

ENSEÑANZA DE PROGRAMACION ORIENTADA A OBJETOS MEDIANTE EL DESARROLLO DE APLICACIONES GRAFICAS INTERACTIVAS

D. RODRÍGUEZ LOSADA¹, M. HERNANDO¹, R. GONZÁLEZ¹, C. PLATERO², L. DÁVILA¹, P.L. CASTEDO¹, P. SAN SEGUNDO¹, S. LÓPEZ¹, C. SANTOS¹

¹Grupo de Sistemas Telemáticos Aplicados a la Educación (GSITAE).

^{1,2}Departamento de Electrónica, Automática e Informática Industrial. Escuela Universitaria de Ingeniería Técnica Industrial. Universidad Politécnica de Madrid. España.

Contacto: diego.rlosada@upm.es

En la asignatura de Informática Industrial de la EUITI-UPM se ha optado por desarrollar un método docente basado en el diseño y desarrollo de una aplicación gráfica interactiva como son los videojuegos, los simuladores o herramientas de ingeniería. La enseñanza de la Programación Orientada a Objetos se ve facilitada en gran medida por un enfoque unificado de la asignatura que engloba desde las practicas realizadas en laboratorio a las tutorías, el desarrollo de trabajos personales por los alumnos, la docencia teórica e incluso el material docente. Se cree a raíz de los buenos resultados obtenidos, que la motivación e interés que despierta este enfoque en el alumnado redundan claramente en una mejora de la docencia y en los resultados académicos.

1. Introducción

La asignatura de Informática Industrial forma parte del actual Plan de Estudios de la Escuela Universitaria de Ingeniería Técnica Industrial (E.U.I.T.I.) de la Universidad Politécnica de Madrid (U.P.M). Se imparte en la titulación de Ingeniero Técnico Industrial – Especialidad en Electrónica Industrial durante el segundo curso (cuarto cuatrimestre), siendo el responsable de su docencia el Departamento de Electrónica, Automática e Informática Industrial [1] de la U.P.M. La asignatura tiene asignada una carga docente de 10.5 créditos totales distribuidos en 7.5 créditos teórico-prácticos correspondientes a 5 horas semanales de clase en aula y 3 créditos de prácticas que equivalen a una sesión semanal de 2 horas, impartida en el Laboratorio de Informática de dicho departamento. Es una asignatura de carácter obligatorio, por lo que deben cursarla todos los alumnos de la titulación (110 alumnos/curso aprox.). Los alumnos son evaluados en un examen teórico-práctico al final del cuatrimestre, así como en las sesiones de Laboratorio durante el transcurso del mismo. Además, se les propone la realización de un trabajo en grupo de carácter obligatorio cuyo peso es significativo, y acorde al esfuerzo y dedicación empleados, en la calificación global de la asignatura.

En cuanto a contenidos programáticos, enlaza en su comienzo con la asignatura de primer curso (segundo cuatrimestre) de Fundamentos de Informática, donde se propone un recorrido bastante completo por la Programación Procedural, utilizando como sintaxis de programación el Lenguaje C. Se recorren todos y cada uno de los aspectos básicos de la programación, preparando al alumno para poder afrontar con garantías de éxito suficientes la siguiente etapa centrada en la asignatura objeto del trabajo.

Una vez cubiertos los aspectos de la programación procedural, se afronta en la asignatura de Informática Industrial, el paradigma de la Programación Orientada a Objetos y el Modelado Software (Proceso Unificado), así como, también se proporciona una perspectiva bastante amplia de las diferentes herramientas CASE utilizadas en las etapas de desarrollo. Contenidos todos ellos imprescindibles hoy en día, en la actual concepción de la Ingeniería del Software para el desarrollo metodológico de aplicaciones, no sólo en el entorno industrial, sino en cualquiera de las áreas o disciplinas de la industria, gestión o comercio.

Los alumnos, una vez cursada la asignatura de Informática Industrial, están capacitados para afrontar (aquellos que opten por ello) la asignatura de Ingeniería del Software. Incluida también en el plan de estudios para aquellos alumnos que decidan cursar el bloque de optativas de tercer curso (sexto cuatrimestre) que lleva por denominación Automatización Industrial y Avanzada. En la que recibirán una amplia visión de aspectos avanzados de programación, como desarrollo de aplicaciones en Tiempo Real, programación multihilo, etc.

