INFORME TPE

Programación Orientada a Objetos

Asteroids

Integrantes

Lucas Casagrande 55302

Magdalena Vega 55206

Nicolás Kuyumciyan 55165

Gabriel Silvatici 53122

Introducción

Se trata de un juego con varios modos, para 1, 2 y 3 jugadores, el escenario del juego es el espacio, y luego están las naves espaciales (pueden ser 1, 2 o 3) y los asteroides.

El juego presenta un menú donde se puede elegir: Nuevo juego, opciones(de pantalla), ayuda(controles) y puntuaciones.

El juego principalmente adopta una modalidad de arcade, siendo el principal modo de juego, un jugador vs la computadora donde el jugador es una nave que debe ir esquivando los asteroides que son lanzados inteligentemente, a medida que avanza el tiempo los asteroides son lanzados a mas velocidad y a mayor cadencia de tiro, aumentando así la dificultad resultando un juego desafiante, el jugador también debe aprovechar los powerups que se irán apareciendo en pantalla aleatoriamente, para proveerle mejoras en las habilidades de la nave, o por ejemplo vidas extra.

Luego, se tienen los modos de 2 jugadores y 3 jugadores.

Con 2 jugadores se puede optar por jugar con 2 naves contra la computadora que lanzara los asteroides hacia las naves donde el jugador que mas tiempo perdure sera el ganador, se puede elegir también que un jugador maneje una nave y el otro lance los asteroides, cuando los asteroides derriben la naves habrá un cambio de lados donde el que manejaba la nave ahora lanza los asteroides, ganara el jugador que perduro mas tiempo con la nave.

Con 3 jugadores, similar a 2 jugadores, se pude optar por jugar con 3 naves por cada jugador donde el jugador ganador será el que perdure mas tiempo sin ser derribado, la otra opción es que haya 2 naves manejadas por 2 jugadores, y el tercer jugador lance los asteroides aquí nuevamente habrá un cambio de lados y el ganador ser el que haya perdurado mas tiempo sin ser derribado (manejando la nave).

Los powerups aparecen en pantalla aleatoriamente en todos los modos de juego, el que maneje la nave debe alcanzarlos, y los asteroides deberán destruirlos (así perjudicando a la/s nave/s)

Diseño

Haciendo uso de las librerías libGDX, se diseño una jerarquía de clases donde, en 9 diferentes paquetes se implementaron sus respectivas clases, con un total de 53 clases:

* Paquete para las clases que conforman el motor del juego: contiene excepciones, manejo del motor físico (clase Collisionable) de la clase Collisionable extienden todas las clases que puedan colisionar, en este caso particular son las naves y los asteroides, Se encuentra la clase Assets donde ser cargan los archivos del proyecto (tipografías, imágenes, sonidos), clases utilitarias(clase ArrayMap y Connector). Dentro de este paquete están embebidos todos los paquetes restantes del proyecto:
* Clases que conforman los asteroides: en el paquete se implemento una clase para el lanzado de asteroides, para la inteligencia del lanzador de asteroides y para el entorno de usuario del asteroide.
* Clases que conforman los managers del juego: dentro del paquete se encuentran clases para el manejo de menú (MenuManager), para el manejo de los puntajes(HighScoreManager), manejo de la partida de juego(GameManager), y sus respectivas clases de interfaz visual. Dentro del mismo paquete están embebidos otros 2 paquetes, uno para el manejo de HUD's, y otro paquete para el manejo de los mundos, una clase para cada modo de juego.
* Clases que conforman los PowerUps: en el paquete hay una clase por cada PowerUp diferente(para extra vida, extra aceleración, extra masa e invencibilidad), cada powerup extiende de la clase padre PowerUp y su respectiva clase de interfaz visual.
* Clases que conformas las spaceships(naves): en el paquete se encuentran; una clase para crear las naves, otra que define el tipo de la nave (hay 5 tipos diferentes), luego la clase SpaceShip que extiende de Collisionable, donde implementa comportamientos para colisionar.
* Clases que conforman los test del proyecto: en este paquete se implementaron las pruebas Junit que testean diferentes comportamientos del proyecto.
* Clase DesktopLauncher dentro del paquete Desktop, esta clase debe ser implementada como requisito que pida la librería libGDX

Decisiones

* Los nombres de los jugadores están limitados a 7 caracteres por una cuestión estética.
* A la hora de cargar las naves para jugar el GameManager funciona como almacenador temporal tanto del back como del front, para una vez inicializado el juego cargarlo en sus respectivas clases (WorldManager y WorldManagerUI).
* Decidimos tener una sola clase SpaceShip y un generador que carga la clase SpaceShip con los parámetros dependiendo de la nave seleccionada (patrón de diseño Factory)
* Los Highscores se guardan en un TreeMap serializable, en vez de guardarlos como texto plano.