

TP Videojuegos

Práctica 6

En esta práctica, vamos a empezar desde el código que hemos visto en clase para el juego **Asteroids** con SDLNET y ampliarlo para que tenga funcionalidades adicionales. Recuerda que el número de jugadores es fijo (codificado en la clase GameManager), sin embargo, durante la corrección se probará con diferentes números.

1) Añadir Asteroides al Juego

La versión actual del código tiene sólo cazas y balas. Añadir asteroides al juego con el siguiente funcionalidad:

1. Cuando el juego se encuentra en estado `RUNNING` (ni `WAITING` ni `OVER`), el **“jugador master”** añade 1 asteroide cada 10 segundos al juego, con la siguiente configuración:

```
Vector2D vel((1-2*(rand() % 2))*((rand() % 10)+1),((rand() % 10)+1));
vel.normalize();
vel = vel * 0.5;
Vector2D pos(rand() % getGame()->getWindowWidth(), rand() % 30);
Vector2D dir(0, -1);
int width = (rand() % 10) + 20;
int height = (rand() % 10) + 20;
```

2. Al añadir un asteroide, el **“jugador master”** debe enviar la información correspondiente a los demás jugadores para que cada uno cree un asteroide idéntico.
3. Como en la práctica 4, los asteroides giran mientras se mueven y una vez salen de los bordes de la pantalla, se deben deshabilitar y volver a utilizarse, etc.
4. Los asteroides deben ser visibles sólo cuando el estado del juego es `RUNNING`.
5. Cuando una bala choca con un asteroide, la bala desaparece pero no el asteroide. Los asteroides no se dividen como en la práctica 4.
6. Cuando un caza choca con un asteroide, el caza muere (el asteroide no).

El código proporcionado ya incluye una clase `AsteroidManager`, complétala y modifica el método `initGame` en `AsteroidsGame.cpp` para añadir ese nuevo componente al juego.

2) Manejar la Desconexión de Clientes

La versión actual de código no maneja el caso en el que un cliente se desconecta. Esto puede suceder al pulsar `ESC` o debido a algunos problemas de red. En esta parte, debes modificar el juego para manejar esta situación.

Primero modifique la clase `Server` de tal manera que cuando un cliente se desconecta, envíe un mensaje correspondiente a todos los clientes, y luego modifique el juego de modo que cuando un cliente recibe dicho mensaje, detenga el juego y cambia su estado a `WAITING`.