

PRÁCTICA 2

DISEÑO E IMPLEMENTACIÓN DE UN JUEGO BASADO EN “BATTLE CHES”

NOMBRE DEL JUEGO

SCARLET CHES

PERSONAS QUE FORMAN EL GRUPO DE PRÁCTICAS

Nombre de la Estudiante:

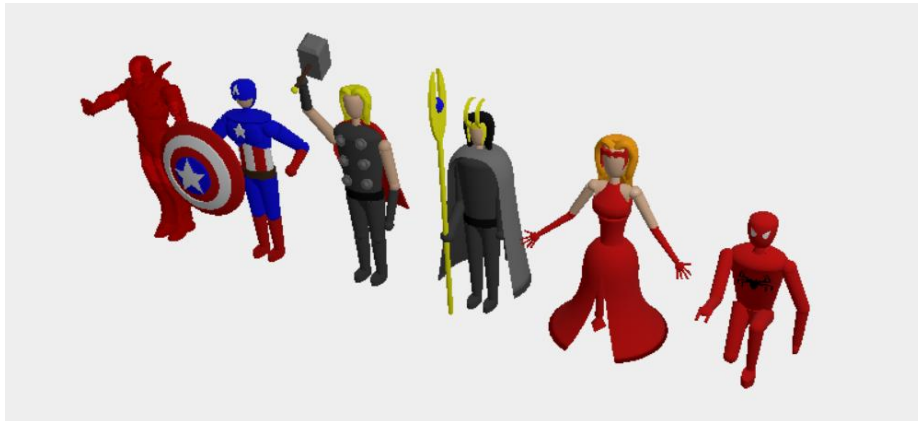
Natalia García Ortega

INTRODUCCION



En esta práctica, he realizado un ajedrez basando el modelado de las figuras en los superhéroes típicos. La batalla toma lugar en una ciudad, aunque con todo esto entraremos en detalle más adelante.

MODELADO DE LAS PIEZAS



Voy a comentar brevemente como se ha realizado el modelado de las piezas. Comenzando por las partes que comparten entre todas o varias:

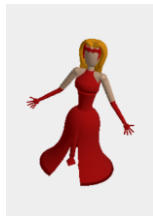
Para la hacer **cabeza** se ha utilizado un shape revolucionado, añadiendo a los que llevan **mascara** el shape de la misma quitándole el bisel.

Otra parte que comparte muchas figuras es el **torso y las piernas**, el cual se ha realizado con esferas y cilindros escalados para conseguir la forma deseada.

Y por último el **pelo**, aquellas figuras que tienen pelo, se ha construido realizando una revolución del shape principal, del cual nos quedamos con la forma redondeada y de atrás realizando una operación de intersección y otra de sustracción de una esfera y una caja sobre el shape principal. Y para la parte delantera, se ha conseguido con el barrido de un tubo para que siga esa forma.

Ahora continuamos con las piezas en más detalle:

Reina:



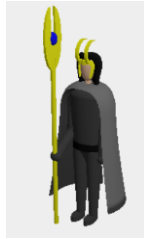
Para hacer la falda, se ha realizado una revolución del shape base, a la cual se le sustrae un clone de la misma trasladado un poco mas abajo, para dejarla hueca. Y para la apertura de la falda, se ha extruido un shape con esa forma y se le ha sustraído a la pieza principal. El abdomen y el torso son revoluciones del shape, quitándole al torso algunos recortes con shapes extruidos y cilindros, añadiéndole otro shape sin bisel para hacer la forma de las clavículas. Y por último para los brazos, se han construido con esferas y cilindros.

Caballo:



Las estrellas y la letra A se han hecho de la misma forma, realizando el shape y quitándole el bisel. Para el casco se ha partido de la cabeza base, quitando el barrido de un shape para hacer la apertura del casco, sumándole por ultimo la letra A. Para el escudo se han ido sumando cilindros de distintos tamaños y la estrella, para a continuación hacer una intersección de una esfera para hacer la forma curva del exterior y una sustracción de otra esfera para la forma interior. Para el abdomen se han hecho dos cilindros base del mismo tamaño, a uno de ellos el quitamos dos cajas y le

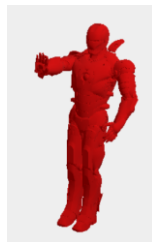
sumamos el otro cilindro, quedando esas líneas verticales. Para el torso se han extruido dos cuadrados con bisel sumándole un shape revolucionado y escalado para hacer las clavículas.

