

UNIVERSIDAD DE LAS AMERICAS

NOMBRE: Julio Aguirre

CURSO: MULTIMEDIA E INTERACTIVIDAD I

TEMA: EXAMÉN FINAL- DESARROLLO DE UN JUEGO

DATOS DEL JUEGO

JUEGO: Simon Says - SHAKE

PLATAFORMA: PROCESSING

DESCRIPCIÓN DEL JUEGO:

Basado en el juego tradicional SIMON SAYS (Simón dice) este juego tiene una variación la cual consiste en que dentro del juego encontraran cuatro esferas de diferentes colores, las cuales vibraran aleatoriamente y el usuario deberá presionar una tecla de acuerdo al color que corresponde. El juego consta de 25 niveles, mientras se vaya pasando los niveles el tiempo es más corto.

DESCRIPCIÓN DEL CODIGO:

Para desarrollar este juego, se aplico distintos temas dentro de programación como son figuras geométricas, textos, variables enteras, decimales y booleanas además condicionales, variantes de asignación de color, KeyPressed y contadores de tiempo.

VARIABLES

```
int color1R;  
int color1G;  
int color1B;
```

Variable int para colores que varien en modo RGB además, se utiliza esta variante cuando el objeto tiene añadido una variante más en este caso la vibración. Se debe repetir la misma opción para las 4 esferas.

```
int ellipse1number;  
int ellipse2number;  
int ellipse3number;  
int ellipse4number;
```

Variable ellipse#number para definir posición y vibración aleatoria de cada ellipse.

```
int currentColor = 0;  
int roundNumber = 1;  
int instructionNumber = 0;  
int countdown = 5;
```

Int:

currentColor: Actual color que se muestra en pantalla. Se lo utiliza para else if tomando el color de matriz o colorArray.

roundNumber: Variable que calcula el valor del entero de un número decimal, dentro del código se aplica para contador de nivel y el calculador de niveles else if dentro del void checkPressed.

instructionNumber: Sirve para declarar las pag. de inicios del juego, en este caso están distribuidas desde el 0 al 3.

countdown: Sirve para la cuenta regresiva y funciona conjuntamente con el timer.

```
float adjustedNumber = 2.0;  
float timer1;  
float timer2 = 0;  
float timer3;  
float timer4 = 0;
```

Float:

adjustedNumber: Da el número indicado para cada esfera además actúa en el else if y el color de la matriz.

timer: Sirve para designar un cierto lapso de tiempo para se pueda tener una respuesta o para que aparezca un mensaje en pantalla.

```
boolean playerTurn = false;  
boolean keyBeenPressed = false;  
boolean numbersMatch = false;  
boolean instructions = false;
```

Boolean:

playerTurn: Sirve como condicional para que el timer siga cada vez que se presione el botón indicado.

keyBeenPressed: Es una condición que cada vez que se presione el botón erróneo el juego se detenga.

numbersMatch: Compara los dos valores, el de los botones y el de los números.

instructions: Comienza a ejecutar las páginas de inicio.

VOID SETUP

```
void setup()
{
    size(650, 600);

    setEllipseColors();
    for (int i = 0; i < 25; i++)
    {
        colorArray[i] = getColor();
    }
}
```

Dentro del setup se declara la variable que se va establecer en todo el juego.

Se declara la colocación del color de cada elipse siempre aplicando la condición con la variable i , además se declara la matriz (en este caso de color) en base a i .

VOID DRAW

```
void draw()
{
    background(#32C6FA);
    fill(0);
    if (instructions == false)
    {
        textSize(40);
        fill(#FAFEFF);
        text("Presionar ENTER");
        if (instructionNumber == 1)
        {
            (50);
            fill(#FAFEFF);
            text("SIMON SAY-SH");
            textSize(30);
            text("PROGRESO FI");
        }
        if (instructionNumber == 2)
        {
            (50);
            fill(#FAFEFF);
            text("SIMON SAY-SH");
            textSize(30);
            text("PROGRESO FI");
        }
    }
}
```

Dentro del draw, van todas las condiciones del juego, se han creado mas condicionales if, else if .

El primer condicional que se declara es de las instrucciones, la condición dice que si esta es falsa en la pantalla aparece el texto. Como se tiene tres pantallas antes del juego la condición sigue hasta:

```
if (instructionNumber == 3)
{
    instructions = true;
}
```

```
if (instructions == true) {
    if (countdown > 0) {
        timer1 += .0167;
        countdown = 7 - (int)timer1;
        textSize(30);
        fill(#FAFEFF);
        text("COMIENZA EN " + countdown , 0, 150);
    }
}
```

Si la condición instrucciones es verdadera se empieza a ejecutar la pantalla del juego.

