**PARCIAL**

**I. REFACTORIZANDO: BBD-MDD XP**

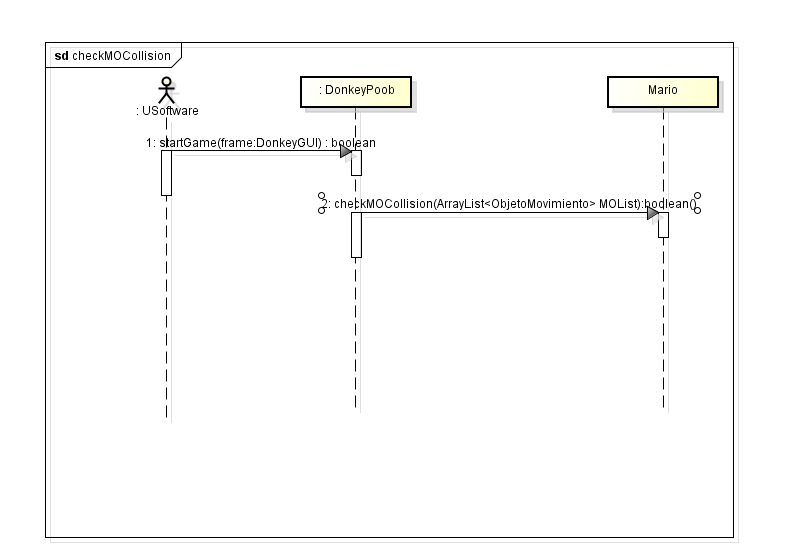
**Seleccione el comportamiento mejor implementado hasta el momento entre interacción del rescatador con los barriles o con las sorpresas.**

Método CheckMOCollision(ArrayList<ObjetoMovimiento>){}

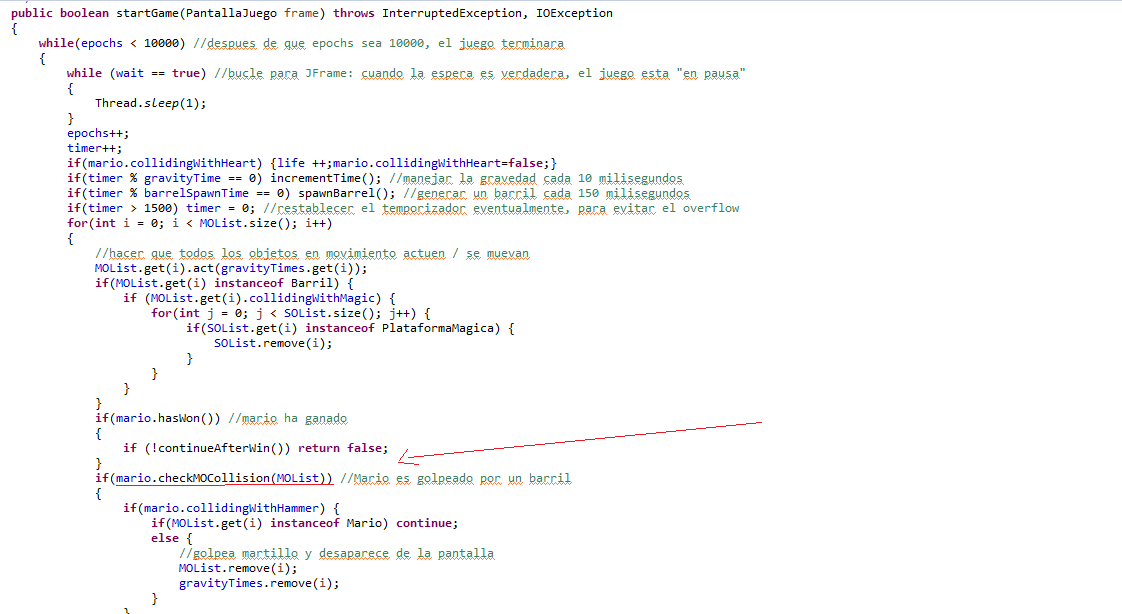
**1. Justifique su selección.**

Lo seleccione debido a que es un método que ayuda mucho, mi rescatador “Mario” o “Luigi” extienden de una clase llamada objetoMovimiento la cual tiene este método que lo que hace es por medio de las posiciones y el tamaño de cada objetoMovimiento en los cuales también incluyen los barriles podemos verificar si hay colisión con el rescatador, además que cumple con los estándares de código que no ocupa una pantalla y esta documentado.

