

ICC343 - Programación Avanzada 2021-1

Tutorial #6 - Clases y Objetos PHP.

Tal como los otros tutoriales, este le ofrece un conjunto de actividades para ser desarrolladas EN EL ORDEN en que están propuestas. Al realizar estas actividades se pide un conjunto de actividades de evidencia, entre ellas fotos de pantalla (screenshots) en las cuales se debe ver, claramente, (1) el código en un editor HTML-Javascript, (2) el resultado en un navegador, y (3) parte de su fondo de escritorio, es decir, parte de la imagen de fondo y el reloj, de forma de verificar la secuencia de las actividades.

La forma de entregar los resultados es en el canal Slack del curso, pegando sus resultados en el canal privado para revisión del profesor, en el momento que termine cada programa. El profesor reportará su puntaje y comentará elementos técnicos de su respuesta. Eventualmente le puede pedir re-construir su solución para que se adhiera a lo solicitado, pero esto no significa una disminución del puntaje, porque el porcentaje otorgado está orientado a marcar el resultado de avance.

1. (25%) Copie y estudie los siguientes 3 programas PHP. Replique la salida de la forma que se indica.

A. ProgramaBase icc343clases.php

```
1  <?php
2  include_once "class.Ambiente.php";
3  $amb = new Ambiente(1000,800);
4  $amb->generaEntesAlAzar(50,"#33AA44");
5  $amb->generaEntesAlAzar(1,"#AA3344");
6  echo $amb->vistaSVG();
7  ?>
```

Este es el programa principal y usa una clase llamada Ambiente. En este “ambiente” “viven” “entes”. Usaremos una clase para representar el ambiente y otra para representar los entes.

La forma de acceder a un elemento del objeto es con el operador “->” (es como el . de Java).

Conceptos contemporáneos de documentación de código invitan a que la mejor documentación del código son buenos nombres de objetos, variables y clases. En este caso, el programa principal genera un ambiente gráfico de 1000x800. En él, habrá 50 entes de un color y 1 de otro color. El método vistaSVG realiza un “render” gráfico de la salida. Básicamente se ofrece una estructura básica de animación de entes, que con poco trabajo adicional, podría configurarse como la mostrada por el [Washington Post para el caso de contagios COVID-19](#) y que podría

usarse para revisión de los efectos de vacunaciones parciales (sólo grupos inmóviles o sólo de vectores).

B. Clase Ambiente (Archivo class.Ambiente.php)

```
1  <?php
2  include_once ('class.Ente.php');
3  class Ambiente {
4      private $ancho;
5      private $alto;
6      private $entes;
7      private $radio;
8
9      function __construct($ancho_x,$alto_y) {
10         $this->ancho = $ancho_x;
11         $this->alto = $alto_y;
12         $this->radio = 7;
13         $this->entes = [];
14     }
15
16     function generaEnte($color) {
17         $delx = rand(0,10) -5 ;
18         $dely = rand(0,10) -5 ;
19         $cel = new Ente(rand(0,$this->ancho),rand(0,$this->alto),$delx,$dely);
20         $cel->fijaColor($color);
21         array_push($this->entes,$cel);
22     }
23
24     function generaEntesAlAzar($cantidad,$color) {
25         for($j=0 ; $j<$cantidad ; $j++) {
26             $this->generaEnte($color);
27         }
28     }
29
30     function vistaSVG() {
31         $ret ='<svg width="'. $this->ancho. '" height="'. $this->alto. '">'. "\n";
32         foreach($this->entes as $ente) {
33             $ret .= $ente->svg(). "\n";
34         }
35         $ret .= '</svg>';
36         return $ret;
37     }
38 }
39 ?>
```

