



UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS

PROPUESTA DE PROYECTO FINAL DE CARRERA
INGENIERÍA EN INFORMÁTICA

**Desarrollo de algoritmos de teselación
adaptativa en GPU**

Alumno: Fernando Nellmaddin

Director: Dr. Néstor Calvo

Palabras clave: refinamiento dinámico, programación en GPU, nivel de detalle

Santa Fe, Septiembre de 2013

Índice

1. Introducción	2
2. Justificación	3
3. Objetivos perseguidos	4
3.1. Objetivo general	4
3.2. Objetivos específicos	4
4. Alcances	4
5. Metodología	5
6. Plan de tareas	5
7. Cronograma	7
8. Puntos de control y entregables	7
9. Riesgos y estrategias de mitigación	8
10. Recursos disponibles y necesarios	9
11. Presupuesto	10
12. Bibliografía	13

1. Introducción

Las tecnologías de hardware han tenido un crecimiento exponencial desde la concepción misma de la informática, y en particular desde su aceptación por el público no especializado. En los primeros años, el objetivo era minimizar el espacio ocupado por las máquinas; en esta etapa se sitúa el lanzamiento de la primera PC de IBM en el año 1981 [1]. Más tarde la industria del hardware se concentró en dotarlas de más potencia, principalmente en sus microprocesadores; este paradigma se mantuvo durante más de 20 años [2]. Desde ese momento se bifurca la tendencia, por un lado hacia máquinas más potentes, con sus problemas de peso y refrigeración; y por el otro en dispositivos livianos, portátiles y agradables al usuario hogareño.

Durante la década de 1980, y en paralelo con el avance en la tecnología de los microprocesadores, las placas aceleradoras de video evolucionaron de manera de brindarles una distintiva capacidad para realizar cálculos geométricos a gran escala y velocidad [3]. Fue en aquella época cuando los videojuegos comenzaron a mostrar un gran nivel de detalle en entornos tridimensionales. A diferencia de los microprocesadores para la CPU, los coprocesadores gráficos o GPU, pudieron evolucionar con arquitecturas totalmente innovadoras, debido a que no tenían el requisito de compatibilidad hacia atrás.

Los entornos 3D se generan a partir de una gran cantidad de polígonos conectados entre sí formando mallas. Los polígonos elegidos son triángulos y cuadriláteros, debido a su simplicidad. Así, cada escenario y personaje de una animación cinematográfica o de un videojuego, está representado por un conjunto de polígonos que le dan forma.

La cantidad de elementos utilizados para mostrar mundos virtuales crece año tras año, siguiendo las exigencias del consumidor. Cuanto más polígonos contenga un modelo en particular, se apreciará un mejor nivel de detalle. En la misma línea, los ingenieros de hardware se enfrentaron a la tendencia de crear mundos de gran tamaño y complejidad para la industria de videojuegos. En ésta, se necesita renderizar simultáneamente un gran volumen de polígonos que den vida a toda la escena con un nivel de detalle adecuado. Por otra parte, todo lo anterior se refiere al renderizado de una sola imagen en 3D; en un videojuego, se renderizan varias imágenes por segundo, con la geometría cambiante en forma fluída, de manera que la experiencia visual del usuario sea placentera.

Para aliviar esos inconvenientes, surge la tecnología de teselación adaptativa, una solución de hardware y software que permite generar mundos grandes (muchos polígonos) y con gran detalle. Esta tecnología de teselación comenzó a utilizarse ya desde su implementación en la API multimedia Direct3D 11, presentada en 2008 [4].

La teselación adaptativa consiste en actualizar dinámicamente el nivel de detalle de una región en particular de la escena. Se parte de una malla gruesa (de poca densidad) que se refina a medida que se requiere. Para los objetos que estén próximos a la cámara, y por lo tanto deban verse de cerca, se genera un mallado denso que los representa con gran cantidad de polígonos. Por otra parte, a los objetos alejados de la cámara y que no requieran una alta precisión, se los representa con una malla de pocos elementos. Cuando la cámara se mueve, el mallado se regenera dinámicamente para obtener más detalles de los nuevos objetos centrados en la pantalla y reducir la carga de los que ya no están en foco.

En la GPU, la flexibilidad atenta contra la velocidad de procesamiento que tiene el software rígido implementado directamente en el hardware. Para permitir programar teselación en forma dinámica, fue necesario introducir una etapa programable específica en el rígido *pipeline* gráfico de la GPU, y modificar el software para que lo interprete.

