Angel Mart

A continuación se detallan los requisitos para la elaboración del proyecto de programación obligatorio de la asignatura Introducción a la programación para el curso 2017/2018.

El objetivo de este proyecto es diseñar, implementar en C++ y documentar una versión del juego **Angel Mart** 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 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

Como requisito previo en la evaluación del proyecto está 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.

La evaluación 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, la nota del bloque del proyecto será la nota de evaluación del proyecto más 1 punto adicional. En cualquier otro caso, tendrán 1 punto menos en la convocatoria en la que superen la prueba de evaluación.

DESCRIPCIÓN DEL JUEGO *ANGEL MART*

Angel Mart es un juego que mezcla características de juegos clásicos de memoria y del conocido Tetris.

En su versión estándar se juega en un tablero de 7 filas y 8 columnas. Las celdas de las dos filas inferiores están ocupadas inicialmente por fichas que tienen dos caras diferentes; el resto de las celdas están vacías. De forma similar a las tradicionales cartas (españolas o de poker), las fichas tienen un reverso común a todas ellas y un anverso en el que aparecen diferentes imágenes. Inicialmente, solo es visible el reverso común de las fichas.

El jugador debe seleccionar dos fichas para que se den la vuelta, dejando ver su anverso durante un pequeño periodo de tiempo. Cuando el jugador selecciona dos fichas cuyo anverso coincide, ambas fichas desaparecen. Entonces, las que ocupan posiciones superiores “caen” ocupando los huecos dejados por las fichas borradas (la situación es similar a lo que ocurre en el juego Tetris cuando una fila de fichas desaparece). Si los anversos de las fichas son diferentes, éstas se giran de nuevo y no pasa nada.

El jugador dispone de un tiempo determinado, que llamaremos “jugada”, para eliminar el máximo número de parejas de fichas que sea posible. Cuando este tiempo finaliza, se añade una nueva fila de fichas al tablero, y comienza de nuevo el tiempo de una nueva “jugada”.

El juego termina cuando no se puede colocar una nueva fila de fichas en el tablero por falta de espacio.

La puntuación depende del número de fichas eliminadas del tablero.

Para ver el juego original se puede consultar la web http://www.juegos.com/juego/angel-mart?

NUESTRA VERSIÓN DE *ANGEL MART*

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 10) para jugar con distintos tamaños de tableros. El número **n** indica que habrá **n x n** celdas. Los tableros siempre son cuadrados, es decir, tienen el mismo número de filas y de columnas. El tamaño del tablero es siempre un número par para que las fichas (generadas aleatoriamente) que se colocan en el tablero, al comienzo de la partida o después de cada jugada, puedan estar siempre emparejadas.
- Cuando el juego comienza se colocan fichas en las dos filas inferiores del tablero. Las fichas están colocadas de forma que solo se ve su reverso. Las fichas deben generarse aleatoriamente pero garantizando que todas tengan pareja.
- El anverso de las fichas contiene un dígito del 0 al 9.
- El jugador utilizará las teclas de los cursores del teclado para desplazarse por el tablero y seleccionará una ficha pulsando **Enter**. Debe seleccionar dos fichas en posiciones diferentes del tablero (si selecciona dos veces la misma ficha el juego no debe hacer nada) a las que se da la vuelta durante un pequeño periodo de tiempo:
    - Si los dígitos del anverso de las fichas son iguales:
        - ambas fichas desaparecen,
        - se actualiza el marcador de la puntuación añadiendo 10 puntos y
        - las fichas que hay encima de las eliminadas caen hacia abajo.
    - Si los dígitos del anverso de las fichas son diferentes se vuelven a colocar del revés.
- El juego debe controlar que ha pasado el tiempo correspondiente a una "jugada". Cuando esto ocurre, se generan aleatoriamente **n** nuevas fichas (garantizando que estén emparejadas) que caen desde la fila superior, es decir, en cada columna la nueva ficha se colocará en la celda vacía que ocupe la posición más baja.
- El juego terminará cuando no se pueda colocar una nueva colección de fichas por falta de espacio, cuando el tablero quede completamente vacío o cuando se pulse la tecla Escape. Si el tablero queda completamente vacío, el jugador recibe 50 puntos más.
- Cuando el juego termina debe mostrar en un mensaje la puntuación obtenida por el jugador.

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 12 (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:

- **Pista 1:** Cuando se pulse la tecla X, se muestra durante un corto periodo de tiempo la imagen del anverso de todas las fichas. Se perderá un número determinado de puntos (indicados en el fichero de configuración) cada vez que se use la pista. (Si no hay puntos suficientes, pulsar la tecla X no hará nada.) (Hasta 0.5 puntos.)
- **Pista 2:** Cuando se pulse la tecla Y, se da la vuelta, una a una y durante un corto periodo de tiempo, a cada ficha. Se perderá un número determinado de puntos (concretamente, la **mitad** de los que indica el fichero de configuración) cada vez que se use la pista. (Si no hay puntos suficientes, pulsar la tecla Y no hará nada.) (Hasta 1 punto.)
- **Pista 3:** Una vez seleccionada y volteada una ficha, si se pulsa la tecla Z, se dará la vuelta brevemente a todas las fichas que tengan el mismo número. Se perderá un número determinado de puntos (concretamente, el **doble** de los que indica el fichero de configuración) cada vez que se use esta pista. (Si no hay puntos suficientes, pulsar la tecla Z no hará nada.) (Hasta 1 punto.)
- **Inserción inferior:** Cuando finaliza una jugada, las nuevas fichas generadas aleatoriamente ocupan la fila inferior del tablero, es decir, la fila n-1. Esto implica que el resto de las fichas se desplazan una posición hacia

arriba. Si en ese momento alguna ficha está seleccionada (y podemos ver el anverso) debe también cambiar de posición en el tablero (pasando a la fila inmediatamente superior). (Hasta 1.5 puntos.)