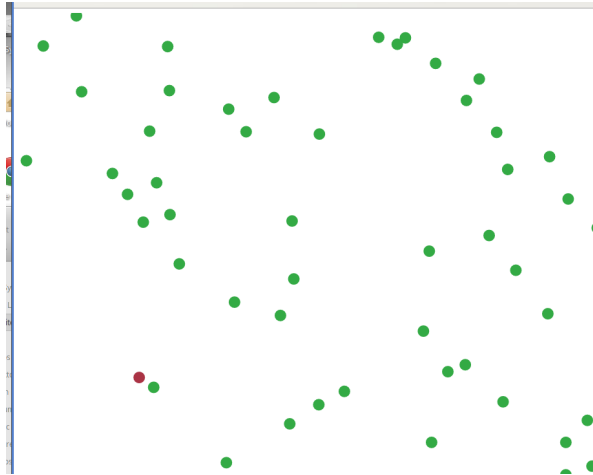
Elementos a destacar es que el constructor en PHP no lleva el nombre de la clase sino que lleva explícitamente el nombre `__construct`. Como PHP es un lenguaje débilmente tipado (flexible en los tipos de variables) entonces no se hace necesario declarar los tipos de los atributos o propiedades de los objetos (línea 4 a 7). Lo que hace la clase simplemente es crear entes (objetos en el arreglo `$entes`) y además es capaz de generar un despliegue SVG de dicho ambiente (método `vistaSVG`).

C. Clase Ente (archivo class.Ente.php)

```
1  <?php
2  class Ente {
3      private $x;
4      private $y;
5      private $deltax;
6      private $deltay;
7      private $color;
8      private $radio;
9
10     function __construct($pos_x,$pos_y,$velocidad_x,$velocidad_y) {
11         $this->x = $pos_x;
12         $this->y = $pos_y;
13         $this->deltax = $velocidad_x;
14         $this->deltay = $velocidad_y;
15         $this->color = "#000000";
16         $this->radio = 10;
17     }
18
19     function fijaColor($nuevoColor) {
20         $this->color = $nuevoColor;
21     }
22
23     function fijaRadio($nuevoRadio) {
24         $this->radio = $nuevoRadio;
25     }
26
27     function svg() {
28         $ret = '<circle cx="&1" cy="&2" r="&3" stroke-width="0" fill="&4" />';
29         $ret = str_replace('&1',$this->x,$ret);
30         $ret = str_replace('&2',$this->y,$ret);
31         $ret = str_replace('&3',$this->radio,$ret);
32         $ret = str_replace('&4',$this->color,$ret);
33         return $ret;
34     }
35 }
36 ?>
```

En el caso de los entes hemos agregamos además de su coordenada de ubicación en el espacio una forma que varíe su hipotética velocidad como dos variables llamadas deltax y deltay, que representan cuántos pixels se moverán en cada tiempo en cada eje.

Al ejecutar el programa principal se visualiza la siguiente salida:



Reporte su propia respuesta en el SLACK del curso etiquetando cda respuesta como se indica (recuerde incluir parte del fondo de escritorio en cada screenshot)

Tut-6-P1-A. Código de la clase Ambiente

Tut-6-P1-B. Código de la clase Ente

Tut-6-P1-C. Código del programa principal

Tut-6-P1-D. Salida en el navegador (note que como la salida es al azar esta no será exactamente la misma).

2 (15%) Haga los cambios necesarios, en el programa principal para que aparezcan 5 puntos rojos y 45 verdes.

Reporte su propia respuesta en el SLACK del curso etiquetando cda respuesta como se indica (recuerde incluir parte del fondo de escritorio en cada screenshot)

Tut-6-P2-A. Nuevo Código del programa principal

Tut-6-P2-B. Salida en el navegador

3 (30%) Ahora tendremos un nuevo programa principal, llamado icc343clases02.php. Su contenido es el siguiente:

```
1  <html>
2  <head>
3  </head>
4  <body>
5  <?php
6  include_once "class.Ambiente.php";
7  $amb = new Ambiente(1000,800);
8  $amb->generaEntesAlAzar(50,"#33AA44");
9  $amb->generaEntesAlAzar(1,"#AA3344");
10
11  for($k=0; $k<100 ; $k++) {
12      echo "\n";
13      echo '<div id="amb_'. $k. '" style="display:none">';
14      echo $amb->vistaSVG();
15      echo "\n</div>';
16      $amb->mueve();
17  }
18  ?>
19  <script>
20      var actual = 0;
21      function muestraSiguiente() {
22          var nuevo = (actual==99)?0:(actual+1); //operador condicional terciario
23          console.log(nuevo);
24          document.getElementById('amb_'+actual).style.display='none';
25          document.getElementById('amb_'+nuevo).style.display='';
26          actual = nuevo;
27          setTimeout(muestraSiguiente,300);
28      }
29      muestraSiguiente();
30  </script>
31  </body>
32  </html>
```

Básicamente se genera un conjunto de div **sin** visualización (líneas 11 a 17), pero cada uno conteniendo una vistaSVG. Las vistas cambian porque se usa el nuevo método mueve de Ambiente. Se produce un efecto de animación porque, vía javascript (función muestraSiguiente) se oculta un div y muestra el siguiente.