Aún siendo una asignatura enmarcada dentro del área de Informática, como se deduce claramente de la ubicación e interrelación con las asignaturas mencionadas, no es menos cierto comentar, que la cohesión y protagonismo con otras asignaturas, como el Control de Procesos, la Instrumentación Electrónica, la Robótica y Visión Artificial, etc., es claramente manifiesta. El empleo de los modelos es una característica generalizada en cualquier disciplina técnica. El uso de modelos en la concepción del software, aporta una visión metodológica y pragmática muy eficiente para los futuros ingenieros técnicos industriales. La implementación de algoritmos está estrechamente ligada a disciplinas como el Álgebra, el Cálculo y la Física, por lo que alumno es capaz de percibir puntos de enlace con dichas materias desde los diferentes paradigmas estudiados o tratados. La abstracción, el encapsulamiento, la modularidad o la herencia son conceptos y herramientas generalizables y utilizables en otras áreas de concepto, facilitando de esta manera, la asimilación de conocimiento y el asentamiento de conceptos como base de la formación del Ingeniero Técnico Industrial.

2. Estado del arte

Hoy en día prácticamente nadie se cuestiona el uso del ordenador como una herramienta eficaz para el aprendizaje. Multitud de estudios pedagógicos confirman que los sistemas interactivos basados en computador facilitan los procesos de asimilación de conceptos en multitud de disciplinas como Historia, Geografía, Matemáticas, Física, Idiomas, etc. Y siendo una herramienta, es decir, un útil, un medio, para conseguir un objetivo, desde el departamento se ha planteado el reto de cómo conseguir motivar a los alumnos para que se “enganchen” en la disciplina de la asignatura. En este caso, se trata de incentivar la práctica mediante la propuesta del desarrollo de aplicaciones que motiven al alumno. En algunos casos, la algorítmica es por sí sola un canal conductor para conseguir que asuman objetivos y retos para superar la funcionalidad planteada. Sin embargo, la posibilidad de interactuar de forma gráfica por ejemplo, añade una componente visual incrementando los recursos y destrezas a utilizar. Ante estas premisas, se considera que la realización de un trabajo que cubra los objetivos de desarrollo algorítmico, gráfico y multimedia en general, es un buen reclamo motivador, que cumple a su vez, los objetivos docentes necesarios para cubrir todos y cada uno de los contenidos de la asignatura.

De entre las posibles alternativas, que cubren estos requisitos, se encuentran los videojuegos, simuladores y herramientas gráficas de ingeniería. Todos ellos, son desarrollos que permiten ejercitar casi sin límite a priori, salvo los que el profesor y el alumno se fijen, todos y cada uno de los aspectos anteriormente mencionados.

Los videojuegos poseen una amplia expansión y predominio en el ámbito doméstico. Con la profusión de los PCs y las consolas dedicadas, su difusión es ampliamente conocida por los alumnos, llegando a considerarse un elemento familiar o cercano. El poder desarrollar tu propio videojuego, es de por sí, un factor de partida muy motivador en la mayoría de los casos. Este tipo de aplicaciones se prestan fácilmente al planteamiento de retos que involucran la participación de varios individuos, por lo que se fomenta, el trabajo en grupo, factor este último tremendamente positivo de cara al futuro o actual desempeño profesional del alumno, cada vez más orientado hacia el trabajo en grupos multidisciplinares. Como alternativa a los videojuegos, para abrir en cierta medida, el abanico de posibilidades, también se plantean aplicaciones de tipo simulador, donde también se pueden ejercitar de manera amplia la algorítmica y el resto de disciplinas anteriormente mencionadas. De hecho, muchos de los videojuegos que pueden plantearse, están clara y directamente relacionados con la simulación de comportamiento de sistemas físicos, donde se pone a prueba la destreza del usuario o se pretende el entrenamiento y aprendizaje del sistema en cuestión (automóviles, aviones, barcos, submarinos, etc.).

Existen ya algunas experiencias en una línea similar, donde también se les propone a los alumnos la realización de trabajos basados en aplicaciones de entretenimiento como en la Universidad Complutense [1] o la Universidad de Alcalá de Henares [2], ambas de Madrid. No obstante, en estos casos no se centran en la Programación Orientada a Objetos (POO), sino en los aspectos gráficos o algorítmicos.

