### Informática Gráfica II

# Object-oriented Graphics Rendering Engine Práctica 1.[4-7] Suelo, texturas, HUD y luces

Alberto Núñez Departamento de Sistemas Informáticos y Computación Universidad Complutense de Madrid

# El suelo

El suelo del laberinto es donde reposan Los bloques Los personajes Está situado debajo de los bloques Las medidas se pueden calcular utilizando las dimensiones del laberinto Se debe calcular en tiempo de ejecución Es posible utilizar otros laberintos con distintas dimensiones La llamada para crear la malla del plano Ogre::MeshManager::getSingleton().createPlane Hay que tener en cuenta los parámetros relativos a las texturas iOjo con el nombre! Si se crean más laberintos, el nombre no debe repetirse



Vamos a mostrar por pantalla información del juego en la esquina inferior derecha Como mínimo, debe aparecer: Las vidas restantes Los puntos (10 puntos por perla comida) Utilizaremos dos elementos OgreBites::Label (con un título) y OgreBites::TextBox (vidas y puntos) Para crearlos utilizamos los métodos mTrayMgr->createLabel(OgreBites::TL BOTTOMRIGHT, nombre, texto, ancho); mTrayMgr->createTextBox(OgreBites::TL BOTTOMRIGHT, nombre, texto, ancho, alto); mTrayMgr es de tipo OgreBites::TrayManager y se crea en IG2App El orden en el que se crean los elementos es relevante Primero la etiqueta Luego el cuadro de texto

Práctica 1

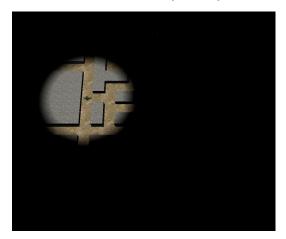
## **Texturas**

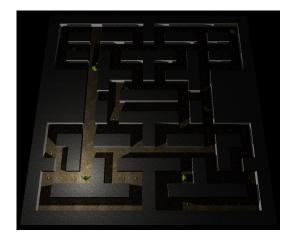
- ☐ En este apartado pondremos color al juego
- Deben asignarse materiales a:
  - El suelo
  - Los bloques
  - Las perlas
  - El villano creado en el apartado 3
    - Cada parte con su textura individual
- Opcionalmente podemos asignar materiales a:
  - Los villanos ogrehead
    - Un color distinto a cada villano
- ☐ Los materiales de las perlas, el muro y el suelo se indican en el fichero del laberinto
  - No es necesario compilar para probar distintos materiales
- Todas los ficheros utilizados deben estar en media/IG2App
  - Incluido el fichero .material

```
NumFilas
NumColumnas
materialPerla
materialMuro
materialSuelo
caracteresFila_1
caracteresFila_2
...
caracteresFila_NumFilas
```

### La luz

- Inicialmente utilizamos una luz direccional
- En este apartado utilizaremos una luz de tipo spotlight
  - Opcionalmente, podéis utilizar una de tipo Point
- La idea es que la luz siga al personaje
  - Limitando la visibilidad en el resto del laberinto
- Podéis utilizar sombras para dar un aspecto más realista
- Se configura en el fichero del laberinto
  - tipoLuz = [directional | spot | point]





NumFilas

NumColumnas

materialPerla

materialMuro

materialSuelo

### tipoLuz

caracteresFila\_1

caracteresFila\_2

• • •

caracteresFila NumFilas