

## LABORATORIO FINAL

### FUNCIONES, SUBROUTINAS Y RECURSIVIDAD

**Materia:** Algoritmia y programación II

**Profesora:** Rocio Ramos Rodríguez

**Monitor:** William García y Juan Solano

**Curso:** IST2089

#### OBJETIVO.

Aplicar conceptos de funciones, subrutinas y recursividad utilizando los arreglos unidimensionales y bidimensionales.

#### ENUNCIADO

##### Cross And Trap

El personaje animado comienza en un risco (columna o plataforma), la misión del jugador es construir un puente que le ayude a nuestro personaje a cruzar al otro lado, donde también encontrará un risco (columna o plataforma). El puente se debe construir verticalmente; manteniendo la barra espaciadora presionada el jugador logrará que el puente crezca hasta que la suelte y sea liberado el puente, el cual pasará a orientarse horizontalmente de tal forma que éste debe cuadrar o encajar perfectamente en el espacio entre los dos riscos, para que el personaje pueda caminar sobre el puente creado. Si el puente no es lo suficientemente largo para superar el risco el personaje caerá y por consiguiente perderá, si su largo supera el vacío del risco también caerá y el juego terminará y tendrá la opción de volver a jugar.

Si el jugador logra cruzar exitosamente, debe enfrentarse ahora a un nuevo reto el cual consiste en controlar al personaje que solo puede moverse de dos formas: **Subiendo o Bajando**. La misión ahora de nuestro personaje principal es atrapar todas las monedas que aparezcan en su camino, tener en cuenta que las monedas solo pueden aparecer sobre el personaje a una altura generada aleatoriamente, sin chocarse con el borde superior o con algún enemigo. Los enemigos, al igual que las monedas, aparecen en **alturas aleatorias posibles** para la captura del personaje, con la diferencia que el radio del cuerpo de los enemigos también es generado aleatoriamente. Si el jugador colisiona con los límites del mundo o con algún enemigo, termina el juego y debe tener la opción de iniciar nuevamente.

En cada cruce con recogida exitosa, se deberá mostrar el número de monedas recogidas por el personaje, el número de riscos cruzados exitosamente y el tiempo que el usuario lleva jugando.

El programa deberá mostrar el logro obtenido y además el jugador volverá a la posición inicial en el risco para continuar jugando.

## INSTRUCCIONES Y CONDICIONES.

Se debe tener en cuenta las siguientes condiciones, instrucciones y restricciones:

- Solo las dos plataformas iniciales se deben generar, las demás plataformas no deben ser generadas al inicio del juego sino cuando el personaje ya ha agarrado las monedas y está a salvo en la plataforma.
- Debe ser el personaje más no el fondo quien se mueva.
- El personaje debe subir estilo Flappy Birds.
- Todas las colisiones generan Game Over.
- Los enemigos pueden tomar las monedas, si esto sucede la moneda tomada deberá aparecer en otro lugar aleatorio (solo eje y).
- La moneda aparecerá en un lugar aleatorio sobre el personaje, en el ancho de la plataforma y desde la altura del personaje, hasta el límite superior de la pantalla.
- Si el personaje toca la parte superior de la pantalla el usuario perderá.
- Las monedas solo aparecerán, cuando el usuario este en la próxima plataforma.
- El número de monedas en una plataforma es variable mínimo serán 3 monedas y máximo 5.
- Para pasar a la siguiente plataforma será necesario tomar todas las monedas.
- Las monedas deben ser agarradas todas antes de volver a la plataforma, es decir el personaje debe quedar en el espacio, aire o volando hasta agarrar las monedas generadas y esquivando los enemigos.
- El juego debe tener la opción de reiniciar.
- Al tomar las monedas los monstruos no desaparecen, solo desaparecen cuando el usuario regresa a la plataforma.
- La creación del puente deberá tener una velocidad aleatoria, por cada plataforma.
- Los monstruos tendrán una velocidad aleatoria, además de posición aleatoria en eje x y el eje y.
- Solo al inicio del juego se crean dos plataformas y cada vez que alcance las monedas y llegue sano y salvo a la plataforma, debe crearse la nueva para seguir su camino. Si se pierde se debe reiniciar completamente el juego.
- El juego puede ser realizado en Netbeans o en Processing
- Utilizar las herramientas vistas en clase, se pueden consultar e investigar otras y antes de utilizarlas solicitar autorización al respecto por medio de un correo externo desde el catálogo.
- La GUI debe ser muy creativa y llamativa para el usuario, no debe estar sobrecargada en colores ni saturada de elementos gráficos.
- El personaje no debe ser copia ni existir de ningún juego ya creado.
- Los enemigos pueden tener formas diferentes y deben ser creados o diseñados por los autores del juego.
- El diseño del mundo debe ser ingenioso y creativo.
- El juego debe ser amigable con el usuario y tener una interfaz agradable.
- Realizar en parejas.
- No seguir instrucciones genera penalizaciones en la nota.
- Las entregas equivalen al 30%, la sustentación al 40% y el programa al 30% del laboratorio.

- El programa debe quedar guardado con el nombre LabFinal\_nombre#1\_nombre#2, debe ser enviado como adjunto el proyecto comprimido (un solo integrante) al link correspondiente en el catálogo y dejarlo guardado en sus respectivas carpetas (ambos integrantes).
- Cualquier intento de copia se penaliza con 0,0 y anotación a hoja de vida.

## **ENTREGAS.**

**Entrega #1:** grupo IST208902 jueves 19 de octubre y grupo IST208903 viernes 20 de octubre de 2017. En esta entrega mínimo se debe mostrar el diseño del mundo y de los personajes del juego, además de la creación del puente.

**Entrega #2:** grupo IST208902 jueves 26 de octubre y grupo IST208903 viernes 27 de octubre de 2017. En esta entrega mínimo se debe mostrar el cruce del personaje por el puente, y la recogida de las monedas.

**Entrega Final y sustentación:** grupo IST208902, jueves 2 de noviembre y grupo IST208903, viernes 3 de noviembre de 2017.

**El éxito se adquiere luchando por lo que se quiere.... Tienes todo para ser mejor que los demás... esfuérzate y cree en ti, verás los resultados que obtendrás.**