3. Método docente

Como se ha descrito en el apartado primero, la asignatura de Informática Industrial se centra en la enseñanza del paradigma orientado a objetos, o Análisis, Diseño y Programación Orientada a Objetos. Si bien es cierto que el vehículo o lenguaje de programación es C++, el aprendizaje de las reglas de sintaxis de dicho lenguaje es relativa o comparativamente sencillo, contrariamente a lo que sucede con el Análisis y Diseño Orientado a Objetos. Enseñar o transmitir esos conceptos supone comunicar una filosofía, una forma de trabajar, con un nivel de conceptualización y abstracción muy elevado, lo que presenta una curva de aprendizaje mucho más dura para los alumnos. Se podría decir que la parte crítica de la asignatura es precisamente esta. El resto de la asignatura se articula en torno a este concepto.

El desarrollo de videojuegos presenta un gran atractivo para los alumnos, a la vez que esta relacionado con otras muchas disciplinas, tal y como se comentaba en el apartado anterior. Pero el desarrollo de juegos o simuladores presenta una ventaja añadida importante: al ser representado un mundo físico de forma virtual en el ordenador, hay un alto paralelismo entre los objetos reales del mundo y los objetos software. Cuanto más cercana sea la representación hecha en código al mundo real, más fácil es de entender por los alumnos. En el Análisis y Diseño Orientado a Objetos, cuando las clases del modelo del dominio se trasladan casi directamente al diseño, se dice que hay una “reducción en el salto de la representación”. Estas ideas aplican principalmente a juegos y simuladores con gran componente de simulación física, con objetos físicos que se mueven, con cinemática, dinámica, etc. Sin embargo, los juegos lógicos como las 4 en raya, el ajedrez, las damas, etc. presentan poco modelado físico pero una gran carga de lógica y algorítmica.

Inicialmente se comenzó en la asignatura a plantear trabajos de diversa índole, así como unas prácticas genéricas de POO. Cuando se comenzó a proponer algunos trabajos tipo videojuego o simulador, se vio el potencial de esta idea como metodología de enseñanza. Actualmente, gran parte de la arquitectura se estructura en torno al desarrollo de estas aplicaciones como caso de estudio:

- El trabajo obligatorio de la asignatura es realizado por parejas, consistiendo en el mayor número de casos en un juego típico de computador, del que los alumnos deben entregar al

final del cuatrimestre el código y toda la documentación y diagramas UML generados. Se podría decir que el desarrollo de este trabajo es el motor principal de motivación de los alumnos, y centraliza la mayor parte de su dedicación durante el curso. La gran ventaja es que este desarrollo les obliga a aprovechar las practicas de la asignatura y a revisar constantemente los contenidos teóricos de la misma, por lo que cuando acaban el trabajo también tienen gran parte de la asignatura ya asimilada.

- Las prácticas de la asignatura han sido rediseñadas completamente, tomando como línea conductora desde la primera a la última practica el desarrollo de un videojuego típico. Las prácticas van introduciendo y explicando los conceptos y aplicándolos a la aplicación. El videojuego seleccionado como temática en las prácticas es el juego del Pang, en el que un muñeco se mueve en la parte inferior de una caja en la que rebotan pelotas. Si una pelota impacta en el jugador, este pierde la partida. El jugador puede disparar a las pelotas, que son partidas en dos o desintegradas en caso de que sean muy pequeñas.

