

Mesa de pool

Tema 6

Grupo: 27

Integrantes:

- Pedro Ignacio Palacios Rossi
- Sebastián Ignacio González Manríquez

Profesor:

- Geoffrey Jean-Pierre Christophe Hecht

Enunciado

En el panel central, con vista aérea, deben aparecer bolas en posiciones randómicas, una blanca y otras de color. El taco debe aparecer automáticamente apuntando a la bola blanca y debe ser manejado con troles GUI (teclado y mouse), para golpear bola blanca. Las bolas deben tener la física de impactos, inercia y roce. En las esquinas debe haber troneras donde pueden caer. La cantidad de bolas debe ser definible por interfaz GUI. Habrá bandas para rebote de las bolas y si caen en las troneras, otorgan puntos. Reiniciar se debe hacer por controles GUI. Si se cae la bola de color y la blanca no hay puntaje, si cae la blanca se resta puntaje.

Diagrama UML

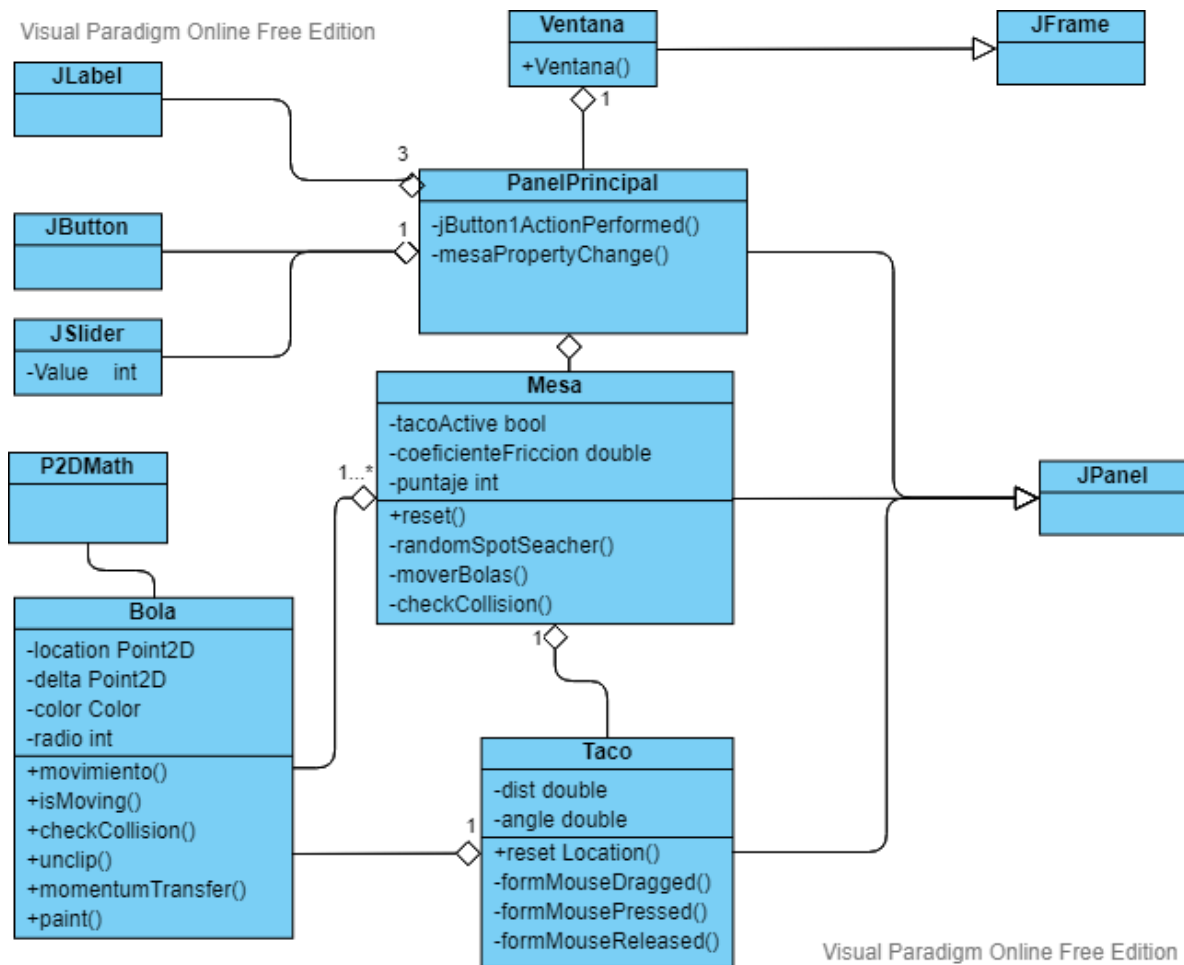
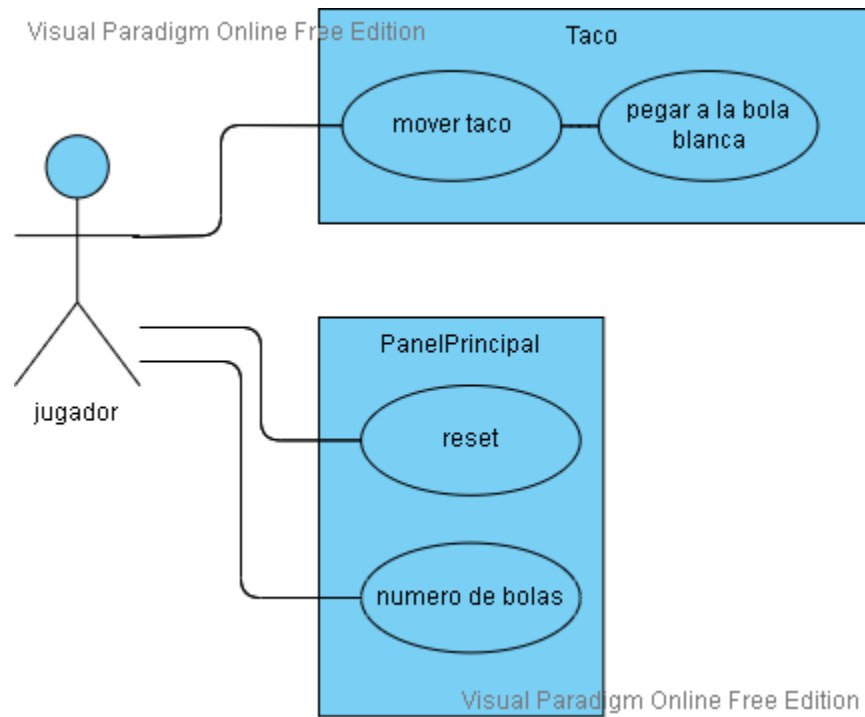


