
Grupo SG2

Marina Hernández Bautista
Román Larrosa Lewandowska

Diseño de la aplicación

What a Night

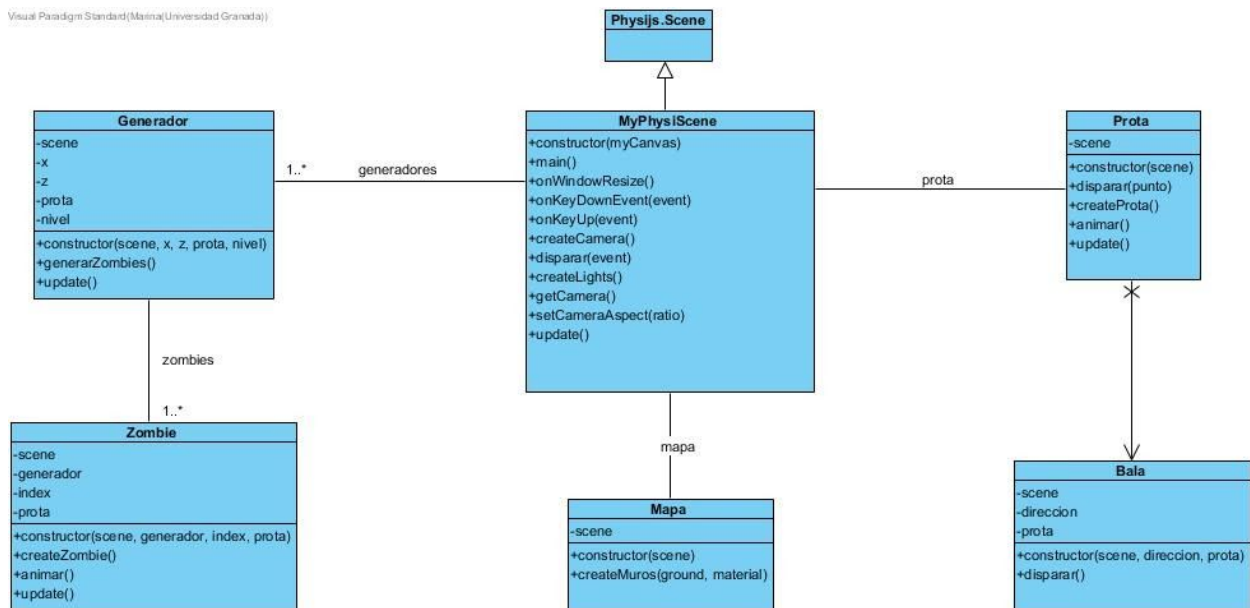
DISEÑO

Vamos a hacer un recorrido explicando las diferentes clases que conforman el juego. Se trata de una descripción simple puesto que todo el código se encuentra extensamente comentado:

- **MyPhysiScene**: esta escena tiene el funcionamiento básico de una escena de Physijs. Añadimos algunos elementos adicionales como: una cámara isométrica, un raycaster, unas luces fijadas al personaje, eventos para manejar teclado y ratón.
- **Prota**: define una caja física que contiene un mesh que conforma el objeto jerárquico (aspecto visual del personaje). El personaje anima su movimiento mediante Tween y gestiona las colisiones con los zombies. Además, genera una bala y la dispara cuando un evento de ratón llama a la función `disparar()`. Controla su movimiento mediante eventos de teclado asociados a las teclas WASD.
- **Mapa**: define el suelo, las paredes y los muros mediante cajas físicas. Los muros se dispondrán aleatoriamente en cada partida.
- **Zombie**: define una caja física que contiene un mesh que conforma el objeto jerárquico (aspecto visual del zombie). El zombie anima su movimiento mediante Tween. Controla su movimiento y orientación en función de la posición del protagonista.
- **Generador**: se trata de una esfera sin física que conforma su aspecto. El generador se encarga de crear un array de zombies y define la posición en que se generan los zombies al cabo de un tiempo que viene marcado por el nivel del juego. Se encarga también de llamar a la función `update` para cada uno de sus zombies asociados.
- **Bala**: esfera física que crea el protagonista en su método `disparar` y que se encarga de gestionar su colisión con los objetos de la escena, eliminando a un zombie y destruyéndose a sí misma cuando se produce la colisión.

DIAGRAMA DE CLASES

Visual Paradigm Standard (Marina@Universidad Granada)



El diagrama de clases no recoge con exactitud el contenido total del juego ni su funcionamiento, debido a que Javascript no es un lenguaje orientado a objetos y algunos conceptos escapan a la representación mediante el diagrama de clases.

Al final, las clases **Zombie**, **Prota**, **Bala** y **Mapa**, representan objetos físicos modelados con objetos físicos que proporciona **Physijs**.