**Análisis y Diseño**

**Ejercicio 17: Si queremos representar personajes o power ups (premios) en la pantalla debemos**

**primero ubicarlos en alguna posición dentro de la pantalla. Imagine que está en un juego donde un**

**power up desaparece porque el personaje se acerca a una distancia de x unidades, sin importar por**

**dónde se acerque. Por tanto, para que desaparezca, en primer lugar, hay que determinar esa**

**distancia. La forma de representar la posición de un objeto en la pantalla es a través de las**

**coordenadas de un punto. Suponga que la posición de Link está representada por la coordenada (��1,**

**��1) , mientras que las de la caja de tesoro se hallan en la posición (��2, ��2). Si observa con detenimiento**

**se observa la conformación de un triángulo rectángulo, por lo que es posible aplicar Pitágoras para**

**obtener la distancia**

**Resultado:**

**ANÁLISIS DEL PROBLEMA:** Deberá configurarse para la consola lea el código basado en un proceso de variables otorgando distancias y coordenadas entre link y el tesoro para cumplir con los parámetros

especificados en el ejercicio.

**Análisis:**

**Datos de entrada:**

Las coordenadas de link (x1,y1).

Las coordenadas del tesoro(x2,y2)

**Datos de salida:**

La distancia entre link y la caja del tesoro

**Proceso:**

**¿Quién realiza el proceso?:** puede ser una persona o un programa

**¿Cuál es el proceso que realiza?:**

1 – Calcular las diferencias entre las coordenadas x de link y la caja del tesoro: dx =

x2-x1.

2 – Calcular las diferencias entre las coordenadas y de link y la caja del tesoro: dy=

y2- y1.

3 – Aplicar la fórmula de distancia utilizando las diferencias calculadas en los pasos 1

y 2 para obtener la distancia entre el link y la caja del tesoro.

Realizar las operaciones.

**PSEUDOCÓDIGO:**

| **NOMBRE DE LA ENTIDAD QUE RESUELVE EL PROBLEMA:** PROGRAMA |
| --- |
| **Variables:** // definir las variables  //Inicio  Definir las coordenadas de link  Definir x1, y1 como números  X1valor  Y1 valor  //Definir las coordenadas de la caja del tesoro  Definir x2, y2 como números  X2 valor  Y2 valor  //Calcular la distancia entre link y la caja del tesoro  Definir distancia como número  Distancia raiz cuadrada de ((x2 – x1)^2 + (y2 – y1)^2)  //Mostrar la distancia  Escribir “La distancia entre link y la caja de tesoros: “distancia, “” unidades.”  //Decidir si el power up debe desaparecer  Si distancia &lt;= 50 entonces  Escribir “¡Power up obtenido!”  Sino  Escribir “ Link debe acercarse más a la caja del tesoro.”  FinSi  **fin** |
| **NOMBRE DEL ALGORITMO:** link\_buscandotesoro  **PROCESO DEL ALGORITMO:** Busca tesoro a partir de variables y establece parámetros  //Inicio  PVector posLink;  PVector posTesoro;  public void setup() {  size(500, 500);  posLink = new PVector(100, 100);  posTesoro = new PVector(width / 2, height / 2);  }  public void draw() {  background(0);  dibujarLink();  posLink.x = mouseX;  posLink.y = mouseY;  validarDibujarTesoro();  }  public void dibujarLink() {  circle(posLink.x, posLink.y, 40);  }  public void dibujarTesoro() {  rectMode(CENTER);  rect(posTesoro.x, posTesoro.y, 100, 70);  }  public float obtenerDistancia() {  float distancia = 0;  float cateto1 = posTesoro.x - posLink.x;  float cateto2 = posTesoro.y - posLink.y;  distancia = sqrt(pow(cateto1, 2) + pow(cateto2, 2));  return distancia;  }  public void validarDibujarTesoro() {  if (obtenerDistancia() 100) {  println(Debe desaparecer el tesoro);  } else {  dibujarTesoro();  }  }  Fin |
|  |