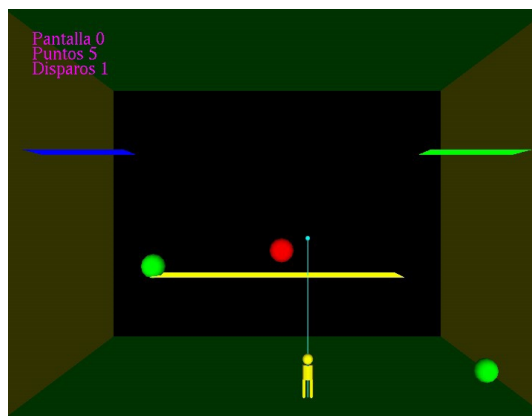


Figura 1. Juego del Pang

- En la página Web de la asignatura se cuelgan actualmente algunos manuales de OpenGL y otros recursos relacionados.
- En la docencia teórica, sobre todo en la parte relacionada con ADOO se explican también ejemplos sencillos relacionados con los videojuegos, se desarrollan y escriben artefactos típicos, casos de uso, modelos del dominio, diseño, uso de patrones, etc. De hecho, tal y como se explica posteriormente, hasta en los exámenes teóricos escritos de la asignatura, aparecen enunciados muy relacionados.
- El desarrollo de estos trabajos requiere de un soporte tutorial muy importante, pero altamente productivo. En las tutorías el profesor dirige y supervisa la ingeniería software del proyecto, así como ayuda a solucionar problemas de implementación.

Se ha optado por utilizar OpenGL como librería grafica ya que esta constituye un estándar de la industria de gráficos por computador. Además, estas librerías encuentran implementaciones compatibles en prácticamente todos los sistemas existentes (tanto hardware, como software y sistemas operativos). Para la creación de ventanas de forma sencilla y gestionar los eventos de teclado, temporización, etc., se

ha optado por utilizar el gestor GLUT, que a su vez también es 100% portable a otros sistemas, como Linux.

Aunque el lenguaje de programación podría ser Java por ejemplo (que de hecho es más orientado a objetos que C++), se desarrolla la asignatura en C++. Esto se debe a que la titulación es ingeniería industrial, no informática. En el entorno industrial, el código existente es en su mayoría C y C++, ya que Java ha presentado tradicionalmente el inconveniente de una menor eficiencia debido a la máquina virtual que interpreta el bytecode.

4. Prácticas de la asignatura

Según se ha expuesto en el apartado tercero, la asignatura se ha diseñado de forma que los conocimientos adquiridos puedan ser llevados a la práctica progresivamente. Con esta perspectiva se ha diseñado un conjunto de diez prácticas que incrementalmente van ilustrando los conceptos expuestos en las clases teóricas a la vez que dotan de herramientas para el desarrollo del trabajo de curso que consiste en un juego de ordenador. El conjunto de prácticas, así como una breve descripción de los objetivos docentes se describe a continuación

1. Programación en C: Uso del depurador.
Familiarización con MSVS6.0 y las funciones de depuración. Operaciones con vectores.
2. Programación en C: Uso de herramientas.
Introducción a la potencia de la POO. Uso de objetos. MFCs y ActiveX.
3. Gráficos tridimensionales con OpenGL.
Introducción a OpenGL. Uso de las librerías Glu y Glut. Estructuración de la aplicación.
4. Concepto de Clase en C++.
Diseño e implementación de las primeras clases. La clase Esfera y la clase Plano.
5. Ingeniería directa e inversa con *Rational Rose* y *Visual C++*.
Relación entre diseño e implementación. Diagramas de clases y de Secuencia. Añadir a la aplicación disparos, el jugador, bonus, etc.
6. Creación y destrucción de objetos. Interacciones entre objetos
Contenedores de objetos, vectores que contienen Esferas. Patrones para implementar interacciones entre clases. Rebotes entre esferas y esferas con planos.
7. Concepto de herencia.
Reutilización de código. Clase base para los objetos gráficos. Nuevos objetos derivados.
8. Polimorfismo y herencia.
Clases especializadas con interfaces comunes. Copia polimórfica. Implementación de una máquina de estados. Fin del Juego.
9. Diseño con patrones
Revisión de todo el código comprobando la aplicación de diversos patrones: Creador, Bajo Acoplamiento, Experto en Información, etc.
10. Examen de Laboratorio.

La última práctica es una práctica de examen, que consiste en la realización de un sencillo programa con filosofía de POO. Con las notas de las distintas prácticas y la del examen, se obtiene la calificación del

laboratorio que constituye un 25% de la nota final de la asignatura. Para aprobar las prácticas es obligatorio aprobar el examen práctico independientemente de las notas obtenidas durante el curso.

Las prácticas de laboratorio están constituidas por sesiones de 2 horas semanales en un aula habilitada con 22 puestos, de los cuales uno es el del profesor que tiene la particularidad de que lo que se muestra en el monitor es proyectado sobre una pantalla para que así los alumnos puedan seguir con facilidad las explicaciones. En los dos últimos años, cada práctica de laboratorio ha tenido una media de 12 estudiantes, siendo el total de grupos 7. (84 alumnos).

