

Crossy Road - Análisis

Quiz 2

Contexto

Se desea realizar un prototipo del juego Crossy Road. Se deben crear los primeros objetos a partir de un archivo .txt con la información de cada uno. Al menos otros 30 vehículos distribuidos en 6 calles deben ser creados. La partida finaliza si el personaje logra llegar al otro lado de la calle o si uno de los vehículos impacta con él. Debe haber un indicador en la pantalla que vaya indicando el tiempo. El juego debe guardar la fecha, la hora y el tiempo para finalizar la partida. Hay una tercera pantalla en la que se pueden organizar las partidas, naturalmente por fecha y hora, y parcialmente por la duración de la partida.

Requerimientos

Requerimientos funcionales

RF1

Descripción	El programa debe poder cargar la información del archivo .txt en una lista.
Entradas	Archivo .txt y arreglo de texto.
Salidas	
Precondición	Que se haya iniciado el programa
Postcondición	Cadena de texto cargada.

RF2

Descripción	El programa debe poder crear los objetos con la información obtenida del archivo .txt
Entradas	Cadena de texto.
Salidas	
Precondición	Que se haya iniciado el programa y la información del .txt haya sido cargada en la lista.
Postcondición	Vehículos y personaje creados

RF3

Descripción	<i>El programa debe agregar a los vehículos a una lista.</i>
Entradas	Vehículo.
Salidas	
Precondición	Que se haya iniciado el programa y se hayan creado los vehículos.
Postcondición	Vehículo agregado a la lista de vehículos.

RF4

Descripción	<i>Cada vehículo deberá moverse por la pantalla a través de la ejecución de un hilo.</i>
Entradas	Posiciones de los vehículos.
Salidas	
Precondición	Que se haya iniciado el programa y la lista de vehículos no esté vacía.
Postcondición	Vehículos moviéndose.

RF5

Descripción	<i>El programa debe crear un indicador que mida la duración de la partida.</i>
Entradas	
Salidas	Duración de la partida.
Precondición	Que se haya iniciado el programa.
Postcondición	

RF6

Descripción	<i>El programa debe visualizar el indicador de la duración de la partida en pantalla.</i>
Entradas	Indicador.
Salidas	
Precondición	Que se haya iniciado el programa.
Postcondición	Indicador visualizado en la interfaz del programa.

RF7

Descripción	<i>El programa debe lanzar una excepción propia que diga "Perdiste" si un vehículo impacta con el personaje</i>
Entradas	Posiciones de los vehículos y el personaje.
Salidas	Excepción
Precondición	Que se haya iniciado el programa.
Postcondición	Excepción lanzada, juego finalizado.

RF8

Descripción	<i>El programa debe implementar una excepción del sistema al leer el archivo .txt con la información inicial.</i>
Entradas	Archivo .txt
Salidas	Excepción
Precondición	Que se haya iniciado el programa.
Postcondición	

RF9

Descripción	<i>El programa debe guardar la fecha y la hora la partida</i>
Entradas	
Salidas	Fecha y hora de la partida.
Precondición	Que se haya iniciado el programa.
Postcondición	

RF10

Descripción	<i>El programa debe guardar cada partida a una lista de partidas</i>
Entradas	Partida y lista de partidas
Salidas	
Precondición	Que se haya iniciado el programa.
Postcondición	Partida guardada en lista de partidas.

RF11

Descripción	<i>El programa debe poder organizar las partidas naturalmente por fecha-hora.</i>
Entradas	Lista de partidas.
Salidas	Lista de partidas organizada por fecha-hora.
Precondición	Que se haya iniciado el programa y la lista de partidas no esté vacía.
Postcondición	Lista organizada según criterio en la interfaz.

RF12

Descripción	<i>El programa debe poder organizar las partidas parcialmente por duración de la partida</i>
Entradas	Lista de partidas.
Salidas	Lista de partidas organizada por duración.
Precondición	Que se haya iniciado el programa y la lista de partidas no esté vacía.
Postcondición	Lista organizada según criterio en la interfaz.

RF13

Descripción	<i>El programa debe poder reiniciar el juego.</i>
Entradas	
Salidas	
Precondición	Que se haya iniciado el programa y el juego haya finalizado.
Postcondición	Juego reiniciado.

Requerimientos no funcionales

- El diseño del prototipo será en el lenguaje de programación Java.
- Se realizará con el patrón de diseño MVC (Model, View, Controller).
- Deberá hacer uso de herencia.
- Deberá hacer uso de hilos.
- Deberá implementar excepciones.
- Los elementos del prototipo deberán ser hechos con figuras geométricas.