|  |
| --- |
| **Práctica 3** |
| Uso de Interfaces Gráficas en el Problema del Ciclista |
|  |
| **Juan Carlos Marco, Juan Luis Pérez, Emilio Álvarez Piñeiro.** |
| **18/04/2013** |
|  |
|  |
|  |

Contenido

[Preliminares 3](#__RefHeading__250_860113062)

[Requisitos: 3](#__RefHeading__334_1925510426)

[Objetivos 3](#__RefHeading__336_1925510426)

[Introducción 5](#__RefHeading__338_1925510426)

[Estado del Arte 5](#__RefHeading__340_1925510426)

[Fundamentación teórica: 5](#__RefHeading__342_1925510426)

[Propuestas de Solución (Solución implementada) 5](#__RefHeading__344_1925510426)

[Discusión 5](#__RefHeading__346_1925510426)

[Conclusión 5](#__RefHeading__348_1925510426)

[Líneas futuras 6](#__RefHeading__350_1925510426)

[Referencias bibliográficas utilizadas 7](#__RefHeading__352_1925510426)

[Anexos 7](#__RefHeading__354_1925510426)

* + Preliminares

**Título:** Uso de Interfaces Gráficas en el Problema del Ciclista

**Fecha de entrega:** 18 de Abril de 2013.

* + Requisitos:
* Conocer la teoría de los temas 1 a 10 con especial interés en el uso de los interfaces gráﬁcos de usuario en Java.
* Tener claros los conceptos vistos en prácticas anteriores.
* Tener claros los conceptos para el uso de teclado y ﬁcheros en java y su problemática de uso con swing.
* Profundizar en los conceptos sobre polimorﬁsmo, genéricos, etc.
* Conocimientos básicos de ciclismo.(Se aportan por la cátedra)
  + Objetivos

**General**:

* Aﬁanzar los conocimientos adquiridos de Metodología de la Programación y añadir los recibidos sobre el uso de ﬁcheros, interacción con usuarios, etc...
* Incluir lo aprendido sobre genéricos a la práctica 1 como siguiente nivel de desarrollo.

**Particulares**:

* Estudiar la interacción con usuarios mediante teclado y pantalla.
* Estudiar el uso de ﬁcheros, su comportamiento y compartición.
* Aplicar los conocimientos adquiridos sobre genéricos (enumerados, mapas, etc…).
* Ampliar el número de actores y objetivos de **cambios dinámicos** en el programa para orientar la práctica hacia el siguiente nivel, que se corresponderá con la práctica 3.
* Aﬁanzar conceptos de Física básica, aplicados a la bicicleta, claramente enmohecidos.
* Utilizar Eclipse y JUnit.

Resumen:

A una aplicación, basada en el desarrollo realizado en la Práctica Inicial Segunda, se han añadido una salida con un interfaz gráﬁco con una base de usabilidad. Por otra parte, este interfaz ha ampliado la entrada "sin interrupción" de órdenes o comandos desde entrada/salida estándar , ﬁchero(s)y la posibilidad de la entrada "inteligente" con la entrada con swing . Se presenta una salida gráﬁca que simula un ciclista en una carretera con sus pendientes y vientos.

Se ha hecho una validación y veriﬁcación exhaustiva.

Abstract

With the recent modifications several cyclist can “run” at the same time and share de track and weather conditions in the application.

External factors have been added, such as the slope of the track and the direction of the wind and their influence over the bikes´ acceleration and speed. These factors are modified during the execution of the application and, as such, it is reflected on the behavior of the cyclist depending on the time lapsed and the distance traveled.

Also, a Parser class has been added which is able to read predetermined commands from files or console and make them execute on the specified “object” ( the receptor of the command) indicated by the user by console or file.

Observaciones:Ninguna reseñable.

Sugerencias:Creemos que sería positivo ahondar más en la fase de especificación previa a la práctica para dictaminar sin lugar a errores cuáles son las expectativas que se tienen respecto a la misma y para reducir al máximo las posibles, y por otro lado inevitables, interpretaciones erróneas o insuficientes de lo exigido por parte del profesorado.

**Introducción**

Todos los lenguajes de programación actuales tienen algún componente gráﬁco que permite al usuario interactuar con los sistemas: Java también.

Java tiene para usarse como interfaz gráﬁco de usuario dos librerías AWT y SWING.

Además, de salida por ventanas y el uso de otros componentes gráﬁcos, se ha usado como entrada de comandos e información.

1. Estado del Arte

Definiciones:

* Enumerados: Un enumerado es un tipo especial de datos que permite comparar una variable con un conjunto determinado de constantes. La variable debe ser igual a uno de los valores que han sido predefinidos (más información en la bibliografía).
* Listas: Una colección o secuencia ordenada de elementos. El usuario puede acceder a los elementos de la lista a través de un índice de tipo entero y buscar elementos en la lista (más información en la bibliografía).
* Parser: O en castellano, analizador sintáctico, convierte el texto de entrada en otras estructuras que son útiles para su posterior uso.

