

Informe de proyecto Grupo n°24

Mesa de pool



Integrantes:

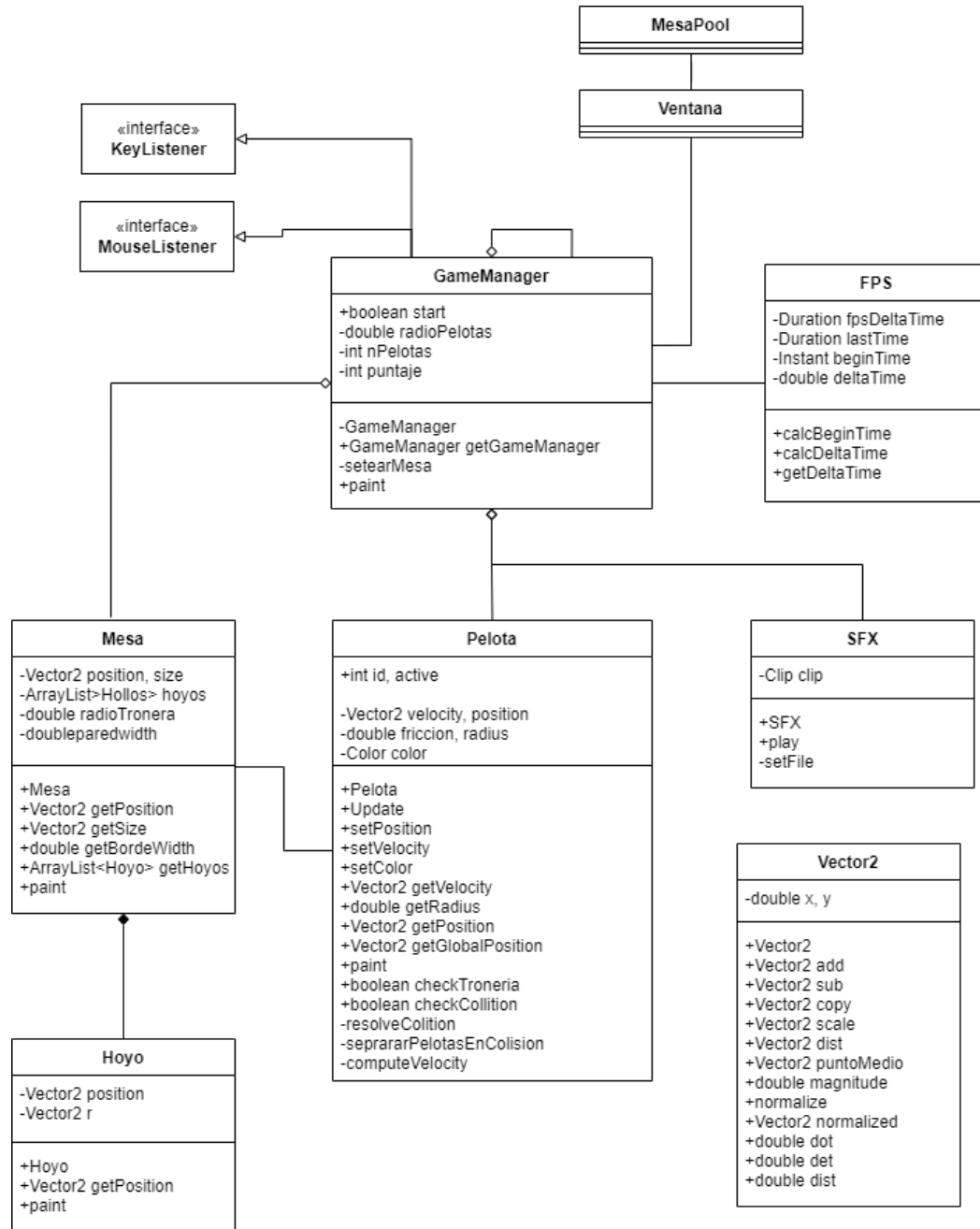
Sebastian Aguayo Segura

