

NORMATIVA del Trabajo de Informática Industrial. Curso 2021-2022

Integrantes de los equipos

Los alumnos formarán equipos de cinco alumnos como máximo, cuatro alumnos como mínimo. Para ello deberán entregar en la tarea dispuesta en MOODLE la ficha adjunta en el anexo con la siguiente información:

- Nombre del equipo
- Nombre de cada componente del equipo
- Repositorio en GitHub usado para desarrollar el trabajo
- Cuenta de usuario de GitHub de cada uno de los componentes del equipo
- Firma de los integrantes

Aquellos alumnos que por lo que sea no han conseguido formar grupo, deben entregar la ficha suelta con una indicación de que no han conseguido grupo. Se intentará agregar alumnos a los grupos de cuatro hasta completar, o se formaran grupos con aquellos alumnos que queden sueltos. SE RECOMIENDA encarecidamente evitar esta situación. Lo ideal es que todos sean capaces de integrarse en un grupo.

La temática del trabajo es un juego que debe desarrollarse íntegramente en C++. Los alumnos deben incluir ideas que reflejen una clara personalización del programa, aunque todos ellos versen sobre el mismo juego que es el **Ajedrez**.

ES OBLIGATORIO el cumplimiento de unos requisitos mínimos para aprobar el trabajo. Estos se incluyen al final de esta normativa. Además se incluyen ideas y aspectos como requisitos opcionales que permitirán graduar la evaluación de los distintos desarrollos.

Procedimiento de desarrollo

Los trabajos seguirán todos el mismo procedimiento, y como tales, aunque se realice su entrega en la convocatoria extraordinaria, deberán contar con el visto bueno del profesor encargado. Concretamente se realizarán los siguientes pasos:

1. Formación de grupos. (Moodle)
2. Publicación de grupos. Semana posterior a la fecha límite de entrega de fichas de grupos.(Moodle)
3. Entrega del trabajo. Se habilitará una tarea en Moodle con una fecha límite en cada convocatoria. La fecha **LIMITE de entrega del trabajo** de la convocatoria ordinaria será en la **penúltima semana de curso** (Moodle)
4. Tutoría de evaluación OBLIGATORIA a discreción del profesor de trabajos. Hasta una semana después del examen de la convocatoria. TEAMS
5. Publicación de evaluaciones de trabajo y ranking. 10 días después del examen.

Procedimiento de entrega

El procedimiento de entrega de los trabajos es telemático. **En la convocatoria ordinaria y extraordinaria** se pondrá una fecha límite siempre anterior al examen final de la correspondiente

convocatoria. Para ello se dispondrá de una tarea en el Moodle de la asignatura en donde los alumnos deberán subir antes del vencimiento de la tarea los siguientes ficheros:

1.- **Un fichero zip con el código fuente** (si no cabe, se excluyen los datos, sonidos, etc. que se podrán incluir con un enlace a un repositorio externo). Este código deberá ser un clonado del los fuentes del repositorio del grupo y se utiliza como “congelado” del mismo.

2.- **Una memoria en PDF** en la que se explique la estructura del diseño, las clases y relaciones entre clases, así como un breve manual de instrucciones. Se deberán incluir las indicaciones necesarias para descargar e instalar las librerías auxiliares que se hayan utilizado si no han podido ser incluidas en el zip. Dada que la mayoría de las evaluaciones se realizarán de forma telemática, LA MEMORIA se considerará una parte muy relevante en la evaluación.

3.- **Un fichero en PDF con la firma de todos los integrantes del grupo** que explique las tareas realizadas por cada miembro del equipo y una evaluación de cada uno atendiendo a su participación en el desarrollo del trabajo. La autoevaluación será tal que todo el equipo dispondrá de un total de 10*número de miembros del equipo puntos. (es decir, que un equipo de 5 personas dispondrá de un total de 50 puntos). Estos puntos los distribuirán entre los miembros del equipo de forma que nunca se repitan dos puntuaciones. Por ejemplo (50 = 20+15+8+5+2).

