Alumno: Gerardo Daniel Naranjo Gallegos, A01209499.

Alumno: Roberto Figueroa Saavedra, A01209689.

Curso: Programación Orientada a Objetos (Java).

Profesor: Norma Amanda Elías S.

ITESM, campus Querétaro.

05 de mayo de 2016.

**UML proyecto final.**

# Desarrollo:

**Estructuras de datos:**

Utilizamos mucho la parte gráfica para mostrar el juego en la pantalla, mostrando al personaje y a los obstáculos a esquivar. También hicimos uso de ArrayList, para el tema de los obstáculos y la plataforma, que ayudan a registrarlas y poder visualizar las requeridas para abarcar la ventana.

**Descripción de métodos:**

Respecto a la clase principal, MiCanvas, utilizaremos un método llamado paint, que mostrará en pantalla todo lo necesario, el fondo, la puntuación, el jugador, etcl. Implementaremos el método de juego, para detener el juego por un tiempo. Haremos uso del teclado, mediante el keyListener, para implementar el método de salto; la idea es que el personaje realice un salto para esquivar obstáculos. Se llamarán a métodos de otras clases, para crear los arraylist necesarios.

En la clase llamada fondo, utilizaremos un método que permita mostrar la imagen de fondo del juego.

En la clase monito se llevará a cabo un método que muestre al personaje.

Dentro de la clase puntuación, se creará un método que permita registrar la puntuación actual; se mostrará en pantalla durante el juego en la parte superior.

La clase obstáculo y plataforma funcionan similarmente, imprimiendo rectángulos de colores que funcionan como el suelo y los obstáculos a esquivar. Se implementan métodos para desplazarlos en el eje X.

Las clases plataformas y obstáculos son ArrayList de plataforma y obstáculo, para crear varios de ellos a lo largo del juego. Se incluyen métodos para ir agregando más.

En la clase ventana, se inicializa.

Para finalizar, la clase main, en donde encontramos lo necesario para inicializar la ventana.

Diagrama UML:

MainR

+miVent MiVentana

-public static void main

Fondo

+x int

+y int

+tamanio int

+imagen BufferedImage

-cargar() //carga la imagen de fondo

-paintComponent(g: Graphics) //imprime la imagen de fondo

MiVentana extends JFrame

+ Canvas MiCanvas

//Solo el constructor

Monito

+x int

+y int

+tamanio int

+imagen BufferedImage

//Setters y getters

-cargar() //carga la imagen de fondo

-paintComponent(g: Graphics) //imprime la imagen de fondo

-saltar() //desplaza la imagen en Y

-bajar() //desplaza la imagen en Y

Obstaculo

+x int

+y int

+tamX int

+tamY int

//Setters y getters

-avanza() //desplaza el objeto en X

-paint(g: Graphics) //imprime la imagen de fondo

Obstaculos

+obts ArrayList<Obstaculo>

-agregar(obt: Obstaculo) //añade un Obstaculo al arreglo

-paint(g: Graphics) //pinta el arreglo

-avanza() //mueve el arreglo

-tam: int //regresa el tamaño del objeto

//-colision() tratamos de hacer un método para detectar el choque entre el monito y un obstáculo, pero no logramos hacer que funcionara adecuadamente, por ello, se quedó comentada

Plataforma

+x int

+y int

+tamX int

+tamY int

+color Color

+r Random

//Setters y getters

-paint(g: Graphics)

-avanza()

Plataformas

+pla ArrayList<Plataforma>

-agregar(pla: Plataforma) //añade un objeto Plataforma al arreglo

-paint(g: Graphics) //pinta el arreglo

-avanza() //mueve el arreglo

-size: int //regresa el tamaño del objeto

Puntuacion

+puntaje int

//setter y getter

-score() //puntaje=puntaje+1

Puntuaciones

+punt int[]

-agrega(a: int) //agrega una puntuación al topten

-topTen() //organiza las mejores puntuaciones (10)

MiCanvas extends Canvas implements ActionListener, KeyListener

+pla Plataforma

+miPlataforma Plataformas

+obs Obstaculo

+obstáculos Obstaculos

+mon Monito

+fondo Fondo

+punt Puntuacion

+reloj Timer

+r Random

+aux boolean

+pausa boolean

+menú boolean

+i int

-paint() //Se comentó el fondo, porque parpadea mucho. Para ver, quitar comentario.

-actionPerformed(ae: ActionEvent)

-keyPressed(e: KeyEvent)

-keyReleased(e:KeyEvent)

-salto() //método para que salte el monito

-jugar() //comprueba si está o no en pausa

Se incluye también un paquete de imágenes.

Se creó la imagen “monito” es un gif, pero desconocemos el porqué no realiza ningún movimiento.

La imagen “back” que es el fondo, fue descargada de internet.