Introducción

Block breaker es un juego donde manejas una pala con la que golpeas una pelota y vas destruyendo bloques evitando que caiga. Mientras vas destruyendo esos bloques pueden aparecer unos powerups que pueden ayudar al jugador añadiendo más pelotas en pantalla, ampliando la pala o reduciéndola…

Arquitectura

Para la implementación de este juego se sigue una arquitectura de clases simples:

- **Vector2D:** Equivalente a Coor, contiene un entero **x** y otro **y**.

- **Paddle:** Representa a la pala, contiene métodos para actualizar su posición mediante el input representado por un **Vector2D** mediante una dirección (también representado por esa misma clase) además un método para dibujarse, quitarse vida en caso de que no queden pelotas en pantalla.

- **Ball:** Es la pelota que tenemos que evitar que se caiga y con la que tenemos que destruir los bloques, contiene una posición y una dirección al igual que Paddle, también contiene métodos para dibujarse en pantalla e inicializar su dirección a una random.

- **Bloque:** Clase que representa los bloques a destruir contiene un Vector2D para representar su posición, métodos para dibujarse y en caso de colisión quitarse vida

- **Reward:** Clase que representa los powerups que podemos coger con la pala, contienen como todo elemento en el juego una posición y para moverse una dirección que siempre será hacia abajo y un enum **RewardID** que representa el tipo de powerup a activar en caso de que el jugador

Para organizar los elementos del juego al principio se iban a representar en arrays pero debido a que tenía que andar eliminando elementos continuamente al final están representados en Lista enlazadas denominadas **ListaBolas, ListaBloques** y **ListaPremios** (aunque técnicamente se podría haber usado la estructura List de la librería System de modo que no se necesitaría estas clases).

En cuanto a representación de todos estos elementos tenemos la clase **Tablero,** la cual contiene las tres listas mencionadas anteriormente, un Paddle representando al jugador un array de colores (formando una paleta de colores que en un principio se compone de los tres colores primarios) métodos para renderizar todos los elementos, mover los elementos de cada una de las listas comprobando a su vez colisiones entre los distintos elementos del juego (ej cuando se llame a mover las pelotas se comprobarán las colisiones entre el player, los bloques y los límites del juego). Además, su constructora lee archivos de modo que da lugar a la implementación de múltiples niveles.

Después dado que tenía varios niveles como quería que guardará la progresión, poder bloquear algunos niveles y demás monté un sistema de estados con un minimenu donde tú introduces el nivel a cargar, el guardar usuario, cargarlo ver los scores más altos por nivel con sus usuarios y ver los tuyos propios.