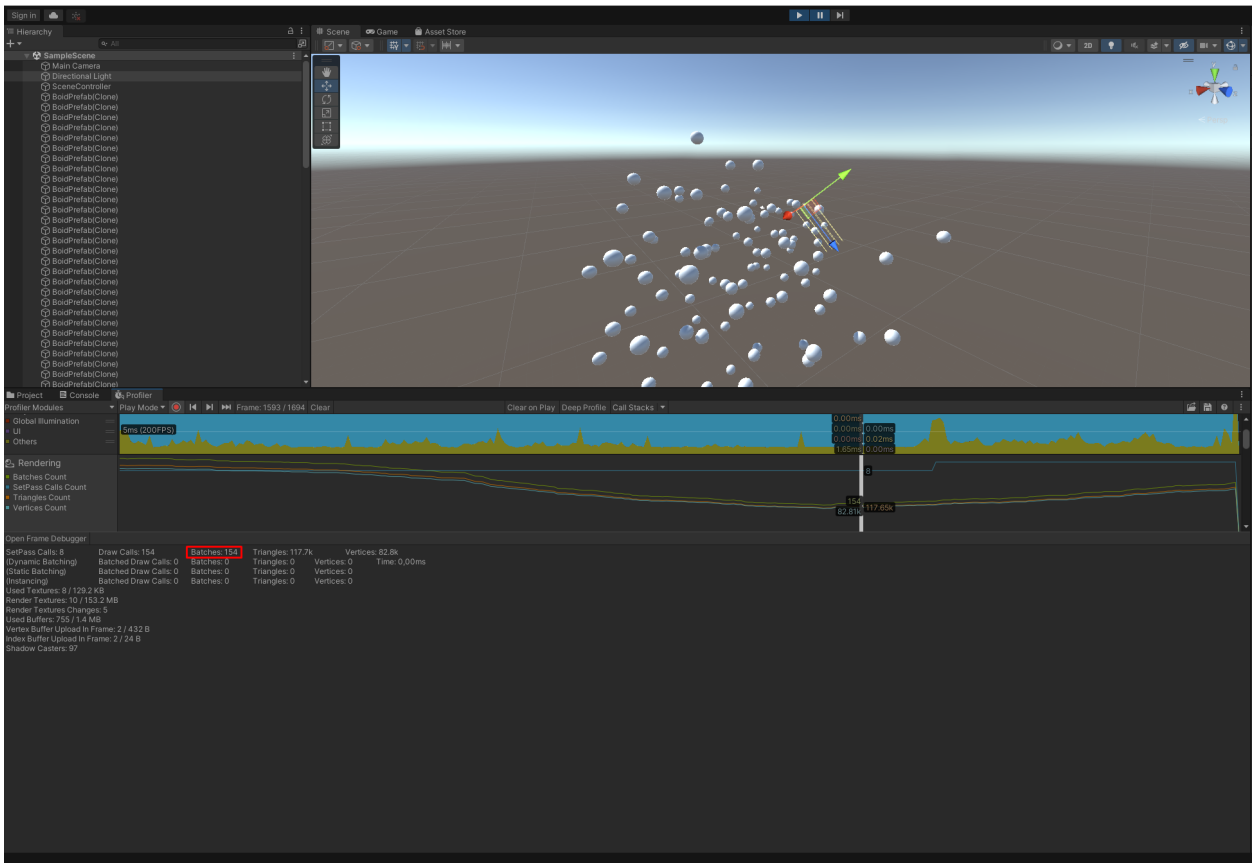
Como el BoidController no tiene ningún método Update, el único script que está gestionando el comportamiento es el SceneController, el cual se encarga de manera privilegiada de actualizar el comportamiento de los 100 boids que generamos en la escena. Por lo tanto, no se está teniendo que llamar a cada Boid y su comportamiento dentro de la CPU sino que con el SceneController se modifica el comportamiento de todos los boids.

Problema de optimización a solucionar

Como la simulación consiste de múltiples objetos renderizando en la cama, uno de los mayores inconvenientes era el de la cantidad de batches con los que tenía que lidiar la GPU. Al principio se estaban utilizando cápsulas ya que eran lo que parecería más similar al cuerpo de un "pájaro" sin embargo, el cálculo de las cápsulas dentro de Unity consume muchas más primitivas y por lo tanto, se vuelve un objeto mucho más pesado de renderizar.



Para solucionar este problema, la ruta obvia era elegir un objeto 3D que fuera más liviano en cuanto a renderización. Por lo tanto, para que tuviera uniformidad y siguiera teniendo algo de cohesión visual la simulación, se escogió la esfera para el prefab del boid, la cual de inmediatamente mejoró el rendimiento ya que consumía menos de la mitad de batches que los que hacían las cápsulas. Si se quisiera mejorar mucho más, se podría utilizar un cubo, el cual al ser un objeto tan simple, reduciría en gran tamaño la cantidad de primitivas que se utilizarían dentro de la escena.


Hierarchy
SampleScene
Main Camera
Directional Light
SceneController
BoidPrefab(Clone)
Scene
Game
Asset Store
Project
Console
Profiler
Profiler Modules
Play Mode
Frame: 1593 / 1694
Clear
Clear on Play
Deep Profile
Call Stacks
Global Illumination
UI
Others
5ms (200FPS)
Rendering
Batches Count
SetPass Calls Count
Triangles Count
Vertices Count
Open Frame Debugger
SetPass Calls: 8
Draw Calls: 154
Batches: 154
Triangles: 117.7k
Vertices: 82.8k
(Dynamic Batching)
Batched Draw Calls: 0
Batches: 0
Triangles: 0
Vertices: 0
Time: 0,00ms
(Static Batching)
Batched Draw Calls: 0
Batches: 0
Triangles: 0
Vertices: 0
(Instancing)
Batched Draw Calls: 0
Batches: 0
Triangles: 0
Vertices: 0
Used Textures: 8 / 129.2 KB
Render Textures: 10 / 153.2 MB
Render Textures Changes: 5
Used Buffers: 755 / 1.4 MB
Vertex Buffer Upload In Frame: 2 / 432 B
Index Buffer Upload In Frame: 2 / 24 B
Shadow Casters: 97