Rullo

A continuación se detallan los requisitos para la elaboración del proyecto de programación obligatorio de la asignatura Introducción a la programación para el curso 2018/2019.

El objetivo de este proyecto es diseñar, implementar en C++ y documentar una versión del juego **Rullo** con ayuda de una serie de módulos ya escritos proporcionados por los profesores de la asignatura.

IMPORTANTE

- Se deben cumplir todos los requisitos que se indican en este documento.

Este proyecto puede realizarse **en grupos de dos o individualmente**. La cantidad de trabajo que hay que realizar está pensada para dos estudiantes pero, opcionalmente, se puede hacer de manera individual.

En el apartado correspondiente del aula virtual de la asignatura se puede encontrar toda la información sobre el proyecto, un ejecutable de prueba como modelo de lo que debe entregarse, un proyecto base con los archivos del entorno, una plantilla para la documentación y la rúbrica de evaluación. Además, se puede usar el foro de la asignatura para preguntar dudas y aclaraciones.

Es responsabilidad del estudiante la custodia y protección de su proyecto. **Solo las personas del grupo y los profesores de la asignatura pueden ver el código desarrollado para un proyecto**. ***Si ves el programa de otra persona, estás copiando. Si dejas tu proyecto a otra persona, estás copiando.***

Se utilizará un software de detección de copias en los programas entregados. En caso de encontrar similitudes en **partes significativas** de los programas, **todos** los implicados tendrán una nota de SUSPENSO (0) en la asignatura.

ENTREGA Y PRUEBA DE EVALUACIÓN

La entrega final del proyecto de programación se realizará mediante la actividad correspondiente del aula virtual.

El plazo de presentación se avisará con suficiente antelación. En cada convocatoria será siempre uno o dos días antes del día del examen oficial de la asignatura.

Cada miembro del grupo deberá entregar exactamente la misma información. Tanto en la documentación interna como en la externa deberán aparecer los nombres de las personas que han realizado el trabajo.

El día del examen final de la asignatura se realizarán las correspondientes pruebas de evaluación del proyecto, en el horario que se publicará en la convocatoria oficial. Esta prueba solo hay que hacerla si se ha entregado el proyecto en el plazo indicado.

La prueba se realizará de manera individual, independientemente de que el proyecto se haya realizado en pareja o individualmente.

Es necesario comprobar que se ha enviado la información correctamente y que está accesible en el aula virtual.

MATERIAL QUE SE DEBE ENTREGAR

Al entregar el proyecto se debe presentar un único fichero comprimido (tar.gz) con el nombre de las personas que han hecho el programa, **nombre1_nombre2.tar.gz**, que contenga los siguientes archivos:

- La documentación externa del proyecto en formato ODT, DOC o PDF.

  El directorio del proyecto comprimido con todos los archivos necesarios para poder compilarlo y ejecutarlo.

  Otros proyectos comprimidos con las ampliaciones (si se hacen).

  Nombre1 representa el nombre completo con los dos apellidos del primer componente del grupo (por orden alfabético) y nombre2 al del segundo. Por ejemplo, si el proyecto lo realizaran dos profesoras de la asignatura el fichero debería llamarse MariscalAraujoMAngeles_VicenteChicoteCristina.tar.gz

CALIFICACIÓN DEL PROYECTO

Un requisito previo a la evaluación del proyecto es la superación de la **prueba de evaluación**.

La prueba de evaluación del proyecto consistirá en unas preguntas sobre el proyecto y una modificación del funcionamiento básico. Las respuestas y la modificación se harán en papel, aunque se podrá consultar el código del proyecto en el ordenador.

Si no se supera la prueba de evaluación, la nota será 2.

Si es copia de otro proyecto, la nota será 0 en el bloque y en toda la asignatura para todos los implicados.

La calificación se hará en función del programa, de la documentación y de la prueba de evaluación del proyecto.

En los proyectos realizados de forma individual, la nota del bloque de proyecto se corresponderá con la nota de la evaluación del proyecto.

En los proyectos realizados en pareja, si ambos miembros del grupo superan la prueba de evaluación en la misma convocatoria y la calificación del proyecto es superior a 5, la **nota final del bloque del proyecto** se aumentará en 1 punto. En cualquier otro caso, se restará 1 punto a la calificación obtenida en el bloque del proyecto en la convocatoria en la que superen la prueba de evaluación

DESCRIPCIÓN DEL JUEGO *RULLO*

Rullo es un juego matemático para un solo jugador. (Se pueden encontrar muchas versiones para jugar en un navegador o en el móvil.)

En su versión clásica se juega con tableros cuadrados donde el número de filas y columnas oscila entre 5 y 8.

