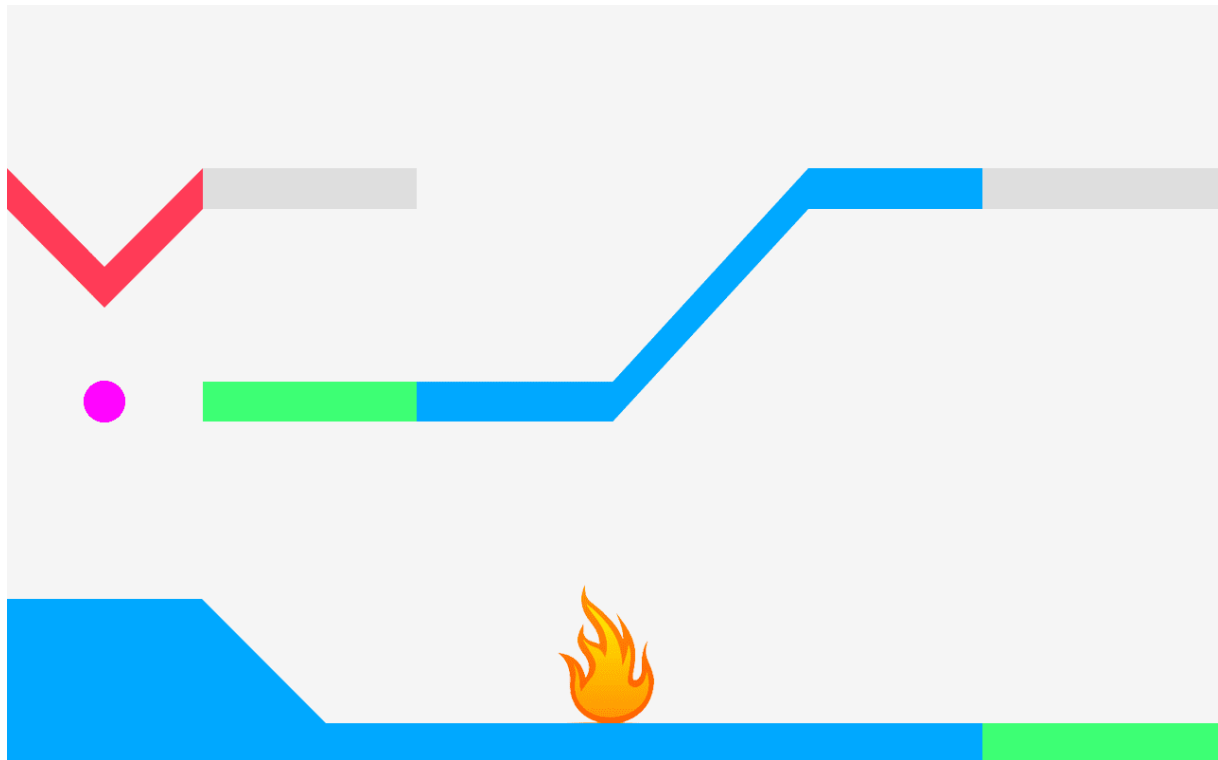


## PRÁCTICA OBLIGATORIA DEL PRIMER SEMESTRE

### REQUISITOS

Dada la siguiente escena animada:



se pide implementar la escena y uno o varios mecanismos basados en física que dejen caer el círculo magenta **sobre** la V roja teniendo en cuenta las siguientes indicaciones:

- El círculo magenta cae hacia abajo por sí solo atraído por la gravedad.
- Las plataformas deben colisionar con cualquier objeto en cualquier dirección.
- Los mecanismos que mueven el círculo pueden ser controlados (opcionalmente) con el teclado y/o el ratón.
- No se pueden añadir nuevos cuerpos estáticos (static) o cinemáticos (kinematic), solo dinámicos (dynamic).
- Los rectángulos verdes son plataformas que ascienden hasta los rectángulos grises que tienen encima cuando hay un objeto sobre ellos. Si no hay nada sobre ellos, descienden hasta su posición original (los rectángulos grises no son objetos, solo marcan hasta dónde ascienden las plataformas).

- En el trayecto de llevar el círculo magenta a su destino se debe pasar por las dos plataformas verdes.
- Las rampas inclinadas tienen una pendiente de 45°.
- El fuego se debe implementar como un sistema de partículas contenido dentro de un área rectangular similar a la del dibujo de muestra.
- Si cualquier cosa toca el fuego, se debe reiniciar la escena por completo.
- El dibujo de la escena es representativo. No tiene porqué hacerse exactamente igual en cuanto a aspecto o proporciones.
- La programación se debe realizar enteramente con C++ sin usar un engine de juegos de terceros. Sin embargo, sí se pueden usar librerías auxiliares como SFML, SDL2, Box2D, etc. Por ejemplo, SFML es suficiente para dibujar la escena y Box2D para controlar la animación. Cualquier aspecto de los que se evalúan debe ser implementado con código propio.
- El software se debe poder ejecutar en Windows, Linux o Mac OS X.
- El trabajo debe ser individual y original.

## ENTREGABLES

Se debe entregar lo siguiente comprimido todo en un archivo ZIP:

- Dentro de una carpeta llamada `code`: el código fuente completo (propio y de terceros) que compile sin errores.
- Dentro de una carpeta llamada `libraries`: cualquier librería precompilada de terceros que se haya usado.
- Dentro de una carpeta llamada `project`: un proyecto de Visual Studio 2015, un proyecto de XCode o un proyecto de Netbeans (si se trabaja en Linux). No se deben incluir archivos temporales del proyecto.
- Dentro de una carpeta llamada `documents`: cualquier documentación que describa el código fuente (arquitectura, uso, notas, etc.). No hay un mínimo ni un máximo de documentación que se deba entregar.
- Dentro de una carpeta llamada `binaries`: el software compilado para que se pueda ejecutar en alguna de las plataformas establecidas en los requisitos incluyendo todos los elementos necesarios (librerías de enlace dinámico, archivo o archivos de datos de la escena, etc.).

## VALORACIÓN

- Ausencia de errores y warnings: 0,5 puntos.
- Calidad del código y del diseño del código: 1 punto.
- Documentación: 0,5 puntos.
- Modelo de la escena: 1 punto.
- Uso de motores: 1 puntos.
- Mecanismos que transportan el círculo: 1 punto.
- El círculo llega al destino: 1 punto.
- Modelo y control de las plataformas ascendentes: 1 punto.
- Paso por las dos plataformas ascendentes: 0,5 puntos.
- Modelo y control del sistema de partículas: 2 puntos.
- Restablecimiento de la escena si se toca el fuego: 0,5 puntos.

## ACLARACIÓN DE DUDAS

[angel.rodriguez@esne.edu](mailto:angel.rodriguez@esne.edu)