|  |  |  |
| --- | --- | --- |
| Texto  Descripción generada automáticamente con confianza media | TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS  FACULTAD DE INGENIERÍA  Universidad Nacional de Jujuy |  |

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

*Ing. Carolina Cecilia Apaza*

*Año*

Trabajo Práctico/Actividad

N°1

Apellido y Nombre – LU /

Flores Noemi Cintia- TUV000758

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Indice

Caratula \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-pag1

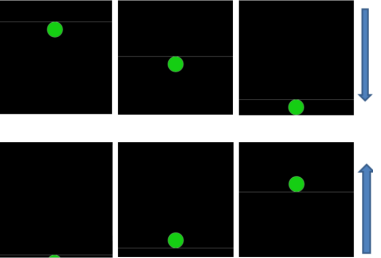
Indice \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pag2

Ejercicio 19\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pag3

Ejercicio 19\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pag4

Conclusion y bibliografia\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_pag5

Ejercicio 19: Declare las variables necesarias para dibujar una línea que se dibuja desde las coordenadas iniciales del lienzo y se extiende por todo el ancho. Sobre el punto medio de la línea y a una distancia de 40px (en sentido vertical desde la línea) dibuje una elipse que tenga como ancho 80px y de alto 80px. Dentro de la función draw(), actualice las variables necesarias para que la línea desde su inicio se mueva en dirección hacia abajo arrastrando la elipse. Mantenga en cero el valor para background(). Cuando la línea supere la posición de la altura del lienzo, debe invertir su sentido, es decir dirigirse hacia arriba arrastrando la elipse. Cuando la línea alcance nuevamente el valor 0 para su posición en y, el desplazamiento debe ser hacia abajo y así sucesivamente. El lienzo debería verse como en las siguientes figuras



Desarrollo del punto

Definicion del problema: dibujar una línea horizontal en el lienzo, la cual se mueva hacia arriba y hacia abajo. Sobre el punto medio de la línea y a una distancia de 40 píxeles (en sentido vertical desde la línea) se debe dibujar una elipse de 80x80 píxeles. Cuando la línea supere la posición de la altura del lienzo, debe invertir su dirección y dirigirse hacia arriba, arrastrando consigo la elipse. Cuando la línea alcance nuevamente la posición 0 en Y, el desplazamiento debe ser hacia abajo, y así sucesivamente.

Fase de análisis

* Datos de entrada: Coordenadas iniciales del lienzo.
* Tamaño de la elipse.
* Velocidad de movimiento de la línea.

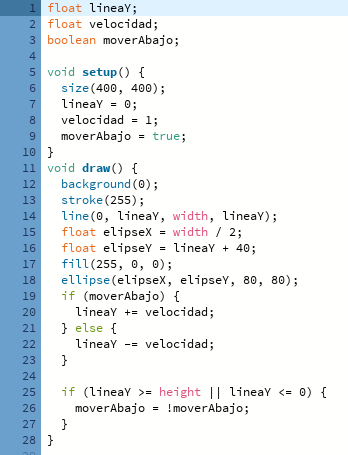
Datos de salida: Dibujo de la línea y la elipse en el lienzo.

Proceso:

* Dibujar una línea horizontal en el lienzo.
* Dibujar una elipse centrada sobre el punto medio de la línea, a una distancia vertical de 40 píxeles.
* Mover la línea hacia arriba y hacia abajo en función de la velocidad.
* Invertir la dirección de la línea cuando alcance los bordes superior o inferior del lienzo

Fase de diseño

|  |
| --- |
| Entidad Que Resuelve El Problema: Algoritmo en Processing |
| Variables   * lineaY: Posición Y de la línea en el lienzo. * velocidad: Velocidad de movimiento de la línea. * moverAbajo: Variable booleana para controlar la dirección del movimiento de la línea. |
| Nombre Del Algoritmo: DibujarLineaConElipse  Proceso Del Algoritmo:  Inicializar el lienzo y las variables necesarias.  En el bucle draw():  a. Dibujar la línea horizontal en la posición especificada por lineaY.  b. Calcular la posición de la elipse en función de lineaY.  c. Dibujar la elipse en la posición calculada.  d. Actualizar la posición de la línea y verificar los límites del lienzo.  e. Invertir la dirección de la línea si alcanza los bordes del lienzo. |



Conclusión

En este ejercicio numero 19, logré crear un algoritmo en Processing que dibuja una línea horizontal moviéndose hacia arriba y hacia abajo en el lienzo. También agregué una elipse que sigue a la línea a una distancia fija verticalmente. Cuando la línea alcanza los bordes superior o inferior del lienzo, cambia de dirección. Este ejercicio me ayudó a comprender cómo utilizar variables y estructuras de control para crear animaciones simples y dinámicas en Processing.

Fuentes bibliográficas

Bohnacker, H. et al. (2012). Generative Design: Visualize, Program, and Create with Processing. Nueva York\_ Princeton Architectural Press. Glassner, A. (2010). Processing for Visual Artists: How to Create Expressive Images and Interactive Art. Natick, MA: AK Peters, Ltd. Greenberg, I. (2007). Processing: creative coding and computational art. Nueva York: Apress. Reas, Casey & Fry, B. (2007). Processing: a programming handbook for visual designers and artists. Cambridge: MIT Press. Reas, Casey & Fry, B. (2010). Getting Started with Processing. California, EUA: O’Reilly Media, Inc. Shiffman, D. (2009). Learning Processing: a beginner’s guide to programming images, animation, and interaction. Massachusetts: Morgan Kaufmann,. Terzidis, K. (2009). Algorithms for visual design using the processing language. Indianapolis: John Wiley & Sons. Recursos web\*: http://processing.org http://www.joan.cat/processing/cs/ http://www.learningprocessing.com/tutorials/ Recursos Youtube\*: https://www.youtube.com/watch?v=9UcL8B0GQuE&list=PL374E5107CB62B2BE https://www.youtube.com/watch?v=ZByppaYSJB8

Fundamentos de Programación: Algoritmos, estructuras de datos y objetos. Cuarta edición. Luis Goyanes Aguilar. ISBN: 978-84-481-6111-8

<https://www.youtube.com/watch?v=Nr3NdAxjqsE> PROCESSING | Estructuras de control | Condicionales | Programación videos de youtube subidas por el canal de [STEM con Pablo](https://www.youtube.com/@STEMconPablo)