La estructura que se sigue en cada práctica es la siguiente:

Durante los 15 primeros minutos el profesor corrige los ejercicios propuestos en la práctica anterior. Estos son variaciones que debe realizar el alumno sobre el programa final generado tras la última sesión, en la que debe utilizar los conceptos que entonces se explicaron. Al ser un programa, su corrección es rápida, dado que es fácil verificar por el resultado que la programación se ha realizado. Análogamente, al dar libertad al alumno en cuanto al resultado más accidental, se obliga a que en cierta medida las soluciones sean originales. En cualquier caso, siempre se realiza una inspección rápida del código para verificar que se ha seguido la filosofía de POO y se evalúa el trabajo realizado por el alumno.

Finalizada la corrección el profesor invierte entre 5 y diez minutos en explicar el marco en el que desarrolla la práctica, y una vez situados los estudiantes, se van siguiendo paso a paso los puntos del guión. En estos se intercalan ejercicios guiados totalmente por el profesor, con soluciones que debe programar el alumno aplicando lo que se ha ido viendo.

Por último, la práctica termina con un programa más evolucionado que sirve de base para la realización de los ejercicios de esa práctica, y que a su vez serán la plataforma sobre la que se comience la práctica siguiente.

5. Docencia teórica

Las aplicaciones gráficas, como los videojuegos, son un buen hilo conductor para la enseñanza de la POO. De un lado, tiene la propia motivación para los estudiantes al diseñar un programa con mucha realimentación, ya que los resultados son palpables y la mayoría han sido usuarios de estos paquetes informáticos. Por otro, estas aplicaciones requieren el uso de software prefabricado, del manejo de conceptos abstractos y un conjunto de iteraciones dentro del marco del Proceso Unificado. En estas aplicaciones se exige emplear muchos de los artefactos colocados por el Proceso Unificado: Casos de Uso, Modelo del Dominio, Diagramas de Clases de Diseño, Modelo de la Implementación e Ingeniería Inversa. Pero sobre todo del diseño mediante patrones.

Por otro lado, el paradigma de la programación orientada a objetos y en concreto con su implementación en C++, permite abordar el juego de una forma ordenada y exponiendo durante su desarrollo los problemas más habituales que se encuentran durante el desarrollo de un programa en este lenguaje. Cabe destacar como se orienta al alumno para respetar el encapsulamiento de clases, mostrando la importancia de cuidar los procesos de construcción y destrucción de objetos. De igual forma se insiste en el ahorro de código utilizando para ello la herencia que nos da el lenguaje y la capacidad polimórfica de los métodos virtuales. Se insiste en el uso de contenedores de objetos que permitan simplificar al máximo el contenido de las funciones principales de la aplicación, y se enseñan los distintos procedimientos que existen para implementar interacciones entre objetos de nivel de abstracción similar. C++ ha demostrado

ser una herramienta asequible para el alumno en el conjunto del curso que se expone, y prueba de ello son los programas reflejo de una fase de diseño que se entregan al final de curso.

Sirva como ejemplo, el ejercicio propuesto en el examen de septiembre de 2007. Se pretende desarrollar un juego basado en la batalla entre hombres y dragones. Los hombres lanzan cuchillos y los dragones bolas de fuego. Los dragones se mueven en el área izquierda de la pantalla y los hombres en el lado derecho. En mitad de la batalla aparecen paredes móviles que se desplazan en el centro de la pantalla. El número de luchadores puede ser variable y dinámico. El planteamiento de solución consiste que el alumno se guíe empleando el Proceso Unificado, UP. En primer lugar se pide que extraiga las características principales de la aplicación empleando el artefacto de jerarquía a dos niveles:

1. Video juego de los hombres que luchan contra los dragones

- 1.1. Los hombres se mueven en un área restringida de la derecha.
- 1.2. Los dragones se mueven en un área restringida de la izquierda.
- 1.3. Los hombres lanzan cuchillos que se clavan en la pared o que matan al dragón o que pasan sin hacer daño.
- 1.4. Los dragones lanzan bolas de fuego que no pueden atravesar las paredes y que si tocan a un hombre lo mata.
- 1.5. Los dragones desaparecen de la pantalla al morir.
- 1.6. Los hombres desaparecen de la pantalla al morir.

