

SNOOKER REBOOT

Francisco Miguel Galván Muñoz

- 1.-*Temática de juego*
- 2.-*Diagrama de clases*
- 3.-*Ecuaciones físicas utilizadas*
- 4.-*Efectos incorporados*

1.- *Temática de juego*

El snooker es un juego derivado del billar que se juega en una mesa cubierta de un paño verde y rodeada por bandas. Las mesas de esta mesa son 3,56 metros de largo por 1,78 metros de ancho (esta proporción ha sido mantenida en el proyecto). En el snooker tenemos 21 bolas de colores diferentes. Dependiendo del color de la bola esta reportará más puntos al meterla por alguno de los hoyos.

Las puntuaciones son:

- Rojas : 1 punto
- Amarilla: 2 puntos
- Verde : 3 puntos
- Marrón: 4 puntos
- Azul: 5 puntos
- Rosa : 6 puntos
- Negro : 7 puntos

En el snooker tradicional se juega primero una bola roja para poder tener la opción de meter una de otro color, y solo se pierde el turno si no se mete ninguna bola o si metes/tocas una bola de color diferente a la roja antes de meter/tocar una roja. Cuando no queden bolas rojas en mesa, se meterán todas las restantes para acabar el juego.

En esta revisión del snooker, he decidido adaptar la forma de jugar para hacerla más accesible, para que el público objetivo sea más amplio.

En todo momento vas a poder disparar a cualquier tipo de bola. Estas conservan sus puntuaciones antes descritas.

Solo podremos disparar de nuevo una vez todas las pelotas estén quietas, al no poder asegurar esto siempre, a los máximo 10 segundos se pararán todas.

El jugador tendrá 5 tiros máximos para puntuar el máximo posible.

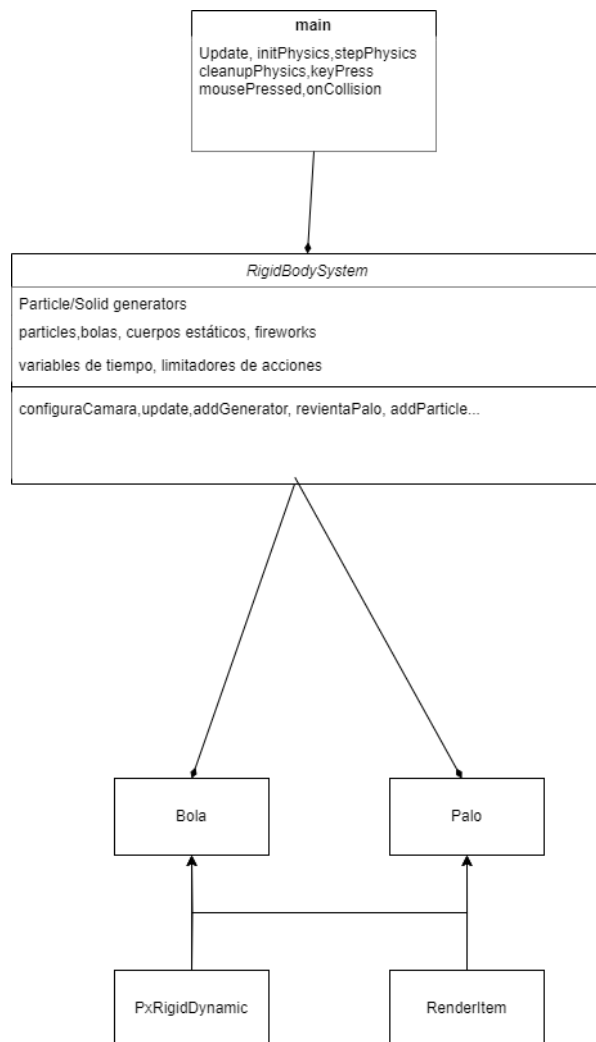
Solo se perderá turno si al disparar no se emboca ninguna bola, sea se, no se puntúa.

El juego terminará por dos condiciones:

- Has conseguido embocar todas las bolas rojas y por lo tanto has ganado y se mostrará tu puntuación en consola.
- Has gastado tus tiros máximos y has perdido.

En cualquier caso, se instanciará un torbellino y se llevará todas las bolas terminando así con el juego.

2.- Diagrama de clases



3.- Ecuaciones físicas utilizadas

Las ecuaciones físicas las vemos presentes en los generadores de fuerzas.

De todos los presentes he utilizado tres (cuatro si contamos la gravedad implícita en los objetos sólidos de Physx).

3.1.-Torbellino

Por una parte he utilizado el torbellino que se aplica en las bolas de billar cuando el juego termina, llevándose consigo todo lo que encuentre por delante.ç

Este torbellino funciona calculando la fuerza en función de la posición del cuerpo con relación a la posición del torbellino, y se aplica como fuerza de arrastre en la dirección opuesta al movimiento del objeto.

Saber que si la masa es mínima, no se aplica la ecuación.

```
Vector3 solidPosition = bola->getBola()->getGlobalPose().p;  
if (fabs(1 / bola->getBola()->getMass()) < 1e-10) return;  
Vector3 velocidadtorbellino = k1 * Vector3(-(solidPosition.z - this->position.z), 75 - (solidPosition.y - this->position.y), solidPosition.x - this->position.x);  
Vector3 v = bola->getBola()->getLinearVelocity() - velocidadtorbellino;  
float drag_coef = v.normalize();  
drag_coef = k1 * drag_coef + k2 * drag_coef * drag_coef;  
Vector3 dragF = -v * drag_coef;  
bola->getBola()->addForce(dragF);
```

El torbellino, recibirá por su constructora el vector de la fuerza aplicada, la constante k la cual determinará la magnitud del coeficiente de arrastre (junto con su cuadrado) la posición del torbellino y lo que este abarca de espacio.