4.- **Un fichero que incluya al menos dos enlaces externos:**

a.- Un enlace a un video corto (4-5 minutos como máximo) en el que se muestren las principales funcionalidades del proyecto. EN EL VIDEO se deberán resaltar los aspectos más específicos del desarrollo, poniendo especial énfasis en lo más original del mismo. De igual forma, aquellos aspectos que se detallan al final que se tendrán en cuenta particularmente para la evaluación.

b.- Un enlace a un zip externo que permita la ejecución directa del programa. Es decir, una vez descomprimido en una carpeta de un ordenador con Windows y Visual Studio 2019 instalado, se puede ejecutar y testear sin problemas el resultado.

Presentación telemática/presencial

A discreción del profesor encargado de los trabajos, se procederá una vez realizada la entrega a convocar a los integrantes del equipo para que expliquen el desarrollo realizado. Salvo clara incompatibilidad que deberá ser justificada, los alumnos tendrán la obligación de asistir a la defensa de su trabajo en la fecha y horario en el que se les convoque. Esta presentación será presencial en principio pudiendo derivarse a telemática por teams en función de la situación pandémica.

Plagio / copia

En el trabajo se admiten las prácticas habituales en la programación de software, como es la consulta de ciertos elementos de código que realizan tareas específicas. Sin embargo, el desarrollo debe ser original en su conjunto lo cual es fácil de ver por la estructuración y organización de clases y código así como por las especificidades de implementación. Además, se debe observar el desarrollo incremental reflejado en el repositorio del proyecto.

En caso de detectarse la entrega de un trabajo en el que se considera por parte del tribunal que no es mayoritariamente de los integrantes del grupo, o no haberse realizado INCREMENTALMENTE con contribuciones (*commits*) recurrentes, el tribunal podrá aplicar en su caso con la máxima rigurosidad la normativa UPM actual. Esto es, en caso de considerarse un plagio, además de evaluarse con cero el trabajo y por tanto suspender automáticamente la asignatura, se procederá a elevar al Rector los hechos para que así se tomen las medidas disciplinarias oportunas.

Evaluación

El profesorado evaluará el conjunto del trabajo con una nota común denominada *NotaTrabajo*. Esta nota será de 5 a 10 en caso de considerarse APTO (en caso contrario no se aceptará la entrega del trabajo). Esta calificación se pondrá en base a:

- Principalmente atendiendo al correcto uso de las técnicas de programación orientada a objetos.
- Lógica y algoritmia empleada.
- La complejidad del motor de Inteligencia Artificial si la hubiera
- La jugabilidad, originalidad y lo bien terminado que se encuentre el trabajo (interfaz gráfica, interacción del jugador, ...).
- La defensa y exposición del trabajo.
- Documentación interna del código
- Memoria del trabajo
- Interacción con el sistema de control de versiones Git (número de commits, uso de ramas de desarrollo, etc.)
- Desacople entre el motor gráfico (librería glut, SDL, etc.) y la parte lógica (las estructuras de datos en memoria que codifican el juego y sus reglas). Para el desacople entre las estructuras de datos y el motor gráfico se puede consultar el siguiente enlace (<https://moodle.upm.es/titulaciones/oficiales/mod/url/view.php?id=879617>).

Se valorará también el uso de herramientas adicionales para el desarrollo de software (CMake, gestión de dependencias binarias, etc.). Atendiendo a la autoevaluación y a lo apreciado por el tribunal éste podrá modificar la nota individualmente de los miembros del equipo. El tribunal puede reducir al mínimo su calificación e incluso podrá considerar a alguien particularmente NO APTO dentro de un equipo.

Además se procederá a establecer un **ranking de trabajos en cada convocatoria**. Se asignará hasta un punto adicional absoluto (se suma a las medias una vez que se cumplen los criterios mínimos de evaluación) a los componentes de los 10 mejores trabajos (1 punto al mejor, y 0.9 al segundo mejor, y así sucesivamente hasta el décimo que se premiará con una décima).

