

28. HERRAMIENTAS LIBRES PARA GRÁFICOS Y SISTEMAS INTERACTIVOS: EL CASO CONCRETO DE LA ASIGNATURA JUEGOS Y REALIDAD VIRTUAL

F. Aznar Gregori, M. Pujol López,

R. Rizo Aldeguer, M. Sempere Tortosa, C. Villagrà Arnedo

*Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante*

RESUMEN

Seleccionar las herramientas docentes para las sesiones prácticas de una determinada asignatura no es tarea fácil. Por ejemplo estas deben ser suficientemente potentes para poder desarrollar los objetivos del curso, pero además no deben de ser excesivamente complejas ni requerir mucho tiempo de aprendizaje por parte de los alumnos.

En este artículo deseamos plasmar nuestra experiencia desde el curso 04/05 en la utilización de software libre para la docencia de la asignatura de Juegos y Realidad Virtual de las carreras de Ingeniería Informática de la Universidad de Alicante. Esta puede extrapolarse directamente a cualquier asignatura que esté relacionada con los gráficos por computador o sistemas interactivos.

1. INTRODUCCIÓN

El conjunto de las asignaturas de gráficos por computador (y muy especialmente la asignatura Juegos y Realidad Virtual) disfrutan de una base eminentemente práctica. Aunque existen diversas alternativas para orientar su docencia nosotros consideramos que el aprendizaje basado en proyectos es una muy buena aproximación para fomentar el aprendizaje de los alumnos. Es por esta razón que las sesiones prácticas son fundamentales y por tanto la elección del software a utilizar no es tarea fácil. Por una parte ha de ser suficientemente potente para poder desarrollar los objetivos del curso, pero además no debe de ser excesivamente complejo ni requerir mucho tiempo de aprendizaje por parte de los alumnos. También es deseable que el software propuesto se esté utilizando en el mundo empresarial proporcionando de esta manera una transición más gradual al mundo laboral, aunque no hay que dejar de lado que la universidad no puede estar supeditada al mundo de la empresa. Por último la utilización de software libre proporciona diversas ventajas respecto al software privativo y además se acerca mucho más a la filosofía de la universidad. Obviamente es complejo ajustar todos estos parámetros y más aun encontrar las aplicaciones que cumplan todas estas características.

En este artículo deseamos plasmar nuestra experiencia en la utilización de software docente para la asignatura de Juegos y Realidad Virtual. Esta puede extrapolarse a cualquier asignatura que esté relacionada con los gráficos por computador o sistemas interactivos. Deseamos mostrar por tanto las aplicaciones libres que hemos seleccionado para esta tarea, justificando en todo momento su elección y comparándolas con las herramientas más comunes destacando sus ventajas y desventajas.

Inicialmente comentaremos muy brevemente la asignatura de Juegos y Realidad Virtual de la Universidad de Alicante situándola en el plan de estudios de Ingeniería Informática. A continuación reflexionaremos sobre el aprendizaje basado en proyectos (motivo principal que nos ha llevado a realizar este análisis), siguiendo por la exposición de las herramientas software seleccionadas. Por último finalizaremos este artículo comentando las conclusiones de este trabajo.

2. SISTEMAS GRÁFICOS E INTERACTIVOS

Dentro del rápido crecimiento que en los últimos años ha sufrido la informática en general, el caso de los Juegos por Computador (videojuegos) y la Realidad Virtual resulta especialmente llamativo por las repercusiones que ha tenido no solo en el ámbito científico sino también en buena parte de la sociedad. La industria del videojuego ha experimentado en los últimos años altas tasas de crecimiento, debido al desarrollo de la computación, capacidad de procesamiento, imágenes más reales y la estrecha relación entre películas de cine y los

videojuegos. Los videojuegos cuentan con una muy exitosa industria, que en los últimos años ha estado generando más dinero que la del cine. La industria del videojuego representó mundialmente un valor de 27.000 millones de euros en el año 2005.

