

n en raya

Plazo de entrega	1
Material que se debe entregar	2
Descripción del juego <i>4 en raya</i>	2
Nuestra versión: <i>n en raya</i>	2
Normas de implementación	3
Tipos abstractos de datos	4
Librería entorno.....	4
TAD casilla.....	5
TAD tablero.....	5
TAD juego.....	6
Programa principal	6
Planificación	6
Documentación externa	7
Introducción	7
Análisis y diseño	7
Planificación y tareas	7
Conclusiones y principales problemas.....	7

A continuación se detallan los requisitos para la elaboración del proyecto de programación obligatorio de la asignatura 'Introducción a la programación' para el curso 2012/2013.

El objetivo de este proyecto es diseñar, implementar en C++ y documentar una versión del juego **4 en raya** o **conecta 4** 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 ningún caso se mejorará la calificación de un estudiante por el hecho de haber realizado el trabajo de forma individual.
- 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 obtendrán una calificación final de SUSPENSO (0) en la asignatura.
- Es responsabilidad del alumno la custodia y protección de su proyecto.
- Los criterios de evaluación se encuentran en el programa de la asignatura.
- 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 el foro de la asignatura para aclaraciones adicionales.

PLAZO DE ENTREGA

La entrega del proyecto de prácticas final se realizará **antes del 18 de enero de 2012, viernes, a las 14:00 horas**, mediante la actividad correspondiente del aula virtual.

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 dos personas que han realizado el trabajo (excepto si se ha hecho de forma individual.)

El lunes 21 de enero se realizarán las correspondientes pruebas de evaluación del proyecto, en el horario que se publicará en la convocatoria oficial de la asignatura. **Esta 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.

Si no se entrega el proyecto en la fecha indicada, habrá que presentarlo (para realizar su recuperación) en las fechas que se comunicarán oportunamente en las convocatorias extraordinarias oficiales del curso 12/13.

MATERIAL QUE SE DEBE ENTREGAR

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

- La documentación externa del proyecto en formato ODT o PDF.
- El directorio del proyecto comprimido con todos los archivos necesarios para poder compilarlo y ejecutarlo.
- 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 los profesores de la asignatura, el fichero debería llamarse MariscalAraujoMAngelos_PulidoGalanFernando.tar.gz

DESCRIPCIÓN DEL JUEGO 4 EN RAYA

El juego *4 en raya* o *conecta 4* es un conocido juego de estrategia para dos jugadores. Generalmente se juega en un tablero vertical de 6 filas por 7 columnas.

Por turnos, cada una de las personas que juega va poniendo una ficha de su color en el tablero. Sólo hay que escoger la columna, ya que la ficha cae hasta el fondo del tablero o hasta que otras fichas impiden que siga bajando.

Gana el jugador que consigue hacer una línea de 4 o más fichas de su color horizontal, vertical o diagonalmente. Cuando ya no se pueden colocar más fichas en el tablero, el juego termina en empate.



Una referencia y un sitio donde poder jugar:

- http://en.wikipedia.org/wiki/Connect_Four
- <http://www.disfrutalasmaticas.com/juegos/4-en-linea.html>

NUESTRA VERSIÓN: N EN RAYA

Versión básica

- El programa servirá para una única partida de dos jugadores.
- Existirá un fichero de configuración donde se podrán determinar algunas características del juego.
- Se podrá configurar el tamaño del tablero para jugar con distinto número de filas y de columnas.
- Se podrá configurar la longitud de la línea con la que se gana el juego.

- El programa sólo permitirá movimientos válidos según las reglas anteriores y llevará a cabo automáticamente la gestión del juego (comprobaciones, control de los turnos y finalización del juego).
- Con las teclas del cursor se puede seleccionar la columna deseada. La jugada se realizará al pulsar la tecla Enter.
- El juego finalizará cuando uno de los dos jugadores haya conseguido n fichas de su color en línea, ya sea horizontal, vertical o diagonalmente. Si ya no caben más fichas en el tablero también se acabará el juego con un empate.

La nota máxima que se puede obtener en el proyecto con esta versión básica es de 8.

Ampliaciones

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

Con estas ampliaciones se podrá llegar a obtener la calificación de 10 en el proyecto, siempre que el proyecto original esté correctamente implementado y documentado.