En segundo lugar, se pide el Modelo del Dominio del juego. El alumno debe de aplicar las técnicas de detección de conceptos (test lingüísticos y listas de categorías conceptuales) y sus asociaciones (“necesito conocer” y tablas de asociaciones). También se da la metodología para su transformación en un diagrama de UML.

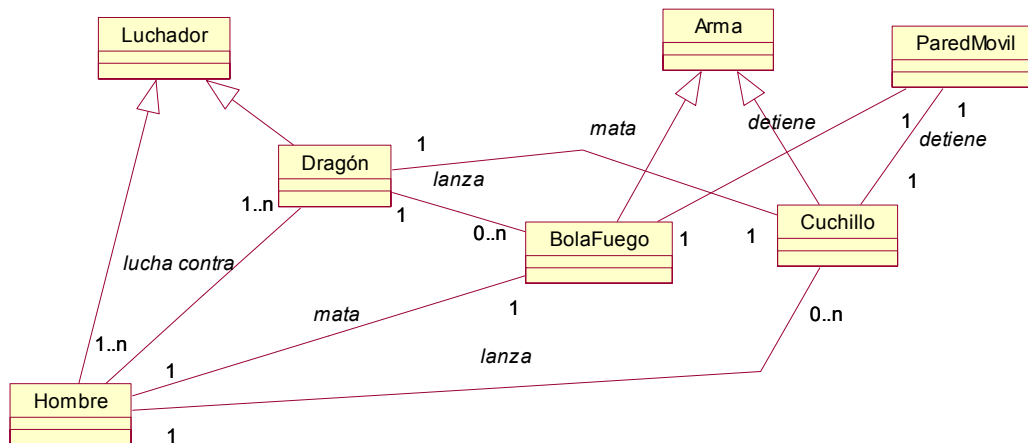


Figura 2. Modelo del Dominio en UML

Más de la mitad de los créditos de AOO/D se dedican al diseño mediante patrones GRASP y GOF. Este ejercicio es un buen ejemplo de la aplicación de patrones Singleton (GoF), Factoria Abstracta (GoF)

y Métodos de Fabricación (GoF), los cuales incluyen las buenas prácticas de Polimorfismo (GRASP), Fabricación Pura (GRASP) y Variaciones Protegidas (GRASP).

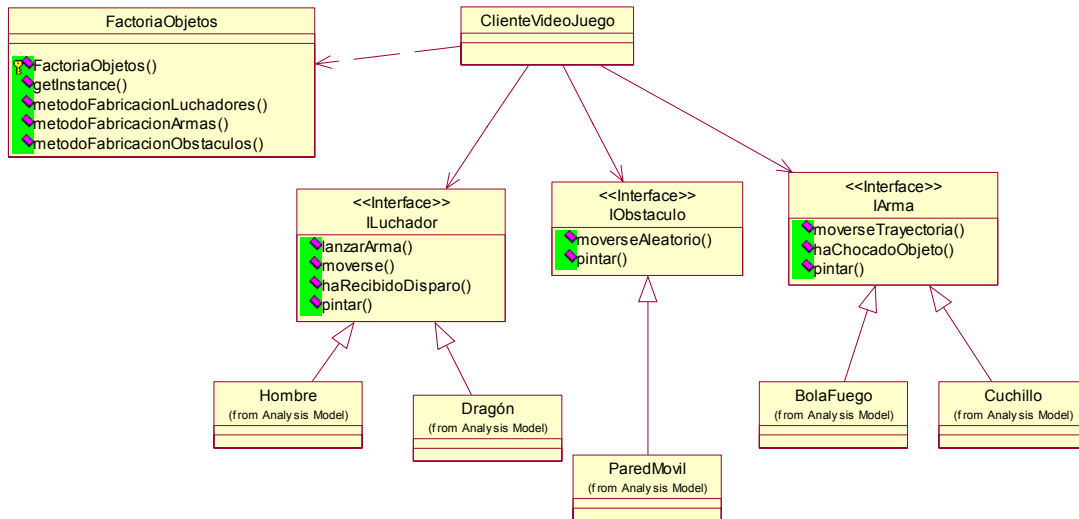


Figura 3. Diagrama UML de diseño. Solución de examen de teoría.

