

Práctica 2

Por:

Pablo Quesada rojo
Guillermo Medialdea Burgos

Descripción original

El jugador comienza en el patio de una iglesia, donde ha llegado con su coche. Para abrir la puerta y acceder a la habitación, se encuentra con una cadena, que puede romper con las tenazas que lleva en el coche.

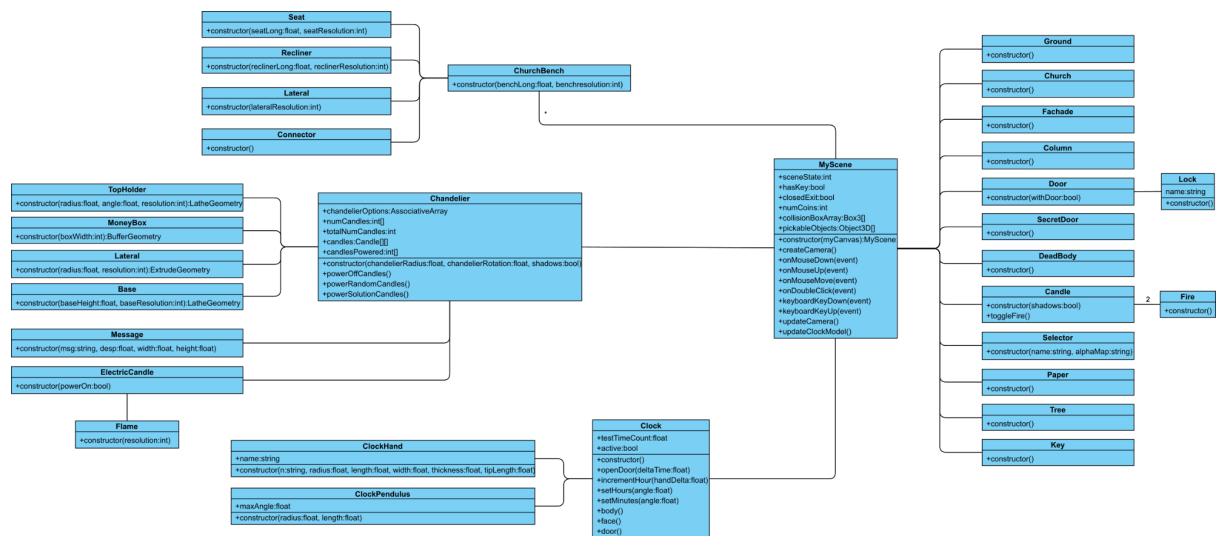
Al acceder, se cerrará la puerta y, al fondo, a los pies del altar, hay un cadáver, se debe tomar la nota que hay sobre él, lo que dará la pista para el siguiente puzzle: dado que consiste en encender los candelabros que hay en las columnas con la combinación correcta, la nota contiene la información necesaria para poder hacerlo, estos se van a poder encender con un mechero que posee el jugador.

Una vez superado el puzzle, se abrirá la puerta de una pequeña sala lateral en la que hay una serie de monedas que serán necesarias para el puzzle que se debe resolver a continuación. En este puzzle, hay que poner un número determinado de monedas, que se puede saber por uno de los cuadros que se encuentran en el muro opuesto al actual, en una máquina con velas eléctricas: las monedas se introducen por una ranura (se seleccionan las monedas y se hace click en la ranura) y se acciona un interruptor que enciende las velas si la combinación es correcta o devuelve las monedas si no lo es. Si se encienden las velas van a permitir saber una hora que va a ser necesaria para el puzzle final en el que habrá que modificar la hora de un reloj a la dada en el puzzle anterior para poder acceder a la llave que abra la puerta de la iglesia y el jugador pueda salir, esto se hará pulsando en el reloj, lo que accede a un modo de modificación de la hora de manera que se debe salir de este modo para verificar si la hora es correcta. En el momento en que se introduzca la hora necesaria se abrirá un compartimento del reloj con la llave.

Cambios

Respecto a la descripción original se han hecho algunos cambios. El puzzle de las tenazas junto al coche, ha sido eliminado. El cadáver ha sido sustituido por un dibujo. El puzzle de las monedas ha sido simplificado.