Tanto los videojuegos como la realidad virtual se presentan en mayor o menor medida en casi todos los campos de la Informática. No obstante su carácter multidisciplinar, que hace que por ejemplo partes de estas materias se vinculen a la Inteligencia Artificial o la Ingeniería del Software, su eje troncal se centra en los métodos de representación gráfica e interfaces (y por tanto con las asignaturas del bloque de especialización de Gráficos por Computador e Interacción Hombre Computador).

La versión disponible más reciente del Computing Curricula [1] nombra como capacidades necesarias para los licenciados e ingenieros en informática explícitamente el producir gráficos o videojuegos, enmarcándolo dentro del área de interacción hombre máquina. No obstante en el [2], la materia Interacción Hombre Máquina hace únicamente hincapié en el desarrollo de interfaces, no quedando claro realmente donde se enmarcan los contenidos relativos a los Juegos por Computador.

Como ya se ha comentado anteriormente aunque la asignatura Juegos y Realidad Virtual disfrute de una base multidisciplinar su eje central se basa en los gráficos por computador. Desde nuestro punto de vista se podría concluir que dentro del grupo de asignaturas sobre conceptos y sistemas gráficos, se pueden distinguir tres niveles de profundidad:

- Introducción a los Gráficos, generalmente pertenecientes a los ciclos primero o segundo, que incluyen los fundamentos de la disciplina. Las asignaturas aparecen con nombres como Gráficos por Ordenador, Introducción a los Gráficos por Ordenador, Gráficos Interactivos, Informática Gráfica, etc.
- Gráficos Avanzados, generalmente de segundo o tercer ciclo, donde se encontraría enmarcada la asignatura de Juegos y Realidad Virtual. Estas incluyen gráficos 3D y algunos conceptos avanzados. Los nombres utilizados son Gráficos por Ordenador, Gráficos Interactivos, Ampliación de Gráficos, Gráficos 3D, Juegos y Realidad Virtual etc.
- Asignaturas especializadas, generalmente de tercer ciclo, cuyos contenidos son específicos y avanzados. Entre ellas destacan Técnicas Avanzadas de Visualización, Visualización, Realismo, etc.

En cuanto a la Universidad de Alicante contamos con tres asignaturas cuatrimestrales. Se recomienda cursar inicialmente Gráficos por Computador:

- Gráficos por Computador, obligatoria en tercer curso de Ingeniería Informática, con 4,5 créditos docentes (2,25 créditos de teoría y 2,25 créditos

de práctica): se corresponde con una asignatura introductoria, perteneciente al primer nivel de profundidad.

- Juegos y Realidad Virtual, optativa en Ingeniería Informática y de las dos ingenierías técnicas (3 créditos de teoría y 3 créditos de prácticas): Es la asignatura objeto de este artículo y puede considerarse como perteneciente al segundo nivel de profundidad.
- Gráficos Avanzados y Animación, optativa en Ingeniería Informática, con 6 créditos docentes (3 créditos de teoría y 3 créditos de práctica).

2.1 Juegos y realidad virtual

La asignatura Juegos y Realidad Virtual forma parte de los planes de estudio de Ingeniería Informática, Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Informática de Sistemas como asignatura optativa que se imparte en el segundo cuatrimestre. Por sus contenidos, se recomienda cursarla en los últimos cursos. Los descriptores de dicha asignatura son: programación gráfica de juegos (efectos, texturas, visualización en tiempo real, multiresolución), modelado de fenómenos naturales, realidad virtual.

Dicha asignatura tiene una relación clara y estrecha con varias asignaturas de cursos anteriores donde se imparten conceptos relacionados, y cuyo entendimiento es necesario para abordar con éxito la materia que nos ocupa. Concretamente dichas asignaturas son: gráficos por computador (forma parte del primer cuatrimestre del tercer curso de Ingeniería Informática como asignatura obligatoria, y se oferta también como asignatura optativa en las Ingenierías Técnicas). Entre sus descriptores se encuentran, transformaciones 2D y 3D, proyecciones y vistas, visualización. Esta asignatura puede considerarse una primera parte de Juegos y Realidad Virtual y de Gráficos Avanzados y Animación y se recomienda a los alumnos que la cursen con anterioridad, ya que presenta los conceptos básicos sobre gráficos.

