

## Trabajo Práctico: Fase 1 - Parte Gráfica

### Objetivo

Desarrollar la interfaz gráfica de un juego de plataformas. Tomar como guía el juego “**Super Mario Bros**” (“<https://www.youtube.com/watch?v=rLI9XBg7wSs>”)

### Aplicación

Se debe desarrollar una aplicación, la cual será lanzada a través de CLI (interfaz de línea de comando). La aplicación tomará los parámetros de configuración inicial a través de un archivo de configuración.

Todos los elementos visuales a emplear son de libre elección, simplemente considerar semejanzas con los originales.

### Personaje principal

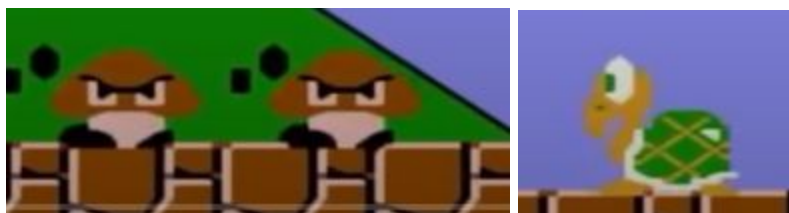
En todo momento el jugador deberá poder controlar al personaje principal, este debe poder agacharse, saltar y caminar hacia la derecha o izquierda (incluir saltos con desplazamiento lateral).

Considerar el tamaño del personaje crecido.



### Enemigos

En el escenario deben aparecer dos tipos de enemigos diferentes distribuidos de forma aleatoria, la cantidad de los mismos debe ser configurable. En esta etapa del desarrollo los mismos se encontrarán desplazándose por el escenario, pero no tendrán ningún tipo de interacción.



## Los Controles

El jugador debe poder desplazar a su personaje por el escenario mediante las cuatro teclas direccionales y sus combinaciones.

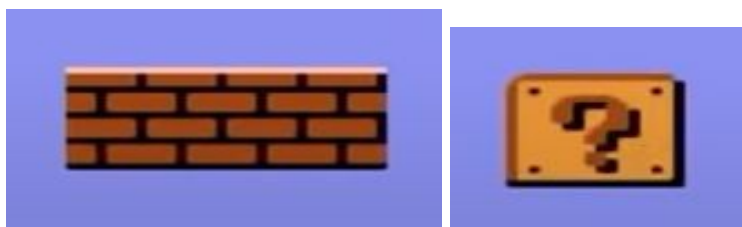
## Escenario

Se deberán implementar 3 escenarios. Con fondos fijos (sin parallax).

En esta etapa del desarrollo no se implementarán pozos ni plataformas móviles.

Se deberán implementar 2 tipos de plataformas diferentes (ladrillos y sorpresas). Deberán implementarse las plataformas en al menos 2 alturas diferentes como se visualiza en la **\*Imagen escenario**, que se encuentra más adelante en este texto.

La distribución de las plataformas deberá ser completamente configurable, indicando para cada plataforma el tipo, coordenada (x,y) y cantidad de elementos que la componen.



Implementar indicación de nivel y tiempo restante (tiempo configurable).



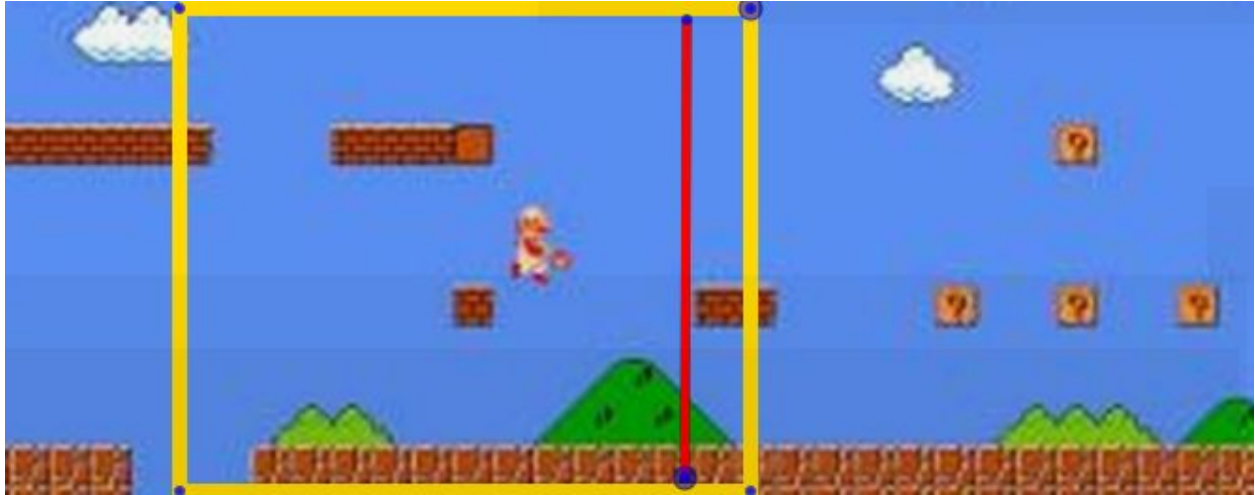
En los escenarios deberán aparecer monedas distribuidas aleatoriamente (cantidad configurable).