En la actualidad, la necesidad de generar mundos de gran tamaño, no sólo se halla en los videojuegos. Los entornos tridimensionales también se encuentran, por ejemplo, en los simuladores de vuelo necesarios para el entrenamiento de pilotos y otros productos de similares características. Además, distintas áreas de ingeniería hacen uso de simulaciones para el estudio de fenómenos específicos; ya sea para la visualización o para el cálculo numérico de sistemas de ecuaciones diferenciales. En todos estos casos, la posibilidad de adaptar la geometría de la malla, refinando y desrefinando sectores según el interés variable, presenta una notable y deseable mejora en el procesamiento de la escena y/o el cálculo numérico.

2. Justificación

La vertiginosa velocidad de desarrollo de las tecnologías dificulta que un profesional se mantenga actualizado en la totalidad de los últimos avances de su área. En el caso concreto de computación gráfica, y si bien se cuenta con un curso de grado en nuestra facultad sobre el tema, los contenidos allí brindados abarcan los fundamentos del área. Estos fundamentos son de importancia primaria para el desarrollo del alumno que desee dedicarse a computación gráfica, ya que sin un manejo adecuado de éstos, no se podrían abordar los temas más avanzados. Sin embargo, poco espacio queda en el planeamiento general del curso y de la carrera en sí, para la adición de tópicos más avanzados de computación gráfica, ya que éstos deben convivir con cursos en otras áreas.

El principal propósito del presente proyecto es el de realizar un estudio minucioso sobre las técnicas de teselación para la generación y actualización dinámica de mallas. La presentación de esta tecnología, a partir del análisis, desarrollo e implementación de algunos modelos ejemplificadores, será la clave para analizar sus ventajas y desventajas. Adicionalmente, y como subproducto, se busca acercar a la comunidad universitaria los últimos conocimientos desarrollados en el área. A partir de la presentación cuidadosamente estudiada y comprendida de tecnologías innovadoras, se pretende estimular a la comunidad local en el desarrollo e investigación en el área de computación gráfica, mostrando cuáles son las posibilidades de las tecnologías recientes.

Previamente al agregado de teselación en GPU, los desarrolladores trabajaron en diseñar técnicas que generaran nivel de detalle dinámico con algoritmos puramente en CPU, o utilizando las capacidades reducidas de las GPU antiguas. Una vez disponible la tecnología de teselación, se espera que estos algoritmos hayan quedado en desventaja frente a una implementación pura y directa en la GPU. Con este proyecto, se pretende analizar la aceptación y utilización de esta tecnología en la industria, y desarrollar e implementar pruebas propias para obtener una conclusión referente a sus ventajas y desventajas.

Este trabajo permitirá cumplir con los objetivos, demandando muy poco dinero para su desarrollo, pues se dispone del hardware requerido. Por otra parte, los objetivos están plenamente justificados en lo individual, por la ganancia en conocimientos aplicables en la profesión, y en lo institucional, por la trasmisión de dichos conocimientos a la comunidad uni-

versitaria, tanto en herramientas didácticas como en puntas para iniciar futuros desarrollos productivos o de investigación.

Por último, y a nivel personal, el interés del alumno en el estudio y desarrollo de tecnologías para videojuegos, y la predilección por el desarrollo de videojuegos en general, resulta como principal motivación para la realización del proyecto. La oportunidad de comprender cómo los profesionales solucionan problemas que surgen de las exigencias de la misma industria, resulta en un gran atractivo para el alumno, quien siempre se ha planteado soluciones propias ante los problemas que ha sabido identificar.

3. Objetivos perseguidos

3.1. Objetivo general

Investigar, desarrollar e implementar un conjunto de técnicas de teselación para la actualización de mallas y aplicarlas en modelos-ejemplo que requieran un nivel de detalle variable.

3.2. Objetivos específicos

Los objetivos específicos propuestos son:

- Investigar bibliotecas que permitan la teselación en placas aceleradoras de video.
- Analizar, desarrollar e implementar técnicas de teselación.
- Investigar, desarrollar e implementar aplicaciones de teselación.
- Adquirir experiencia con el trabajo de mallas y con teselación adaptativa.
- Colaborar con la comunidad universitaria local al presentar tecnologías innovadoras del área de computación gráfica.