Al final cada alumno contará con una calificación individual que se incluirá como nota de trabajo según la fórmula siguiente:

$$NOTA\ TRABAJO = NotaTrabajo\ (0-10) + PuntuaciónAdicional(0-1)*10/3$$

O sea que un trabajo de 10, que sea el mejor de los presentados podrá sacar como máximo un 13.3 que se usará para obtener la nota final con el 30% correspondiente. Nótese que esta nota se guardará para convocatorias futuras con el punto adicional integrado.

NOTA ACLARATORIA: En un equipo que funcione como tal y en el que todos hayan contribuido, aun habiendo quien sabe más o ha ejercido de líder, lo normal es que todos compartan la misma calificación.

Temática

El trabajo consistirá en **un juego realizado en C++ con parte estratégica y parte gráfica**. La perspectiva del juego podrá ser en vista aérea. Podrán ser tanto en 3D como en 2D. Deberá incluir una pantalla de introducción, ranking (consultable), sistema de puntuación, menús, etc.

El trabajo se realizará sobre Visual Studio 2019. Se admite el uso de CMake y Conan para la gestión del proyecto y de las dependencias, aunque no es para nada obligatorio ni necesario.

Se permite el uso de librerías de apoyo para la solución de aspectos concretos, pero no se permite el uso de frameworks completos de juegos (por ejemplo, no se puede usar UNITY o el motor Unreal, ni soluciones análogas). En cualquier caso es importante consultar con los profesores de la asignatura las posibilidades o no de utilizar librerías concretas.

Este año la temática será uniforme y consistirá en el desarrollo del juego del ajedrez.



REQUISITOS MINIMOS

- Deberán implementarse al menos 4 tipos de fichas, con sus reglas de movimiento correspondientes
- El juego deberá tener dos agentes -jugador vs. Máquina y/o jugador vs. jugador- que se alternarán durante el juego, representados claramente de forma diferente (color o tipo de personajes)
- Representación gráfica (2d o 3d) e interactividad con el juego, bien por ratón como por teclado
- Deberá detectarse la posición de finalización de juego, por lo que una de las piezas a implementar tendrá que ser el Rey
- Uso de una estructura Orientada a Objetos. Las piezas deberán implementarse como objetos que puede ser conveniente que sigan una jerarquía de clases.
- Incluir algún aspecto original en el desarrollo. Por ejemplo piezas temáticas ETSIDI, películas (ogros contra elfos), modos de realizarse los movimientos, temática en torno al juego, etc.

REQUISITOS OPCIONALES

- Inclusión de una IA
- Se valorará particularmente el nivel de desarrollo de la IA en caso de incluirse
- Inclusión del juego completo, incluidas las reglas más excepcionales
- Menú inicial del juego
- Pantalla final de juego
- Gestión de rankings con jugadores, puntuaciones (tiempos) , históricos, etc. que se guarden de forma persistente (en un fichero por ejemplo)
- Posibilidad de guardar una partida y cargarla
- Variabilidad en la representación del tablero, el entorno y las fichas
- Realización de una temática original y en particular relacionada con la ETSIDI o la UPM.
- Inclusión de sonidos
- Cambios de vista 3d/2d en función de quien juega
- Interactividad por medio del ratón (selección de piezas y movimientos)
- Dibujo traslúcido o alámbrico de las piezas durante el proceso de decisión de movimiento. El juego marca gráficamente las alternativas de esta forma u otra gráfica como ayuda al jugador.
- Reproducción animada de una partida ya realizada
- Animación de los movimientos de forma que hagan desplazamientos continuos.
- Efectos especiales o animaciones en las piezas cuando se mueven , cuando ganan y cuando pierden.
- Inclusión de sonidos ambientales y circunstanciales
- Temporizadores y puntuación.
- Incluir una inteligencia que sugiera movimientos al jugador
- Posibilidad de jugar con dos instancias del juego a través de una comunicación por la red local