El contenido de la asignatura queda estructurado en tres grandes bloques. Estos están compuestos tanto por todos los contenidos de carácter teórico, que se impartirán en aula grande en las sesiones teóricas, como por las prácticas que se llevan a cabo en los laboratorios y que implican la utilización de software específico.

Su programa cubre un total de 6 créditos, distribuidos en 3 créditos de teoría y 3 créditos de prácticas. A continuación incluimos como referencia el contenido de cada uno de los bloques temáticos de la asignatura:

- Introducción a los juegos por computador
 - Tema 1. Introducción a los juegos por computador
 - Tema 2. Arquitectura de un juego por ordenador
 - Tema 3. Estructuras de datos y algoritmos
- Programación de juegos por computador
 - Tema 4. Patrones de diseño

- Tema 5. Programación de juegos en 2D
- Tema 6. Inteligencia Artificial en Juegos
- Juegos tridimensionales,
 - Tema 7. Bases de un juego por computador tridimensional
 - Tema 8. Motor gráfico 3D para juegos de interior
 - Tema 9. Motor gráfico 3D para juegos de exterior
- Realidad virtual
 - Tema 10. Introducción a la realidad virtual

3. APRENDIZAJE BASADO EN PROYECTOS

El aprendizaje basado en problemas y su variante aprendizaje basado en proyectos (ABP) es uno de los métodos renovadores del proceso de enseñanza-aprendizaje que más se ha consolidado en las instituciones de educación superior del mundo occidental en los últimos años y que en particular permite una excelente aproximación al replanteamiento de la enseñanza desde la óptica de los créditos ECTS.

El camino que recorre el proceso de aprendizaje convencional se invierte al trabajar en el ABP. Mientras tradicionalmente primero se expone la información y posteriormente se busca su aplicación en la resolución de una situación real, en el caso del ABP primero se presenta el problema, se identifican las necesidades de aprendizaje, se busca la información necesaria y finalmente se regresa al problema. En el recorrido que viven los estudiantes desde el planteamiento original del problema hasta su solución, trabajan de manera cooperativa en pequeños grupos, compartiendo en esa experiencia de aprendizaje la posibilidad de practicar y desarrollar habilidades y competencias genéricas de carácter transversal, y de observar y reflexionar sobre actitudes y valores que en el método convencional expositivo difícilmente podrían ponerse en acción.

La experiencia de trabajo en el pequeño grupo orientado a la solución del problema es una de las características distintivas del ABP. En estas actividades grupales los alumnos toman responsabilidades y acciones que son básicas en su proceso formativo. Por todo lo anterior se considera que esta forma de trabajo representa una alternativa congruente con el modelo del rediseño de la práctica docente necesario en la formación de profesionales, especialmente en el lado de técnicos.

Concretamente en esta asignatura los alumnos plantearán un proyecto relacionado con la asignatura. Los profesores aceptarán o modificarán esta versión inicial del proyecto dándole el visto bueno para su ejecución. Serán los alumnos los que decidirán los distintos hitos de entrega del proyecto que verifican los profesores de la asignatura. El proyecto, que es la parte más importante de la evaluación de la asignatura, se realizará en las horas especificadas por los créditos ECTS de trabajo en casa y se tutorizará en horario de prácticas o tutorías.

Además este proyecto estará coordinado con otro grupo de asignaturas (Gráficos Avanzados y Animación, Razonamiento y Modelos de Fabricación Asistida por Computador) de manera que los alumnos que cursen las cuatro asignaturas realizarán un proyecto común.

Por esta razón es necesario el proporcionar un conjunto de herramientas básicas para que los alumnos puedan realizar el proyecto y además puedan intercambiar información entre ellos de manera sencilla. En este artículo nos centraremos en las aplicaciones que los profesores de la asignatura hemos seleccionado para el desarrollo de un proyecto de gráficos interactivos.