Note que el PHP se cierra en la línea 18. Es decir, hasta la 17 se ejecuta en el servidor, repetidas veces hasta que se termina de generar el código. A partir de la línea 19 es código que se ejecuta en el navegador. Luego puede corroborar exactamente lo que genera el servidor mirando el código generado eligiendo la opción “Ver código fuente” que le ofrece su navegador.

Las clases se modificaron de la siguiente manera.

- Se creó método mueve en Ambiente (código siguiente)

```
38
39     function mueve() {
40         foreach($this->entes as $ente) {
41             $ente->mueve(0,0,$this->ancho,$this->alto);
42         }
43     }
```

- Se creó método mueve en Ente (código siguiente)

```
36     function mueve($minx,$miny,$maxx,$maxy) {
37         $nuevo_x = $this->x + $this->deltax;
38         $nuevo_y = $this->y + $this->deltay;
39         if($nuevo_x > $minx && $nuevo_x < $maxx) {
40             $this->x = $nuevo_x;
41         }
42         else {
43             $this->deltax *= -1;
44             $this->x += $this->deltax;
45         }
46         if($nuevo_y > $miny && $nuevo_y < $maxy) {
47             $this->y = $nuevo_y;
48         }
49         else {
50             $this->deltay *= -1;
51             $this->y += $this->deltay;
52         }
53     }
```

Queda un poco largo el método mueve de Ente porque se ha programado el efecto de rebote en el borde, lo que se logra cambiando el signo del desplazamiento ya sea en x o en y, dependiendo del límite con el cual se choca. Esta función es importante porque podría simularse otras situaciones no hechas en la simulación mencionada, por ejemplo, sería posible similar que ciertos entes, pese a tener libertad de movimiento, sus movimientos están restringido a ciertos cuadrantes específicos.

Este programa funciona bien si se ve una animación en el navegador. Reporte su propia respuesta en el SLACK del curso etiquetando cda respuesta como se indica (recuerde incluir parte del fondo de escritorio en cada screenshot)

Tut-6-P3-A. Nuevo Código del programa principal

Tut-6-P3-B. Nuevo Código de la clase Ente

Tut-6-P3-C. Salida en el navegador usando 4 fotos en posiciones diferentes.

4. (30%) Haga los cambios necesarios para que:

- (a) un ente pueda ser catalogado como “contagiado” o como “sano” (nueva propiedad del ente).
- (b) resuelva en un método de la clase Ente, el cálculo de la distancia de un ente (this) a otro objeto (parámetro). Considere los centros como los puntos para calcular la distancia.
- (c) resuelva en un método (booleano), en la clase Ente, para verificar si un objeto está en colisión con otro (distancia menor a un diámetro).
- (d) resuelva en un método (contagia) si un ente contagia a otro, para lo cual el que contagia debe estar contagiado y en colisión con otro ente. El ente contagiado cambiará de color verde a rojo.
- (e) modifique la clase mueve de Ambiente, para verificar todos los contagios posibles cada vez.

El programa funcionará bien si se ve una animación donde los círculos verdes se vuelven rojos al chocar con círculos rojos. Reporte su propia respuesta en el SLACK del curso etiquetando cada respuesta como se indica (recuerde incluir parte del fondo de escritorio en cada screenshot)

Tut-6-P4-A. Nuevo Código del método mueve de la clase Ambiente.

Tut-6-P4-B. Nuevo Código de la clase Ente (debe aparecer la propiedad y los tres métodos nuevos).

Tut-6-P4-C. Salida en el navegador usando 4 fotos en posiciones diferentes.