4. Alcances

El presente trabajo tiene como objetivo investigar, desarrollar e implementar algoritmos de teselación en placas aceleradoras de video de última generación. Por lo tanto y a pesar de que existen técnicas de teselación ya desarrolladas que no hacen uso de las nuevas tecnologías, se excluye del trabajo planificado el desarrollo de algoritmos que no utilicen exclusivamente las capacidades de teselación de la GPU.

Para probar dichos algoritmos, se propone el uso de modelos tridimensionales simples que sean suficientes para mostrar las características de las técnicas desarrolladas. La complejidad de estos modelos se decidirá acorde a las técnicas que se desarrollen, y se intentará que se

mantengan lo más simple posible para no desviar así la atención en el desarrollo. Adicionalmente, se evaluará la posibilidad de realizar pruebas de teselación en escenas complejas propias de un videojuego, de manera de analizar sus características en esta industria.

Para finalizar, se propone realizar un análisis cuantitativo y cualitativo con respecto a la velocidad y complejidad de renderizado de estas escenas. De este modo, se busca analizar las ventajas y desventajas de utilizar teselación en diversidad de aplicaciones.

5. Metodología

Para el desarrollo del presente proyecto se utilizará la metodología de trabajo del modelo en espiral. Este modelo propone agrupar las actividades en fases que se repiten en forma de ciclos a medida que se avanza. Al inicio de cada iteración, se propone un conjunto de objetivos a corto plazo a cumplimentar. Al finalizar cada bucle, se realiza un contraste entre los objetivos propuestos y los alcanzados y se efectúa un análisis de riesgos, de manera de plantear los objetivos para la próxima iteración.

El proceso comienza con el análisis del problema, por lo que el alumno debe realizar una búsqueda bibliográfica sobre el estado del arte en la temática. A partir de la comprensión del problema, las etapas 2 y 3 consisten en el análisis y selección tanto de las herramientas a utilizar (etapa 2), como de las técnicas de teselación propiamente dichas (etapa 3). Luego de esta selección, se procederá al desarrollo de técnicas de teselación específicas en la etapa 4. En la quinta etapa, al finalizar el desarrollo, se realizará un estudio comparativo para evaluar las características del método estudiado. Pero todo este proceso se realiza en forma espiral con pequeños avances en cada ciclo. De esta manera, se realizan varias iteraciones abarcando ciertas actividades de las etapas 3 y 4, donde el alumno debe repetir el ciclo: análisis-diseño-desarrollo de cada técnica, realizar las pruebas correspondientes y llevar a cabo la integración con el trabajo ya realizado. Además, antes de avanzar, el alumno debe adquirir nuevos conocimientos que le sean necesarios para el desarrollo de la próxima iteración.

El modelo de trabajo en espiral es elegido para el presente proyecto, ya que su flexibilidad brinda oportunidades de elegir los objetivos en base al avance que se realiza. Además, en cada iteración se tiene un prototipo funcional, de manera de poder evaluar el desempeño al finalizar cada ciclo, sin esperar a la finalización del proyecto. Si durante el trabajo se encontraran nuevas técnicas de teselación, se evaluará el tiempo y dificultad de desarrollarlas según el estado de avance.

6. Plan de tareas

Para la realización del proyecto, se propone el siguiente conjunto de etapas, discriminado por las tareas que se detallan en cada uno. Las etapas 1 y 2 se realizarán en forma secuencial. En la actividad 3.2 se definirá cuántas iteraciones del modelo en espiral se van a realizar, y cuáles son las técnicas de teselación que se van a implementar en cada etapa. Se utilizarán las iteraciones del modelo en espiral para la realización de las etapas 3.3 (análisis), 3.4 (diseño), 4.1 (desarrollo) y 4.2 (pruebas unitarias), donde se iterará por cada técnica o grupo de

técnicas a desarrollar. En este grupo de tareas es en donde se debe iterar, ya que allí se encuentra la mayor carga de desarrollo de software para el alumno.

