

# TP1: Interfaz Gráfica y Simulación

*Taller de Programación I - Cátedra Ing. Diego Azcurra - 1er cuatrimestre 2015*

## Objetivo

Desarrollar un programa en C/C++ que muestre un escenario. El escenario consta de un personaje animado y controlable.

## Desarrollo

Se deberá desarrollar un programa C/C++ que:

1. Lea desde un archivo la definición de un escenario. Si el archivo no existe, se creará un escenario por defecto. El nombre del archivo será pasado por línea de comandos.
2. Muestre dicho escenario gráficamente en una pantalla. El escenario estará compuesto por:
  - Un fondo. Compuesto por dos o más imágenes de diferentes tamaños. Las imágenes utilizadas para mostrar el fondo se obtendrán de archivos externos de imagen.
  - Un protagonista que puede ser controlado por el usuario

## Controles

- **Flecha Izquierda/Derecha:** el protagonista camina hacia la derecha o izquierda.
- **Flecha arriba:** el protagonista salta.
- **Tecla R:** se vuelve a cargar el archivo de escenario (el cual puede haber sido modificado por un editor externo) y se reinicia la simulación.

## Archivo de configuración

El escenario se carga a partir de un archivo de configuración en formato JSON.

Todos los objetos que integran el escenario se especifican en las denominadas "coordenadas del mundo". Este sistema permite que el motor del juego sea independiente de la capa que dibuja la pantalla (y que debe considerar entre otras cosas la posición de la "camara" o scroll en cada momento). Nos referiremos a este sistema como "coordenadas de mundo" o "coordenadas lógicas". Las coordenadas lógicas se expresan con números reales.

La ventana es el único elemento que tiene un par de dimensiones en píxeles y otro par de dimensiones en coordenadas lógicas. No necesariamente existe una relación de proporcionalidad entre estas dimensiones. Estos 4 valores (ancho y alto en píxeles y en coordenadas lógicas) permiten realizar la transformación de todos los elementos del engine del juego a sus dimensiones y posiciones en la ventana.

Tanto el personaje como el escenario tienen un alto y un ancho (en coordenadas lógicas). El personaje está confinado al rectángulo del escenario, es decir, si  $x_{izquierdo}$  y  $x_{derecho}$  son las coordenadas  $x$  de los bordes izquierdo y derecho del personaje, las siguientes son invariantes durante la ejecución:

$$0 \leq x_{\text{izquierdo}} \leq \text{ancho\_escenario} - \text{ancho\_personaje}$$

$$\text{ancho\_personaje} \leq x_{\text{derecho}} \leq \text{ancho\_escenario}$$

El ancho lógico de la ventana siempre será menor o igual que el ancho lógico del escenario.

El escenario también tiene definida la coordenada del nivel del piso que es por donde el personaje va a caminar. Asimismo el personaje está afectado por la gravedad lo que lo mantiene en el piso.

## Vista multicapa

El escenario está formado por un array de “capas”. Las mismas serán imágenes que tendrán transparencias.

El orden en el array define cuales capas estan delante de otras. Consideramos que las primeras capas del array se encuentran detrás de las subsiguientes. Definimos como z-index de la capa al indice dentro del array.

Por definición (para las coordenadas lógicas):

$$\text{alto\_escenario} = \text{alto\_lógico\_ventana} = \text{alto de todas las capas}$$

Por lo tanto este valor solo se define en el objeto “escenario” del archivo de configuración y se omite en la especificación de la ventana y de las capas por ser redundante.

Cuando el personaje camina y sobrepasa un margen de la ventana se produce un desplazamiento horizontal (scroll) de la cámara.

Todas las capas se mueven a una velocidad proporcional al ancho de la misma. Por lo tanto el ancho de cada capa define su velocidad de desplazamiento.

## Protagonista

El protagonista podrá caminar y saltar. Al hacerlo se debe mostrar una animación que represente dicho movimiento.

Se define en el archivo de escenario el z-index del personaje. El personaje se dibuja por delante de todas las capas con z-index menor o igual a la del personaje, y detrás de todas las capas con z-index mayor.

## Estado inicial

El estado inicial del juego es la ventana centrada en el escenario, y el protagonista centrado en el escenario.

## Aclaraciones

- Las imagenes del fondo pueden estar en cualquier formato (por ejemplo: bmp, png, jpg) y puede ser cargada con las bibliotecas de extensión de SDL.
- Los archivos de escenarios deben respetar la sintaxis JSON. El diseño de la estructura del mismo queda a criterio de los alumnos. Se recomienda pensar

en un formato extensible que permita su reutilización en los siguientes trabajos prácticos.

- Los alumnos deben considerar todas las situaciones de error posibles, ante las cuales el programa deberá reaccionar de la manera que sea más apropiada (por ejemplo: mostrar un mensaje de error, no hacer nada, tomar una acción por defecto, etc).

## Restricciones

- Para la representación gráfica se deberá utilizar la biblioteca SDL 2.0 (<http://libsdl.org>).
- Para la lectura y escritura de archivos JSON debe utilizarse, preferiblemente, la biblioteca json-cpp (<http://jsoncpp.sourceforge.net>). La utilización de otras bibliotecas no está prohibida pero debe ser consultada. **No se permite la utilización de un parser propio.**
- Todo el código debe ser desarrollado íntegramente por cada grupo. No se permite la reutilización de código de cuatrimestres anteriores o de otras materias. Ante cualquier duda se deberá consultar con los docentes. La reutilización de código sin consulta previa será condición suficiente para la desaprobación de la materia.

**Este enunciado no es definitivo. Si se realizan cambios en clase se respetarán y evaluarán los mismos.**

## Fechas

Semana #	Fecha	Tema
1	11 de marzo	Presentación de la materia
2	18 de marzo	Presentación enunciado TP 1
3	25 de marzo	Consultas
4	1 de abril	Consultas
5	8 de abril	Consultas
<b>6</b>	<b>15 de abril</b>	<b>Entrega TP1. Presentación enunciado TP 2</b>
7	22 de abril	Primer recuperatorio TP1. Consultas TP 2
8	29 de abril	Segundo recuperatorio TP1. Consultas TP 2
9	6 de mayo	Consultas
<b>10</b>	<b>13 de mayo</b>	<b>Entrega TP2. Presentación enunciado TP 3</b>
11	20 de mayo	Primer recuperatorio TP2. Consultas TP 3
12	27 de mayo	Segundo recuperatorio TP2. Consultas TP 3
13	3 de junio	Consultas
<b>14</b>	<b>10 de junio</b>	<b>Entrega TP 3</b>
15	17 de junio	Primer recuperatorio TP 3
16	24 de junio	Segundo recuperatorio TP 3

## Anexo: Archivo de escenario de ejemplo

```
{
  "ventana": {
    "ancho-px": 800,
    "alto-px": 600,
    "ancho": 200
  },
  "escenario": {
    "ancho": 1000.5,
    "alto": 150,
    "y-piso": 20
  },
  "capas": [
    {
      "imagen_fondo": "fondo1.png",
      "ancho": 500
    },
    {
      "imagen_fondo": "fondo2.png",
      "ancho": 1000.5
    }
  ],
  "personaje": {
    "ancho": 20,
    "alto": 35,
    "z-index": 1,
    "sprites": ?
  }
}
```