El tablero está relleno con una serie de valores aleatorios entre 1 y 9. El objetivo del juego es ir ***desactivando*** valores del tablero de forma que cada fila y columna sume una cantidad indicada al principio y al final de cada fila y columna.

Cuando los valores que quedan en una fila o columna suman el objetivo, el recuadro que rodea al objetivo cambia de color.

Se pueden bloquear posiciones activas o inactivas de manera que no se puedan activar o desactivar hasta que no se elimine ese bloqueo. Esto se usa para ir bloqueando los números cuyo estado se conoce en la solución final. Los valores bloqueados cambian de color de forma que son una ayuda visual para el jugador.

El juego termina cuando todos los objetivos se han alcanzado.

NUESTRA VERSIÓN DE *RULLO*

Versión básica

El programa servirá para una única partida.

Hay un fichero de configuración donde se determinarán las características del juego. Su utilización se explica en el apartado correspondiente a la librería "entorno".

Se podrá configurar el tamaño del tablero **n** (un valor entre 4 y 8) para jugar con distintos tamaños de tableros. El número **n** indica que habrá **n x n** números en el tablero. Los tableros siempre son cuadrados, es decir, tienen el mismo número de filas y de columnas.

Se iniciará el juego con un tablero generado aleatoriamente donde los números del tablero estarán en el rango entre 1 y 9. También se podrá iniciar el juego con un tablero leído del fichero de configuración.

El jugador utilizará las teclas de los cursores del teclado para desplazarse por el tablero. Con la tecla **Espacio** activará y desactivará los distintos números del tablero. Con la tecla **Enter** se bloquearán los números del tablero para que no se puedan cambiar con **Espacio** (y así tendrá una ayuda visual de los valores que ya sabe que formarán parte o no de la solución). Con la misma tecla **Enter** podrá eliminarse el bloqueo de los números.

El juego terminará cuando los números activos sumen los valores objetivos en cada fila y en cada columna. También se acabará cuando el usuario pulse la tecla **Escape**.

La nota máxima que se puede obtener en el proyecto con esta versión básica es de 7.

Ampliaciones

Existe la posibilidad de presentar, junto con el proyecto básico explicado anteriormente, versiones ampliadas.

Con estas ampliaciones se podrá llegar a obtener la calificación de 10 (personas que presenten individualmente el proyecto) o 13 (parejas), **siempre que el proyecto básico esté correctamente implementado y documentado.** Si existen errores graves en la versión básica o en la documentación **no** se tendrán en cuenta las ampliaciones.

Ampliaciones propuestas para todos:

Como máximo se pueden incluir dos de las propuestas de este bloque. Por tanto, el máximo de puntos que se puede conseguir es 3.

Máximo de jugadas: Se leerá del fichero de configuración un entero positivo que indicará el número máximo de jugadas (números del tablero que se activan o desactivan) para terminar. Una vez que se alcance ese número, el juego finalizará. Se deberá ir mostrando el número de jugadas que quedan. (Hasta 1 punto.)

Ayuda: Cuando se pulse la tecla **Y**, se mostrará en la pantalla, durante un corto periodo de tiempo, la suma actual de los valores activos de la fila y de la columna en la que se encuentra el valor resaltado. (Hasta 1 punto.)

Puntos: Se puede añadir un sistema de puntuación a la partida. Cada vez que se desactive un valor, se sumarán 2 puntos. Cada vez que se vuelva a activar, se perderán 3 puntos. Cuando se pulse la tecla Y de ayuda se perderán la mitad de los conseguidos (si se ha implementado esa ampliación). La puntuación mínima es 0; nunca se podrán tener puntos negativos. Habrá que mostrar siempre los puntos actualizados en la pantalla. (Hasta 1 punto.)

Inicio: Se establecen tres niveles en la generación aleatoria de los tableros iniciales (leído del fichero de configuración): en el nivel 1 (sencillo) no podrá haber valores repetidos en filas o columnas; el nivel 2 (normal) será el nivel normal, donde simplemente se generarán los valores aleatorios sin ninguna comprobación sobre los repetidos; en el nivel 3 (difícil) deberá haber números repetidos en la mitad de las filas y columnas (y así será más difícil decidir si un número debe estar activado o no) . (Hasta 2 puntos.)

Ampliación propuesta para parejas:

Resolver: Cuando se pulse la tecla **X**, el juego se irá resolviendo automáticamente. Se debe ir viendo poco a poco la solución del juego. (Hasta 2 puntos.)

No se admitirá ninguna ampliación distinta a las propuestas.