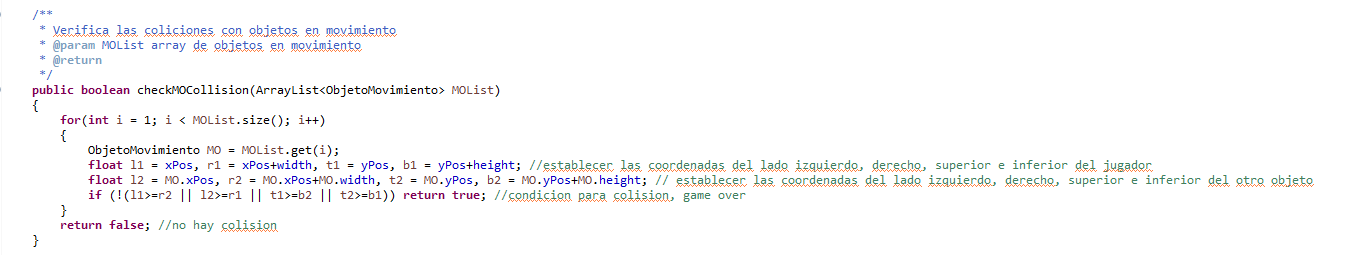
**2. Incluya en el documento los componentes BDD-MDD : prueba aceptación, diseño de comportamiento, código presentación, código aplicación y pruebas de unidad.**

****

**Presentación**

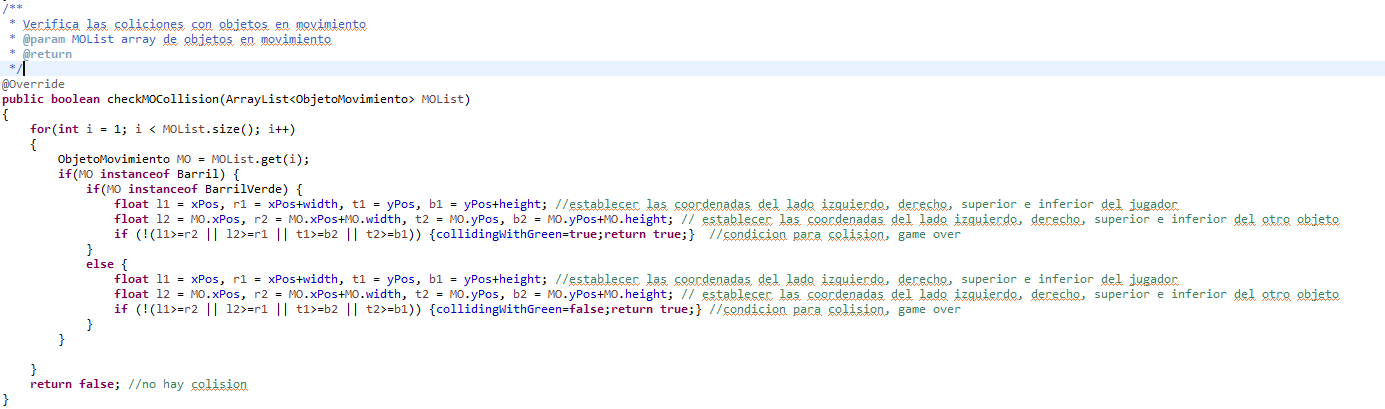
****

**aplicación**

****

**3. Refactorice este comportamiento. Explique e incluya los componentes que cambiaron.**

En la clase objetoMovimiento n ose tiene en cuenta el tipo de barril, para Mario creamos un condicional que si es de barril verde en vez de hacernos perder nos de una vida. De lo contrario haganos perder, esto lo hacemos mediante el atributo collidingWithGreen que nos dice si hubo colision y la DonkeyPoob se encarga de brindarnos la vida.



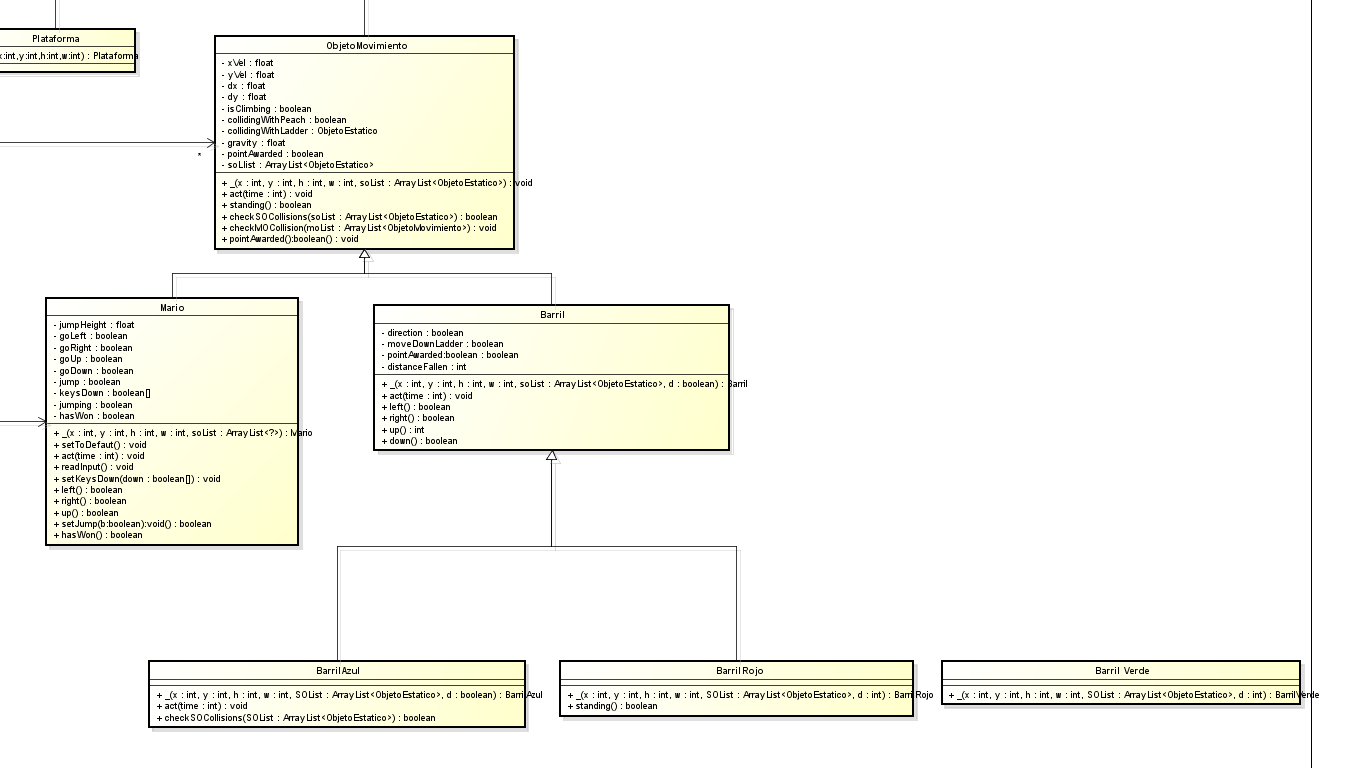
**Seleccione el área del proyecto donde mejor aplica el patrón Polimorfismo**

