

# TP2: Interfaz Gráfica

*Taller de Programación I - Curso Ing. Diego Azcurra - 1er cuatrimestre 2016*

## Objetivo

El objetivo de este trabajo práctico es extender el programa desarrollado en el TP1 agregando funcionalidad gráfica. Para ello se debe mostrar una pantalla similar a la del juego 1942 (<https://youtu.be/AIXf77Hheeo>), que le permita a múltiples usuarios mover su avión por la misma utilizando el teclado mientras el fondo se desplaza de forma constante.

## Desarrollo

Podrán participar de 1 a 4 jugadores simultáneamente en red. Todos pueden interactuar simultáneamente moviendo sus aviones por la pantalla. Asimismo, todos los participantes pueden ver en tiempo real los movimientos de los aviones de los otros jugadores.

*Nota: Incentivamos la creatividad! La temática y arte del juego está a cargo de los alumnos. Los gráficos de los personajes previamente mencionados pueden reemplazarse por lo que cada grupo desee. No obstante, las reglas del juego y funcionalidad en general no puede cambiar.*

## Red

El sistema consistirá de un servidor y varios clientes que se conectan a él. Queda a criterio de cada grupo si el servidor y los clientes son programas separados, o un mismo ejecutable que puede asumir los distintos roles. Cabe aclarar que de optar por esto último, un cliente no puede ser simultáneamente servidor en la misma ejecución.

El servidor define el escenario: al iniciar el servidor se especifica el archivo XML con la configuración del escenario y, por lo menos en este trabajo, el mismo no cambiará en ningún momento. Si el archivo no existe, se creará un escenario por defecto.

Una vez cargado el escenario se ejecuta el nivel, lo que consiste en realizar un desplazamiento vertical completo del escenario. Al terminar el desplazamiento se reinicia el nivel (no se sale del programa).

No es necesario que el servidor tenga una interfaz gráfica. Puede iniciarse por parámetros de la línea de comandos.

Cada escenario tendrá un máximo de jugadores que está especificado en el XML de configuración. Alcanzado ese límite, el escenario no admite más jugadores.

Cada jugador al conectarse debe especificar un nombre de usuario y servidor al que desea conectarse (ip y puerto). Si ese servidor ya tenía un usuario con ese nombre y el mismo no se encuentra conectado, entonces se acepta el nuevo jugador asignándole el estado guardado para dicho nombre. Si el servidor no contenía ese nombre de usuario y aún queda espacio disponible para un nuevo jugador, se crean los personajes y el jugador entra al escenario.

Cuando un jugador se desconecta del escenario su avión se mostrará grisado y se indicará en un mensaje a los demás que el jugador se ha desconectado.

Es indistinto si el jugador se desconecta por una falla de red o porque voluntariamente salió del programa.

En todo momento cada jugador puede ver los movimientos de los aviones de otros jugadores en tiempo real.

## Interfaz gráfica

El archivo de configuración tiene especificado el ancho y alto de la ventana del juego en píxeles (tipo entero). Este es el tamaño de la ventana que se le abre al usuario cuando inicia el juego.

Todos los elementos del juego tienen sus coordenadas espaciales especificadas en píxeles.

Existe el concepto de cámara, que es un rectángulo contenido en el escenario y representa la parte visible del escenario en cada momento. Al variar la posición de la cámara se obtiene el efecto de desplazamiento (o scroll).

**Restricción 1:** como en este juego no habrá scroll horizontal el ancho de la cámara siempre va a coincidir con el ancho del escenario.

**Restricción 2:** Para simplificar la implementación, la relación ancho/alto de la ventana debe ser igual a la relación ancho/alto de la cámara.

## Biblioteca de sprites

El archivo de configuración tiene una sección de sprites. Cada sprite está identificado por un id tipo string.

Los sprites son animados y consisten de una secuencia de fotogramas (un sprite no animado es el caso particular de un sprite con un solo fotograma).

Cada sprite se almacena como un único archivo de tipo imagen que contiene todos los fotogramas uno al lado del otro en una tira.

Todos los fotogramas de un sprite tienen el mismo ancho (al que llamaremos ancho del sprite). El ancho del sprite se especifica en píxeles.

**Restricción:** El ancho de la imagen tiene que ser igual al ancho del fotograma multiplicado por la cantidad de fotogramas en la animación. Por ejemplo, si la animación tiene 10 fotogramas de 130 píxeles, el ancho de la imagen en píxeles será 1.300 .

Por lo tanto para cada sprite de la biblioteca se especifica:

- id
- path a la imagen
- cantidad de fotogramas
- ancho (en unidades lógicas)
- alto (en unidades lógicas)

El ancho y alto indica cómo se debe escalar la imagen. Notar que no es el ancho y alto en píxeles de la imagen (éstos se pueden obtener del archivo de imagen)

## Escenario

El escenario representa la vista superior del terreno por sobre el cual vuelan los aviones. El mismo consta de un fondo (por ejemplo el océano o tierra) y diversos elementos (como ser islas, barcos, portaaviones, etc).

El escenario define un sistema de coordenadas (en píxeles) que se utilizará para especificar las posiciones de los demás elementos. Definimos este sistema de coordenadas como las **coordenadas del mundo**.