Rey:

Para los cuernos de la máscara se ha sustraído un cilindro interior al exterior, colándolo a los dos lados de la máscara. Para la capa se ha revolucionado un shape al cual con intersecciones y recortes de cajas se le ha dado la forma deseada. Para el cetro se han sumado dos cilindros, uno de ellos escalado y rotado, para luego quitarle una caja y sumarle una esfera.

Alfil:

Las partes únicas del alfil son el martillo, el cual es una caja a la cual se le han sustraído varias cajas para hacer la forma que tiene la piedra en los extremos, a la pieza principal le sumamos un cilindro y una esfera. Para la capa, hacemos un shape revolucionado y con una intersección de una caja conseguimos que sea solo por detrás. Y la decoración de torso es simplemente un cilindro y una esfera sumándose en cada una de esas posiciones.

Torre:

Es la pieza cargada de disco, la cual he sacado de: <https://free3d.com/3d-model/ironman-rigged-original-model--98611.html>

Peón:

Las partes únicas que tiene el peón son la araña del traje y los ojos, los cuales son shapes sin bisel.

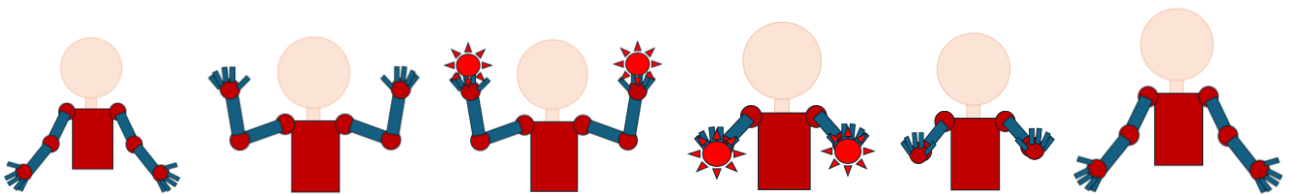
ANIMACIÓN

La pieza que he articulado ha sido la reina. El modelo de la pieza va a estar basado en un personaje llamado 'Scarlet Witch'.



La animación va consistir, en que la figura subirá los brazos con una flexión en el codo de 90º, donde en sus manos, dos esferas irán haciendo un escalado simulando que la figura está haciendo unas bolas de energía. A continuación, la pieza estará los brazos hacia la posición de la figura que esta atacando y las bolas de energía desaparecerán cuando se haya alcanzado la posición del enemigo. Para finalizar, la pieza vuelve a su posición original.

Animación:



Para visualizar la animacion de la reina se ha proporcionado la carpeta reinaAnimacion, en la cual en el panel de control encontramos un botón para activarla.

Y tambien he realizado la animacion para el movimiento de las piezas. El cual consiste en que la pieza se eleva, se mueve a la posicion que le corresponde horizontalmente y finalmente baja hasta colocarse en ella. Simultaneamente, si el movimiento era una captura, la pieza que ha sido comida tambien se anima, se mueve horizontalmente hasta el lateral del tablero que le correspoanda y se coloca mirando hacia el tablero.

EXPLICACIÓN DE ALGORITMOS Y LÓGICA DEL JUEGO

Comenzamos con la creación del tablero, para ello he creado la clase **Board**, a la cual le pasamos en su constructor el tamaño de las casillas (por defecto 0.5) y las filas y columnas (8x8). Creamos las casillas como cajas de ese tamaño y una altura de 0.05, las cuales se crean con dos bucles anidados, asignándole un material blanco o negro dependiendo si es par o impar. Calculamos las posiciones 'x' y 'z' para centrar el tablero en (0,0) y trasladamos también hacia abajo para que el tablero quede justo por debajo de los ejes 'x' y 'z', de esta forma se nos facilita el proceso de colocar las piezas, ya que las he modelado por encima de estos.

La creación del tablero la realizamos con un método en MyScene.js el cual se encarga de crearlo y añadirlo a la escena.