4. ANÁLISIS DE HERRAMIENTAS LIBRES

La finalidad de este análisis de herramientas libres es múltiple, por una parte se trata de analizar aquellas herramientas que consideramos útiles para el desarrollo de las asignatura, fruto de la experiencia y de probar multitud de herramientas distintas. Por otra parte se pretende mostrar como se pueden utilizar herramientas libres para la docencia de una asignatura sin por ello perder ninguna característica importante del software privativo.

En esta sección revisaremos brevemente algunas de las razones que motivaron la elección del software libre sobre el software privativo. A continuación comentaremos el conjunto de herramientas seleccionadas para la asignaturas relacionadas con los sistemas gráficos e interactivos y más concretamente a la asignatura Juegos y Realidad Virtual.

4.1 ¿Por qué software libre?

Como bien se comenta en [3] a pesar de que el software libre está en auge, el desconocimiento de su existencia por parte del gran público limita su expansión. Las universidades son plataformas ideales en las que es posible su utilización y promoción, constituyendo además un entorno particularmente adecuado para el crecimiento de aplicaciones desarrolladas bajo esta filosofía. El estado actual de las aplicaciones existentes en el software libre permite satisfacer en gran medida las necesidades docentes de la mayoría de la comunidad universitaria.

El software libre es aquél que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios; sin embargo, no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente. Así, se considera que el software es libre si garantiza las siguientes cuatro libertades: libertad de ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.); libertad de estudiar y modificar el programa (para lo cual es necesario poder acceder al código fuente); libertad de copiar el programa; y libertad de

mejorar dicho programa y hacer públicas las mejoras, de forma que se beneficie toda la comunidad [6].

Son tres las principales condiciones que la universidad debe exigir al software que utiliza: en primer lugar, debe operar en la lengua propia del lugar en que se va a utilizar (localización); en segundo lugar, debe poder garantizar el acceso a la información en todo momento, en el presente y en el futuro (perenidad); y, en tercer lugar, no ha de permitir que personas no autorizadas tengan acceso a los datos confidenciales de los particulares o a información reservada (seguridad). En definitiva, como indica en [4], el aprendizaje se promueve a partir de un acceso libre a la información. Por esta razón, en la mayoría de países existen escuelas y bibliotecas públicas. Hoy en día, el software no sólo es información en sí mismo, sino que es el principal canal de acceso a la información de cualquier tipo. Además, el software también es un entorno de aprendizaje, en el que se pide a los estudiantes que desarrollen las competencias propias de los estudios que realizan [5].

Existen multitud de artículos que destacan las ventajas de utilizar software libre en la docencia universitaria [3], [4], [5] con lo que no lo vamos a repetir todas aquí. Finalizaremos esta sección recordando las ventajas más elementales antes de adentrarnos en el estudio de las herramientas seleccionadas:

- Distribución y uso. El software libre se puede distribuir a los alumnos. Estos pueden continuar utilizando en casa las mismas herramientas que en las sesiones de prácticas. Además la universidad puede invertir el dinero destinado anteriormente al pago de las licencias de software privativo en, por ejemplo, mejorar las infraestructuras docentes...
- Independencia del proveedor. En relación con la anterior premisa, el software libre garantiza una independencia respecto al proveedor del servicio gracias al acceso garantizado al código fuente.
- Trabajo cooperativo. En el modelo del software libre prima el hecho de compartir información y el trabajo cooperativo. Este modelo es parecido al que solemos utilizar tradicionalmente en el mundo académico y científico.
- Adaptación. Las aplicaciones libres se pueden adaptar a los requerimientos del uso final. Una gran parte de la industria del software se basa, precisamente, en desarrollar proyectos en los que se requiere software personalizado