Las ampliaciones tienen que estar en un proyecto diferente al de la versión básica del juego. También debe incluirse en la documentación externa un pequeño resumen de lo que se ha cambiado con respecto a la original.

¡Atención! Algunas de estas ampliaciones pueden requerir una inversión considerable de tiempo por lo que cada estudiante debe sopesar si abordarlas o no, considerando su carga de trabajo en ésta y otras asignaturas.

NORMAS DE IMPLEMENTACIÓN

El proyecto debe implementarse usando exclusivamente programación imperativa con C++.

El proyecto debe ejecutarse correctamente en la máquina virtual utilizada en la asignatura y en los ordenadores instalados en los laboratorios.

Se debe usar la librería *entorno* entregada por los profesores para gestionar el entorno del juego **sin ninguna modificación**. Para que esta librería funcione, es necesario tener instalada la librería ***allegro5***. (En la máquina virtual y en los laboratorios ya está instalada.)

Se deben definir, como mínimo, un tipo abstracto de datos para representar el tablero completo, otro para representar cada casilla del tablero y otro para representar el juego.

Cada TAD o librería debe estar definido usando un fichero .h y uno .cpp.

Cada librería o TAD utilizado debe venir acompañado de la implementación de la correspondiente librería con un **juego de pruebas completo.** No es necesario hacer juegos de prueba de la librería *entorno* ni del *TAD juego*.

En el aula virtual hay un proyecto de ***eclipse*** comprimido, ***rulloBase.tar.gz***, con los archivos del entorno (entorno.h y entorno.cpp) y un programa simple de ejemplo que puede usarse como punto de partida para desarrollar el proyecto. Este proyecto está configurado ya con la información sobre las librerías necesarias.

En el aula virtual también hay un programa ejecutable del juego para tener una idea del aspecto final del proyecto.

TIPOS ABSTRACTOS DE DATOS

Tras un análisis del problema, surge la necesidad de modelar los siguientes elementos que se manejan con los tipos abstractos correspondientes:

El **entorno** gráfico de la aplicación, responsable de la interacción con el usuario.

La **casilla**, que tiene un número y puede estar activada o desactivada y bloqueada o no.

El **tablero**, compuesto por un conjunto más o menos grande de celdas, según lo que se indique en el archivo de configuración. Sobre el tablero se hacen las operaciones para iniciarlo aleatoriamente o a partir del fichero de configuración, calcular lo que suma una fila o una columna, etc.

El gestor del **juego**, que se encarga de controlar todo el proceso de juego: el tablero, la representación gráfica, la respuesta a las teclas, etc.

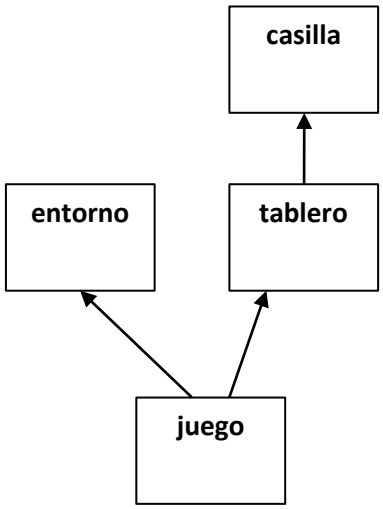
La librería **entorno** es la que se encarga de gestionar la interfaz del programa.

El TAD **Casilla** gestiona el estado de una celda (su valor y si está activa o no y bloqueada o no) y las operaciones correspondientes.

El TAD **tablero** gestiona, mediante las operaciones correspondientes, toda la información del tablero: su inicialización, su modificación y su estado en cada momento de la partida.

El **juego** realiza el control de todo el proceso, encargándose de gestionar las teclas que se pulsan, los movimientos del cursor, el fin del juego. y de mantener la coherencia entre la información que se almacena en el tablero y lo que se representa en pantalla en cada instante del juego.

Este es el esquema de relación entre los tipos abstractos de datos que se pueden definir y utilizar, y la librería principal del entorno que debe usarse para programar la interfaz del juego. Si se considera necesario, se pueden implementar más tipos. Este esquema es una simple recomendación.



LIBRERÍA ENTORNO

Con el fin de facilitar la labor de desarrollo se entrega, en un proyecto de ***eclipse***, una librería con los módulos necesarios para gestionar el entorno gráfico. Desde los módulos desarrollados por el programador solo se deben usar las operaciones definidas en ***entorno.h***. Estos módulos no pueden modificarse.

En el aula virtual hay un fichero ***rulloBase.tar.gz*** con un proyecto base con los ficheros del entorno y un pequeño ejemplo de funcionamiento (no se usan todas las operaciones del entorno), que debe usarse como proyecto base para añadir el resto de módulos.