Para conseguir los movimientos validos de cada pieza, cada clase tiene un método `getValidMoves()` el cual calcula las casillas a las que se puede mover.

Seguimos con el **estado interno del juego**, el cual guardamos en una serie de variables:

- `this.boardState = Array(8).fill(null).map(() => Array(8).fill(null));` //tablero
- `this.turn = 'white';` // jugador activo
- `this.selectedPiece = null;` // pieza actualmente seleccionada
- `this.highlightMeshes = [];` // meshes de casillas resaltadas
- `this.pieceMeshes = [];` // meshes de piezas para picking
- `this.capturedWhite = [];` // piezas capturadas lado blanco
- `this.capturedBlack = [];` // piezas capturadas lado negro

En las cuales nos apoyamos para posicionar las piezas y actualizar sus movimientos.

Para posicionar las piezas al inicio de la partida usamos el metodo `colocaPiezas()`, el cual se ayuda en dos de sus funciones internas, `Place` y `PlaceTorre` (esta ultima ya que la anterior daba problemas con la pieza cargada de disco), instancia una, la posiciona en la casilla correspondiente, la orienta si es negra y la añade a la escena y al sistema de picking.

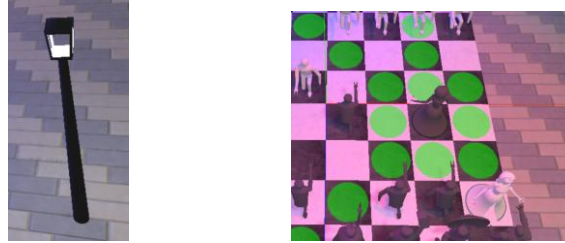
Se definen los dos materiales que pueden ser las piezas, blanco o negro, se calcula la posición (x, z) de cada pieza en función del tamaño de las casillas (`casillaSize`) y se centran respecto al tablero. Cada pieza se almacena en la matriz `boardState`. Se colocan las piezas.

Para actualizar el tablero, utilizamos el metodo `updateBoardState(from, to)` cada vez que un jugador realiza un movimiento, es decir, se encarga de mover una pieza de la posicion 'from' a la posicion 'to', manejando si hay captura y aplicando las animaciones cuando sean necesarias. Cuando una pieza es capturada se elimina del sistema de picking

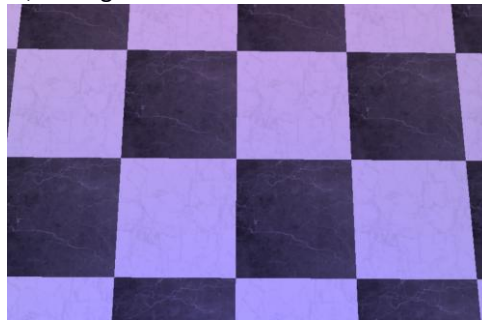
Para hacer el '**picking**', utilizamos `THREE.Raycaster`, inicializamos en el constructor todo lo necesario para utilizarlo, además de un grupo para guardar las casillas a las que se puede mover una pieza y añadimos un listener. Seguimos con el metodo `onPointerDown()`, el cual se encarga de calcular y normalizar las coordenadas según el tamaño del canvas, si el jugador ha clicado sobre una pieza, el metodo con ayuda de otros metodos que ya hemos visto y con el metodo `colocaHighlights(moves)` coloca sobre el tablero unos circulos que simbolizan las casillas a las que esa pieza se puede mover, si el jugador ahora clikea sobre uno de estos circulas, se considera como un movimiento y se llama a la funcion que actualiza el tablero con esta informacion.

LUCES, TEXTURAS Y DECORACION

Para las luces, he añadido dos fijas de colores: una luz azul sobre el lado de las blancas y una morada sobre el lado de las negras. Hay otras cuatro luces fijas, que se sitúan sobre unas farolas colocadas a modo de decoración. Hay una luz que va cambiando durante el juego: cada vez que selecciona alguna de las reinas, aparecerá una luz roja encima de ella, no desaparecerá hasta que o bien se seleccione otra pieza o el movimiento de la reina.