4.2 Compilador y entorno de desarrollo

Adentrándonos ya en la revisión de las herramientas software, el primer paso a seguir por los profesores de la asignatura fue seleccionar el lenguaje de programación. Para el desarrollo del proyecto se han elegido los dos lenguajes, C/C++, más utilizados para la realización de videojuegos. Su principal ventaja

es la velocidad y el control de la máquina, necesaria muchas veces cuando estamos hablando de aplicaciones en tiempo real (como son los videojuegos). La primera duda con la que nos enfrentamos fue decidir si queríamos un sistema multiplataforma o por contra nos interesaba centrarnos en algún sistema operativo concreto. Consideramos positivo dejar libertad a los alumnos en la elección del sistema operativo donde realizar el desarrollo no obstante valoramos conveniente el que los proyectos planteados por los alumnos (la mayoría videojuegos) se pudiesen ejecutar en entorno Windows, ya que el mercado de los videojuegos para PC se centra en este sistema operativo. Como veremos a continuación estas características hicieron que se desestimaran la mayoría de las herramientas comerciales para el desarrollo de los videojuegos.

Mingw32. Tal y como lo definen sus autores Mingw32 (<http://www.mingw.org/>) es una colección de libre distribución de cabeceras y librerías combinadas con herramientas GNU que permiten producir aplicaciones de Windows nativas que no requieren DLLs de terceras partes.

Se trata por tanto de un compilador que produce ejecutables para los sistemas operativos W32 (Windows 95/98/2000/NT4/XP) adaptando los compiladores del proyecto GNU GCC. De esta manera este conjunto de programas llevan toda la potencia y versatilidad del software GNU a las plataformas Windows.

De hecho el funcionamiento del compilador es prácticamente el mismo que su homólogo GCC, compartiendo las mismas herramientas comunes. Es por ello que los profesores de la asignatura elegimos este compilador para la realización de las prácticas. Por una parte se trata de una familia de compiladores de sobrado prestigio y funcionalidad. Por otra los alumnos ya conocen las herramientas GNU GCC en el entorno Linux con lo que el tiempo requerido para aprender su funcionamiento es mínimo. Finalmente el compilador, aun funcionando prácticamente de la misma manera que su homólogo GCC genera ejecutables para W32 (pudiendo portar el código fácilmente a entornos UNIX).

No obstante Mingw32 es únicamente un compilador, necesitamos un entorno de desarrollo integrado (IDE por sus siglas en inglés) para ayudar a la programación. Es fundamental que los alumnos utilicen un IDE por varias razones: por una parte les facilita la realización de tareas más o menos tediosas (como puede ser la compilación del proyecto mediante makefiles...) y por otra les permite disfrutar de un entorno de depuración completo y compacto.

Eclipse. Eclipse (<http://www.eclipse.org/>) es una plataforma de software de Código abierto independiente de plataforma. Las Herramientas eclipse pretenden ser un conjunto lo suficientemente amplio de herramientas para la creación de cualquier tipo de contenido, lenguaje, arquitectura, etc... Utilizando el plugin de desarrollo para C/C++, conocido como CDT se posibilita el utilizar este potente editor para generar aplicaciones en estos dos lenguajes. No obstante la configuración de este plugin no es sencilla y por otra parte los requerimientos

para su ejecución son elevados por lo que por ahora descartamos su uso en la asignatura, aunque desde nuestro punto de vista no deja de ser una buena alternativa.

DevC++. Bloodshed Dev-C++ (<http://www.bloodshed.net/devcpp.html>) es un IDE para programar en lenguaje C/C++ que usa MinGW como su compilador. Dev-C++ puede además ser usado en combinación con Cygwin y cualquier compilador basado en GCC. El Entorno está desarrollado en el lenguaje Delphi de Borland. La ventaja fundamental de este entorno de desarrollo es la posibilidad de descargarse y preinstalar librerías de una manera muy sencilla, evitándonos todos los problemas relacionados con las dependencias y compilación de las mismas.

No obstante este entorno tiene el problema de no tener correctamente ajustado el depurador de código, con lo que muchas veces podemos tener problemas a la hora de buscar y corregir fallos de compilación.