1. Fundamentación teórica:

Sobre el uso del Parser:

Las diferentes opciones para la identificación del objeto sobre el que se va a ejecutar un comando determinado es uno de los principales problemas de la práctica. Las soluciones posibles que ha tenido en cuenta el grupo han sido, o pasarle al Parser una lista con referencias a los objetos en ejecución desde una instancia superior, o hacer que, de alguna manera diferente, el Parser pueda determinar cuál es el objeto sobre el que se va a ejecutar el comando y poder llamar así al método indicado.

Sobre los Factores externos:

La principal dificultad de la programación de los factores externos radicaba en cómo convertir una entrada de una cadena desde fichero en una entidad que fuese capaz de modificar la velocidad de los corredores. La solución propuesta se basa en el uso de enumerados para la identificación de los factores desde fichero.

1. Propuestas de Solución (Solución implementada)

Sobre el uso del Parser:

Dado que la mayoría de las soluciones encontradas para la vinculación de comandos con los objetos sobre los que ejecutarse se basaban en el uso de Threads, y dado que estos no habían sido impartidos todavía, se ha optado por pasarle por parámetro la lista de objetos “ejecutables” al Parser como solución **provisional**. Para las sucesivas prácticas se está estudiando la implantación de una solución mediante el uso de **Dispatchers** de Akka (para más información consular la bibliografía), que permitirían la redirección al objeto deseado sin el paso de listas por parámetro.

Sobre los Factores externos:

En la solución implementada se reconocen los factores externos desde ficheros mediante el uso de enumerados y se almacenan como atributos de cada corredor, siendo tomados en cuenta para el cálculo de la velocidad, y siendo actualizados varias veces a lo largo de la ejecución de la aplicación.

1. Discusión

La dificultad principal de la práctica radica en dos puntos:

- La dificultad de incluir, identificar y codificar elementos leídos desde ficheros y consola para que influyan de manera dinámica en la aplicación y sean capaces de modificar las condiciones para cada corredor.

- La dificulta de identificar y reconocer objetos en ejecución y pasarles mensajes de forma relativamente dinámica dependiendo de cómo se desarrollen las condiciones en las que se ejecutan.

Además, se ha pretendido aplicar los factores de calidad del software en lo máximo posible para facilitar la legibilidad y la extensibilidad del código. La salida de datos está centralizada, permitiendo así poder usar otros métodos (Swing, etc…).

Se han realizado un juego de pruebas que es suficiente para demostrar la corrección del programa.

1. Conclusión

Esta práctica, nos ha servido para acercarnos más al lenguaje Java, a los conceptos de la POO, y a crear un buen código, ya que una vez más, hemos aplicado los factores de calidad del software.

Creemos que los resultados obtenidos de la solución implementada son positivos ya que, aunque no es la implementación deseada, cumple con la especificación de la práctica y nos sirve como solución provisional mientras ahondamos en el estudio de Threads que nos servirán para mejorar la implementación en las prácticas sucesivas.

Por lo dicho anteriormente, pensamos que hemos cumplido con los objetivos marcados.

1. Líneas futuras

Además de la implementación de Dispatchers para la redirección de los comandos y la ejecución de varios ciclistas en una simulación de una carrera, en un futuro se añadirá una interfaz gráfica capaz de visualizar por pantalla de forma dinámica a cada uno de los ciclistas, su situación en el recorrido y los datos referentes a cada ciclista, cómo la velocidad o la distancia recorrida.

**Referencias bibliográficas utilizadas**

1. Estado del Arte -> JDK:

<http://en.wikipedia.org/wiki/Java_Development_Kit>

Estado del Arte -> JDK:

<http://docs.oracle.com/javase/6/docs/technotes/tools/windows/jdkfiles.html#jdk1.6.0>

Estado del Arte -> JDK:

<http://www.lab.dit.upm.es/~fprg/entorno/mipc/jdk.htm>

Estado del Arte -> Javadoc:

<http://www.lab.dit.upm.es/~lprg/material/apuntes/doc/doc.htm>

Introducción -> Paradigmas de programación:

[http://labsys.frc.utn.edu.ar/ppr-2009/Unidad%20I%20-%20POO%20Avanzada/Apunte/Unidad%20I%20-%20POO%20Avanzada.pdf](http://labsys.frc.utn.edu.ar/ppr-2009/Unidad%20I%20-%20POO%20Avanzada/Apunte/UnidadI-POOAvanzada.pdf)

Documentación sobre enumerados en Java:

<http://docs.oracle.com/javase/tutorial/java/javaOO/enum.html>

Documentación sobre listas en Java:

<http://docs.oracle.com/javase/6/docs/api/java/util/List.html>

Documentación Dispatcher de Akka:

[http://doc.akka.io/docs/akka/snapshot/java/dispatchers.html](http://www.lab.dit.upm.es/~lprg/material/apuntes/doc/doc.htm)

1. Anexos