1. Lectura de bibliografía [60 horas]

- 1.1 Investigación sobre el estado del arte en teselación. [40 horas]
- 1.2 Investigación sobre representación de modelos tridimensionales. [20 horas]

2. Análisis y selección de herramientas [105 horas]

- 2.1 Búsqueda bibliográfica sobre bibliotecas y herramientas de desarrollo de software. [20 horas]
- 2.2 Análisis de bibliotecas y herramientas a utilizar. [20 horas]
- 2.3 Selección de bibliotecas y herramientas. [10 horas]
- 2.4 Adquisición de experiencia en el desarrollo utilizando las bibliotecas y herramientas seleccionadas. [40 horas]
- 2.5 Redacción de primer informe de avance. [15 horas]

3. Análisis, selección y diseño de técnicas y modelos [135 horas]

- 3.1 Análisis general de técnicas. [20 horas]
- 3.2 Selección de las técnicas a desarrollar. [10 horas]
- 3.3 Análisis de cada una de las técnicas. [40 horas]
- 3.4 Diseño de cada una de las técnicas. [20 horas]
- 3.5 Redacción de segundo informe de avance. [15 horas]
- 3.6 Análisis de modelos tridimensionales para la evaluación de las técnicas. [20 horas]
- 3.7 Selección de modelos. [10 horas]

4. Desarrollo de técnicas [255 horas]

- 4.1 Desarrollo de cada una de las técnicas de teselación seleccionadas. [100 horas]
- 4.2 Pruebas de funcionalidad unitarias de cada técnica desarrollada. [30 horas]
- 4.3 Redacción de tercer informe de avance. [15 horas]
- 4.4 Desarrollo de los modelos tridimensionales. [50 horas]
- 4.5 Desarrollo de interfaz para la aplicación de las técnicas a los modelos. [20 horas]
- 4.6 Pruebas integrales del software desarrollado. [40 horas]

5. Análisis comparativo [45 horas]

- 5.1 Comparación de las técnicas según complejidad algorítmica, tiempos de ejecución y apreciación subjetiva de la calidad. [30 horas]
- 5.2 Redacción de cuarto informe de avance. [15 horas]

6. Presentación [100 horas]

- 6.1 Redacción de informe final. [60 horas]
- 6.2 Presentación de los desarrollos en la universidad y en congresos. [20 horas]
- 6.3 Publicación de resultados que lo justifiquen. [20 horas]

7. Cronograma

En la siguiente figura, se detalla el cronograma propuesto para la realización del proyecto. Se fija como fecha de inicio el día primero de Julio de 2013. Se estima que para la ejecución del proyecto serán necesarias 36 semanas, por lo que la fecha esperada de finalización es el 14 de Marzo de 2014.

