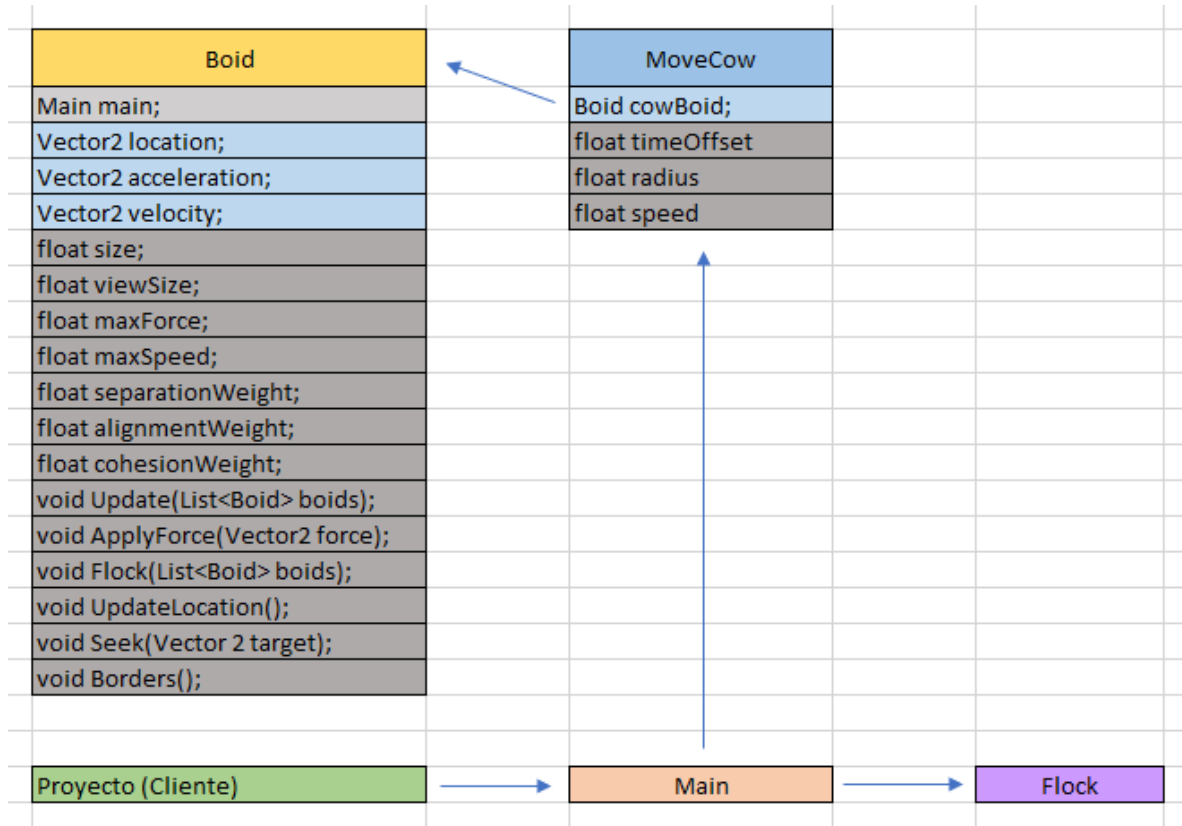
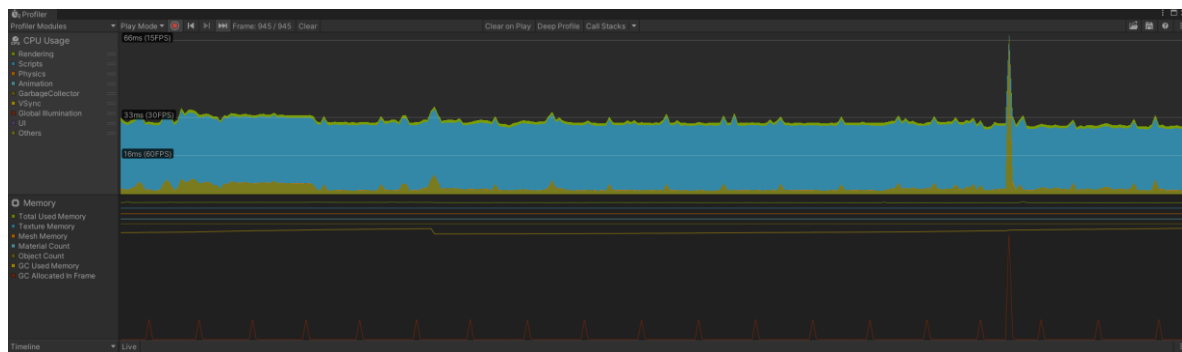


Boids Julio Echavarría 000419350

Implementación flyweight



Profiler Antes



Antes de mejorar el problema de colisiones entre las vacas, el desempeño del programa bajó mucho, para arreglarlo se puso un nuevo vector que evita que se choquen entre los vecinos.

