



Introducción al desarrollo de videojuegos con Unity3D

Tema 4: Interacción con objetos



Vídeo T4_V09

Normas de estilo en programación

Sobre los comentarios

- Comentar bien para explicar lo que se hace

```
// -----  
// Función ObtenerPuntuacion  
// Esta función devuelve la puntuación de un jugador partiendo de  
//   los puntos que se otorgan por ganar una carrera y  
//   de las carreras ganadas  
// -----  
int ObtenerPuntuación  
(  
    int numero_carreras_ganadas,  
    int puntos_por_ganar  
)  
{  
    int puntos_obtenidos = numero_carreras_ganadas * puntos_por_ganar;  
    return puntos_obtenidos;  
}
```

Sobre los comentarios

- **NUNCA** comentar líneas con código fuente

```
// -----  
// Función ObtenerPuntuacion  
// Esta función devuelve la puntuación de un jugador partiendo de  
//   los puntos que se otorgan por ganar una carrera y  
//   de las carreras ganadas  
// -----  
int ObtenerPuntuación  
(  
    int numero_carreras_ganadas,  
    int puntos_por_ganar  
)  
{  
    //int puntos_obtenidos = numero_carreras_ganadas * puntos_por_ganar;  
    int puntos_obtenidos = numero_carreras_ganadas/2 * puntos_por_ganar;  
    return puntos_obtenidos;  
}
```

Sobre los nombres de las variables

1. El nombre debe tener relación con el concepto.
2. No importa si el nombre es un poco largo

```
// -----  
// Función ObtenerPuntuacion  
// Esta función devuelve la puntuación de un jugador partiendo de  
//   los puntos que se otorgan por ganar una carrera y  
//   de las carreras ganadas  
// -----  
int ObtenerPuntuación  
(  
    int numero_carreras_ganadas,  
    int puntos_por_ganar  
)  
{  
    int puntos_obtenidos = numero_carreras_ganadas/2 * puntos_por_ganar;  
    return puntos_obtenidos;  
}
```

Sobre los nombres de las variables

- Convenciones que podemos usar:
 - Si la variable es “numero_...”, usar “n_...”
 - Usar i, j, k para índices de bucles
 - Personal:
 - Para índices **i_concepto**.
 - i_jugadores, i_balas, i_vueltas, etc.

```
...  
for (int i_jugadores=1; i_jugadores <= n_jugadores; i_jugadores ++)  
{  
...  
}  
...
```

Paréntesis extra

- En ocasiones poner más paréntesis de los estrictamente necesario ayuda a entender un código.

```
...  
int posición = posición_x * velocidad_x + (1/2) * aceleración_x - 2 * factor_rozamiento;  
...
```

Paréntesis extra

- En ocasiones poner más paréntesis de los estrictamente necesario ayuda a entender un código.

```
...  
int posición = posición_x * velocidad_x + (1/2) * aceleración_x - 2 * factor_rozamiento;  
...
```

```
...  
int posición = (posición_x * velocidad_x) + ((1/2) * aceleración_x) - (2 * factor_rozamiento);  
...
```


Paréntesis extra

- En ocasiones poner más paréntesis de los estrictamente necesario ayuda a entender un código.

```
...
int posición = posición_x * velocidad_x + (1/2) * aceleración_x - 2 * factor_rozamiento;
...
```

```
...
int posición = (posición_x * velocidad_x) + ((1/2) * aceleración_x) - (2 * factor_rozamiento);
...
```

```

...
int posición = posición_x * velocidad_x      +
                (1/2)      * aceleración_x    -
                2          * factor_rozamiento;
...

```

Alineación

- Los “espacios” son gratis

```
...  
int n_jugadores = 34;  
float puntos_por_partida = 56.7;  
Int n_partidas_jugadas = 21;  
...
```

Alineación

- Los “espacios” son gratis

```
...  
int n_jugadores = 34;  
float puntos_por_partida = 56.7;  
int n_partidas_jugadas = 21;  
...
```

```
...  
int    n_jugadores           = 34;  
float  puntos_por_partida    = 56.7;  
int    n_partidas_jugadas    = 21;  
...
```

Resumen

- Piensa en tus normas
- Aplícalas siempre