Condición countdown es mayor a 0 se ejecuta el timer que corresponde a la cuenta regresiva y el texto que le acompaña.

Cada vez que el countdown sea menor a 0, el usuario no podra presionar ninguna tecla, ademas cuando el timer2 llegue hasta uno y el número del adjustedNumber sea menor a 2.0 la elipse empezara vibrar, caso contrario si escoge el mismo color y el número del adjustedNumber la elipse se detendra. Esta acción se debe repetir para cada elipse.

```
if (countdown <= 0 )
{
if (playerTurn == false)
{
timer2 += .0167;

if (timer2 <= 1);
{
if (colorArray[currentColor] == 1 && timer2 <= adjustedNumber)
{
ellipse1Shake();
displayEllipse1();
}
else if (colorArray[currentColor] == 1 && timer2 > adjustedNumber)
{
ellipse1StopShake();
setEllipseColors();
}
}
```

Cuando el playerturn sea verdadero 0, el usuario podra presionar las tecla correspondiente a cada elipse con su color respectivo, y se evaluara mediante dos condicionales. El primer if evalua si el boton presionado y el número coinciden y el otro lo contrario. Segun esto pasan a un checkFalse y un checkTrue

```
if (playerTurn == true )
{
timer3 += .0167;
 textSize(30);
 fill(#0D76A5);
 text("Your Turn", 50, 150);
 textSize(11);

if (colorArray[currentColor] == playerSelect)
{
numbersMatch = true;
}
if (colorArray[currentColor] != playerSelect)
{
numbersMatch = false;
}
}

checkFalse();
checkTrue();
}
drawEllipse();
}
}
```

VOID CHECKFALSE

Dentro de void checkFalse se encuentra una condición que dice que si el numbersMatch es falso y timer3 es mayor a dos Ó numbersMatch es falso y el keyBeenPressed es verdadero, sale en pantalla “ Perdiste, reiniciar”.

```
void checkFalse()
{
  if (numbersMatch == false && timer3 >= 2 || numbersMatch == false && keyBeenPressed == true )
  {
    fill(0);
    textSize(30);
    text("PERDISTE, REINICIAR" , 0, 200);
    noLoop();
    timer3 = 0;
  }
}
```

VOID CHECKTRUE

Lo controtario es el checkTrue ya que en las dos condiciones el numbersMath como el keyBeenPressed son verdaderos en caso que se de esa condición se da ciertos criterios para que el usuario vaya pasando de nivel en este caso del uno al veinticinco.

```
void checkTrue()
{
  if (numbersMatch == true && timer3 >= 2 || numbersMatch == true && keyBeenPressed == true )
  {
    playerTurn = false;
    timer3 = 0;
    playerSelect = 0;
    numbersMatch = false;
    keyBeenPressed = false;
    roundNumber++;
    currentColor++;
    if (roundNumber >= 1 && roundNumber <= 5)
    {
      adjustedNumber = 2;
    }
  }
}
```

VOID KEYPRESSED

Dentro del keyPressed se puede encontrar la condición de que cuando se presione el número uno y sea correcto el juego seguira. Se encuentra la condición del uno al 4.

```
void keyPressed()
{
  if (key == '1' && playerTurn == true)
  {
    playerSelect = 1;
    displayEllipse1();
    keyBeenPressed = true;
  }
}
```

VOID DRAWELLIPSE

Aquí se define las coordenadas x,y además del tamaño de la esfera con una variante en la coordenada y ya que se necesita hacer esta variante para que la esfera pueda vibrar.

```
void drawEllipse()
{
    ellipse(275, 400, 250, 250);
    fill(color1R, color1G, color1B);
    ellipse(400 + ellipse1number, 400, 150, 150); //azul
    fill(color2R, color2G, color2B);
    ellipse(275+ ellipse2number, 525, 150, 150); //verde
    fill(color3R, color3G, color3B);
    ellipse(275, 300 + ellipse3number, 150, 150); //rojo
    fill(color4R, color4G, color4B);
    ellipse(150, 400 + ellipse4number, 150, 150); // amarillo
}
```

VOID ELLIPSESHAKE AND STOPSHAKE

Se define cuando la esfera empieza a vibrar y cuando tiene que detenerse. En el void ellipseshake una de las elipses debe ser igual a un valor randomico. Por otro lado en el stopshake todos los valores deben ser 0. Esto se lo debe repetir para cada caso.

```
void ellipse3Shake()
{
    ellipse1number = 0;
    ellipse2number = 0;
    ellipse3number = (int)random(-5, 5);
    ;
    ellipse4number = 0;
}
```

```
void ellipse3StopShake()
{
    ellipse1number = 0;
    ellipse2number = 0;
    ellipse3number = 0;
    ellipse4number = 0;
}
```