Los materiales basados en texturas para el canal de color los he aplicado en el tablero, las casillas blancas tienen una textura de mármol blanco, y las negras, de negro:



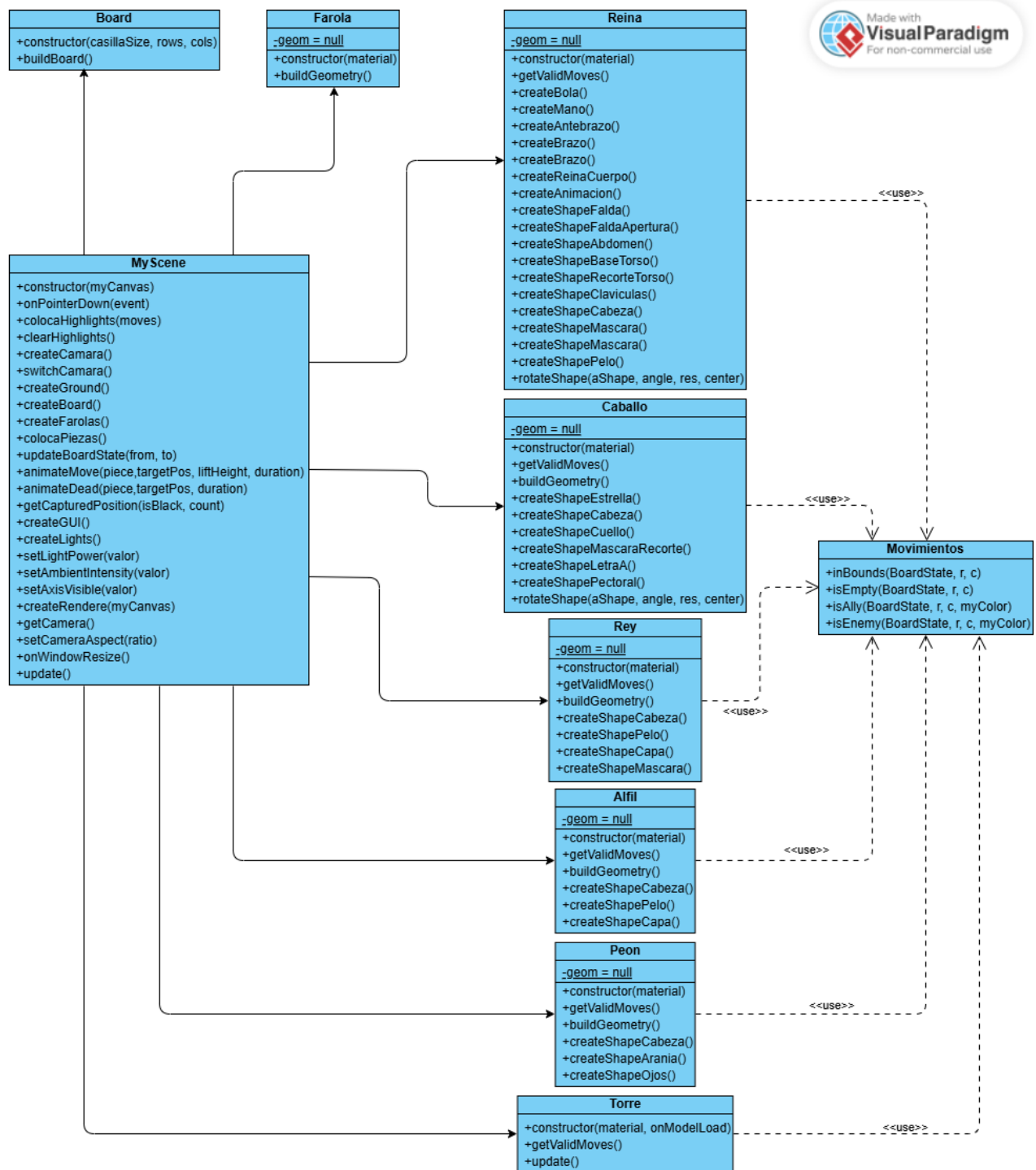
Para el material con una textura en el canal de relieve, he creado un suelo, a modo de decoración y para las piezas capturadas, el cual tiene relieve en las baldosas:



Para el fondo he puesto una imagen de una ciudad, ya que encaja con la tematica del proyecto:



DIAGRAMA DE CLASES



MODELO JERÁRQUICO