**El área de objetoMovimiento y sus objetos que lo heredan**

1. **Justifique su selección**

Es un área donde se evidencia el polimorfismo ya que objetoMovimiento es la clase padre y todas sus clases hija comparten una cantidad de atributos enorme lo cual nos permitió ahorrar demasiado tiempo y código, cada herencia evidencia que se necesitó menos código ya que todo lo necesario se extendía en objetoMovimiento el cual directamente habla con DonkeyPoob donde se maneja toda la lógica del juego y la comunicación están fácil entre clases ya que muchas veces solo se necesita saber si es un objeto movimiento para llamar barriles o al mismo rescatador. Y lo mas interesante es que a su vez objetoMovimiento extiende a objetoEstatico pero le doy prioridad a ObjetoMovimiento ya que es una clase con una herencia mas fuerte y a su vez vital para el funcionamiento del juego.

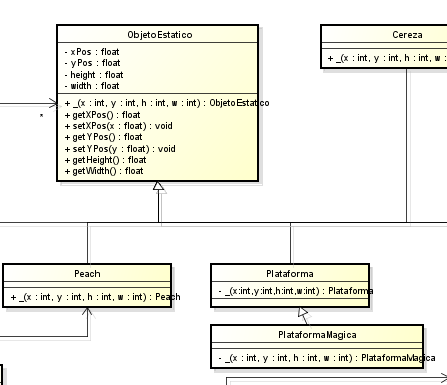
1. **Cómo se evidencia el patrón en el diseño estructural y el diseño dinámico. Incluya en el documento estos diseños.**

****

**3. Analice el diseño con respecto a los patrones variaciones protegidas y plantilla.**

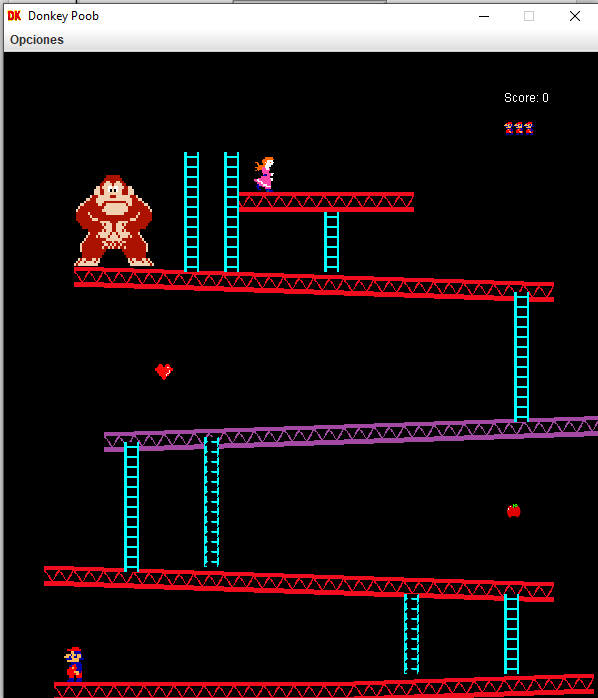
**4. Evalué las ventajas y desventajas de aplicar este patrón.**

**III. EXTENDIENDO. Plataformas mágicas**

****

Prueba de aceptación

Evidenciamos que se crean de forma aleatoria y solo una por partida, además de que la inferior nunca se torna como plataforma magic



Acontinuacion veremos como es que al tocar un barril la plataforma mágica, esta se destruye junto al barril.