Diagrama de clases



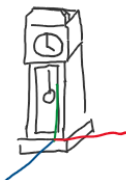
Modelos jerárquicos

Puerta de la iglesia



hinge (Rotación $(0, y, 0)$)
 ↳ door (Posición $(x, 2, 0, 0.3)$)
 ↳ key (Posición $(1.84, y, z)$) Rotación $(0, 0, z)$
 ↳ lock (Posición $(1.84, 1.6, 0)$)

Reloj con manecillas y péndulo



clock ()
 ↳ pendulum (Posición $(0, y, 0)$) Rotación $(x, 0, 0)$
 ↳ door hinge (Posición $(-0.65, 0, 0.6)$)
 ↳ door (Posición $(0.65, 0, -0.6)$)
 ↳ glass (Posición $(-0.6, 2.05, 0)$)
 ↳ face ()
 ↳ clock Hand Minute (Posición $(-0.92, y, 0)$) Rotación $(x, 90^\circ, 0)$
 ↳ clock Hand Hour (Posición $(-0.94, y, 0)$) Rotación $(x, 90^\circ, 0)$

Candelabro



```

candle ( )
├── fire 0 ( Posición (19.45, 5.7, -0.58) Rotación (0, 4, 0) )
├── fire 1 ( Posición (19.45, 5.7, 0.58) Rotación (0, 4, 0) )
└── light ( )
  
```

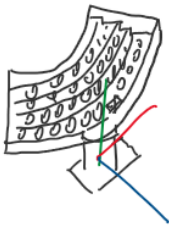
Jugador (porta la cámara)



```

jugador (x, 0, y)
├── cameraObj ( Posición (0, 2, 0) Rotación (0, 4, 0) )
└── camera ( Rotación (x, 0, 0) )
  
```

Lampadario



chandelier

```

├── moneyBox ( Posición (0, 8, 3) )
├── topHolder ( Posición (0, 8, 0.1) )
├── T (0.0, 8, -8)
│   ├── R(-30°)
│   │   ├── T (0.0, 0.0, 8)
│   │   └── lateral 0
│   └── T (0.0, 8, -8)
│       ├── R(30°)
│       │   ├── T (0.0, 0.0, 8)
│       │   └── lateral 1
├── base
└── *candle ( Escala (0.25, 0.25, 0.25) Posición (x*, y*, z*) )
  
```

Modelos y clases

Modelos cargados de disco (realizados con Blender)

- Church
- Fachade
- Tree
- Ground
- Column
- Painting
- SecretDoor
- Key

Door

Se trata de un modelo cargado desde disco realizado con Blender que incluye texturas. En este caso, el constructor permite determinar si se quiere añadir a la puerta un cerrojo o no (clase *Lock*).

Lock

Se trata de un modelo cargado desde disco realizado con Blender que incluye texturas. Se asigna al atributo *name* de la instancia un nombre y se asigna también el parámetro *userData* para poder realizar correctamente el *picking*.

Candle

Se trata de un modelo cargado desde disco realizado con Blender que incluye texturas. En este caso, se añaden además luces (tipo *PointLight*) para representar la llama de la vela encendida.

Por otro lado, la textura de la vela se incluye como una instancia de la clase *Fire*. Esta llama es posible activarla o desactivarla con el método *toggleFire*. Además, se incluye un método *update* que hace que la textura de la vela siempre mire a la cámara, haciendo que el jugador perciba que es una llama en 3D, también se modifica la escala de la llama en el eje Y, simulando que el fuego está activo.

Fire

Se trata de un *Mesh* creado a partir de una geometría *PlaneGeometry* junto con un material del tipo *MeshPhongMaterial* al que se le asigna una textura a partir de una imagen de una llama.

Table

Se trata de un modelo cargado desde disco realizado con Blender que incluye texturas. Además, se añade encima de la mesa un objeto (*Object3D*) que contiene un grupo de monedas y al que se le asigna un nombre (parámetro *name*). Estas monedas son de la

clase *Coin* y se les asigna el parámetro *userData* para poder seleccionarlas mediante *picking*.

Se hace de esta manera para que el movimiento del grupo de monedas dependa siempre del de la mesa.

Coin

Se trata de un *Mesh* creado a partir de una geometría cilíndrica (*CylinderGeometry*) con ambos extremos cerrados y a la que se le ha asignado un material de tipo *MeshPhongMaterial* que incluye una textura de una moneda.

A esta textura se le añaden transparencias con una textura de transparencias asignada al atributo *alphaMap*. También incluye un mapa para las normales (atributo *normalMap*) para representar relieve.

Clock

Se trata de un modelo que incluye varias partes:

- *ClockHand*: se trata de un modelo de *Three* creado a partir de un *Shape* mediante extrusión. Se le asigna un material basado en un color de tipo *MeshPhongMaterial*. Se asigna el atributo *userData* para poder ser seleccionada en cualquier momento.
- *ClockPendulus*: se trata de un modelo de *Three* creado a partir de un cilindro (*CylinderGeometry*) para el cuerpo y un cubo (*BoxGeometry*) para la sujeción. Adicionalmente, se añaden cuatro animaciones de *Tween* para simular el movimiento del péndulo de la forma más realista posible, para ello, por ejemplo, la animación dura exactamente 1 segundo.
- Puerta: se trata de un modelo cargado desde disco realizado con Blender que incluye texturas. Pero, en este caso, también se añade un cristal como un *BoxGeometry* con un material *MeshPhysicalMaterial* que permite un mayor control de las transparencias.
- Cara: se trata de un modelo cargado desde disco realizado con Blender que incluye texturas.