En cuanto a los valores pasados son todos estándar menos la fuerza, la cual es muy pequeña para apreciar cómo se forma el torbellino en la mesa y las pelotas no salen volando repentinamente.

3.2.-Explosion

También encontramos la explosión. Esta ha sido añadida para darle otro toque al juego. Se puede accionar siempre que se quiera hace que las bolas se muevan del centro del tablero hacia afuera, por lo que resulta muy útil para conseguir puntos. La explosión funciona, primero comprobando que no tenemos masas ínfimas, luego aumenta el radio de la explosión con el tiempo y dependiendo de la distancia entre el objeto y el centro de la explosión se aplica más o menos fuerza.

```

void ExplosionGenerator::updateForce(physics::PxRigidBody* solid, double t)
{
    if (fabs(1 / solid->getMass()) < 1e-10) return;

    radius += velocity * t; // El radio aumenta con el tiempo

    Vector3 aux = solid->getGlobalPose().p - position;
    double r = sqrt(aux.x * aux.x + aux.y * aux.y + aux.z * aux.z);

    if (r < radius)
    {
        Vector3 f = (k / (r * r)) * aux * exp(-t / timeConstant);
        solid->addForce(f);
    }
}

```

3.3.-Muelle

El muelle es otro de esos elementos cómicos implementados en el juego. Este juega el papel de multiplicador de puntos. Este multiplicador se activará cuando el muelle que une a las dos partículas que están sobre nosotros se rompe.

```

Vector3 f = p->getPos()-p;
f -= other->getPos()-p;

// Length
float length = f.normalize();
length = (length - this->length);
if (length > 3.46133e+17)
{
    multiply = true;
}
if (length > 0.0f) {
    f *= -(length * k);
    p->addForce(f);
}

```

Este sistema recibe como valores la constante K, la variable length la cual describe la distancia máxima del muelle en reposo, y por último V que es la posición de la otra partícula.

3.4.- Gravedad

Esta fuerza se aplica por defecto en los sólidos rígidos y viene definida por:

```

//Mira si la partícula tiene masa infinita
if(fabs(1 / p->getMass()) < 1e-10)
    return;

p->addForce(_gravity * p->getMass());

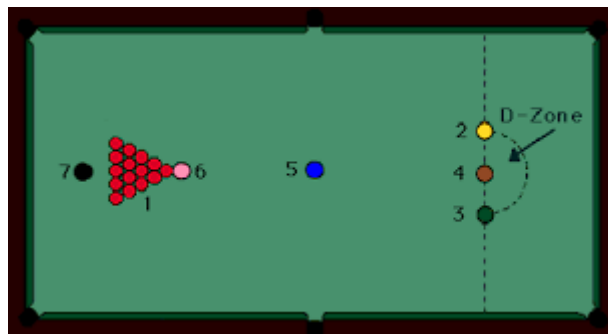
```

Aunque no sea un generador de fuerzas al uso, el *Palo* usa una clase *Fuerza Instantánea* para aplicar fuerza sobre la bola blanca. (!Sin necesidad de tocarla, como un Jedi!).

4.-Efectos incorporados

En este juego encontramos varios actores principales como podrían ser las bolas y el taco de billar, pero algunos otros que aunque están ahí puedes pasarse por alto.

- Como **escenario** encontramos una mesa de billar clásica con sus 6 hoyos donde poder puntuar, esta, de color verde, está rodeada por una franja dorada que hace de filo de la mesa (muchos recuerdos de apoyarme en esa franja cuando era pequeño porque no llegaba a golpear bien).
- El **palo** es uno bastante minimalista que cumple su función, rota alrededor de la pelota a la hora de poder golpearla, se mueve con ella cuando esta ha sido desplazada y por último no interactúa con ningún otro elemento de la mesa, para hacer la experiencia lo más cercana a la realidad posible. Podremos cargar el golpeo todo el tiempo que queramos, por lo que medir los tiempos es crucial para clavar un buen golpe.
- Las **bolas** están colocadas exactamente como pide el reglamento del snooker. Cada una tendrá su puntuación asociada y, exceptuando las rojas, siempre que metamos una esta volverá a aparecer en su sitio inicial.



Demostración de una mesa de snooker

- Otro elemento a tener en cuenta es el muelle que tenemos sobre nuestras cabezas, ya que en el momento en el que ese muelle se rompa... ¡puntos dobles!
- También tenemos dos ejemplos de efectos especiales, uno de ellos los fuegos artificiales que saldrán disparados al cielo una vez hayamos metido alguna pelota y por otra parte tenemos unas partículas que aparecen de nuestro palo mientras cargamos el disparo, a modo de feedback para el usuario.
- Mencionar también la pintura blanca del campo, ya que ahí es donde van colocadas tres de las pelotas y donde aparecerá la pelota blanca en un principio. Tendremos la capacidad de mover la pelota blanca a nuestro antojo

(dentro del rectángulo blanco) al principio de cada partida, para así preparar más cómodamente nuestro tiro.