

Análisis - quiz

Juan Felipe Blanco Tigreros

Contexto:

Se deberá crear un programa que pueda crear y mover (constantemente) dos tipos de figura (cuadrado y círculo), ya sea por medio de la lectura de un archivo de texto plano (con datos predeterminados) o presionando clic derecho (generando una figura con datos aleatorios). Los atributos de estas figuras son: tamaño, posición en ambos ejes, dirección y valor. Dichas figuras no podrán salirse del lienzo y en caso de que se choquen entre sí, crean una nueva figura (triángulo, que también deberá de moverse constantemente) de tamaño aleatorio y que contenga en su interior la suma de los valores de las dos figuras que se fusionaron. Cabe resaltar que la fusión puede suceder entre cualquier tipo de figura y una vez sucede, las dos formas que chocan son eliminadas del sistema. Finalmente, si se hace un clic normal sobre una figura, esta deja de moverse y si se presiona de nuevo se vuelve a mover.

Requerimientos funcionales:

El programa deberá estar en la capacidad de:

R1 - Leer un archivo de texto plano

Descripción: El programa deberá leer un archivo de texto plano de tal forma que extraiga información acerca de las figuras que van a ser creadas, es decir, sus atributos (tamaño, posición en el eje X y Y, dirección y valor).

Precondición: El archivo de texto a leer debe existir.

Entradas: Archivo de texto plano.

Salidas: Información acerca de cuántas figuras deben crearse y sus atributos.

Poscondición: --

R2 - Crear figuras con base en texto plano

Descripción: El programa deberá crear una figura de dos posibles (cuadrado y círculo) con base en la información extraída del archivo de texto.

Precondición: Lectura de archivo exitosa

Entradas: Datos extraídos del texto plano

Salidas: --

Poscondición: Se crean las figuras de acuerdo a la información suministrada

R3 - Crear figuras con clic derecho

Descripción: El programa deberá generar una figura aleatoria entre círculo y cuadrado con dirección, tamaño, valor y posiciones aleatorias (que estén dentro del lienzo); siempre y cuando el usuario haya presionado clic derecho.

Precondición: presionar clic derecho

Entradas: --

Salidas: --

Poscondición: Crear una nueva figura (cuadrado o círculo).

R4 - Mover figuras

Descripción: El programa deberá mover las figuras creadas por el lienzo sin que estas se salgan del mismo; sus valores deben ser visibles y moverse con ellas.

Precondición: --

Entradas: --

Salidas: --

Poscondición: --

R5 - Crear un triángulo al impactar dos figuras

Descripción: El programa deberá crear una nueva figura (triángulo) si hay una colisión entre cualquier figura. Este triángulo tendrá valores aleatorios para sus atributos con la excepción de su valor, el cual será la suma de los valores de las dos formas que chocaron.

Precondición: Colisión entre dos figuras

Entradas: posiciones de ambas figuras que colisionan al igual que sus valores

Salidas: --

Poscondición: Se deberán eliminar las dos figuras que colisionaron y crear una nueva figura (triángulo)

R6 - Congelar y descongelar figuras

Descripción: Si se hace clic sobre una figura esta dejará de moverse hasta que se le haga clic nuevamente.

Precondición: Haber hecho clic sobre una figura.

Entradas: posición en X y Y del mouse y de las figuras

Salidas: --

Poscondición: La figura seleccionada se detendrá o se volverá a mover dependiendo del caso.

Requerimientos no funcionales:

El programa debe ser creado en Java.

El programa debe ser creado con colores RGB.

Paso a paso:

1. Leer archivo de texto

- a. Separar el texto por líneas y guardarlas en un arreglo de String
 - b. separar cada línea (posición del arreglo) por espacios y guardarlo en una lista de String
2. Crear las figuras iniciales del archivo
 - a. (idea inicial) Recorrer la lista de String tomando la información de la posición actual de la variable de avance más las siguientes 5 y almacenarlas en atributos temporales haciendo los correspondientes casteos al tipo de variable que requiere cada atributo (ya que se conocen cuántos y de qué tipo son). Teniendo un avance de 6 en lugar de uno en dicho ciclo.
 - b. Al final de cada ciclo crear un nuevo objeto figura, dependiendo del caso con las variables temporales recién asignadas.
3. Mover las figuras
 - a. Crear un método que usando la información del máximo y mínimo del lienzo sepa cuándo cambiar de dirección si es requerido.
4. Crear figuras con clic
 - a. Al hacer clic normal se deberá llamar al método encargado de crear figuras de manera aleatoria el cual generará un número aleatorio para decidir si se creará un círculo o un cuadrado. Posteriormente se crearán los demás campos aleatoriamente (procurando que sean valores posibles, como posiciones dentro de los límites del lienzo y tamaños razonables).
5. Congelar y descongelar con clic
 - a. Al hacer clic derecho, verificar que las posiciones del mouse en ambos ejes corresponden a la posición de una figura del lienzo (por medio de un ciclo)
 - b. Si en efecto se le hizo clic, hacerle set a una variable de las figuras que controla si se mueve o no se mueve (que posiblemente se verifique dentro de su propio método de movimiento); este set se hará obteniendo el valor que posea actualmente esta variable y asignándole su opuesto.
6. Verificar impactos
 - a. Verificar que la distancia entre los centros de las figuras sea menor a la suma de sus radios (todavía no se ha decidido la mejor manera de detectar las colisiones con el cuadrado y triángulo). Esto se haría todo el tiempo.
7. Crear un triángulo

- a. Al detectar un impacto, se crearán atributos temporales aleatorios para la dirección y tamaño de la nueva figura, se sumarán los valores de las dos que chocaron y se tomará el punto medio entre ambas al momento del impacto. Con lo anterior se creará una nueva figura (triángulo) en las posiciones X y Y del punto medio hallado, con un valor igual a la suma realizada y sus demás atributos aleatorios.
8. Eliminar figuras que colisionan
- a. Al detectar un impacto se tomarán las posiciones de las figuras que chocaron y con ellas se buscarán en la lista de figuras y serán removidas.