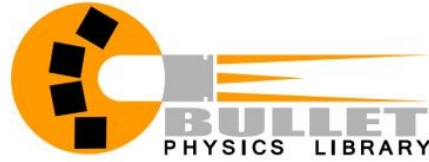


## TEMARIO

Diseño y programación de videojuegos – 150 horas



## Tema 1: TEORÍA E INTRODUCCIÓN A BENNUGD COMO LENGUAJE 100% ORIENTADO A LA PROGRAMACIÓN DE VIDEOJUEGOS

**Duración: 9 horas**

- ✓ Características que lo hacen distinto a otros lenguajes.
  - ◆ Lenguaje semi-interpretado.
  - ◆ Lenguaje modular.
  - ◆ Programación orientada al proceso.
  - ◆ Herencia.
  - ◆ Multiplataforma (PC, Mac, Wii, Wiz, etc.)
  - ◆ Ejecución desde navegador.
  
- ✓ Utilidades que lo hacen 100% orientado al diseño y programación de videojuegos.
  - ◆ Sincronización con FRAME.
  - ◆ El buffer de vídeo.
  - ◆ Variables locales predefinidas.
  - ◆ collision, father, son, etc.
  
- ✓ Herramientas asociadas de Bennupack.
  - ◆ Editor FPG y FNT.
  - ◆ Terrain editor y md2 model viewer.
  - ◆ Explogen y otras utilidades de interés general.
  
- ✓ Sencillo programa de ejemplo e introducción a las plantillas del DVD.

## Tema 2: PRÁCTICAS DE IMPLEMENTACIÓN DE UN SENCILLO VIDEOJUEGO EN 2D

**Duración: 15 horas**

- ✓ Creación y destrucción de procesos.
  - ◆ Declaración e invocación de procesos.
  - ◆ Break y signal.
  
- ✓ El buffer de vídeo.
  - ◆ Resoluciones y modos de color.
  - ◆ Buffer 2D y buffer 3D.
  
- ✓ Ámbito de las variables: GLOBAL, LOCAL, PRIVATE y PUBLIC
  
- ✓ Gestión de efectos de sonido y melodías de fondo.
  
- ✓ Fuentes de texto.
  - ◆ Fuentes FNT en 16 bits.
  - ◆ Fuentes TTF en 32 bits.
  - ◆ Control de interfaz y textos en pantalla.

- ✓ Animación avanzada basada en máquinas de estados.
  - ◆ Estados de animación.
  - ◆ Condiciones de cambio de estado.
  - ◆ Elaboración de diagramas de cambio de estado.
- ✓ Mapas de dureza.
  - ◆ Interacción con el escenario.
  - ◆ Gravedad.
  - ◆ Inteligencia artificial y búsqueda de caminos.

### Tema 3: GRANDES PROYECTOS EN LA INDUSTRIA DEL SOFTWARE

**Duración: 30 horas**

- ✓ Código de terceros y librerías dinámicas (DLL).
  - ◆ Reutilización de código de terceros.
  - ◆ Legibilidad, mantenibilidad y simplicidad
  - ◆ Introducción a las dll y al manual de referencia de BenuGD.
- ✓ Módulos de código y gestión de dependencias.
  - ◆ Uso adecuado de CONST.
  - ◆ Uso adecuado de GLOBAL.
  - ◆ Cabeceras
  - ◆ Punteros.
- ✓ Diagramas de análisis y diseño y documentación general de un proyecto.
  - ◆ La etapa de análisis.
  - ◆ La etapa de diseño.
  - ◆ La etapa de implementación.
- ✓ Prácticas de programación de un proyecto real de videojuego en equipos.
  - ◆ Medición de la eficiencia del trabajo en equipo: El juego de la NASA
  - ◆ Reparto de roles en un proyecto de software.
  - ◆ Gestión automática de proyectos con DropBox.
  - ◆ Prácticas.
- ✓ Extensiones de interés general
  - ◆ Reconocimiento de voz y reconocimiento facial.
  - ◆ Port para usar WiiMote como periférico,
  - ◆ Netlib para aplicaciones cliente/servidor.
- ✓ El entorno laboral
  - ◆ Empresas del sector en España y en Europa.
  - ◆ Portales de búsqueda de empleo.
  - ◆ Puestos de trabajo.
  - ◆ El currículum.

## Tema 4: INTRODUCCIÓN A BULLET PHYSICS

**Duración: 15 horas**

- ✓ Modelos 3D: Personajes, escenarios y sus operaciones básicas.
  - ◆ Carga de modelos estáticos y animados.
  - ◆ Modelos de alta y baja densidad poligonal.
  - ◆ Aplicación de texturas.
  - ◆ Movimiento, rotación y escalado.
  
- ✓ El buffer de vídeo.
  - ◆ Implementación de un motor 3D real.
  - ◆ Métodos de render (Software, DirectX, OpenGL, etc.)
  - ◆ Conceptos sobre eficiencia.
  
- ✓ Introducción a las características de Bullet y aplicación real en videojuegos de última generación.
  - ◆ Formatos de modelo soportados (.obj, .3ds, .md3, etc.).
  - ◆ Formatos de textura soportados (.png, .jpg, .bmp, etc.).
  - ◆ Uso de Bullet 3D en videojuegos y películas actuales.
  
- ✓ Diferencias principales con Benu en 2D.
  - ◆ Control de varios elementos gráficos desde un mismo proceso.
  - ◆ Reutilización de variables locales predefinidas.
  - ◆ Definición de nuevas variables locales.
  
- ✓ Introducción básica al programa de modelado 3D Wings3D.

## Tema 5: IMPLEMENTACIÓN DE UN SENCILLO VIDEOJUEGO EN 3D

**Duración: 45 horas**

- ✓ Características de los materiales e iluminación en tiempo real.
  - ◆ Modelos de iluminación (Total, Gouraud y Phong).
  - ◆ Reacción a la niebla y mapas de niebla.
  - ◆ Simulación de día y noche en tiempo real y meteorología.
  - ◆ Control de la opacidad y del canal alpha.
  
- ✓ Gestión de animaciones en videojuegos comerciales.
  - ◆ Prácticas con el formato de modelo .md3 de Quake III.
  - ◆ Gestión de modelos hijo y equipamiento de items sobre el modelo.
  - ◆ Introducción a bones: El formato .md de World of Warcraft.
  
- ✓ Emisores de partículas, matrices de agua, splines y fades.

- ✓ Interfaz gráfica de usuario.
  - ◆ Textos y marcadores.
  - ◆ Gráficos en 2D.
  - ◆ Uso del ratón (mod\_mouse).
  - ◆ Gestión de colisiones.
  
- ✓ Detección eficiente de colisiones en 3D.
  - ◆ Identificadores de modelo e identificadores de proceso.
  - ◆ Rayos de colisión (Juegos de acción).
  - ◆ Grids y matrices (Juegos de estrategia).
  
- ✓ Física, listas de colisiones y rendimiento en Bullet.
  - ◆ Bounding Boxes.
  - ◆ Listas de colisiones.
  - ◆ Modelo físico.
  - ◆ Ragdoll Physics y física de carros.
  
- ✓ Introducción a la extensión 3Dit para Bullet Physics.
  
- ✓ Introducción a Ogre3D como motor de render de última generación.

### Tema 6: PRÁCTICAS Y CIERRE DE PROYECTOS

**Duración: 39 horas**

- ✓ Cierre del proyecto individual en 3D.
- ✓ Cierre de un proyecto colectivo en 3D.
- ✓ Presentación, evaluación y entrega de proyectos.