En ***entorno.h*** aparecen las constantes definidas y las funciones de manejo del entorno (con su especificación con pre/postcondiciones). Podemos suponer que todas las funciones del entorno tienen un coste constante, O(1).

También se define el tipo enumerado con los valores que devuelve la función **entornoLeerTecla**, con los valores de las únicas teclas aceptadas (cursores, Enter, Espacio, X, Y y Escape). Si se pulsa cualquier otra tecla, la función devuelve TNada. (Inicialmente, algunas teclas no se utilizan para nada, pero se reconocen por si se quieren usar en alguna ampliación o modificación del proyecto.)

El juego se configurará a partir de la información presente en el fichero ***rullo.cnf*** que debe estar situado en el directorio del proyecto.

La estructura de este fichero de texto, que nos permite cargar la configuración inicial del juego, es la siguiente:

En la primera línea, el tamaño del tablero, **n,** que será un valor entre 4 y 8.

En la segunda línea, un valor entero no negativo, **i,** 0<= i < n*n**,** que determinará cómo se generará el tablero inicial. Si **i** es 0, el tablero se leerá de este fichero de configuración. Si **i** es positivo, habrá que generar el tablero aleatoriamente, teniendo en cuenta que en el tablero solución deberán estar desactivados **i** números.

En la tercera línea, un valor positivo **m**, que indica el número máximo de jugadas (para la ampliación Máximo de jugadas). Si no se implementa esta ampliación, ese número no se usará para nada.

En la cuarta línea, un valor positivo que podrá tomar el valor 1, 2 o 3 para la ampliación Inicio. Si no se implementa esta ampliación, ese número no se usará para nada.

Si en la segunda línea hay un 0, a partir de la quinta línea habrá la siguiente información en el fichero:

- n líneas con n números cada una (entre 1 y 9), separados por un espacio, con los valores iniciales del tablero. Y luego, el valor objetivo de esa fila.
- una línea con n números que serán los valores objetivos de cada una de las columnas (empezando por la de la izquierda)

Se puede suponer que el fichero de configuración es correcto. Si es incorrecto, el comportamiento de la aplicación no tiene que ser correcto.

Hay una operación en la librería entorno que devuelve toda la información que hay en el fichero de configuración.

TAD CASILLA

El TAD Casilla gestionará la información de una casilla del tablero. En cada casilla hay un número que puede estar activo o no y bloqueado o no.
Las operaciones que pueden ser útiles con una casilla son:

- iniciar una casilla
- poner un valor en una casilla
- cambiar el estado de activación de una casilla
- cambiar el estado de bloqueo de una casilla
- obtener el valor de una casilla
- obtener si la casilla está activa o no
- obtener si la casilla está bloqueada o no

TAD TABLERO

El TAD Tablero es el que va a gestionar la información del tablero en cada momento del juego. En el tablero no se gestionan las teclas que se pulsan durante el juego, ni se actualiza la pantalla, sino que se definen las operaciones necesarias para modificar y obtener la información del tablero.
La estructura de datos seleccionada para representar el tablero condiciona el desarrollo e implementación del resto del proyecto. Es importante decidir qué estructura de datos es la más adecuada.
La estructura de datos correspondiente al tablero debe guardar:

- Espacio capaz de almacenar tantas casillas como se puedan definir con el tamaño máximo del tablero completo.
- La dimensión real del tablero (número de filas y columnas) con el que se está jugando.
- Los valores objetivos de las filas y de las columnas

Algunas de las operaciones que pueden ser útiles son las siguientes (pueden ser necesarias algunas más, o se pueden agrupar o dividir en otras similares; no siempre se indican todos los parámetros que deben pasarse):

- Dado el tamaño del tablero, iniciarlo aleatoriamente.
- Dado el tamaño del tablero, iniciarlo con la información obtenida del fichero de configuración.
- Obtener el tamaño real del tablero.
- Cambiar el valor de una celda concreta del tablero.
- Cambiar el estado de activación de una celda concreta del tablero.
- Cambiar el estado de bloqueo de una celda concreta del tablero.
- Obtener el valor de una celda concreta del tablero.
- Obtener el estado de activación de una celda concreta del tablero.
- Obtener el estado de bloqueo de una celda concreta del tablero.
- Obtener el resultado de sumar todas las casillas activas de una fila o de una columna
- Comprobar si el tablero está resuelto o no.

En el TAD Tablero se almacenará toda la información de los valores almacenados en la memoria. Los cambios que se produzcan en la pantalla se realizarán desde el gestor del juego, mediante las operaciones de la librería Entorno.