La ampliación debe acordarse previamente enviando un mensaje al profesor de prácticas, explicando brevemente qué cambios se pretenden hacer. El profesor aceptará o no la ampliación, indicando también la nota máxima que se podrá alcanzar con esa modificación. No se considerará ninguna extensión que no haya sido aprobada previamente.

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

Algunas ampliaciones posibles: jugar con más de 2 jugadores, animar la caída de las fichas cuando se selecciona la columna deseada, ganar por haber realizado más líneas cuando se haya llenado todo el tablero, etc.

Ampliación extra

También existe la posibilidad de presentar una ampliación extra del juego que permite obtener hasta una calificación de 12.

En este caso, cada vez que se consigue una línea de n fichas del mismo color, esa línea desaparece y las fichas que están sobre esas fichas, caen. Ganará el jugador que consiga antes x líneas, siendo x un valor leído en el fichero de configuración. Si el tablero se llena antes de que alguno de los dos jugadores consiga las x líneas, ganará aquel que tenga más o quedarán empatados.

Esta ampliación tiene que estar en un proyecto diferente al de la versión original de la práctica. También debe incluirse en la documentación externa un resumen de lo que se ha cambiado con respecto a la práctica original.

Además de la defensa habitual con la versión básica del proyecto, esta ampliación requiere una entrevista con el profesor de prácticas para debatir sobre el trabajo realizado.

Una nota superior a 10 en este bloque permitirá subir la nota final de la asignatura, compensando notas más bajas en los otros módulos, siempre que estén superados.

¡Atención! Esta ampliación requiere una inversión considerable de tiempo por lo que cada estudiante debe sopesar si abordarla 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 que se usan.
- Se debe usar la librería *entorno* (compuesta por un archivo .h y otro .cpp) entregada por los profesores para gestionar el entorno del juego **sin ninguna modificación**. Para que esta librería funcione, es necesario tener instalada la librería *allegro*. (En la máquina virtual y en los laboratorios ya está instalada.)
- Se deben definir, como mínimo, los tipos abstractos de datos que se explican en esta documentación: casilla, tablero y juego. Si se considera necesario, se pueden definir TAD adicionales.
- Cada librería o TAD utilizado debe venir acompañado de la implementación de un **juego de pruebas completo**.
- Cada TAD debe estar definido usando un fichero .h y uno .cpp.

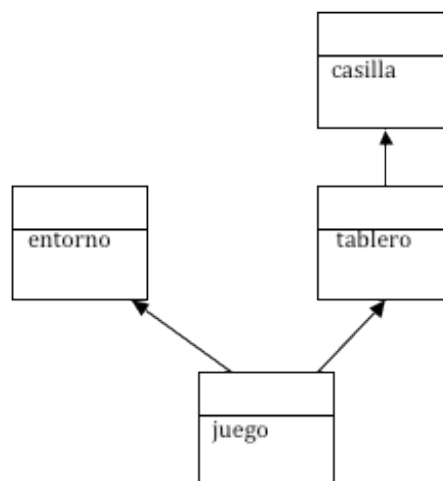
- En el aula virtual hay un proyecto de *eclipse* comprimido, *nenraya_base.tar.gz*, con los archivos del entorno (entorno.h y entorno.cpp), una implementación del TAD Casilla y un programa breve de pruebas que debe usarse como punto de partida para desarrollar el proyecto. Este proyecto está configurado 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 de programación.

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 puede estar vacía o tener una ficha de un color determinado.
- El **tablero**, compuesto por un conjunto más o menos grande de casillas.
- El gestor del **juego**, que se encarga de controlar todo el proceso.

Este es el esquema de relación entre los tipos abstractos de datos que, como mínimo, se deben 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.



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

El TAD **casilla** gestiona el estado de una casilla (vacía u ocupada con una ficha de alguno de los jugadores) 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 TAD **juego** realiza el control del juego completo, encargándose de gestionar los turnos, el estado del tablero, comprobar la finalización de la partida y mantener la coherencia entre la información que se almacena en el tablero y lo que se representa en pantalla en cada instante del juego.

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 sólo se deben usar las operaciones definidas en *entorno.h*. Estos módulos no pueden modificarse y deben utilizarse como se entregan.

En el aula virtual hay un fichero *nenraya_base.tar.gz* con un proyecto base con los ficheros del entorno y un pequeño programa de prueba (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).