En este modelo, se ajusta la hora del reloj aplicando un ángulo determinado a las manecillas. Además, en el método *update* de esta clase se actualiza la hora automáticamente y en tiempo real para que el reloj avance con el tiempo; o bien, si se seleccionan las manecillas, permite cambiar la hora a la deseada.

Esta clase contiene algunos métodos internos que permiten diferenciar el proceso para asignar la hora de forma manual o siguiendo el transcurso del tiempo, junto con un método que verifica que la hora sea válida.

ChurchBench

Se trata de un modelo realizado con *Three* que se compone de diferentes partes:

- *Seat*: se trata de un modelo creado a partir de un *Shape* mediante extrusión. Es el asiento.

- *Recliner*: se trata de un modelo creado a partir de un *Shape* mediante extrusión. Es el reclinatorio para arrodillarse.
- *Lateral*: se trata de un modelo creado a partir de un *Shape* mediante extrusión. Es el lateral del banco.
- *Connector*: se trata de un modelo creado a partir de dos geometrías de tipo *BoxGeometry* y unidas mediante CSG. Es el conector entre el reclinatorio y los laterales del banco.

En el constructor de la clase se relacionan todos los elementos anteriores mediante un *Object3D*. No se ha hecho con CSG dado que empeora considerablemente el rendimiento por el alto consumo de recursos de esta técnica.

Chandelier

Se trata de un modelo realizado con *Three* que se compone de diferentes partes:

- *TopHolder*: se trata de un modelo creado a partir de un *Shape* mediante revolución. Es el soporte para las velas.
- *MoneyBox*: se trata de un modelo creado a partir de varias geometrías de tipo *BoxGeometry* a las que se le aplican transformaciones mediante CSG. Es el recipiente para insertar monedas. Se asigna el atributo *name* (en el constructor de la clase *Chandelier*) y *userData* para que pueda ser seleccionado.
- *Lateral*: se trata de un modelo creado a partir de un *Shape* mediante extrusión. Es el lateral para el soporte de velas.
- *Base*: se trata de un modelo creado a partir de un *Shape* mediante revolución. Es el soporte de toda la estructura.

De nuevo, todos los elementos anteriores se relacionan mediante un *Object3D* para ahorrar recursos en el caso de haber empleado CSG.

Por otra parte, se incluyen varias instancias de la clase *Candle* (no confundir con la clase *Candle* que crea los candelabros, esta es la que está contenida en el archivo *ElectricCandle.js*) que representa las velas del lampadario y que se posicionan proceduralmente dependiendo del número de estas y el ángulo del soporte. También se incluyen varias instancias de la clase *Message* para representar las notas con las pistas.

Además, se incluye un método que crea las luces del lampadario, se añaden sólo 3 en vez de una por vela para aumentar el rendimiento a la vez que se obtiene un buen resultado visual. Estas luces son actualizadas en el método *update* haciendo que, si están encendidas, la intensidad varíe aleatoriamente dentro de un rango en función del número de velas encendidas en la zona.

También se proporcionan dos métodos para encender velas: uno para encender un número aleatorio y en cualquier posición; y otro para encender las velas de la solución.

Candle

Se trata de un modelo realizado con *Three* que se compone de tres partes:

- *CandleBase*: contiene dos de las partes y ambos consisten en un modelo realizado a partir de un *Shape* mediante revolución. Son el contenedor de aluminio, al que se le

aplica un material de tipo *MeshPhongMaterial* con componente especular; y la cera de la vela, a la que se le aplica un material de tipo *MeshLambertMaterial*.

- *Flame*: se trata de un modelo realizado a partir de un *Shape* mediante revolución. Es la llama de la vela. Se le aplica un material (en ambos casos basado en un color y con *MeshLambertMaterial*) diferente en función de si está encendida o apagada, cuando está apagada se asigna un color rojo y cuando está encendida, un color más blanquecino y con emisividad para simular que está la llama encendida.

En este caso, también se insertan todos los elementos en un *Object3D*.

Message

Se trata de un objeto creado a partir de una geometría plana (*PlaneGeometry*) y al que se asigna un material con un elemento *Canvas* que permite escribir texto con el mensaje deseado. Además, se añade una textura para representar el fondo del papel.