El último apartado conecta con la programación en C++, al transformar el modelo de clases de diseño en código. En el ejercicio sólo se pide los ficheros de cabecera.

6. Tutorías

En la acción tutorial, la mayoría del tiempo se dedica a perfilar los trabajos de curso. Siguiendo el Proceso Unificado y dado el pliego de condiciones de la aplicación a cada grupo, las tres primeras semanas se dedican a qué lo formalicen empleando los documentos de Visión y Alcance, Casos de Uso, Glosario y Especificaciones Complementarias. A tal propósito se les da unas plantillas para escribir dichos informes. En el mes siguiente se procede al análisis empleando el artefacto del Modelo del Dominio. También se les pide que empiecen a manejarse con las librerías gráficas, las cuales se han mostrado en las prácticas de la asignatura. A partir del segundo mes, se le exige los artefactos para el diseño: los diagramas de clases de diseño y los diagramas de secuencia. Es aquí donde las tutorías entran con la inserción de los patrones que se están explicándose en las clases de teoría y en las prácticas. Al principio sólo se les pide Bajo Acoplamiento, Alta Cohesión y Experto en Información, con el objeto de definir los paquetes y en la organización de la producción del software. Más adelante, se les anima a que desarrolle una única factoría de objetos y a un diseño en Variaciones Protegidas.

7. Trabajos realizados

Existen trabajos cuyo enunciado viene dado por el profesor, y otros que puede proponer el alumno libremente. Entre las temáticas seleccionadas destacan videojuegos (vease figura siguiente) como los clásicos de plataformas, en el que el jugador va saltando de plataforma en plataforma librándose de sus enemigos y recolectando objetos, los deportivos como el golf o las carreras de motos o coches, lúdicos como el pinball o los de acción denominados de primera persona.

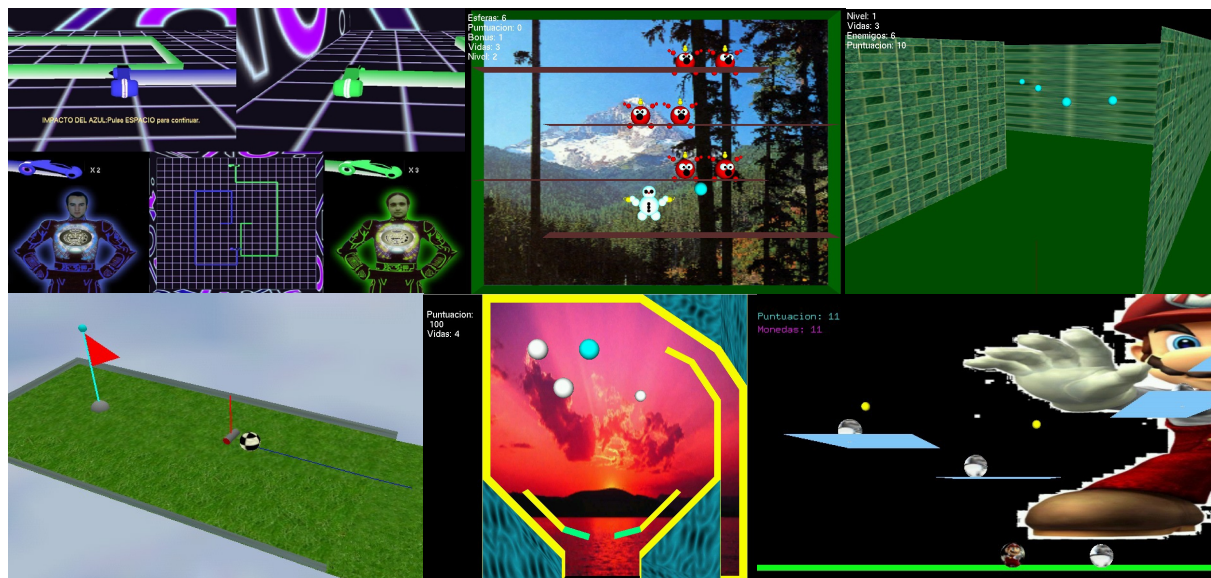


Figura 4. Algunos juegos desarrollados por alumnos del curso 06/07.