JUEGO

El TAD Juego es el que realiza la gestión del juego completo. Será el que gestione el tablero, la interacción con el usuario a través del teclado y la actualización del entorno gráfico del juego (la pantalla).
Como mínimo, deberá tener tres módulos:

- un módulo que inicie la estructura de datos del juego, según la configuración del fichero ***rullo.cnf***
- un módulo que realice la gestión general del juego (gestionar las teclas que se pulsen, actualizar el tablero y la pantalla, dar el juego por finalizado, etc.)
- un módulo que termine el juego, mostrando un mensaje de despedida y cerrando el entorno gráfico.

PLANIFICACIÓN

El trabajo de implementación y documentación de este proyecto está planificado en unas 50 horas de dedicación entre dos estudiantes que hayan realizado un seguimiento correcto de las clases, sesiones de laboratorio y actividades propuestas hasta el momento.

A continuación se presenta una posible planificación, con las horas de dedicación y las principales tareas que hay que desarrollar. (Están detalladas por horas de trabajo individual; algunas serán conjuntas y otras por separado. Por ejemplo, la definición del tablero podría realizarse en una reunión de una hora de duración.)

Hay que ir anotando los tiempos dedicados a cada una de las tareas para incluirlos posteriormente en la documentación externa del proyecto.

Horas	Tarea
4	Lectura de la documentación inicial, planificación del trabajo en grupo (2 horas cada persona)
2	Prueba del proyecto base
2	Diseño general de la aplicación y de los TAD necesarios
4	Diseño e implementación del TAD Casilla (estructura de datos, implementación de operaciones y pruebas)
2	Definición del TAD tablero (estructura de datos y operaciones necesarias)
12	Implementación del TAD tablero (juegos de pruebas y operaciones)
10	Definición e implementación del TAD juego (operaciones)
4	Pruebas de integración del proyecto
6	Ampliaciones del proyecto
4	Escritura final de la documentación (una parte de esta tarea se debe realizar a lo largo de todo el proceso, mientras se implementan los TAD)

Total: 50 horas

DOCUMENTACIÓN

El proyecto debe ir acompañado de su correspondiente documentación interna y externa.

DOCUMENTACIÓN INTERNA

En el fichero .h de cada TAD se debe incluir la especificación con pre/postcondiciones y la complejidad de cada operación.
En el fichero .h de las librerías de prueba de cada TAD deben incluir, con comentarios, el diseño de las pruebas que se hacen en cada módulo.

DOCUMENTACIÓN EXTERNA

En el aula virtual se puede encontrar una plantilla que sirve de guía para la redacción de la documentación externa, con la información que debe contener.
El formato de la página de la cubierta debe ser el indicado en la plantilla de la documentación, que incluye la identificación de los estudiantes, el grupo al que pertenecen y su profesor de laboratorio.
A continuación se explican brevemente los distintos apartados que, como mínimo, debe contener la documentación externa entregada.

Introducción

Toda documentación de programas debe incluir una breve introducción sobre el software desarrollado, explicando el objetivo principal y los requisitos satisfechos.

Análisis y diseño

Una vez que conocemos cuál es el problema en cuestión, descrito en la introducción, debemos comprender el problema y analizar qué entidades intervienen en el mismo. Además, será necesario asociar a cada una de estas entidades las principales acciones que pueden realizar.

La documentación relativa al análisis facilita la comprensión del problema, ya que, sin entrar en demasiados detalles, se describen los tipos abstractos de datos que posteriormente serán implementados y las principales decisiones tomadas.

En el diseño de la aplicación se detalla cada uno de los tipos abstractos de datos utilizados para desarrollar la aplicación. Para cada uno de los tipos abstractos se debe explicar su composición y las operaciones relacionadas.

Esta sección se debe iniciar con el esquema de los tipos abstractos usados (similar al que aparece en la página 4), pasando después a describir cada uno de los tipos de abstractos de datos y el programa principal.

Planificación y tareas

En este apartado de la documentación se debe incluir la planificación inicial del proyecto, las principales tareas realizadas en el desarrollo del programa, quién las ha realizado y el tiempo empleado en cada una de ellas.

Es fundamental incluir entre las tareas, las reuniones mantenidas, el tiempo empleado en cada una de ellas y los principales acuerdos alcanzados.

Conclusiones y principales problemas

En este apartado de la documentación se deben incluir las principales conclusiones extraídas por los autores del trabajo. Además, se debe reflexionar sobre los problemas encontrados a la hora de desarrollar la aplicación y lo que se ha aprendido.

Los listados con el código no deben incluirse en la documentación externa.