

# TP1: Interfaz Gráfica y Simulación

*Taller de Programación I - Cátedra Ing. Diego Azcurra - 2do cuatrimestre 2014*

## Objetivo

Desarrollar un programa en C/C++ que muestre un escenario. El escenario consta de un personaje animado y controlable y un conjunto de plataformas con las que el personaje puede interactuar.

## 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 y lo guardará en dicho archivo. El nombre del archivo será pasado por línea de comandos.
2. Muestre dicho escenario gráficamente en una pantalla. El escenario estará compuesto por:
  - El tamaño del escenario en píxels y en unidades.
  - Un fondo. La imagen utilizada para mostrar el fondo se obtendrá de un archivo externo de imagen.
  - Un conjunto de objetos geométricos que representan entre otras cosas plataformas sobre las cuales el protagonista se puede parar
  - 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.

## Objetos geométricos

Los objetos pueden ser estáticos o dinámicos. Los objetos estáticos conservan su posición inicial durante toda la simulación. Los objetos dinámicos caen por efecto de la gravedad, pueden colisionar con otros objetos e interactuar con el protagonista (eg. ser empujados).

Los objetos tendrán los siguientes parámetros:

- **Posición:** dos números reales representando la posición del centro de gravedad del objeto en unidades.
- **Tamaño:** dos números reales representando el ancho y alto de la figura en unidades. Solo los rectángulos tienen estos parámetros.
- **Escala:** un número real positivo que representa el tamaño de la figura. Los rectángulos no tienen este parámetro.
- **Rotación:** un número entero entre 0 y 359 que representa el ángulo de rotación de la figura, en sentido horario.
- **Color:** una cadena de caracteres que representa el color de fondo de la figura en notación hex triplet

([http://en.wikipedia.org/wiki/Web\\_colors#Hex\\_triplet](http://en.wikipedia.org/wiki/Web_colors#Hex_triplet)).

- **Masa:** un número real positivo indicando la masa del objeto.
- **Estático:** una bandera que indica si el objeto se mantiene estático durante la simulación o no.

Se deben implementar las siguientes formas geométricas:

- **Polígonos regulares.** Podrán tener hasta 6 lados.
- **Rectángulos.** Pueden representar tanto cajas que el protagonista puede empujar como las plataformas fijas del nivel.
- **Círculos.** Serán principalmente dinámicos y podrán rodar por el escenario.
- **Paralelogramos o trapecios** que se utilizarán principalmente para construir rampas)

## Animación del protagonista

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

## Aclaraciones

- La imagen del fondo puede estar en cualquier formato (por ejemplo: bmp, png, jpg) y puede ser cargada con las bibliotecas de extensión de SDL.
- En este trabajo práctico las figuras se deberán pintar de un color sólido de fondo. En próximos trabajos prácticos existe la posibilidad de que sean pintadas con alguna textura.
- Los círculos, al ser de un color sólido, no evidenciarán su rotación. Por lo tanto, los mismos deben tener alguna "marca" para romper su simetría y poder verlos rodar.
- 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.
- Todos los valores numéricos en el archivo de escenario que indiquen unidades de medida (tamaños, posiciones, etc) deberán ser números reales relativos a las dimensiones del escenario en unidades. Estos valores deberá luego ser escalados al tamaño del escenario en píxels.
- 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 simulación física se deberá utilizar la biblioteca Box2D (<http://box2d.org>).
- Para la lectura y escritura de archivos JSON debe utilizarse, preferiblemente, la biblioteca yaml-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         | 20 de agosto            | Presentación de la materia                      |
| 2         | 27 de agosto            | Presentación enunciado TP 1                     |
| 3         | 3 de septiembre         | Consultas                                       |
| 4         | 10 de septiembre        | Consultas                                       |
| 5         | 16 de septiembre        | Consultas                                       |
| <b>6</b>  | <b>24 de septiembre</b> | <b>Entrega TP1. Presentación enunciado TP 2</b> |
| 7         | 1 de octubre            | Primer recuperatorio TP1. Consultas TP 2        |
| 8         | 8 de octubre            | Segundo recuperatorio TP1. Consultas TP 2       |
| 9         | 15 de octubre           | Consultas                                       |
| <b>10</b> | <b>22 de octubre</b>    | <b>Entrega TP2. Presentación enunciado TP 3</b> |
| 11        | 29 de octubre           | Primer recuperatorio TP2. Consultas TP 3        |
| 12        | 5 de noviembre          | Segundo recuperatorio TP2. Consultas TP 3       |
| 13        | 12 de noviembre         | Consultas                                       |
| <b>14</b> | <b>19 de noviembre</b>  | <b>Entrega TP 3</b>                             |
| 15        | 26 de noviembre         | Primer recuperatorio TP 3                       |
| 16        | 3 de diciembre          | Segundo recuperatorio TP 3                      |

## Anexo: Archivo de escenario de ejemplo

```
{
  "escenario": {
    "alto-px": 768,
    "ancho-px": 1024,
    "alto-un": 100,
    "ancho-un": 50,
    "imagen_fondo": "fondo1.png",
    "personaje": {
      "x": 100,
      "y": 100
    },
    "objetos": [
      {
        "tipo": "rect",
        "x": 5,
        "y": 90,
        "ancho": 2,
        "alto": 1,
        "color": "#00FF00",
        "rot": 45,
        "masa": 3,
        "estatico": false
      },
      {
        "tipo": "poli",
        "lados": 3,
        "x": 50,
        "y": 50,
        "escala": 3.5,
        "color": "#FF0000",
        "rot": 90,
        "masa": 1,
        "estatico": true
      }
    ]
  }
}
```