Code:Blocks. (<http://www.codeblocks.org/>), es una herramienta IDE para el desarrollo de programas en lenguaje C++. Está basado en la plataforma de interfaces gráficas WxWidgets, lo que le permite ejecutarse libremente en diversos sistemas operativos, y es de licencia GPL. Debido a que en sí Code:Blocks es sólo la interfaz del entorno de desarrollo, puede enlazarse a una variedad de compiladores para poder desarrollar su trabajo. Por defecto, Code:Blocks buscará una serie de compiladores y configurará los que halle. En particular, como se ha comentado anteriormente usará MinGW32.

Este es el entorno de desarrollo que se ha seleccionado para la asignatura por tres razones. Para empezar, está preparado para utilizar los mismos repositorios de preinstalación que DEVC++, con lo que comparte la misma ventaja de poder instalar de manera muy sencilla librerías o accesorios para el entorno. Además se trata de un entorno multiplataforma con lo que no se limita el sistema operativo utilizado por los alumnos para el desarrollo. Por último la opción de depuración está más elaborada que en la herramienta anterior proporcionando un sistema más sólido para el desarrollo.

Alternativas propietarias:

Visual C++. Microsoft Visual Studio (<http://www.microsoft.com/latam/visualc/>) es una suite comercial de programación para el sistema operativo Microsoft Windows. Ésta se conforma de varios lenguajes de programación y entre ellos C y C++. Se caracteriza por proporcionar un sistema de desarrollo completo para aplicaciones en el sistema operativo Windows. Existe una versión gratuita de su compilador, llamada Microsoft Visual C++ Express Edition disponible aunque con algunas limitaciones respecto a la versión de pago.

La principal ventaja de utilizar este compilador es su total integración en el sistema operativo y sus herramientas de depuración avanzadas. No obstante

el nombre de visual no especifica lo que a priori podríamos esperar, ya que su aprendizaje inicial es bastante costoso. Además se introducen algunas opciones no estándar en los lenguajes C y C++, con lo que el código desarrollado para este compilador puede dar problemas al ser portado a otros distintos. Por otra parte este compilador solo está disponible para los sistemas operativos Windows, limitando el desarrollo exclusivamente a los entornos W32.

Borland C++ Builder. Borland C++ Builder es un completo entorno de desarrollo para windows (<http://www.codegear.com/products/cppbuilder>). Se pueden crear aplicaciones con acceso a bases de datos, componentes para facilitar la programación de internet.

Existe una versión gratuita del Borland C++ Builder 5. Esta versión sólo incluye el compilador (Borland C++ Compiler), no tiene ni el IDE (entorno integrado de desarrollo), ni los componentes visuales (VCL), ni acceso a bases de datos... Como punto negativo este compilador únicamente se encuentra para operativos W32. Aunque existe una versión (ya no soportada por Borland) llamada Kylix, que se centró en los sistemas linux, ésta no está actualmente soportada por la mayoría de distribuciones y además se considera un producto de escasa calidad (ver http://en.wikipedia.org/wiki/Kylix_programming_tool).

4.3 Librerías gráficas

A la hora de desarrollar una aplicación gráfica es interesante trabajar con una librería que nos abstraiga de las tareas más tediosas y menos relacionadas con nuestra aplicación. Las librerías gráficas nos permiten abstraernos del hardware y realizar de manera transparente las operaciones más usuales utilizando una plataforma común. A continuación se revisarán dos librerías libres para el desarrollo de aplicaciones gráficas interactivas.

Por razones de espacio nos centraremos exclusivamente en librerías para el desarrollo de aplicaciones bidimensionales.

Allegro (<http://alleg.sourceforge.net/index.es.html>) es una librería para programadores de C/C++ orientada al desarrollo sistemas gráficos y videojuegos, distribuida libremente, y que funciona en las siguientes plataformas: DOS, Unix (Linux, FreeBSD, Irix, Solaris), Windows, QNX, BeOS y MacOS X. Tiene muchas funciones de gráficos, sonidos, entrada del usuario (teclado, ratón y joystick) y temporizadores. También tiene funciones matemáticas en punto fijo y coma flotante, funciones 3d, funciones para manejar ficheros, ficheros de datos comprimidos y una interfaz gráfica.

