

CONTEXTO

Prototipo del famoso videojuego Crossy Road donde toda la información del estado inicial del juego será cargada a partir de un archivo txt. En el juego deben haber 30 vehículos que se van a distribuir en 6 calles, donde, en cada calle, los vehículos se moverán en diferentes sentidos y diferentes velocidades. Si el jugador cruza todas las calles con éxito, aparecerá una excepción propia donde le indica que ganó y la posibilidad de reiniciar el juego. Si el jugador es tocado por un vehículo, deberá lanzarse una excepción donde le indique que perdió y la posibilidad de reiniciar el juego. El juego debe contar con un cronómetro que contabilice el tiempo de juego del jugador, además el desarrollo debe permitir ordenar naturalmente por fecha y hora y parcialmente por duración de la partida. Se debe tener en cuenta que el movimiento de los autos debe hacerse con un hilo distinto al hilo principal de la ejecución del programa. Para finalizar, el prototipo en sí cuenta con hasta 3 pantallas: la primera donde se muestra el juego, la segunda donde indica si el jugador pierde o gana y la última donde se presenta el ordenamiento de los datos guardados en tiempo de ejecución.

REQUERIMIENTOS FUNCIONALES

RF1

Descripción	El programa debe cargar y leer la información inicial desde un archivo txt
Entradas	-
Salidas	-
Precondición	El archivo txt debe pertenecer al sistema
Postcondición	

RF2

Descripción	El programa debe cargar los enemigos con respecto a la información leída desde el archivo txt
Entradas	-
Salidas	-
Precondición	el archivo txt debe ser leído por el sistema
Postcondición	

RF3

Descripción	En el programa deben haber 6 calles por donde se van a distribuir 30 enemigos
Entradas	-
Salidas	Lienzo con las calles dibujadas
Precondición	el archivo txt debe ser leído por el sistema y dibujar los enemigos
Postcondición	Los enemigos deben ser distribuidos por las calles

RF4

Descripción	El programa debe pintar las calles sobre la pantalla 1 donde se presenta el juego
Entradas	-
Salidas	Lienzo con las calles dibujadas
Precondición	El sistema debe haber creado caller
Postcondición	-

RF5

Descripción	EL programa debe dibujar los enemigos en las calles
Entradas	-
Salidas	Enemigos dibujados y distribuido por las diferentes calles del juego
Precondición	Deben haberse creado y pintado las calles
Postcondición	

RF6

Descripción	Los enemigos deben movilizarse en distintas direcciones y velocidad
Entradas	-
Salidas	Enemigos moviéndose por cada calle
Precondición	Deben haberse creado y pintado los enemigos
Postcondición	

RF7

Descripción	El sistema debe crear un personaje a razón de un txt
Entradas	-
Salidas	-
Precondición	Leer el archivo txt
Postcondición	

RF8

Descripción	El juego debe dibujar al jugador
Entradas	-
Salidas	Jugador dibujado en la pantalla
Precondición	En el sistema debe haberse creado el personaje
Postcondición	

RF9

Descripción	El jugador puede moverse hacia arriba, abajo, derecha y izquierda presionando las teclas del teclado
Entradas	Flechas del teclado "W" "S" "D" "A"
Salidas	Jugador dibujado en la pantalla
Precondición	En el sistema debe haberse creado el personaje y en el lienzo debe estar dibujado el jugador
Postcondición	

RF10

Descripción	El programa debe registrar el tiempo de la partida y ser mostrado en el lienzo
Entradas	-
Salidas	-
Precondición	El programa debe ser ejecutado de manera adecuada
Postcondición	El tiempo de partida se registra cuando el juego termine

RF11

Descripción	El programa debe registrar la fecha y la hora de la partida
Entradas	-
Salidas	-
Precondición	El programa debe ser ejecutado de manera adecuada
Postcondición	la fecha y hora se registra cuando el juego termine

RF12

Descripción	La partida termina cuando el jugador toca a un enemigo
Entradas	Jugado toca a los enemigos
Salidas	Juego terminado
Precondición	El programa debe ser ejecutado de manera adecuada. Toque entre auto y jugador
Postcondición	-

RF13

Descripción	La partida termina cuando el jugador logra llegar al otro lado del lienzo
Entradas	pasar todas las calles sin tocar a un enemigo
Salidas	Juego terminado
Precondición	El programa debe ser ejecutado de manera adecuada.
Postcondición	-

RF14

Descripción	Cuando pierde, el juego debe lanzar una excepción que le indica al jugador que perdió.
Entradas	Jugado toca a los enemigos
Salidas	Juego terminado, excepción
Precondición	El programa debe ser ejecutado de manera adecuada. Toque entre auto y jugador
Postcondición	-

RF15

Descripción	Cuando gane, el juego debe lanzar una excepción que le indique al jugador que ganó.
Entradas	pasar todas las calles sin tocar a un enemigo
Salidas	Juego terminado
Precondición	El programa debe ser ejecutado de manera adecuada.
Postcondición	-

RF16

Descripción	El juego debe reiniciarse al terminar la partida
Entradas	Presión sobre la tecla "R"
Salidas	Juego desde 0
Precondición	El programa debe ser ejecutado de manera adecuada. Estar en alguna pantalla de perder o ganar.
Postcondición	-

REQUERIMIENTOS NO FUNCIONALES:

- Interfaz gráfica clara y acorde con la temática presentada en el juego que le permita al usuario tener una navegabilidad fluida y sin complicaciones.
- El sistema debe tener disponibilidad el 100% de las veces en que un usuario haga ingreso.
- Se realizará con el patrón de diseño MVC (Model, View, Controller).
- El tiempo para dar inicio al sistema no debe tardar más de 1 minuto.
- Se debe hacer uso de hilos.
- Se debe implementar una interfaz propia.
- Se debe manejar excepciones.