



Benemérita Universidad Autónoma de Puebla.

Tarea #8.



GRAFICACIÓN
Pedro Vargas Arenas
21/Febrero/2019

INTRODUCCIÓN.

Siguiendo con el tema de graficación en 2D, esta práctica consiste en realizar un pequeño juego que se basa en el dibujado de cuadrados en diferentes coordenadas y un punto el cual se mueve mediante la pulsación de teclas (que en este caso son: **w**, **s**, **d**, **a**) y este no debe de entrar en los cuadrados antes mencionados.

Para esto, se obtiene la distancia entre puntos y funciones nuevas dentro del lenguaje OpenGL.

CONCEPTOS DESARROLLADOS.

Para realizar esta práctica, se usaron clases, estas fueron la clase de *punto*, *línea* y *polígono*. La clase *polígono* tiene una función para dibujar cuadrados los cuales están dentro del escenario del juego y esta función utiliza funciones de las clases *punto* y *línea*. También se usa una estructura para guardar las coordenadas del punto.

Dentro de la clase *main*, se tiene la función para dibujar el punto y el algoritmo que hace el movimiento del punto mediante la pulsación de teclas, esta se llama *keys*, dentro de la función se tiene un *switch-case* el cual compara la tecla que se presionó con la seleccionadas para la acción. Los movimientos que representan las teclas son:

Arriba -> w

Abajo -> s

Izquierda -> a

Derecha -> d

Para evitar que el punto sobrepase la arista de algún cuadrado, se calcula la distancia entre los vértices del cuadrado, recibe las coordenadas del punto y si las coordenadas del punto están dentro de ese rango, el punto se regresa 20 pixeles hacia la dirección contraria.

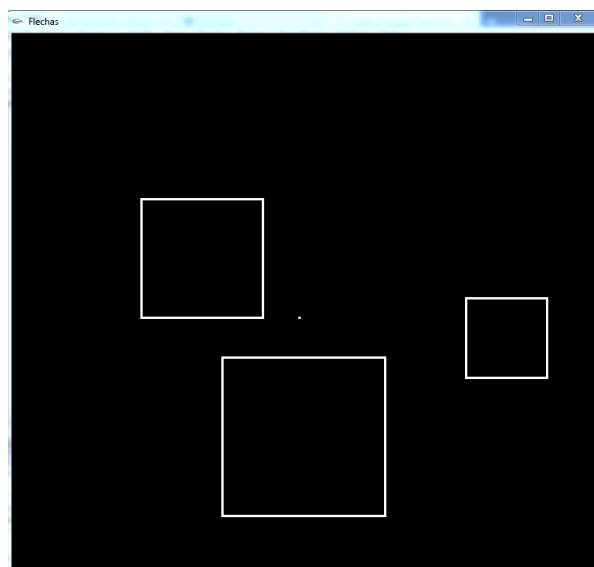
```
void c1(int x, int y){  
    if((x >= -400 && x <= -100) && y == 300){  
        est.cy += 20;  
        return;  
    }  
    if((y <= 300 && y >= 0) && x == -100){  
        est.cx += 20;  
        return;  
    }  
    if((x >= -400 && x <= -100) && y == 0){  
        est.cy -= 20;  
        return;  
    }  
    if((y <= 300 && y >= 0) && x == -400){  
        est.cx -= 20;  
        return;  
    }  
}
```

Se tiene una función como la anterior para cada cuadrado dibujado ya que cada cuadrado tiene diferentes coordenadas. Estas funciones se colocan dentro del *switch-case* que está en la función *keys*.

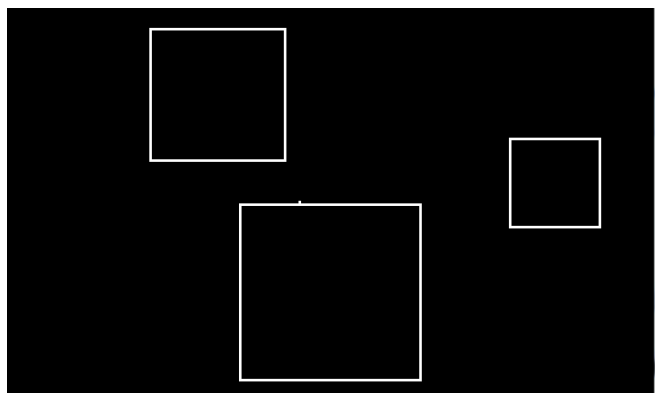
```
switch(key) {  
  //Arriba  
  case 'w':  
    c1(est.cx, est.cy);  
    c3(est.cx, est.cy);  
    c2(est.cx, est.cy);  
    est.cy++;  
    break;  
  //Abajo  
  case 's':  
    c1(est.cx, est.cy);  
    c3(est.cx, est.cy);  
    c2(est.cx, est.cy);  
    est.cy--;  
    break;  
  //Derecha  
  case 'd':  
    c1(est.cx, est.cy);  
    c3(est.cx, est.cy);  
    c2(est.cx, est.cy);  
    est.cx++;  
    break;  
}
```

PRUEBAS REALIZADAS.

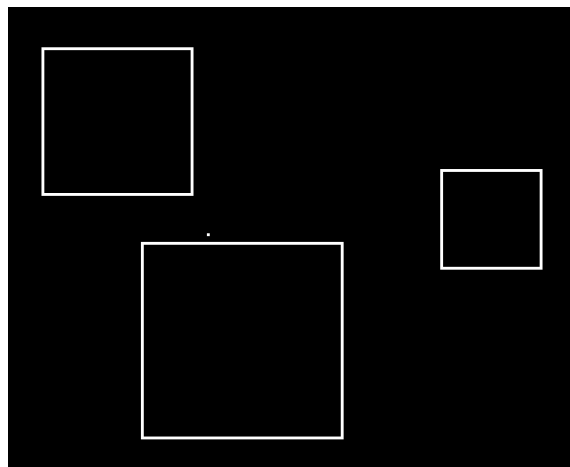
El punto se encuentra en el origen:



Cuando llega al borde el punto bota:



Aquí bota el punto:



CONCLUSIONES.

La dificultad de esta práctica aumentó ya que se tenía que manipular figuras creadas primitivamente con el teclado, pero gracias a ello, se aprendieron los nuevos conceptos que permitieron realizarla satisfactoriamente. Estos conceptos son el uso y creación de clases y objetos en c++ y el método *keys* para datos que se reciben desde el teclado.