

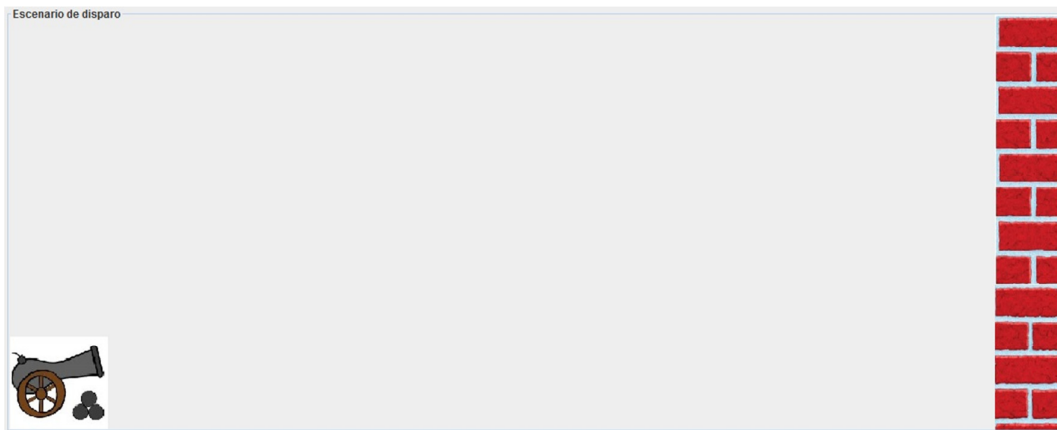
Objetivo:

Al final de esta actividad se espera que el estudiante sea capaz de:

- Identificar los elementos de programación gráfica por medio del reconocimiento de sus utilidades.
- Desarrollar aplicaciones con animación basada en programación gráfica.

Etapas 1: Diseño de la escena

Haciendo uso de las utilidades de programación gráfica realice complete el proyecto para que en el panelAnimacion se dibuje un cañón y una pared, tal como se muestra a continuación:

**Etapas 2: Diseño de la animación**

1. Revise el diagrama de clases que se encuentra en la documentación del proyecto. Entienda las relaciones que allí se presentan.
2. Termine la implementación del método *generarColor():void* de la clase *Bala*, encargado de asignar color aleatorio en formato RGB. Recuerde que cada componente RGB tiene un rango válido desde 0 a 255 incluido.
3. Agregue a la implementación del método *cargarBala():void* de la clase *Canon*, el código necesario para que cree una nueva bala y la agregue a la lista enlazada en la primera posición.
4. Agregue a la implementación del método *paintComponent(Graphics):void* de la clase *PanelCanon*, el código necesario para que dibuje las balas en el panel de acuerdo a las coordenadas x,y de cada objeto *Bala*.
5. En la definición del método *lanzarBala():void* de la clase *InterfazCanon*, agregue:
 - a. Establezca sobre la instancia de *Canon* la velocidad inicial.
 - b. Establezca sobre la instancia de *Canon* el ángulo inicial.

- c. Solicite a la instancia de *Canon* el método *cargarBala():void*

Dentro de la instrucción de Thread

- d. Saque la primera bala de la lista de Canon y asígnela a una variable local.
e. Cree un ciclo que se repita hasta que la *primeraBala* de la instancia de *Canon* se encuentre en el máximo X de la ventana o máximo Y. Y que avance sumando 10 a la posición X de la bala.

Dentro del ciclo realice:

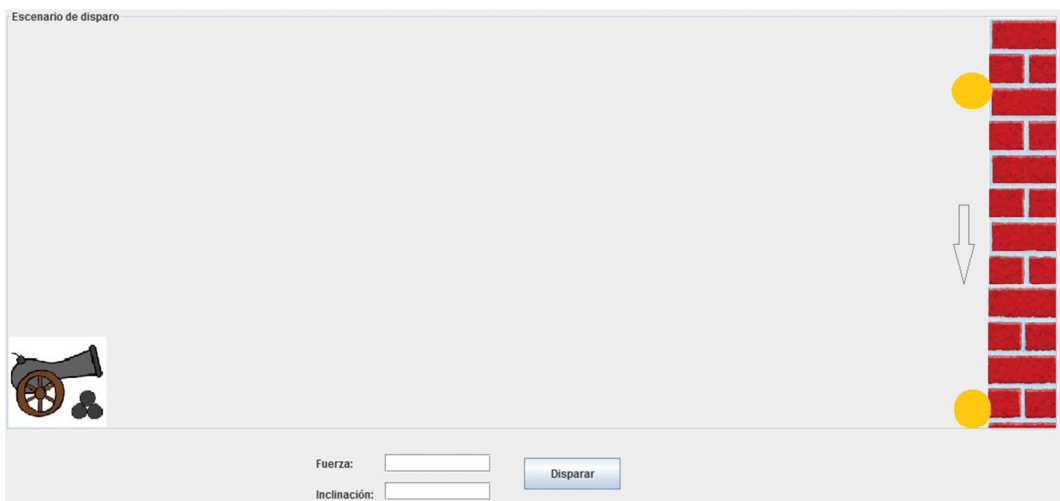
- i. Sobre la instancia de la última bala solicitar que se cambie el valor de la coordenada X.
- ii. Sobre la instancia de *PanelCanon* llame al método *repaint()*;
- iii. Escriba la línea: `Thread.sleep(20);`

- f. Cierre el ciclo

- `repaint()` solicita que se ejecute nuevamente el método `paintComponent(Graphics)`
- `Thread.sleep(2000)` le indica a la aplicación que debe “dormirse” por los milisegundos especificados.

Etapa 3: Extensión

1. Documente e implemente los cambios que considere necesarios en el diagrama de clases y el código de la aplicación, para que cada vez que una bala quede “estrellada” en la pared (porque llego al máximo en X) caiga hacia abajo hasta un Y=0.



2. Documente e implemente los cambios que considere necesarios en el diagrama de clases y el código de la aplicación, para que se utilice la información del campo de texto de ángulo para dibujar el cañón en el ángulo de disparo especificado.