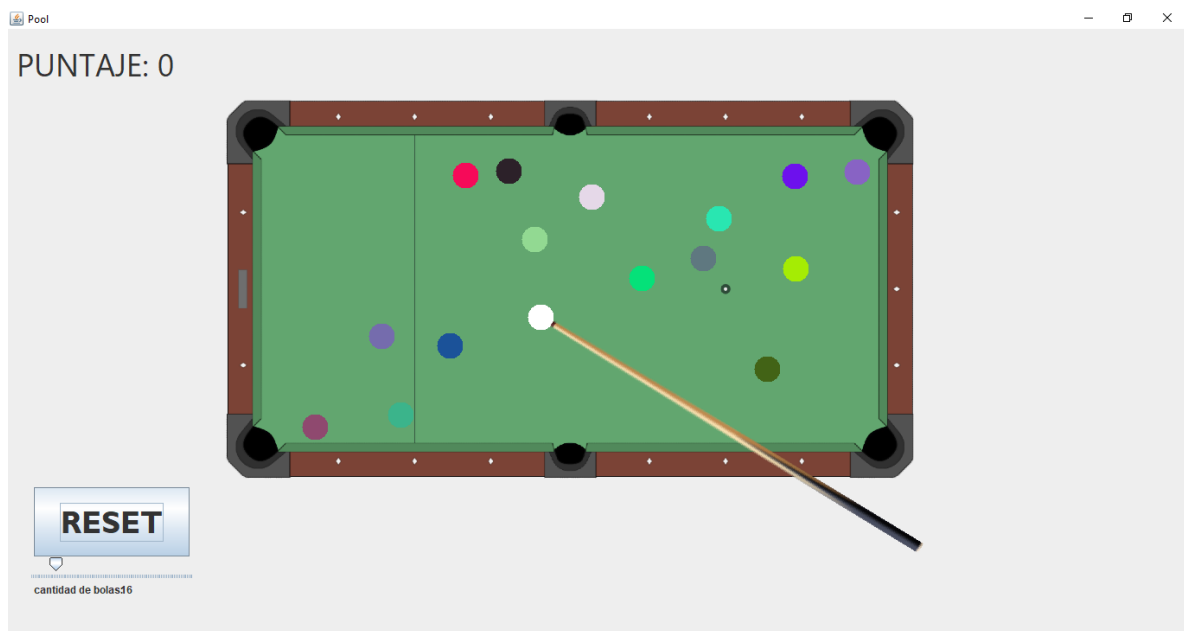
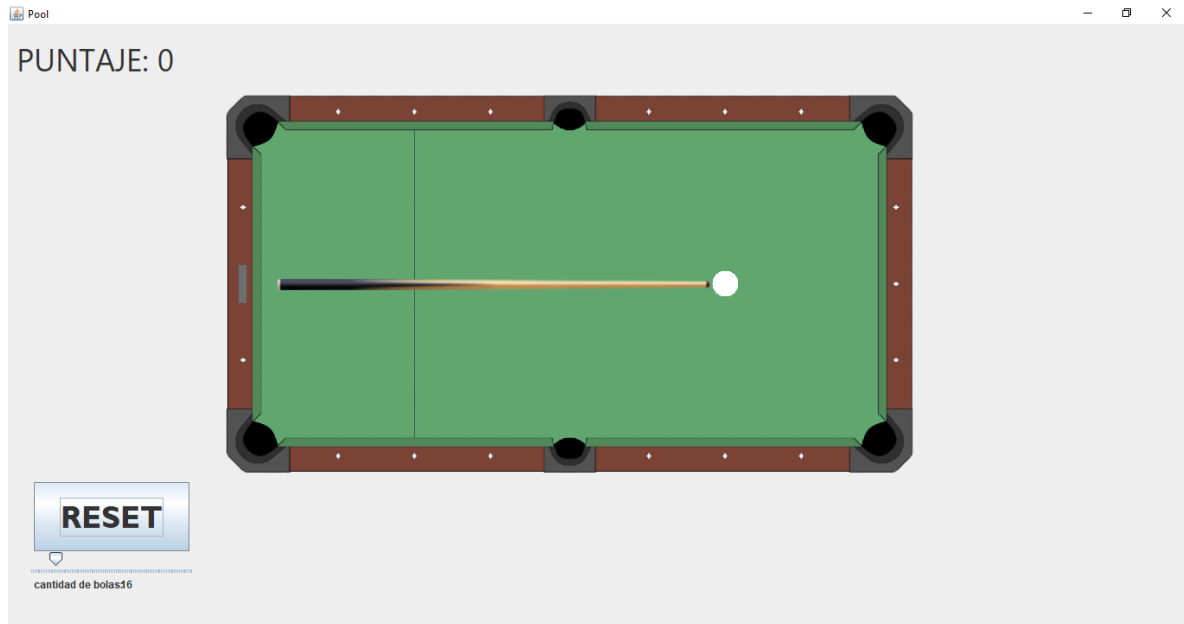
Diagrama de casos de uso