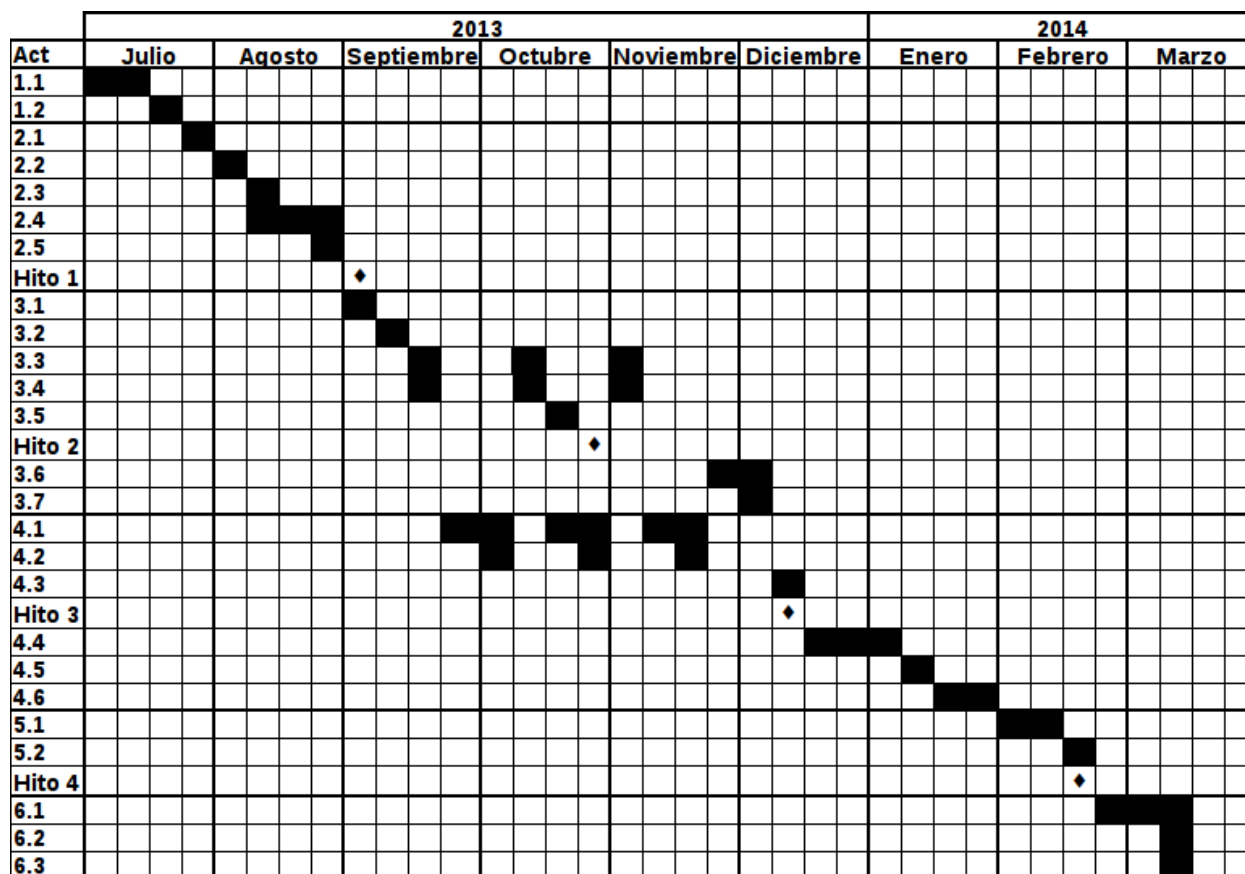


Figura 1: Cronograma por semanas propuesto para llevar a cabo el proyecto.

8. Puntos de control y entregables

En el desarrollo del proyecto, se fijan un conjunto de hitos donde se presentarán informes de avances. A continuación se detalla la lista de los mismos, junto a las etapas que abarca cada uno y la fecha aproximada de realización.

Hito 1: Resumen de la investigación – Etapas 1 y 2 – Semana 9 – 5 de Septiembre de 2013

En este documento se incluirá un resumen de toda la información recopilada con respecto al tema en general, el estado del arte, y las herramientas a utilizar para desarrollar el proyecto. Se propone realizar un análisis comparativo sobre la elección de una u otra biblioteca para el desarrollo del software, exponiendo las razones correspondientes.

Hito 2: Análisis y diseño de técnicas de teselación – Actividades 3.1 a 3.4 –

Semana 15 – 24 de Octubre de 2013

En este hito, se hará entrega de un informe en el que se detalle, con ayuda de los diagramas correspondientes, las técnicas de teselación que se van a desarrollar y sobre qué modelos se van a probar. Se realizará una justificación sobre por qué se elige cierto modelo o técnica. Además se especificarán las iteraciones que se van a realizar y qué técnicas pertenecen a cada ciclo del modelo en espiral. Por último, se entregarán los documentos correspondientes al análisis y diseño de las primeras 2 iteraciones del modelo en espiral.

Hito 3: Desarrollo de técnicas y modelos – Actividades 3.6 a 4.2 – Semana 23 – 12 de Diciembre de 2013

En este hito, se hará demostración de todas las técnicas de teselación adaptativas implementadas. Se hará entrega de un pseudocódigo apoyado con gráficos, donde se explique en que consiste cada técnica. Asimismo, se hará demostración de los modelos tridimensionales implementados para la aplicación de las técnicas.

Hito 4: Integración y comparación de técnicas – Actividades 4.2 a 5.1 – Semana 31 – 20 de Febrero de 2014

En este último hito, se hará demostración del software final que aplica las diversas técnicas de teselación a los modelos tridimensionales previamente seleccionados. Se hará entrega de diagramas que detallen el flujo que realiza el renderizado para cada modelo. Además, se realizará un análisis comparativo sobre las características de los métodos de teselación y se los comparará con el desempeño del mismo hardware en renderizado sin teselación. Se tomarán ciertos parámetros tanto objetivos como subjetivos y se realizará una tabla comparativa.

9. Riesgos y estrategias de mitigación

En esta sección, se detallan los riesgos identificados al inicio del proyecto. Si durante la ejecución del mismo, resultara la aparición de nuevos riesgos, se los agregará en este apartado exponiendo las razones correspondientes.

