

## PLANETARIO ENTREGA DE REALIDAD AUMENTADA.

## **Autores:**

- Samuel Moreno Vincent
- Santiago Rangel Colón



# **PLANETARIO**

## Entrega 3 Imagen Digital

## **Autores:**

- Samuel Moreno Vincent
- Santiago Rangel Colón

## TABLA DE CONTENIDO

Descripción de la Tarea	2
Menús y Teclas de Funcionamiento:	2
Clases:	2
Foco:	2
Modelo3D:	3
Cámara:	3
Mundo:	
Main:	
Parser:	
Parser de XML:	
Metodo de utilizacion:	
Capturas de Funcionamiento:	
Captura del sistema Con Una Marca:	
Captura del Sistema con dos Marcas:	5
Captura del Sistema con 1 marca y activando el modo para desaparecer elipses	6
Ohservaciones:	6



## DESCRIPCIÓN DE LA TAREA

En esta entrega debemos utilizar un sistema de reconocimiento de marcas para que cuando las localice nuestro sistema genere un planetario, exactamente el planetario creado por nosotros en entregas anteriores de la asignatura, por lo cual debemos modificar las clases de nuestro planetario para que este no entre en conflicto con con el reconocimiento de marcas y que este pueda generar un planetario sin problemas, entre las modificaciones necesarias está la necesidad de quitar los swapbuffer de nuestro código, los clear y ahora hay inicializaciones que realizaremos en el main.

Para esta entrega debemos generar encima de la marca encontrada nuestro planetario quedando encima de esta marca, además si encuentra una segunda marca el planetario generado no deberá tener lunas ni el primer planeta del sistema, esto mientras que el esta localizando las dos marcas al mismo tiempo.

## MENÚS Y TECLAS DE FUNCIONAMIENTO:

Los focos serán controlados mediante las teclas 1 a la 8, siendo cada uno de los números los distintos focos que podemos encender o apagar mediante su pulsación, el sistema comienza con el primer foco (luz blanca) encendido cuando ejecutamos el sistema, la tecla 0 es una tecla especial que hemos introducido que apaga todo los focos encendidos.

Como opción especial hemos introducido la opción de **quitar y poner las órbitas** de los planetas pulsando simplemente la **tecla 9**, de inicio las órbitas estarán encendidas.

Tecla	Acción
Teclas 1 a la 8.	Encendido/Apagado de los Focos.
Tecla 0.	Apagado de todos los focos.
Tecla 9.	Poner/Quitar las órbitas de los astros.

#### **CLASES:**

## FOCO:

Esta clase será la encargada de la creación de cada uno de los 8 focos, estableciendo sus valores de posición, color e intensidad lo cual se hará en el constructor de la misma cogiendo los valores del archivo de configuración xml. Además se podrán cambiar el estado de encendido/apagado hacienda una llamada al método "switchFoco". Los cambios entre los focos activos y no activos se harán mediante la pulsación de las teclas 1 a la 8, siendo cada uno de los números uno de los focos y al pulsar en cada número activaremos o desactivaremos el foco asociado a cada tecla, estas pulsaciones serán escuchadas desde la clase "Main", además podremos apagar todos los focos mediante la pulsación.



#### MODELO3D:

Esta clase se encargará de mediante los puntos del modelo que le hemos pasado, en el caso normal, esferas crear nuestro modelo, en esta entrega empezará directamente en modo smooth y no podrá ser modificado mediante el menú que utilizábamos en entregas anteriores.

#### CÁMARA:

En esta entrega no utilizamos las cámaras que usábamos en entregas pasadas, sino que usamos la que venía creada en la clase main proporcionada por el profesor.

#### MUNDO:

En esta entrega el mundo continúa realizando las mismas funciones que en entregas pasadas solo que ya no inicializa tantas cosas, además de que pierde los llamamientos a la creación e inicialización de cámaras, la configuración del sistema sigue siendo por parte del fichero xml solo que en esta ocasión no entra por parámetro, su nombre es config.xml y debe estar en la misma carpeta que el ejecutable.