Se trata de una librería con muchas opciones y muchas veces es recomendada como la librería ideal para iniciarse en el desarrollo de los videojuegos. Esta fue la primera librería que utilizamos en la asignatura. No obstante, como veremos posteriormente, recomendamos el uso de SDL principalmente por dos razones: por una parte SDL proporciona una gestión multiproceso basada

en hilos muy útil en el caso de aplicaciones interactivas. Además implementa un sistema de mensajes, adaptable al usuario, que ayuda sobremanera para el desarrollo de estas aplicaciones.

SDL. LibSDL (<http://www.libsdl.org/>) es un conjunto de librerías desarrolladas con el lenguaje C que proporcionan funciones básicas para realizar operaciones de dibujado 2D, gestión de efectos de sonido y música, y carga y gestión de imágenes. SDL es una abreviatura en inglés de Simple DirectMedia Layer. Pese a estar programado en C, tiene wrappers a otros lenguajes de programación como C++ (que es el que se utilizará en la clase de prácticas). También proporciona herramientas para el desarrollo de videojuegos y aplicaciones multimedia. Una de sus grandes virtudes es el tratarse de una librería multiplataforma, soportando oficialmente los sistemas windows, linux, MacOS y QNX, además de otras arquitecturas/sistemas como Dreamcast, GP32, GP2X... De ahí le vienen las siglas Simple Directmedia Layer que más o menos alude a capa de abstracción. Además tiene como principales bazas el posibilitar de manera sencilla la programación multihilo y el implementar un sistema de gestión de mensajes. Por lo demás la funcionalidad es parecida a la proporcionada por Allegro. Por último remarcar que además SDL proporciona un acceso directo a OpenGL, de manera que se pueden combinar aplicaciones 2D y 3D de manera sencilla utilizando toda la potencia que nos proporciona esta librería.

4.4 Herramientas de interacción y documentación

Por último, además de las herramientas propias para desarrollar las tareas de la asignatura es importante, más aun cuando utilizamos una metodología como el aprendizaje basado en proyectos, proporcionar a los alumnos herramientas para la interacción y la documentación. A continuación se comentará brevemente la herramienta para trabajo en grupo utilizada en la asignatura.

Gforge (<http://gforge.org/>) es una plataforma para el desarrollo de software en forma comunitaria que permite organizar y administrar grandes cantidades de proyectos. Esta herramienta se basó inicialmente en el código liberado por “sourceforge” (uno de los repositorios y herramientas para el desarrollo de software libre mas importantes) y fue evolucionando hasta su forma actual. Esta plataforma proporciona multitud de herramientas para el desarrollo y administración de software, entre otras permite manejar versiones de ficheros utilizando subversion, dispone de un calendario donde distribuir las tareas entre los miembros del grupo, proporciona herramientas para publicar documentos relacionados con el proyecto, foros... Además Gforge proporciona una imagen Vmware <http://www.vmware.com/es/> que permite realizar una instalación de la plataforma en minutos (de otra manera la configuración correcta del sistema puede tardar días debido a la cantidad de herramientas que forman la plataforma).

Doxygen (<http://www.doxygen.org/>) es un sistema libre generador de documentación. Se trata de una aplicación multiplataforma que analiza el código fuente en busca de comentarios en un determinado formato. De esta manera genera informes utilizando la estructura del código fuente junto con la documentación introducida por los programadores. Consideramos esta herramienta como un complemento ideal para la realización de la documentación de un proyecto.

DIA. Por otra parte, para completar los informes de la documentación es fundamental el proporcionar diagramas explicativos del funcionamiento del sistema. La herramienta DIA (<http://www.gnome.org/projects/dia/>) es un potente programa que nos permitirá crear cualquier tipo de diagrama de flujo, UML o similares, de una forma muy sencilla e intuitiva utilizando el ratón en sistemas linux/unix/windows. Con un concepto similar al software propietario Visio, DIA va a permitir desarrollar gráficos vectoriales y diagramas para aquellos documentos que lo requieran. No obstante la desventaja fundamental de esta aplicación es que no permite generar código fuente a partir de los diagramas. Esta es una característica que debemos tener en cuenta a la hora de seleccionar esta aplicación.