El origen de coordenadas del mundo está en el ángulo inferior izquierdo del escenario. Las abscisas crecen de izquierda a derecha y las ordenadas crecen de abajo hacia arriba.

El archivo de configuración debe especificar:

- Su ancho y alto en píxeles.
- **El fondo**, que es un sprite que se repite a modo de mosaico y siempre se encuentra detrás de los demás elementos. Ejemplo: agua.
- **Una lista de elementos de juego**. Cada elemento tiene forma rectangular y los siguientes atributos:
  - **El id de sprite que lo representa**.
  - **Su posición en el escenario**. Es la posición del borde inferior izquierdo del elemento en coordenadas del mundo.

Los elementos pueden quedar en parte fuera del escenario.

Puede haber superposición de elementos. El orden en el que se especifican los elementos en el archivo de configuración define el orden de dibujo, es decir, los elementos que aparecen primero en la configuración están detrás de los que aparecen después.

## Scroll

La cámara se desplaza automáticamente de abajo hacia arriba. El archivo de configuración define la velocidad de la cámara en [píxeles/frames]

## Avión

Cada jugador puede controlar un avión que puede desplazarse libremente por toda la pantalla y disparar.

### Controles

**Flecha Izquierda/Derecha:** el avión se desplaza hacia la derecha o izquierda.

**Flecha Arriba/Abajo:** el avión se desplaza hacia arriba o abajo.

**Tecla espacio:** realiza un disparo del arma.

**Tecla enter:** el avión realiza una vuelta en el aire (muestra una animación)

**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.

El archivo de configuración especifica:

- La velocidad relativa de desplazamiento del avión en [píxeles/frames]
- La velocidad de los disparos en [píxeles/frames]
- El id del sprite del avión
- El id del sprite de la animación del avión dando la vuelta
- El id del sprite de los disparos del avión

Tener en cuenta que si no se presionan teclas, el avión tiene una velocidad igual a la de la cámara. La velocidad que se especifica es la velocidad relativa del avión con respecto a la cámara cuando el jugador utiliza las teclas. Esta velocidad vale tanto para desplazamientos verticales como horizontales.

## Restricciones

- Para la representación gráfica se deberá utilizar la biblioteca SDL 2.0 (<http://libsdl.org>).
- Todo el código debe ser desarrollado íntegramente por cada grupo. No se permite la reutilización de código de cuatrimestres anteriores o de otras asignaturas. 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

<b>Semana #</b>	<b>Fecha</b>	<b>Tema</b>
1	9 de marzo	Presentación de la materia
2	16 de marzo	Presentación enunciado TP 1
3	23 de marzo	Consultas
4	30 de marzo	Consultas
<b>5</b>	<b>6 de abril</b>	<b>Entrega TP1. Presentación enunciado TP 2</b>
6	13 de abril	Primer recuperatorio TP1. Consultas TP 2
7	20 de abril	Segundo recuperatorio TP1. Consultas TP 2
8	27 de abril	Consultas
<b>9</b>	<b>4 de mayo</b>	<b>Entrega TP2. Presentación enunciado TP 3</b>
10	11 de mayo	Primer recuperatorio TP2. Consultas TP 3
11	18 de mayo	Segundo recuperatorio TP2. Consultas TP 3
<b>12</b>	<b>25 de mayo</b>	<b>Feriado</b>
13	1 de junio	Consultas
<b>14</b>	<b>8 de junio</b>	<b>Entrega TP3</b>
15	15 de junio	Primer recuperatorio TP 3
16	22 de junio	Segundo recuperatorio TP 3

## Anexo: Archivo de escenario de ejemplo

```
<ventana>
  <ancho>800</ancho>
  <alto>600</alto>
</ventana>
<sprites>
  <sprite>
    <id>avion</id>
    <path>path/a/avion.png</path>
    <cantidad>1</cantidad>
    <ancho>10</ancho>
    <alto>10</alto>
  </sprite>
  <sprite>
    <id>disparo</id>
    <path>path/a/disparo.png</path>
    <cantidad>1</cantidad>
    <ancho>5</ancho>
    <alto>5</alto>
  </sprite>
</sprites>
<escenario>
  <ancho>100</ancho>
  <alto>1000</alto>
  <fondo>
    <spriteld>agua</spriteld>
    <ancho>10</ancho>
    <alto>10</alto>
  </fondo>
  <elementos>
    <elemento>
      <spriteld>isla</spriteld>
      <posicion>
        <x>10</x>
        <y>15</y>
      </posicion>
    </elemento>
    <elemento>
      <spriteld>isla</spriteld>
      <posicion>
        <x>15</x>
        <y>20</y>
      </posicion>
    </elemento>
  </elementos>
</escenario>
```

```
<avion>
  <velocidadDesplazamiento>10</velDesplazamiento>
  <velocidadDisparos>6</velocidadDisparos>
  <avionSpriteId>avion</avionSpriteId>
  <vueltaSpriteId>disparo</vueltaSpriteId>
  <disparosSpriteId>vuelta</disparosSpriteId>
</avion>
```

---

---