

Programación en Entornos Interactivos.

Enunciado Prácticas febrero-julio 2012.

D.L.S.I.

febrero 2012

Índice

1. Enunciado.	2
1.1. Cometido de la aplicación.	3
2. Optativo.	4
3. Mejoras	4
4. Documentación.	5
5. Plazo, formato, entrega y avisos.	5

1. Enunciado.

Realizar dos versiones de una misma aplicación¹ empleando en cada versión de la práctica:

- una de las bibliotecas siguientes: Qt y Gtkmm.
- la *arquitectura MVC* para estructurar el código fuente de la misma. El código del *Modelo* será **obligatoriamente** el mismo en las dos prácticas y, para que quede claro, se encontrará en un único directorio aparte, como se puede ver en la figura 1.
- IMPORTANTE:

Esta estructura de directorios es la que **obligatoriamente** tendrás que emplear para entregar el código de las prácticas.

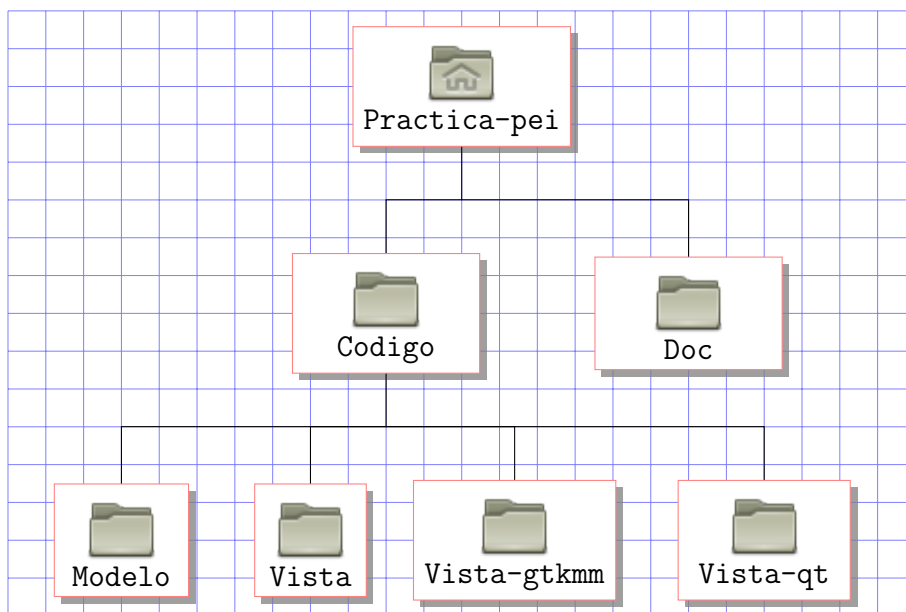


Figura 1: Estructura jerárquica de directorios con el código fuente y documentación de la práctica.

Esta jerarquía de directorios constará de un fichero `Makefile` por cada directorio. El `Makefile` del directorio `Practica-pei` se encargará de descender por cada subdirectorio y ejecutar `make` allí. El resultado final será un

¹Debe compilarse y ejecutarse con el software instalado en los laboratorios asignados para las prácticas de la asignatura.

fichero ejecutable en el subdirectorio `Vista-gtkmm` y otro en `Vista-qt`, evidentemente, cada uno de ellos enlazado con su biblioteca correspondiente.

1.1. Cometido de la aplicación.

Esta aplicación, **presentando un interfaz de usuario coherente y fácil de utilizar**², implementará el *juego de la vida* -diseñado por el matemático británico John Horton Conway en 1970- con ciertas consideraciones, por ejemplo, el tablero donde transcurre el juego tiene un **tamaño fijo** de 256×256 celdas.

Llevaremos a cabo la simulación del juego bajo las siguientes premisas:

1. Asumiremos que una celda puede estar ocupada o no por un solo ser vivo.
2. Una celda vacía con exactamente 3 celdas vecinas ocupadas será ocupada por un ser vivo en el turno siguiente.
3. Una celda ocupada por un ser vivo con 2 ó 3 celdas vecinas ocupadas seguirá ocupada en el siguiente turno, mientras que seguirá vacía o pasará a estar vacía en cualquier otro caso -*muerte por "soledad" o "superpoblación"*-.