- **Pérdida de recursos de hardware:** *Probabilidad: Baja – Impacto: Alto*

Para la realización del proyecto, se necesita disponer de una placa aceleradora de video de última generación, de manera que pueda soportar la tecnología de teselación en GPU. Específicamente, se necesita una placa de video compatible con DirectX 11 [5] y OpenGL 4.0 [6].

Estrategia ante el riesgo: Mitigar

Tanto el alumno (en su hogar) como el director del proyecto (en su oficina en el aula FICH-CIMNE de la Facultad) cuentan con el hardware necesario para la realización del proyecto. Si se presentan inconvenientes en uno de los dos recursos, el desarrollo se puede trasladar al otro. Por otra parte, si se imposibilita la utilización de ambos recursos, se deberá analizar la posibilidad de adquirir uno nuevo.

• **Pérdida del desarrollo o investigación realizados:** *Probabilidad: Baja – Impacto: Medio*

Los recursos relacionados a la investigación y el desarrollo estarán principalmente almacenados en computadoras. Si ocurriese un problema en el que se perdiera la información (rotura de disco rígido, robo, virus informático, etc.), el proyecto se vería comprometido.

Estrategia ante el riesgo: Mitigar

Se propone la utilización del sistema de versionado Git [7] para almacenar el software en servidores online en Internet, y poder recuperarlo ante cualquier eventualidad. Además, se propone hacer un backup semanal en un disco externo o DVD, de manera de tener un punto de control semanal para recuperarse ante el imprevisto.

• **Cambio de tecnologías con las que se realiza el desarrollo:** *Probabilidad: Media – Impacto: Bajo*

Las bibliotecas de desarrollo de software son muy volátiles y se actualizan con frecuencia. Sin embargo, en el caso de APIs y bibliotecas de interfaces gráficas (OpenGL o DirectX por ejemplo), a pesar de que se actualizan frecuentemente, mantienen una buena compatibilidad hacia atrás.

Estrategia ante el riesgo: Evitar

Se propone limitarse a versiones específicas de las bibliotecas y APIs utilizadas durante el proyecto, y no actualizar el software para evitar inconvenientes que escapen al desarrollo propio del proyecto.

10. Recursos disponibles y necesarios

Todos los recursos que el alumno necesitará para llevar a cabo el proyecto, se encuentran disponibles al inicio del mismo.

1. Recursos humanos

- 1.1 Alumno: desarrollo del proyecto.
- 1.2 Director del Proyecto: supervisión del proyecto.

2. Hardware

- 2.1 PC de escritorio con procesador multicore, monitor LCD y placa aceleradora de video compatible con DirectX 11 y OpenGL 4.0.
- 2.2 Impresora para la impresión de papers, apuntes y entregables.

3. Software de disponibilidad gratuita e inmediata

- 3.1 Sistema Operativo: Kubuntu Linux 13.04.

3.2 Entorno de desarrollo: Vim Editor + compilador GCC. Lenguaje C++ y bibliotecas OpenGL o DirectX 11.

4. Equipamiento e insumos

4.1 Ubicación con escritorio y comodidades para el desarrollo.

4.2 Conexión a Internet para consulta de bibliografía y para la realización de copias de seguridad.

4.3 Acceso a publicaciones científicas.

4.4 Insumos varios tales como hojas, lapiceras, cartuchos de tinta para impresora, pendrive para la transferencia de datos, etc.

11. Presupuesto

A continuación se detalla el presupuesto estimado para la realización del proyecto, discriminado por actividad. Primero se detallan unas notas con respecto a la forma de estimación y los detalles de qué significa cada entrada del presupuesto.