4.5 Opinión de los alumnos

Durante el segundo cuatrimestre del curso 2007- 2008 se planteó a los alumnos una encuesta sobre la asignatura donde se especificaban preguntas relacionadas con las herramientas planteadas previamente. Concretamente se les pidió que evaluaran dos de las herramientas fundamentales para la realización de las prácticas, el entorno de desarrollo CodeBlocks (junto con el compilador MINGW32) y la librería SDL. Además los alumnos podían proponer herramientas alternativas o comentar de manera general sus experiencias.

Como resultados generales la herramienta CodeBlocks ha obtenido un 3.9 sobre 5 puntos posibles. La librería SDL un 4.28 sobre también 5 posibles. Cabe destacar que sobre el entorno CodeBlocks hay alumnos que comentan que el debugger integrado a veces no funciona de manera correcta, aunque no es el caso habitual. De esta manera podemos observar como los alumnos evalúan de manera positiva el uso de estas herramientas para el desarrollo de la asignatura.

5. CONCLUSIONES

En este artículo se ha presentado nuestra experiencia en la utilización de software docente libre para la asignatura Juegos y Realidad de la carrera de Ingeniería Informática de la Universidad de Alicante. Se han presentado varias aplicaciones comentando sus ventajas y desventajas. Como se ha comentado previamente la selección del software docente para una asignatura es una tarea

compleja, que debe buscar el equilibrio entre varios requerimientos como el que las herramientas sean suficientemente potentes para poder desarrollar los objetivos del curso, que no sean excesivamente complejas ni requerir mucho tiempo de aprendizaje. Nosotros consideramos que las aplicaciones aquí mostradas se pueden utilizar de manera conjunta proporcionando una plataforma ideal para el desarrollo de aplicaciones gráficas e interactivas.

Por otra parte nos gustaría destacar que durante los 4 años que llevamos impartiendo la asignatura no hemos tenido ningún problema en utilizar software libre lo que pone de manifiesto que actualmente podemos basar nuestra docencia exclusivamente utilizando este tipo de herramientas.

6. REFERENCIAS

RUSSELL SHACKELFORD, ANDREW MCGETTRICK, ROBERT SLOAN, HEIKKI TOPI, GORDON DAVIES, REZA KAMALI, JAMES CROSS, JOHN IMPAGLIAZZO, RICHARD LEBLANC, and BARRY LUNT. *Computing curricula 2005: The overview report*. Technical report, New York, NY, USA, 2006.

Computing curricula 1991. Commun. ACM, 34(6):68– 84, 1991.

ILLA RIVERO, ALBERTO MORALES, PATRICIO GARCÍA, and FRANCISCO DE SANDE. *El software libre en la educación universitaria utilización de las tic como recurso de apoyo a la enseñanza presencial*. Actas de las I Jornadas canarias sobre las TIC en la docencia universitaria, 2003.

XAVIER AMATRIAIN. *El software libre en la educación: guía para su justificación e implementación*. Actas de las III Jornadas de Software Libre. Escuela Politécnica Superior de Ingeniería, Universitat Politècnica de Catalunya, URL: <http://portal.jornadespl.org/biblioteca/iii-jornades>, 2004.

ANA M. OLIVER CUETO, RAFAEL DEGADO GARCÍA. *Algunas experiencias en la utilización de software libre en la educación superior*. Actas del III Congreso online Conocimiento Abierto, Sociedad Libre del observatorio para la cibernsiedad. <http://www.cibersociedad.net/congres2006/gts/comunicacio.php?id=850&llengua=es>, 2006

[http://es.wikipedia.org/wiki/ Software_libre](http://es.wikipedia.org/wiki/Software_libre)