No se pueden hacer las ampliaciones Pista 2 y Pista 3. Es necesario elegir, como mucho, una de ellas. Por tanto, con este bloque de ampliaciones se pueden obtener 3 puntos extra como máximo.

Ampliación propuesta para parejas:

- **Bonos y bombas:** Cada vez que un nuevo conjunto de fichas se coloca en el tablero, una de ellas (elegida aleatoriamente) será especial: puede ser un "bono" o una "bomba". Los bonos y las bombas se reconocerán porque el color del anverso será diferente. Cuando una ficha "bono" se borra, la puntuación se incrementará en 10 puntos. Cuando una ficha "bomba" se borra, también se eliminan todas las fichas que tienen el mismo valor que ella y la puntuación se actualiza adecuadamente. (Hasta 2 puntos.)

No se admitirá ninguna extensión 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 requieren 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

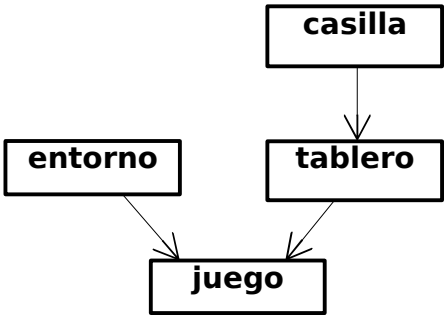
- El proyecto debe implementarse usando exclusivamente programación imperativa con C++.
- El proyecto debe ejecutarse correctamente en la máquina virtual utilizada en la asignatura y en los ordenadores instalados en los laboratorios.
- Se debe usar la librería *entorno* entregada por los profesores para 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 celda 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, ***AngelMartBase.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 **celda**, que puede estar vacía o tener una ficha con un determinado valor (del 0 al 9). Si la celda está ocupada por una ficha es necesario conocer si está visible el anverso o el reverso de la ficha.
- El **tablero**, compuesto por un conjunto más o menos grande de celdas, según el archivo de configuración. Sobre el tablero se hacen las operaciones para comprobar si las dos fichas descubiertas son iguales, se eliminan las fichas emparejadas, se añaden las nuevas fichas cuando pasa el tiempo de una jugada, etc.
- El gestor del **juego**, que se encarga de controlar todo el proceso de juego: el tablero, la representación gráfica, la puntuación, etc.

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.



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

El TAD **celda** gestiona el estado de una celda (vacía o con una ficha con el anverso visible 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 fichas que se colocan, los movimientos del cursor, el fin del juego, la puntuación 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.

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 ***AngelMartBase.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, X, Y, Z 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 ***AngelMart.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, que será un valor par entre 4 y 10.
- En la segunda línea, un valor entero positivo, que indica los segundos que dura una jugada.
- En la tercera línea, la puntuación que se descontará si se utiliza la pista 1 y valores proporcionales para las pistas 2 y 3. Este valor solo se utilizará en el caso de hacer estas ampliaciones.

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

TAD CELDA

El TAD celda gestionará la información de una celda del tablero. Una celda puede estar vacía u ocupada por una ficha. Además, la ficha tiene un determinado valor numérico en el anverso y puede tener dos estados diferentes: anverso (el número está visible) y reverso (el número no está visible). Si se lleva a cabo la ampliación “Bonos y bombas”, se deberá controlar la existencia de fichas “especiales”.
Las operaciones que pueden ser útiles con una celda son:

- la inicialización de una celda como vacía,
- vaciar una celda,
- poner un valor en la celda,
- poner el estado de la ficha: anverso o reverso,
- obtener el valor de la celda,
- obtener si está vacía o no,
- obtener la posición de la ficha: anverso o reverso,
- comparar dos celdas.

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 los movimientos que hace el usuario, ni las puntuaciones, 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 celdas 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.

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 correctamente.
- Determinar si una celda está vacía o no, o qué valor o estado (anverso, reverso) tiene.
- Determinar si cabe una nueva fila de fichas en el tablero.
- Determinar si el tablero está vacío.
- Modificar el estado de una ficha, pasando de anverso a reverso o viceversa.
- Poner una ficha en una determinada posición del tablero.
- Borrar una ficha y hacer que "caigan" las que están encima.
- Obtener el tamaño real del tablero.
- Colocar una nueva colección de fichas en la parte superior del tablero.

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 puntuación 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 ***AngelMart.cnf***
- un módulo que realice la gestión general del juego (gestionar las teclas que se pulsen, actualizar la puntuación, 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 celda (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.