#### MAIN:

Nuestro Main controla el menú que controla el encendido de los focos de nuestro sistema, además de poder quitar las órbitas a los planetas y satélites con la tecla que hemos utilizado para esta función, en esta entrega el main tiene más importancia que en entregas pasadas puesto que ahora debe reconocer marcas y dependiendo de si encuentra y o dos hará distintas cosas, si encuentra solo una de las marcas generará el planetario encima de nuestra marca, pero en el caso de que existan dos marcas, el planetario generado será uno que no tiene lunas ni el primer planeta del sistema.

#### **PARSER:**

Esta clase se encargará de la gestión de los datos obtenidos del documento XML. En el constructor le pasamos el nodo inicial del árbol de datos.

Los métodos "getNumMateriales", "getNumFocos", "getNumCamaras" y "getNumModelos" obtienen el número de materiales, focos, cámaras y modelos respectivamente, que hay en el árbol de datos haciendo una llamada al método recursivo "getNum\_recursivo" al cual se pasa por parámetro el inicio del contador (0), el nodo inicial del árbol de datos, y el nombre del objeto a buscar. Este método hace un recorrido total del árbol y va sumando al contador cada vez que encuentra un objeto con el nombre dado.

Los métodos "getMaterial", "getFoco", "getCamara", "getFrustrum" y "getModelo" obtienen el vector de datos de su respectiva clase con el identificador dado haciendo una llamada al método recursivo "getVector\_recursivo" al cual se le pasa el nodo inicial del árbol de datos, el vector que va a ser llenado con los datos, el identificador del objeto, el tipo de vector dentro del objeto, el nombre del objeto y el número de datos que va a obtener. Este devuelve los datos en el vector haciendo un recorrido por todo el árbol de datos hasta encontrar el objeto del tipo especificado y con el identificador dado.



## **PARSER DE XML:**

Las clases xmlnode, xmlnodelist, xmlparser y útil han sido obtenidas de la página: <a href="http://jok3n.com/xmlparser/descargas/">http://jok3n.com/xmlparser/descargas/</a>

Su función es transformar todos los datos dentro del xml en un árbol de nodos. Para ello se realiza una inicialización del parser en la clase Mundo (InitParser) de la cual se obtiene dicho árbol que se guardará en la clase Parser construida por nosotros y explicada anteriormente.

#### **METODO DE UTILIZACION:**

Para que el xml sea leído correctamente todos los datos deben tener este formato:

```
<Nombre de la clase + identificador>
<Nombre del vector de datos>
<Lista de datos>
```

## Ejemplo:

```
<Modelo id ="0">

<Datos>

<radioGiro>0</radioGiro>

<radioAstro>3</radioAstro>

<velocidad>0.1</velocidad>

<esLuna>0</esLuna>

<velocidadRotacion>0.1</velocidadRotacion>

</Datos>

</Modelo>
```

## **CAPTURAS DE FUNCIONAMIENTO:**

#### CAPTURA DEL SISTEMA CON UNA MARCA:

Captura de como se ve el sistema al ejecutar la aplicación, tiene activado el foco 1 (luz blanca) y el modo Smooth.





## CAPTURA DEL SISTEMA CON DOS MARCAS:

Como se puede observar desaparece el primer planeta y las lunas de todos los planetas.





#### CAPTURA DEL SISTEMA CON 1 MARCA Y ACTIVANDO EL MODO PARA DESAPARECER ELIPSES



## **OBSERVACIONES:**

Cuando abrimos el ejecutable en ocasiones se queda negro hasta k pasan 5 o 10 segundos y la imagen comienza a salir cuando ponemos una marca delante de la cámara, no sabemos por qué nos pasa pero también nos ocurre con la versión original del profesor, por lo que pensamos que debe ser cosa de nuestros equipos, aun así hemos querido dejar por escrito esta incidencia.