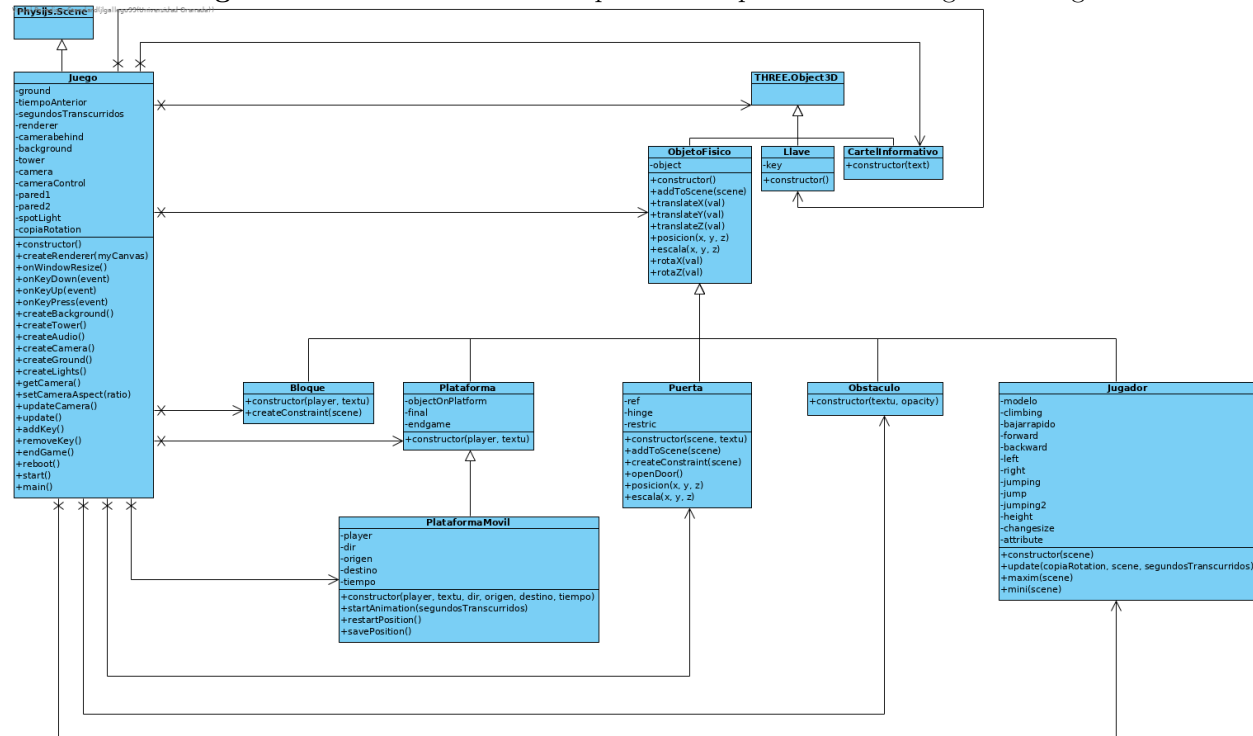


Diseño de la aplicación: La Torre

Jose Luis Gallego Peña La estructura de la aplicación se puede ver en el siguiente diagrama de clases:



Las partes de la aplicación son las siguientes: - **index.html** es el que contiene todas las referencias a todos los scripts y la interfaz de usuario, con los corazones, la llave y las opciones de iniciar partida, reiniciar, game over, etc.

- **Juego** es la clase principal, que instancia una escena Physijs. Crea e instancia todos los objetos que se ven en el juego y llama a los updates correspondientes. También controla la luz, cámara y objetos básicos como la torre, el suelo, el fondo o algunas paredes. Define una animación para el inicio del juego. Recibe inputs de teclado, controlando distintas variables de las clases para que hagan algo.
- **Llave**, es un sprite que representa una llave, que se puede encontrar en algunos lugares y coger con el botón E, lo que la hará desaparecer.
- **CartelInformativo**, es una geometría de texto que se usa al inicio para mostrar los tutoriales del juego dentro de este.
- **ObjetoFisico**, el resto de objetos de la escena a continuación heredan de esta clase. El objetivo de esta clase es proporcionar los métodos de traslación, escalado, posición y rotación para poder tocar estos valores desde la escena al instanciar los objetos de Physijs. Las instancias se colocan y redimensionan y una vez hecho esto se llama al método correspondiente para fijarlos en la escena physijs.
- **Bloque**, es un bloque de piedra con textura personalizada que el personaje podrá mover o subirse encima suya. Tiene un constraint para limitar su movimiento, sobre todo en el eje Y, y que no haga cosas raras.
- **Plataforma**, constituyen la principal estructura de la torre, es por donde va saltando el personaje y gestiona el hecho de que el personaje esté encima suya o no, y si esa plataforma es la última, para que cuando se colisione el jugador con ella se llame al final del juego.
- **PlataformaMovil**, un tipo especial de plataforma que se puede mover en X, Y o Z una distancia empleando un tiempo, usando una animación TWEEN. Tiene el jugador para que este se mueva

automáticamente con la plataforma. Permite definir e iniciar varios tipos de animaciones según dirección. Conoce además su posición inicial, para que el jugador la pueda reiniciar.

- **Puerta**, consiste en un cilindro y una caja que forman una puerta usando un HingeConstraint. Tiene un motor angular que se activa al tener el jugador una llave, estar cerca de una puerta y pulsar el botón de acción.
- **Obstaculo**, un simple bloque duro que no se puede atravesar. Se usa para ponerlo entre la puerta final, o para hacer de techo y que el personaje tenga que agacharse por ejemplo.
- **Jugador**, el personaje principal que mueve el jugador. Se puede mover en el espacio 3d, saltar dos veces, coger llaves y usarlas y agacharse. Para agacharse ha sido necesario guardar el modelo y crear un objeto mas pequeño cuando se agache, puesto que Physijs no permite los escalados. Almacena los datos de dirección que cambia el teclado, para saber qué translate o rotation hacer para el movimiento. Guarda una rotación fija para que el personaje no se vea afectado de más por la física y de vueltas en el aire, cayendo de forma incorrecta. Para los saltos se controla una altura local, para que independientemente de lo alto que esté el personaje en la torre, siempre salte lo mismo. Además, después del salto se obliga al personaje a bajar más rápido, porque si no el motor de físicas actua de forma rara sobre el personaje, haciendo que baje de una forma poco suave y escalonada.