Profiler Luego

```
↑ referencia
Vector2 Separate (List<Boid> boids) {
    float desiredseparation = size;
    Vector2 steer = new Vector2(0, 0);
    int count = 0;
    // se evita que se acerquen demasiado y se buguee
    foreach (var other in boids) {

        Vector2 diff = location - other.location;
        float d = Vector2.SqrMagnitude(diff);

        if (d > 0 && d < desiredseparation*desiredseparation) {

            diff /= d;
            steer += diff;
            count++;

        }

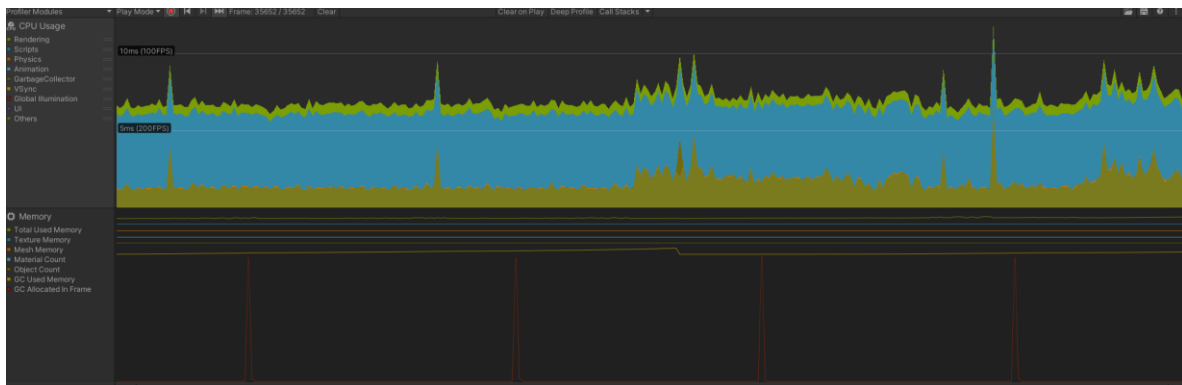
    }

    if (count > 0) {
        steer /= (float)count;
    }

    if (steer.sqrMagnitude > 0) {

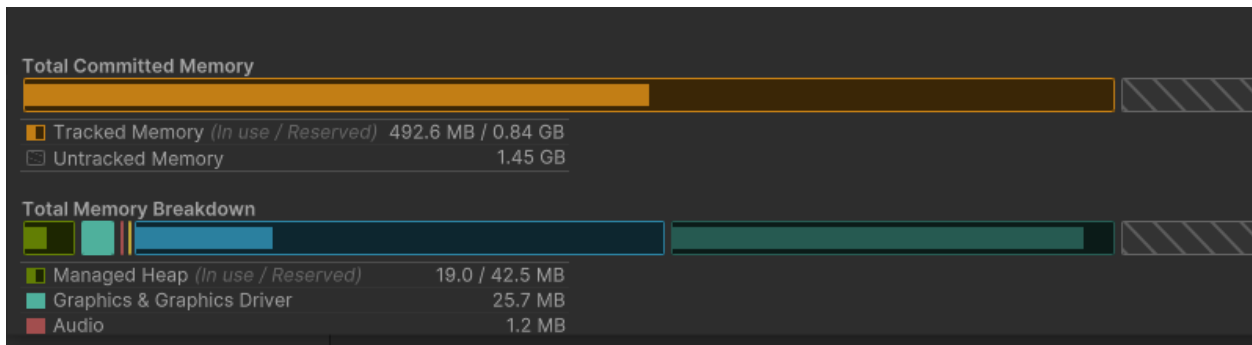
        steer.Normalize();
        steer *= maxSpeed;
        steer -= velocity;
        steer = Limit(steer, maxForce);
    }

    return steer;
}
```



Hacer esto hizo que el rendimiento mejorara, pero hizo que los picos sean más pronunciados a pesar de ser más escasos.

Memoria



El flyweight pone un menor peso en la memoria ya que los elementos que los Boids necesitan se usan directamente desde un script “padre”. Esto evita crear muchas variables en esta, permitiendo un mayor número de Boids sin necesidad de borrar comportamientos.