También se define el tipo enumerado con los valores que devuelve la función *TEntornoLeerTecla*. Según los valores de ese tipo, las únicas teclas aceptadas son las teclas del cursor (arriba, abajo, izquierda, derecha), la tecla X, Enter y Escape. Si se pulsa cualquier otra tecla, la función devuelve *TNada*. (Inicialmente, los cursores arriba y abajo y la tecla X no se utilizan para nada, pero se reconocen por si se quieren usar en alguna ampliación.)

El juego se configurará a partir de la información presente en el fichero *nenraya.cnf* que debe estar situado en el directorio del proyecto junto con las imágenes que se usan para representar las casillas de distintos colores.

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

- En la primera línea, el número de filas del tablero. Debe ser un valor entre 1 y 9.
- En la segunda línea, el número de columnas del tablero. Debe ser un valor entre 1 y 8.
- En la tercera fila, la longitud mínima de la línea que permite ganar el juego. Debe ser, como mínimo, 2.
- En la cuarta fila, el número de jugadores. En la versión básica este valor será siempre 2.
- En la quinta fila, un número que no sirve para nada en la versión básica. Se puede usar para alguna ampliación.
- En la sexta fila, un número que no sirve para nada en la versión básica. Se puede usar para alguna ampliación.
- En las líneas siguientes, los nombres de los jugadores, hasta un máximo de 5.

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

A continuación se muestra un posible fichero *nenraya.cnf* con la información para un juego con un tablero de 6x7, rayas con un longitud de 4 y 2 jugadores.

nenraya.cnf

```
6
7
4
2
0
0
JugadorA
JugadorB
```

TAD CASILLA

El TAD casilla es el que va a gestionar la información de una casilla del tablero. Una casilla puede estar vacía o estar ocupada por una ficha de un determinado color.

Las operaciones que pueden ser útiles con una casilla son la creación de la casilla, cambiarle el color, vaciarla, determinar su estado (vacía o no) y su color actuales.

Se ha incluido en el proyecto base una posible implementación del TAD casilla.

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 tareas del 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 estructuras de datos son las más adecuadas.

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.
- El número de filas y de columnas reales del tablero con el que se está jugando.

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):

- Iniciar el tablero.
- Poner una ficha de un jugador en una determinada columna.
- Comprobar si el tablero está completo.
- Comprobar si hay líneas de una longitud determinada en un determinado sentido.
- Obtener las dimensiones reales del tablero.

En este 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 TAD juego, mediante las operaciones de la librería entorno.

TAD JUEGO

El TAD juego es el que realiza la gestión del juego completo. Será el que gestione el tablero y actualice el entorno gráfico del juego (la pantalla).

La estructura de datos para este tipo deberá contener, como mínimo, un tablero, la longitud de la línea, qué jugador tiene el turno actualmente, etc.

Como mínimo, tendrá que tener dos módulos:

- un módulo que inicie la estructura de datos del juego, según la configuración del fichero *nenraya.cnf*
- un módulo que realice la gestión general del juego.

PROGRAMA PRINCIPAL

El programa principal es muy simple, y sólo tiene que definir una variable del tipo juego, iniciar el juego y jugar.

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.)

Horas	Tarea
4	Lectura de la documentación inicial, planificación del trabajo en grupo (2 horas cada persona)
4	Prueba del proyecto base
4	Diseño general de la aplicación y de los TAD necesarios
2	Definición del TAD tablero (estructura de datos y operaciones necesarias)
10	Implementación del TAD tablero (juegos de pruebas y operaciones)
2	Definición del TAD partida (estructura de datos y operaciones necesarias)

10	Implementación del TAD juego (juegos de pruebas y 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)
50	

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 pormenorizadamente cada uno de los tipos abstractos de datos utilizados para desarrollar la aplicación. Para cada uno de los tipos abstractos se debe documentar su composición y operaciones relacionadas. Además, en el caso de las operaciones, se deben indicar sus precondiciones, postcondiciones y complejidad.

Un apartado importante en la definición del diseño es la descripción de los juegos de pruebas que se desarrollarán para cada tipo abstracto de datos y para validar el programa final.

Esta sección se debe iniciar con el esquema de los tipos abstractos usados, pasando después a describir cada uno de los tipos de abstractos de datos y finalizando con 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 el autor del trabajo. Además, se debe reflexionar sobre los problemas encontrados a la hora de desarrollar la aplicación.

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