Desafíos de SpaceJoust

- **Observable/observer:** Se utilizó la clase Observable y la interfaz Observer implementadas en java como soporte al paradigma MVC. Esta fue la solución a nuestro problema inicial de vinculación entre view y model. Inicialmente se consideraba tener una referencia al objeto en cada view pero esto se volvía incómodo y engorroso.
- **Manejo del estado del juego en view o model:** Se decidió que el estado del juego, esencialmente el sistema de menúes, sería manejado en la view en lugar del model puesto que la intervención de este último solo es necesaria cuando se comienza el juego.
- **Movimiento de las naves:** La solución a este problema fue que las naves tuvieran una única variable de posición nombrada “radialPosition” de donde se derivaría la posición en la pantalla obteniendo el seno y el coseno de esta. Esta posición es variada con una única variable de velocidad de tal manera que el movimiento sea constante y fácil de manejar en lugar de nuestra implementación anterior que consistía de una velocidad en la dirección X y otra en la dirección Y.
- **Manejo de colisiones:** al colisionar las naves, surgía complicaciones con la rotación y el área de colisión de las mismas.
- **Threads diferentes para view y model:** Se utilizaron dos threads distintos para la view y el model puesto que esto es lo indicado por el paradigma MVC.
- **Conexión menú/juego:** el view maneja el funcionamiento general del menú pero se creó un objeto button para poder manejar cada opción del mismo, es decir, le fuimos delegando tareas a cada uno para actuar en conjunto con la view.
- **Estructuración del juego de acuerdo a conceptos OOP:** Inicialmente no resultaba evidente la estructura de herencia del juego. Sin embargo, se advirtieron comportamientos similares de los objetos como posición y velocidad de donde surgió naturalmente una estructura hereditaria, lo cual facilitó inmensamente los cambios y revisiones al código.
- **Animación de las views:** se creó la clase animación para que maneje las animaciones de las diferentes view de los objetos.