INFORME PROYECTO 6: MESA DE POOL

Grupo 11 – Proyecto: Mesa de Pool

Integrantes:

- Nicolás Pino
- Ignacio Barría Concha

Diagrama UML:

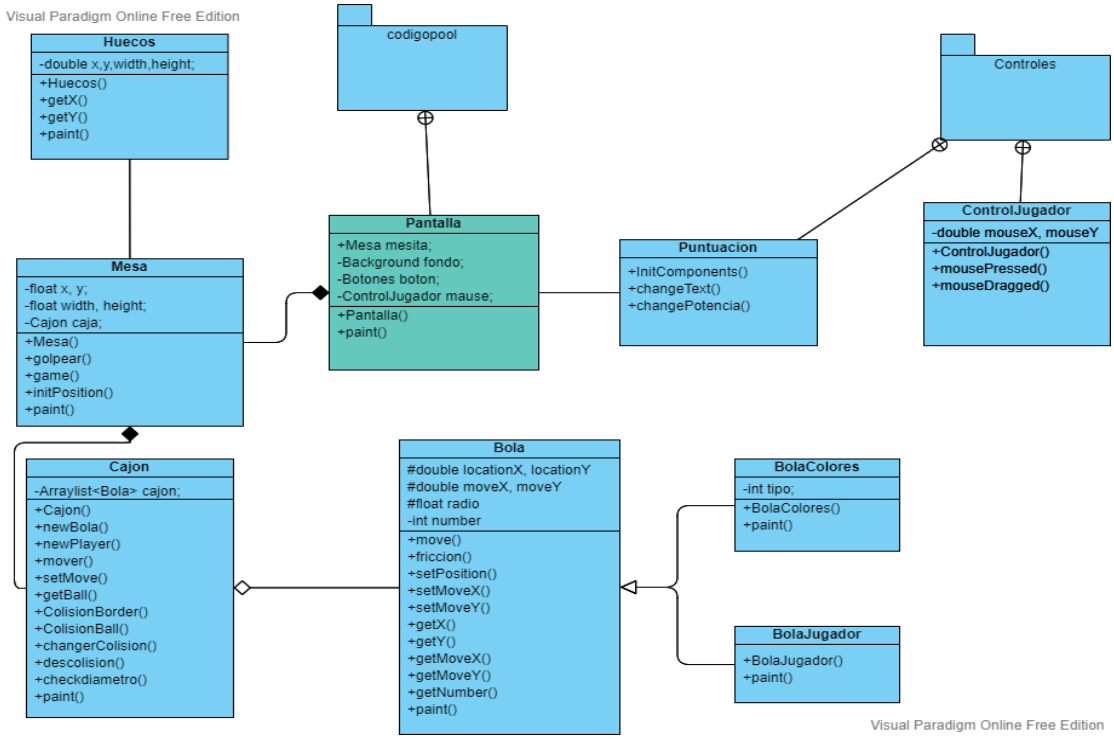
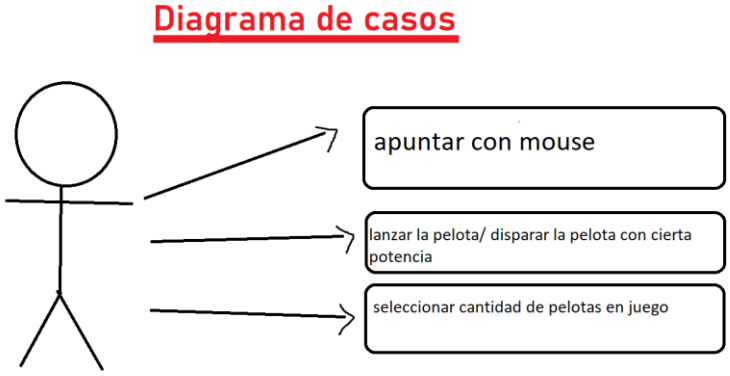


Diagrama de casos de uso:



Justificación de patrones usados

Se utilizaron solo dos patrones de diseño los cuales serían:

1) Diseño Composite. Este diseño se ocupó durante la creación de la mesa y sus partes las cuales son la mesa misma, los huecos, y las bolas de billar.
2) Diseño Builder. Este diseño se utilizó principalmente para la creación de distintas bolas a partir de una bola abstracta, estas son las bolas de colores, la bola jugador y la clase cajón que representa el arreglo de bolas, todas estas clases tienen distintas funciones para el funcionamiento del juego.

Por ultimo todas estas clases se juntan con clases de interfaces para el funcionamiento del programa.

Decisiones

Las decisiones tomadas a lo largo del proyecto se debieron gracias a la experimentación, estas son:

1) Control del mouse: se decidió ocupar el mouse como método de control en la bola del jugador ya que el ocupar el teclado era más complicado, aunque de igual manera se ocupa las "potencias" las cuales se imponen a través del teclado.
2) Iniciar la pelotas en triangulo: se planteó esto ya que el pool se juega con las bolas en posición de triangulo.
3) Dos modos de juego, rápido y completo: consideramos que al dejar las bolas del billar en una posición tendríamos que dejar cantidad de bolas en juego distintas en este caso para un juego más corto dejamos 8 de colores y el juego completo trae 15 bolas de colores.

Problemas encontrados

El mayor problema encontrado fue las colisiones entre las bolas, las cuales tenían que entender la lógica detrás de ella junto con la inercia, este problema se solucionó con mucha ayuda de compañeros y profesores pues esta mecánica es esencial para todo el juego y sin ella no se hubiera podido concretar. A la par de colisiones encontramos problemas con el control del juego el cual debía usar magnitudes de distancia de un punto a otro, en otras palabras, un vector que diera una dirección y velocidad. Finalmente, nuestro mayor conflicto fue un error de "parpadeo" de la pantalla al actualizarla, la cual se solucionó reuniendo todo a un JPanel y añadiendo un Timer para ejecutar la secuencia en un ActionPerformed.

Captura de Pantalla de la Interfaz