



# Máster en Diseño y Creación de Videojuegos

## P3. Programación 2D

Jesús Alonso Alonso

21 Noviembre 2013

Realizar uno de los tres ejercicios siguientes:

1. Continuar proyecto visto en clase y elegir entre opción A y B
2. Aplicar las funcionalidades vistas en clase a un juego en vista isométrica
3. Construcción de prototipo 2D de nuestro proyecto de máster

Podrán utilizarse los siguientes *sprites* así como de otros que se consideren oportunos.

- <http://glstation.free.fr/mine/djgpp.htm>
- <http://glstation.free.fr/mine/djgpp/ecere/w2ecer02.zip>
- <http://glstation.free.fr/mine/djgpp/ecere/w2data01.zip>

Sea cual sea la opción que escojamos:

- El juego tendrá **inicio y final** (nos lo podremos pasar). Como mínimo, tendremos una pantalla de inicio que nos permitirá jugar o salir y cuando nos lo pasemos o fracasemos, volveremos a ésta y podremos volver a realizar otra partida.
- Implementaremos una clase **cSound** que nos permita utilizar música de fondo así como efectos especiales (ver: FMOD, OpenAL u otras librerías).

### Formato de entrega

La entrega se realizará en clase. Fichero comprimido X.zip (siendo X el número del grupo) que contenga las carpetas siguientes:

- Carpeta "src": código fuente
- Carpeta "bin": ejecutable y recursos necesarios
- Fichero autores.txt: nombre y apellidos de cada uno

## EJERCICIO 1

### Parte común: Tiles + Radar + Niebla

Modifica la implementación dada para que:

1. Se vea el mapa utilizando las *tiles* del *Warcraft II* (fichero war2.png).
2. El radar muestre la totalidad del mapa a pequeña escala.
3. Incorporación de al menos dos tipos de edificios de diferente tamaño sobre la escena, de tal manera que conformen *tiles* no "walkables".
4. Visualización de niebla en juego y radar teniendo en cuenta los siguientes niveles:
  - a. Celda visible
  - b. Celda visitada no visible
  - c. Celda no visible
5. Dicha niebla debe soportar contornos redondeados.



6. Visualización del enemigo y los edificios en escena y radar según ubicación en celdas visibles o no.

Observaciones:

- Puede servir de base la configuración del mapa ya existente.
- Debe haber continuidad entre celdas y por tanto que aparezcan *tiles* "redondeadas" donde sea necesario.
- El agua debe estar animada.
- Los edificios en el radar serán imágenes de un solo color, el del enemigo.
- El personaje no podrá "pisar" edificios.
- Un edificio es visible (con niebla) en una celda previamente visitada no visible, el enemigo no.

### Opción A: RPG

Introduce la lógica necesaria para que el juego tenga un desenlace simple tipo RPG: ir a una zona determinada, hablar o luchar, sortear algún obstáculo, conseguir algún ítem y llegar a otra zona.

### Opción B: RTS

Introduce la lógica necesaria para implementar un juego tipo RTS. Deben aparecer diferentes unidades enemigas (ya creadas por defecto) con comportamientos diferentes (ataque a distancia, cuerpo a cuerpo, agresivos, etc). Aplicación del algoritmo A\* para el movimiento de cada uno de ellas de forma correcta y eficiente, esto es, compartiendo la misma estructura de datos para el escenario y comprobando la detección de colisiones mientras los personajes se desplazan.

## EJERCICIO 2

Implementar:

1. Todas las funcionalidades brindadas en el ejercicio de clase para el caso de un juego en **vista isométrica**, con *tiles* que sigan la proporción 2:1.
2. Al menos un personaje animado
3. Escenario con *tiles* apilables.
4. Un mínimo de cuatro *tiles*, una de ellas animada.

Observaciones:

- Los *sprites* deberán ser los adecuados para una vista de este tipo, tal y como se muestra en el siguiente fichero de muestra, pudiéndose utilizar éste mismo: agent.png

## EJERCICIO 3

Construye un **prototipo 2D** de vuestro proyecto final simulando la **lógica** o parte de ella. Lo que el jugador en una zona concreta deberá realizar contando con la ayuda de unos objetos y/o habilidades determinadas.