8. Resultados y conclusiones

La metodología propuesta ha sido aceptada con gran interés por parte del alumnado, despertando una alta motivación en un sector importante de ellos. Cabe resaltar como muchos continúan desarrollando su videojuego una vez superada la asignatura, y que hasta hacen partícipes a sus familias, que utilizan de “probadores” de sus videojuegos.

Por otra parte cabe destacar que la dedicación medida en tiempo empleado en la realización del trabajo incrementa ostensiblemente debido principalmente a la alta motivación. Los profesores de la asignatura estiman subjetivamente que el tiempo empleado en un trabajo por un alumno motivado puede aumentar entre un 25% y un 100% en algunos casos, respecto al mismo alumno frente a un trabajo diferente.

Esta dedicación se traduce a la larga en un incremento de conocimientos y una mejor asimilación de la asignatura, lo que redundará en unas mejores calificaciones en la asignatura, no solo por la nota de los trabajos que suele ser elevada, sino por el aprendizaje realizado durante el desarrollo del mismo.

Existe una unanimidad absoluta respecto al éxito del enfoque adoptado, lo que nos lleva a seguir trabajando con iniciativas similares, tal y como se describe en el punto siguiente.

9. Futuros trabajos

Como futuros trabajos en esta línea se pretende crear una página Web específica para esta temática, en la que se puedan centralizar recursos comunes, preguntados y solicitados muy a menudo por los alumnos. Así, se pretende incluir pequeños tutoriales que permitan lo más fácilmente posible al alumno tareas como incluir gráficos avanzados en su aplicación, como el uso de texturas para crear escenas más realistas, efectos de materiales y luces. También se incluirán otros temas como el uso de música y efectos

sonoros, fundamentos físicos y matemáticos para simulaciones avanzadas, con dinámica, explosiones, etc., y movimientos especiales de cámara. Aunque en general estos elementos escapan del alcance del curso, la afición generada en una gran parte de los estudiantes les lleva a seguir desarrollando su juego hasta darle un acabado prácticamente comercial, incluso una vez terminado y superado el curso.

Esta página Web también servirá para que los alumnos puedan colgar y exhibir sus aplicaciones o videojuegos, de tal forma que se puede establecer un ranking de los mismos por votaciones u algún otro sistema como la cuenta del número de descargas.

La metodología del desarrollo de los videojuegos ha empezado a ser también utilizada en la asignatura optativa denominada ingeniería del Software de último cuatrimestre de la titulación de Electrónica de la EUITI-UPM, en la que se tratan temas de programación concurrente y sistemas distribuidos. En dicha asignatura se dejaba optar a los alumnos entre trabajos convencionales (como un servidor concurrente) y el desarrollo o adaptación de un juego a un juego en red. De tal forma, los alumnos podían coger su juego del curso anterior y convertirlo en una aplicación distribuida en red, con los servidores y clientes de sockets adecuados, programas concurrentes multihilo para gestionar comunicaciones y procesamiento del juego, etc. Los resultados iniciales han sido excepcionales y se pretende (en los próximos 2 cursos) abordar la metodología expuesta en este artículo a la asignatura completa: rediseñar las practicas enfocándolas a dicho objetivo, actualizar los temarios teóricos, adaptando los ejemplos respectivos a un caso de estudio típico.

Agradecimientos

Los autores agradecen el soporte del departamento ELAI-UPM en la realización de todas las actividades descritas, así como la financiación recibida por el grupo de innovación educativa GSITAE por parte de la EUITI y de la UPM

Referencias

- [1] www.elai.upm.es Pagina web del departamento de Electronica, Automatica e Informatica (ELAI) de la Universidad Politecnica de Madrid (UPM)
- [2] Platero C., *Apuntes de Informática Industrial (Análisis y Diseño Orientado a Objetos)*, Servicio de Publicaciones EUITI-UPM, 2006.
- [3] Larman, C., *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*, Segunda edición, Pearson Educación 2002.
- [4] Hernando, M. *Programación C++*. Servicio de Publicaciones EUITI-UPM, 2005.
- [5] Desarrollo de juegos en clase. Universidad Complutense:
http://www.fdi.ucm.es/Guia_Docente/ver_prog_asig.asp?Titulo=450&Asignatura=525&fdicurso=2007-2008
- [6] Desarrollo de juegos en clase. Universidad de Alcalá:
<http://www.etsii.uah.es/Estudios/pdf/guia-docenteII06-07.pdf>