No implementar transiciones entre escenarios.

## Ventana de Juego

La ventana de juego mostrará la zona activa del escenario la cual se desplazará mediante un scroll horizontal que se desplazará a medida que el personaje avance sobre el margen derecho. No podrá retornar hacia la izquierda. Esta ventana tendrá un tamaño mínimo de 800x600 pixels.



**\*Imagen escenario**

## Log de eventos y errores

Se debe registrar información en un archivo de texto sobre la actividad que realiza la aplicación.

Se deben considerar tres niveles: **3-DEBUG**; **2-INFO**; **1-ERROR**. El nivel de log, debe poder modificarse en el archivo de configuración y también debe poder introducirse como parámetro por línea de comando al lanzar la aplicación.

**Cada nivel de log incluye a sus niveles inferiores.**

2020-09-10 08:51:01.233 [INFO]: Se lee archivo de configuración en ruta "/home/test/file.xml" exitosamente.

2020-09-10 08:51:01.345 [ERROR]: No se ha podido cargar la imagen "/home/test/enemigo.bmp" del enemigo tipo 1. Se carga una por defecto.

2020-09-10 08:51:01.711 [INFO]: Se ha finalizado la carga de imagenes.

2020-09-10 08:51:02.154 [INFO]: Inicio del juego.

2020-09-10 08:51:02.234 [DEBUG]: Posicion del personaje 1 (x,y)=(10.23,0)

2020-09-10 08:51:02.234 [DEBUG]: El personaje 1 salta.

2020-09-10 08:51:02.489 [DEBUG]: Posicion del personaje 1 (x,y)=(10.23,5)

2020-09-10 08:51:02.489 [DEBUG]: Posicion del personaje 1 (x,y)=(10.23,2)

2020-09-10 08:51:02.721 [DEBUG]: Posicion del personaje 1 (x,y)=(10.23,0)

## Configuración

El archivo de configuración será un archivo de texto plano en formato **xml**, el cual contendrá los parámetros de configuración inicial de la aplicación. En caso de no encontrar el archivo, se debe leer un archivo de configuración por defecto.

### Ejemplo de configuración:

```
<configuracion>
  <log>
    <level>ERROR</level>
  </log>
  <ventana>
    <ancho>800</ancho>
    <alto>600</alto>
  </ventana>
  <escenario>
    <niveles>
      <nivel>
        <numero>1</numero>
        <fondo>/home/file21.bmp</fondo>
        <monedas>30</monedas>
        <enemigos>
          <enemigo>
            <tipo>1</tipo>
            <imagen>/home/file3.bmp</imagen>
            <cantidad>15</cantidad>
          </enemigo>
          <enemigo>
            <tipo>2</tipo>
            <imagen>/home/file4.bmp</imagen>
            <cantidad>20</cantidad>
          </enemigo>
        </enemigos>
        <plataformas>
          <plataforma>
            <tipo>ladrillo</tipo>
            <coordX>100</coordX>
            <coordY>200</coordY>
            <cantidad>5</cantidad>
          </plataforma>
          <plataforma>
            <tipo>sorpresa</tipo>
            <coordX>200</coordX>
            <coordY>300</coordY>
            <cantidad>6</cantidad>
          </plataforma>
        </plataformas>
      </nivel>
    </niveles>
  </escenario>
</configuracion>
```

Pueden agregarse la cantidad de parámetros que se consideren necesarios, como así también darle la estructura que consideren necesario al archivo **xml**.

## **Restricciones**

- La implementación deberá estar hecha en C/C++.
- Para la lectura y escritura de archivos XML debe utilizarse una biblioteca.
- Para el manejo de gráficos se deberá usar la biblioteca SDL.
- 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.