DeadBody

Se trata de un modelo creado a partir de una geometría plana (*PlaneGeometry*), de forma que al *Mesh* se asigna un material que incluye una textura que representa el material de la silueta, junto con un mapa de normales y un mapa de transparencias. De esta forma, conseguimos que pueda verse el suelo en el interior de la silueta.

Paper

Se trata de un modelo creado a partir de una geometría plana (*PlaneGeometry*), de forma que al *Mesh* se asigna un material que incluye una textura con el dibujo deseado.

Selector

Se trata de un modelo creado a partir de una geometría plana (*PlaneGeometry*), de forma que al *Mesh* se asigna un material que incluye una textura que representa el material de la forma del selector (en nuestro caso es un + y un -), junto con un mapa de transparencias.

Otros detalles

El juego incluye efectos de sonido y ambiente para indicar sucesos.

- Puerta cerrándose

- Puerta secreta abriéndose

- Sonido ambiente de bosque

Se utiliza un efecto de niebla para crear ambiente.

Algoritmos

setHours y setMinutes

setHours(angle) se encarga de cambiar la hora y los minutos del reloj de forma proporcional, pasándole solo al ángulo de las horas.

Para ello se calcula primero el delta de la hora original y de la hora que se intenta poner nueva, después aseguramos de que el valor de dicho delta se encuentra entre -180° y 180° .

Después se cambia tanto la aguja de las horas (delta) y de los minutos (delta*12).

setMinutes(angle) funciona de forma similar.

testTime

testTime(hour,delta) se encarga de comprobar que la hora del reloj es similar a la hora *hour*. Para ello, se transforma *hour* (que es una hora representada en decimal) a un ángulo. Después se transforma la hora del reloj, que por defecto se guarda de $-\infty$ a $+\infty$, a 0° 360° . Por último se comprueba que está en el umbral. Para asegurar que la hora no se prueba por error, la aguja tiene que estar en la posición adecuada durante 2 segundos.

Controles

Usando WASD, el jugador se mueve en 4 direcciones.

Manteniendo pulsado click izquierdo y arrastrando, se puede rotar la cámara del jugador.

Haciendo doble click izquierdo sobre ciertos objetos, se puede interactuar con ellos.

Resolución

El jugador comienza en el patio de una iglesia donde las puertas se encuentran abiertas. Debe acceder a ella, por donde puede moverse con libertad, al hacerlo, se cierran las puertas impidiendo que salga hasta que posea la llave.

Para completar el primer puzle, debe desplazarse al fondo de la iglesia, a los pies del altar, donde se puede observar la silueta de un cadáver en el suelo y, junto a este una nota que contiene un mapa de la iglesia con las columnas marcadas, representando los candelabros. Además, se pueden ver dos números, esos números en binario son la contraseña para abrir una puerta secreta.

Los candelabros se pueden encender y apagar haciendo doble click en ellos. Si los candelabros de la izquierda se colocan con la siguiente configuración: 1-0-1-0-1 (21) y los de la derecha: 0-1-0-1-1 (11), en ambos casos mirando desde el centro de la sala. Se abrirá una puerta secreta a la derecha, junto al reloj. El jugador debe entrar y recoger las monedas que se encuentran en la sala secreta haciendo doble click en ellas.

Después, en la parte de la izquierda de la iglesia hay un cuadro, en el cuadro aparece el número 6. Al lado del cuadro hay un lampadario, si se han recogido las monedas se puede interactuar con la caja de monedas que tiene haciendo doble click. Cuando se interactúe con la caja aparecerá una interfaz. Cada vez que se pulsa (con doble click) el + o el -, se

cambiará la configuración de las velas. Si se pulsa exactamente 6 veces, se encenderán 2 velas, las que marcan las etiquetas de 10 horas y de 30 minutos.

A la izquierda de la sala hay un reloj, haciendo doble click en alguna de las 2 agujas y rotando, podrá cambiarse la hora y los minutos. Seleccionando las 10:30, hará que se abra la puerta del reloj (para parar de cambiar la hora vale con hacer doble click en cualquier parte del cuerpo del reloj). Dentro del reloj habrá un llave que se puede recoger haciendo doble click sobre ella.

Por último, una vez obtenida la llave, se puede hacer doble click sobre la cerradura de la puerta de la iglesia, entonces esta se abrirá y se podrá salir de la iglesia terminando así el juego.

Recursos

Texturas y materiales: <https://3dtextures.me/>

Sonido ambiente:

https://www.youtube.com/watch?v=n_sEDCaS9zU&ab_channel=Lost%26Sound-Topic

Imagen del cuadro:

<https://pixabay.com/illustrations/ship-shipwreck-sea-waves-tall-ship-1366926/>

El resto de sonidos y los modelos han sido realizados a mano.