

## ISW1 - Parte Práctica: “*The Smugglers Run!*”

"It's the ship that made the Kessel Run in less than twelve parsecs..." – Han Solo about the Millenium Falcon, Star Wars Ep IV, 1977.

Nos contactaron para colaborar en la construcción de un nuevo “mini-juego” en el contexto del desarrollo continuo de un juego ya existente de naves espaciales, viajes interestelares y batallas galácticas.

Codenamed “**The Smugglers Run!**”, el juego consiste en que el jugador debe armar una pequeña flota con sus naves espaciales (que pueden ser de diferentes clases y tamaños) y lanzarlas a cruzar una ruta peligrosa que establece un contrincante. Dependiendo de si sus naves logran o no cruzarla, y cuantas llegan a destino, el jugador gana o pierde...

Por ahora debemos construir **sólo** una parte del mismo que sirva de **prueba de concepto**. La idea es la siguiente:

Se debe poder construir una flota de naves no vacía de cualquier cantidad. Por ahora sólo se eligieron 3 tipos de naves que ya son parte del juego para ser incluidas aquí, pero si la prueba es un éxito, se agregaran más de inmediato. Las mismas pueden verse a continuación:

Heavy Cruiser Class Ship	Smugglers Transport C.S.	Light Interceptor C.S.
Maneuverability: 50 Shields: 100 Fuel: 50 MaxHullDamageLevels: 5 Thrusters: Impulse o Graviton o Warp	Maneuverability: 70 Shields: 50 Fuel: 95 MaxHullDamageLevels: 2 Thrusters: Impulse o Graviton o Warp	Maneuverability: 90 Shields: 30 Fuel: 30 MaxHullDamageLevels: 2 Thrusters: Impulse or Graviton o Warp

A la par se define una secuencia de sectores galácticos como la ruta a cruzar por la flota, también de cualquier longitud. Los tipos de sectores por ahora definidos son los siguientes, junto a sus características importantes para el mini-juego:

Planetary Sector	Asteroids System	Nebula Cloud
En general con valores medios de Gravity y Radiation	En general con valores altos de Gravity y bajos de Radiation	En general con valores bajos de Gravity y altos de Radiation

Con la flota y la ruta definida, se lanza a la flota a pasar por la ruta. Las naves deberán intentar cruzarla una a una (de forma independiente) pasando de forma ordenada a cada uno de sus sectores hasta llegar al último. Si una nave es destruida en el proceso, no habrá logrado sobrevivir. De la misma manera, si una nave se queda sin combustible antes de pasar por el último sector de la ruta, o no le alcanza el combustible para pasar por el último sector, tampoco habrá sobrevivido. Sí, se considerará sobreviviente a una nave que quede en exactamente 0 de combustible al pasar por el último sector de una ruta. Se quiere saber cuántas naves de la flota sobrevivieron (si alguna) y cuales.

Para determinar si el casco (hull) de la nave es dañado o el consumo de combustible (fuel) efectivo de una nave al pasar por un sector, se realizan los siguientes chequeos:

- Si el rating de maniobrabilidad de la nave es mayor o igual a la gravedad del sector, la nave no es dañada. En caso contrario el casco de la nave sufre un nivel de daño.
- Si los escudos de la nave (shields) igualan o superan el nivel de radiación del sector, la nave tampoco es dañada. En caso contrario el casco de la nave sufre un nivel de daño.
- Cruzar por un sector normalmente consume  $10^1$  de combustible.

Estas son las reglas **GENERALES** del mini-juego, sin embargo diferentes características de las naves, del sector que atraviesan y del sistema de propulsores que utilizan, produce que los resultados puedan variar un poco.

Todos estos casos están específicamente explicitados en sumo detalle en los Tests que nos incluyó la gente del juego.

## Trabajo a Realizar:

Deben abrir el archivo provisto que dejó la empresa y construir un modelo que los pase (o al menos todos los que puedan!). Dicho modelo debe seguir las heurísticas de diseño vistas hasta ahora en el curso. Se evaluará cómo representa el dominio de problema, su declaratividad, si tiene o no código repetido, el reemplazo de ifs por polimorfismo donde corresponde, la asignación de responsabilidades, los nombres de sus variables, mensajes y categorías, la creación de objetos válidos y completos desde su creación, la no violación del encapsulamiento de los objetos, etc...

---

<sup>1</sup> Deberíamos utilizar unidades para representar todas estas magnitudes. Pero para simplificar, y debido a que no vamos a hacer uso de ningún tipo de conversión entre ellas, ni sabemos exactamente cuales son las unidades de varios de las magnitudes del dominio (10 kg de antimateria por viaje?) POR EL MOMENTO no seguiremos nuestra propia heurística con respecto al uso de las mismas.

## Aclaraciones:

- **No se pueden modificar los tests de la empresa.** Luego, no hace falta que le quite código repetido. El trabajo a realizar está en el modelo, no en los tests. Sin embargo **poseen mensajes de creación que esperan por su implementación.**
- En principio no hay restricciones de que una nave que tenga agotado su combustible entre al próximo sector de una ruta. Es difícil, pero podrían darse las condiciones de que pasar por este no requiera consumo de combustible alguno y lo pase...

## Ayudas:

- Mientras diseñe tener en mente que las **naves** son parte de un juego existente, y que poseen un protocolo de funcionalidades aquí no visible que sirve para que ataquen a otras naves, aterricen dentro de otras naves o planetas, etc... ¿Qué patrón que vimos podría ayudarnos para esta situación?

## Recomendaciones:

- Grabar la imagen cada cierto tiempo.
- Configurar la opción de "autosave" en la categoría de clases de la solución. Es recomendable hacerlo con versioning cada 60 segundos.
- Quite el código repetido que le haya quedado, y mejore la declaratividad de sus métodos. Recuerde que cuenta con un ambiente que posee **refactors automatizados** muy útiles para este paso, como el *rename* o el *extract method*.
- ¡Todo suma! Incluso si no termina, entregue el modelo hasta donde llegó para poder recibir feedback de sus dificultades. Recuerde que el parcial se aprueba con 4, no es necesario que todo esté perfecto.

## Entrega:

1. Entregar el fileout de la categoría de clase **ISW-Parcial1-1C25** que debe incluir toda la solución (modelo y tests). El archivo de fileout se debe llamar: **ISW-Parcial1-1C25.st**
2. Entregar también el archivo (**user** changes!) que se llama **CuisUniversity-nnnn.user.changes**
3. Probar que el archivo generado en 1) se cargue correctamente en una imagen "limpia" (o sea, sin la solución que crearon. Usen otra instalación de CuisUniversity/imagen si es necesario) y que todo funcione correctamente. Esto es fundamental para que no haya problemas de que falten clases/métodos/objetos en la entrega.

4. Deben entregar usando el siguiente form:  
<https://tinyurl.com/isw1-1p-1c-2025>
5. De forma alternativa si no pudiste entregar con el form, realizar la entrega enviando mail a: **entregas@isw2.com.ar** con el **Subject:**  
**LUnnn-aa-Solución 1er parcial 1c2025.**  
En caso de rebotar el envío, reintentar comprimiendo los adjuntos.