Para ello permitirá:

1. Leer de un archivo de texto la representación inicial del juego. Este archivo estará formado por 256 líneas de 256 '0's y '1's -sin separación entre los 0's y 1's ni entre las líneas- y representará la matriz inicial del juego, donde un '1' será una celda ocupada y un '0' será una celda vacía.

```
0111100000000000...101
0000000000000000...000
1010101011111111.. 100
...
0000000000000000...000
0111100000000000...101
```

2. Que el usuario pueda activar o desactivar las celdas del tablero haciendo un click con el ratón sobre la celda correspondiente.

²Se puntuará (positiva o negativamente) el uso de los widgets más adecuados en cada caso.

3. Cuando el usuario lo decida comenzará la simulación, la cual podrá ser parada y reanudada cuantas veces se quiera. Mientras esté parada el usuario podrá activar o desactivar una celda del tablero mediante un click del ratón.
4. La simulación acabará de forma automática cuando el tablero se quede sin ningún ser vivo y se avisará al usuario de ello.
5. En cualquier momento se permitirá guardar en un archivo de texto con el mismo formato que el del punto de partida el estado actual del juego.
6. Que el usuario pueda redimensionar la ventana de la aplicación y por tanto que se redimensione la zona con la representación del juego.

Nota: El programa debe de tener una opción “**Acerca de...**” que nos presente el nombre de los *autores* de la práctica, y las *mejoras incluidas* en la misma.

2. Optativo.

Se dejan como optativas las siguientes características:

- Permitir cambiar la velocidad de la simulación.
- Permitir rellenar aleatoriamente el estado del tablero con un *porcentaje de celdas inicialmente ocupadas*.
- Permitir activar o desactivar una celda del tablero mediante un click del ratón cuando la simulación está en marcha, sin necesidad de pararla.
- Desarrollar la práctica con la biblioteca XForms compartiendo la parte del modelo con las otras dos versiones.

3. Mejoras

El programador podrá hacer las mejoras que considere oportunas, por ejemplo:

- Implementar nuevas reglas de evolución de los seres vivos, puedes ver algunas en la sección *Variantes* de la [página del juego en la Wikipedia](#).
- Implementar la opción ‘Nuevo juego’, la cual pondrá a ‘0’ todas las celdas del tablero.

- Implementar la opción ‘Guardar como’ para guardar el estado actual del juego en un fichero con otro nombre.

4. Documentación.

En tres ficheros de texto ASCII llamados:

1. ‘documentacion_modelo.txt’
2. ‘documentacion_vista_qt.txt’
3. ‘documentacion_vista_gtkmm.txt’

y localizados en el subdirectorio ‘Doc’ de la figura 1 debes indicar:

- Las funciones, métodos y clases que has creado, explicando qué hace cada una de ellas y para qué sirve cada uno de sus parámetros.
- El criterio que has seguido para dividir el código fuente en los distintos ficheros que utilizas.

5. Plazo, formato, entrega y avisos.

1. La entrega de la convocatoria de ‘febrero’ se realizará durante los **días del 28 de mayo al 5 de junio de 2012**. La entrega de la convocatoria de ‘julio’ se realizará durante los **días del 2 al 8 de julio de 2012**. Si alguna de estas fechas tuviera que ser cambiada se indicaría mediante un aviso en el Campus Virtual.

IMPORTANTE:

Este es el enunciado de la práctica de julio al que hay que añadirle como obligatorias las partes optativas excepto la relativa a la versión de ‘xforms’.

2. Se deberá entregar un fichero ‘.tgz’ con nombre ‘pei.tgz’ que contendrá la estructura de directorios de la figura 1.

Dentro de los directorios oportunos estarán:

- Los ficheros con el código fuente apropiado.
- Los ficheros con la documentación.
- Un fichero ‘Makefile’ que facilite la compilación de cada versión.

- El fichero o ficheros creados con ‘qt-designer’ y/o ‘glade’.

Dentro del subdirectorio inicial ‘Practica_pei’ habrá un fichero llamado ‘datos.txt’ que contendrá el nombre de los alumnos.

3. La entrega **sólo** se realizará vía Web³ en la siguiente **página**.
4. **No se corregirán aquellas prácticas que:**
 - No se hayan realizado con las versiones de las bibliotecas instaladas en los laboratorios de prácticas.
 - No se hayan realizado con las especificaciones explicadas en clase (por ejemplo: se debe emplear `gtkmm` en lugar de `gtk+`, cargando dinámicamente el archivo con la descripción del interfaz creada con `glade` y **no** generando el código C/C++ del interfaz).
 - Los archivos ‘Makefile’ deben funcionar en los laboratorios de prácticas y los programas compilarán sin errores.

³No se aceptará ninguna entrega hecha por correo electrónico u otro medio distinto al especificado.