Patrones usados

- **Patrón Holder**
 - Lo utilizamos con el propósito de guardar un registro de los cambios de velocidades de las bolas en el instante que chocan entre ellas, para así ser aplicados al final de los cálculos. Se encuentra en la clase `AsignationDelayer`.
- **Patrón Singleton**
 - Hemos utilizado este patrón con la clase `P2DMath` para poder crear una única instancia que se encargue de realizar cálculos entre componentes de puntos 2D.

Imágenes de la interfaz



Decisiones creativas

- Decidimos crear la mesa de pool, el panel principal y el taco con la herramienta de JPanel de NetBeans, para así tener una mayor facilidad para implementar características como imágenes o listeners.
- Al crear las bolas decidimos utilizar puntos 2D para un mejor manejo de sus componentes, tanto en el apartado de precisión al hacer los cálculos como en el de complejidad al realizarlos, ya que anteriormente probamos con JPanel y con variables tipo double.
- Para no saturar los métodos que calculen datos de puntos 2D con operaciones creamos una clase que se encargue de estos llamada P2DMath.
- Creamos la clase AsignationDelayer para poder guardar los cambios en las velocidades de las bolas, para así ser aplicadas después de revisar todas las colisiones entre ellas en un frame.
- En la modalidad de juego decidimos que se pudiera jugar desde una sola bola (solo la bola blanca) hasta un tope de cien, pudiendo ajustar el número de bolas con una barra.
- En el taco, hicimos que se pudiera alejar y acercar a la bola para ajustar la potencia del tiro.

Problemas encontrados

- Uno de los problemas con el que nos topamos fue la interacción entre las bolas, específicamente al aplicar los efectos de estas en un choque de múltiples bolas debido que después del primer los datos iban variando lo que no representaba el efecto real y la no detección de las colisiones haciendo que las bolas se traspasaran entre ellas.