

Tomando como base el TP anterior lo primero que hice fue desarrollar el TDA singleton y el TDA nave para poder hacerles pruebas individuales en el archivo main. Con esos TDA funcionando correctamente hice los TDA disparo y asteroide, a los cuales les realice pruebas a instancias individuales para asegurar su correcto funcionamiento; seguí con el TDA lista e iterador, y verifiqué poder crear y usar listas de asteroides y disparos. Después con todos los TDA funcionando y las físicas de movimiento implementadas pasé a armar los ciclos de verificación de colisiones; a esa altura el juego se ejecutaba y se podía controlar la nave, disparar, y destruir asteroides.

Constantemente realizaba pruebas con Valgrind para ir controlando que no existieran fugas de memoria y revisando los TDA en busca de errores y optimizaciones.

Armé la estructura del juego (cuando aparece la nave, que pasa si no hay vidas restantes, que pasa si no hay asteroides, etc) y agregué la interfaz de puntuación. En ese momento el código main tenía cerca de 350 líneas, me puse a agregar todas las validaciones de memoria y quedó como esta ahora por arriba de 500. Lo ultimo fue acomodar los módulos y pulir el makefile.

Algunas consideraciones a tener en cuenta: No verifiqué los llamados a funciones de dibujo (no le encontré sentido); el modulo\_asteroids recolecta algunas funciones sueltas para las cuales no valia la pena armar 4 o 5 módulos para 1 o 2 funciones. En la función que verifica la colisión con un asteroide hago un llamado a ajustar variables (de singleton.h) esto es porque al no importarle la posición a ningún modulo se generan desfases donde dos objetos (asteroide, disparo, nave) estan en el mismo lugar pero sus coordenadas no (como en un círculo el ángulo  $\pi$  y el ángulo  $3\pi$  son lo mismo); caracteres y diccionario estan reutilizados del tp anterior. De todas las pruebas que hice con valgrind ninguna tenia fugas de memoria (dejando de lado los 1462 bloques de SDL), pero si “errores de contexto” los cuales aumentan a medida que aumenta el tiempo de juego (llegando a bastantes millones de errores), me parecio que podia tener que ver con el archivo binario y los padding de las estructuras así que puse graficador\_inicializar afuera del while principal y el resultado fue una disminución a un total de 135 errores (los mismos que el tp1 base), de cualquier manera volví a poner graficador\_inicializar en el while porque asumí que se pedía eso. Las escalas y varios números relacionados a la estética estan puestos a ojo.