Franco Vidal Carvajal

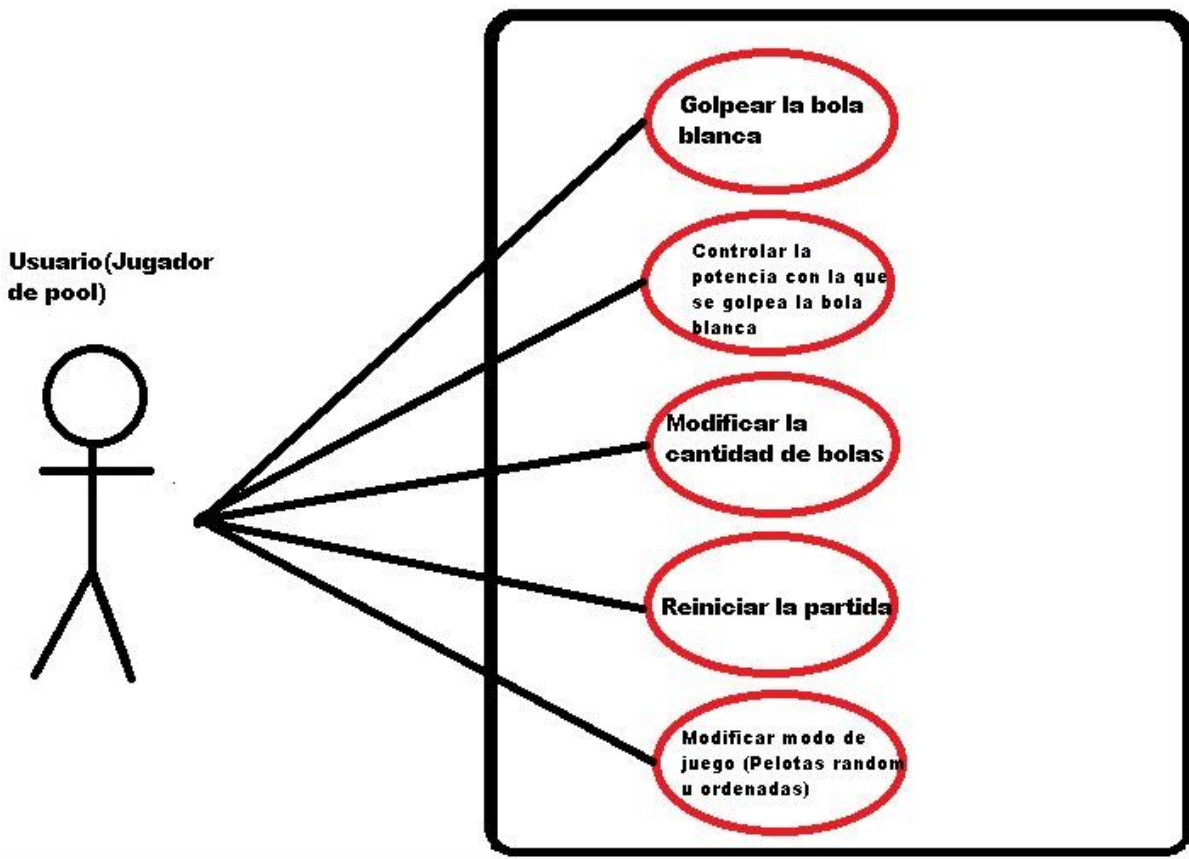
Enunciado

Mesa de pool: 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.

