

Análisis y diseño de software

Contexto

Crear un prototipo del juego Crossy Road, del cual toda la información del estado inicial del juego, como el número de carros, su posición y dirección será cargada a partir de un archivo .txt. Además, el juego deberá implementar el guardado de las estadísticas de partida como la fecha, hora y tiempo de juego. Deberá guardar dos archivos .txt uno organizando estas estadísticas de forma natural por fecha y hora y parcialmente por duración de partida. Deberá implementar hilos para el movimiento de los carros. El juego debe lanzar una excepción propia que indique si el usuario perdió o ganó el juego indicándole como reiniciar.

Requerimientos funcionales

RF1	
Descripción	El programa debe cargar la información de los objetos carro y jugador a partir de un archivo .txt
Entradas	-.txt
Salidas	-el archivo .txt ha sido cargado con éxito en una lista de strings
Precondición	-que el txt exista, txt contiene información para crear objeto carro y jugador
Postcondición	-

RF2	
Descripción	El programa debe crear 30 vehículos que se distribuyan en las 6 calles de Crossy Road.
Entradas	-.txt
Salidas	-30 carro han sido creados
Precondición	-que el txt exista, txt contiene información para crear objeto carro
Postcondición	-

RF3	
Descripción	El programa debe crear un hilo diferente al hilo principal del programa para el movimiento de los carros
Entradas	- ArrayList carros-> dirección del carro
Salidas	- el hilo para mover los carros está movimiento los carros
Precondición	- ArrayList carros debe existir
Postcondición	-

RF4	
Descripción	El programa debe pintar en el lienzo los carros
Entradas	- ArrayList carros -> posX, posY, color
Salidas	- los carros pintados
Precondición	- ArrayList carros debe existir
Postcondición	-

RF5	
Descripción	El programa debe pintar las calles donde circularán los carros
Entradas	- posX, posY, color
Salidas	- las calles pintadas
Precondición	-
Postcondición	-

RF6	
Descripción	El programa debe cuadrar la posición de los carros si su posición en el txt no coincide con el tamaño de la pantalla
Entradas	- posX, posY de todos los carros del ArrayList carros
Salidas	- los carros están en una posición que coincide con el tamaño de la pantalla
Precondición	-
Postcondición	-

RF7	
Descripción	El programa debe poder tener la opción de reiniciarse al ganar o perder la partida
Entradas	- posicionYJugador, conocer si mi jugador se chocó con un carro
Salidas	- el juego se reinició
Precondición	- jugador debe haberse creado, saber si mi jugador se chocó con un carro
Postcondición	- presionar la tecla de reinicio

RF8	
Descripción	El programa debe poder detectar si mi jugador se chocó con un carro
Entradas	- posición del jugador, posición de todos los carros.
Salidas	- boolean de si el jugador se chocó con un carro
Precondición	- carros y jugador deben haberse creado
Postcondición	-

RF9	
Descripción	El programa debe guardar un archivo .txt que contenga tiempo de juego, fecha y hora tras finalizar la partida
Entradas	- tiempo juego, fecha del juego y la hora en el que fue jugado
Salidas	- se ha guardado un archivo .txt con la información de la partida
Precondición	- otorgar valores a tiempo de juego, fecha del juego y hora
Postcondición	-

RF10	
Descripción	El programa debe mostrar el tiempo de juego en el lienzo
Entradas	- tiempo juego
Salidas	- se ha mostrado el tiempo de juego en el lienzo
Precondición	- tiempo de juego tiene un valor
Postcondición	-

RF11	
------	--

Descripción	El programa debe poder detectar si el jugador ganó el juego
Entradas	- posicionYJugador
Salidas	- boolean que dice si el jugador ganó el juego
Precondición	- jugador debe existir
Postcondición	-

RF12	
Descripción	El programa debe poder detectar si el jugador perdió el juego
Entradas	- posición x y y del jugador, la posición x y y de todos los carros
Salidas	- boolean que dice si el jugador perdió el juego
Precondición	- carros y jugador deben haberse creado
Postcondición	-

RF13	
Descripción	El programa debe lanzar un excepción propia cuando el jugador perdió el juego
Entradas	- boolean que me dice si el jugador perdió el juego
Salidas	- se lanzó la excepción propia
Precondición	- RF12 funcione correctamente
Postcondición	-

RF14	
Descripción	El programa debe ordenar naturalmente por fecha y hora los datos dentro del txt que se guarda al finalizar la partida
Entradas	- .txt que se guarda tras finalizar la partida
Salidas	- los datos dentro del txt han sido organizados naturalmente.
Precondición	- haber jugado al menos 1 partida
Postcondición	-

RF14	
Descripción	El programa debe ordenar parcialmente por duración de partida los datos dentro del txt que se guarda al finalizar la partida
Entradas	- .txt que se guarda tras finalizar las partida
Salidas	- los datos dentro del txt han sido organizados parcialmente.
Precondición	- haber jugado al menos 1 partida
Postcondición	-

RF15	
Descripción	El programa debe implementar una excepción del sistema al leer el archivo .txt con la información de los carros y el jugador.
Entradas	- .txt con la información de los carros y el jugador.
Salidas	- se ha detectado la excepciones del sistema al leer y cargar un archivo txt.
Precondición	-
Postcondición	-

RF15	
Descripción	El programa debe poder mostrar en el lienzo los datos guardados de las partidas ordenadamente
Entradas	- datos ordenados según lo requiera
Salidas	- se han mostrado los datos ordenados en la pantalla
Precondición	-
Postcondición	-

RF16	
Descripción	El programa debe poder pasar entre tres pantallas, una para el juego, otra para el decir si se ganó o se perdió y otra para mostrar el ordenamiento de los datos guardados
Entradas	- variable pantalla, boolean si ganó o perdió, una tecla para pasar a la pantalla de ordenamiento de los datos guardados.
Salidas	- se ha pasado de pantalla
Precondición	-
Postcondición	-

Requerimientos no funcionales

RNF1	
Descripción	El programa debe recrear el video juego CrossyRoad

RNF2	
Descripción	El usuario debe poder diferenciar visualmente entre él y los carros con los que no se debe chocar

RNF3	
Descripción	Las carreteras deben estar abajo de los carros

RNF4	
Descripción	El programa debe estar programado en JAVA