UML



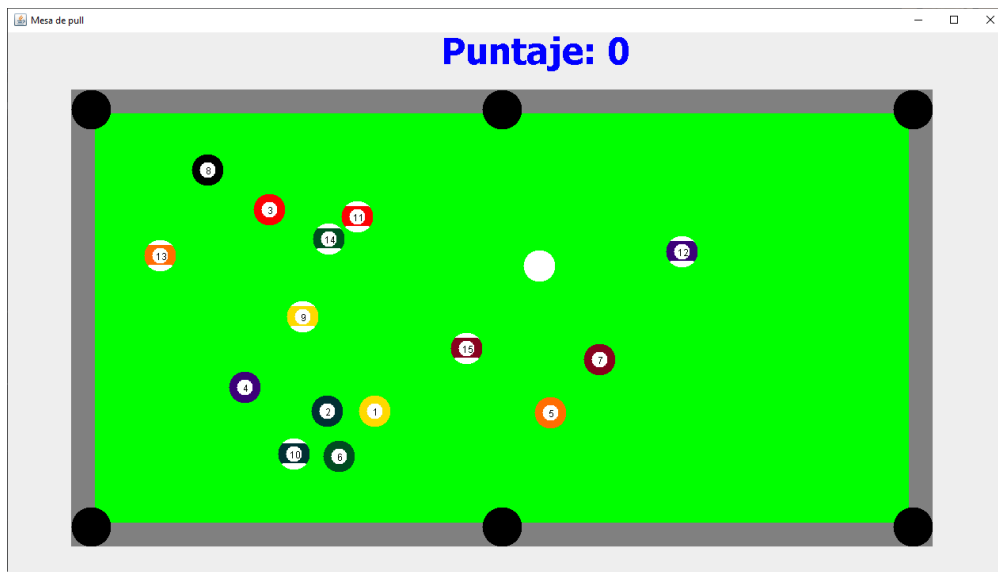
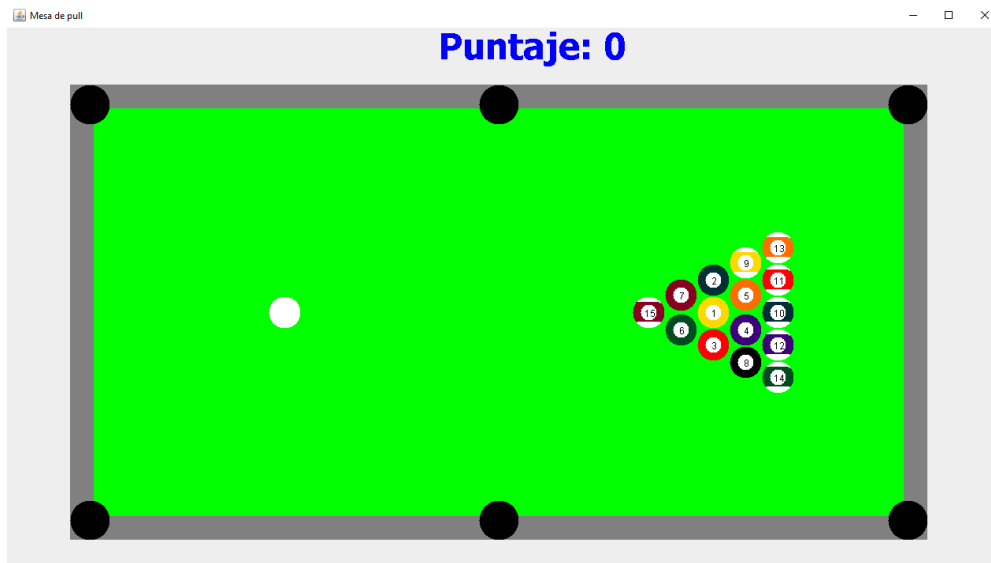
Use case diagram



Patrones utilizados:

El único patron utilizado fue singleton en la clase GameManager porque solo necesitamos uno.

Interfaz



Como se puede observar en la interfaz, tenemos dos “modos de juego”, en los cuales uno es posicionando las bolas de manera random y el otro de manera ordenada.

Para jugar con las bolas en posiciones random se pulsa la tecla (R).

Para jugar con las bolas ordenadas se pulsa la tecla (T), la cual a la vez sirve para reiniciar la partida.

Decisiones tomadas:

Al comienzo de este proyecto nuestra primera idea era hacerlo de una manera mas realista a través de la fisica pero debido a la complejidad y la falta de tiempo cambiamos a un formato mucho mas simple y rapido.

Antes de cambiar todo el proyecto utilizamos los siguientes patrones:

- Decorator: Todo partía de un objeto básico llamado gameObject y desde ahí se le iba añadiendo distintas funcionalidades gameObject <|- rigidbody <|- pelota o gameObject <|- mesa, etc.
- Singleton: Solo necesitabamos un game manager y un physics engine o si no se podían generar problemas.
- State: Las pelotas se movian de acuerdo a un estado, los cuales calculaban el movimiento según matemáticas.

Luego del cambio de idea todo funcionó mucho mejor y pudimos desarrollar el proyecto de forma mas organizada y preguntarle mas cosas a nuestro referente, que en la primera etapa no recurrimos mucho.

Problemas encontrados y autocrítica

Nuestro principal problema radica en nuestra primera intencion de hacer el proyecto demasiado realista que no iba de la mano con el tiempo que disponiamos para desarrollarlo.

La interaccion con nuestro referente para poder desarrollar el proyecto, ya que como teniamos topon de horario con nuestro referente, recurrimos a otra practica y siempre le consultabamos a él en lugar de a quien nos correspondia.

El no manejo al 100% de la plataforma git, ya que algunas veces tuvimos complicaciones al momento de hacer pull y push, como es en el caso del final en donde forzamos un push y se borraron otros 2, que logramos recuperar.

Como autocrítica, principalmente nuestro error fue confiarnos de que podiamos desarrollar el proyecto en poco tiempo, lo que nos jugó una mala pasada ya que se nos juntaron muchas cosas de la Universidad, tambien nos faltó comunicarnos mas con nuestro referente y sentimos que tuvimos que haber buscado a un tercer integrante para poder hacer el proyecto.