- El alumno trabajará 4 horas por día en el aula FICH-CIMNE, ubicada en la Facultad de Ingeniería y Ciencias Hídricas. El coste unitario por hora de la remuneración ficticia al trabajo del alumno se estimó en \$50.
- Se propone una remuneración ficticia al Director de Proyecto de \$90 por hora, estimando que se le consultará dos horas por semana sobre los pasos a seguir en el desarrollo.
- El coste de impresiones se refiere a la impresión de libros o publicaciones para el estudio y formación del alumno. Además, se refiere a la impresión de todos los informes, tanto los reportes de avance, como el informe final a presentar a los evaluadores. Por último, en este apartado se incluyen las impresiones de paneles para la presentación en congresos si correspondiera
- El transporte se calcula como si cada jornada de trabajo, que consta de 4 horas, se toman dos colectivos de ida y vuelta a \$3,50 cada pasaje. Este coste se divide por la cantidad de horas diarias, dando un coste de transporte de \$1,75 por hora de trabajo.
- El costo asociado a “Viajes a Congresos” toma como referencia el coste del viaje a Buenos Aires y hotelería, en el caso de que se desarrollen conocimientos adecuados para su presentación en congresos.
- El coste de hardware se realizó en base a la estimación del precio amortizado al día de realización del proyecto, de las dos computadoras que se van a utilizar en el desarrollo: la del alumno en su hogar estimada en \$2000, y la computadora ubicada en el aula FICH-CIMNE de \$3000.
- En el apartado “Servicios y otros insumos”, se incluye el coste de energía eléctrica e insumos de oficina tales como papeles, tinta de impresión o lapiceras, entre otros.

Actividad	Recurso	Hs. hombre	Total (\$)
1.1	Alumno Impresiones Transporte	40 — —	2000 100 70
1.2	Alumno Impresiones Transporte	20 — —	1000 50 35
2.1	Alumno Impresiones Transporte	20 — —	1000 100 35
2.2	Alumno Transporte	20 —	1000 35
2.3	Alumno Transporte	10 —	500 18
2.4	Alumno Transporte	40 —	2000 70
2.5	Alumno Transporte Impresiones	15 — —	750 26 30
3.1	Alumno Transporte	20 —	1000 35
3.2	Alumno Transporte	10 —	500 18
3.3	Alumno Transporte	40 —	2000 70
3.4	Alumno Impresiones Transporte	20 - —	1000 100 35
3.5	Alumno Transporte Impresiones	15 — -	750 26 30
3.6	Alumno Transporte	20 —	1000 35
3.7	Alumno Transporte	10 —	500 18
4.1	Alumno Transporte	100 —	5000 175
4.2	Alumno Transporte	30 —	1500 53
4.3	Alumno Transporte Impresiones	15 — -	750 26 30
4.4	Alumno Transporte	50 —	2500 88

Actividad	Recurso	Hs. hombre	Total (\$)
4.5	Alumno Transporte	20 —	1000 35
4.6	Alumno Transporte	40 —	2000 70
5.1	Alumno Impresiones Transporte	30 — —	1500 50 53
5.2	Alumno Impresiones Transporte	15 — —	750 50 26
6.1	Alumno Impresiones Transporte	60 — —	3000 500 105
6.2	Alumno Impresiones Viajes a congresos	20 — —	1000 300 2000
6.3	Alumno Impresiones Transporte	20 — —	1000 200 35
Costos indirectos	Hardware Conexión a Internet 5M, 8 meses Servicios y otros in- sumos	— — —	5000 1000 500
	Total	—	6500
Proyecto	Alumno Director de Proyecto Impresiones Transporte Viajes a congresos Costos indirectos	700 70 — — — —	35000 6300 1540 1192 2000 6500
	TOTAL		\$ 52532

12. Bibliografía

- [1] IBM, “*Press Release*”, Information Systems Division, Entry Systems Business. 12 de Agosto, 1981.
- [2] Radhakrishna H., “*From Moore’s Law to Intel Innovation: Prediction to Reality*”. Technical Marketing Engineer. Intel Corporation. 2005.
- [3] Blythe, D., “*Rise of the graphics processor*”. Proceedings of the IEEE Vol. 96, No. 5, pp. 761-778. Mayo de 2008.
- [4] Gee, K., “*DirectX 11 Tessellation*”. Microsoft GameFest, 2008.
- [5] DirectX: API para facilitar el manejo de recursos multimedia. 1995-2013. Disponible en <http://windows.microsoft.com/es-AR/windows7/products/features/directx-11>. [Consultado el 10 de Junio de 2013].
- [6] OpenGL: API para el renderizado de gráficos de computadora en 3D. 1992-2013. Disponible en <http://www.opengl.org>. [Consultado el 10 de Junio de 2013].
- [7] Git: sistema distribuido de control de versión. 2005-2013. Disponible en <http://git-scm.com>. [Consultado el 10 de Junio de 2013]
- [8] CPlusPlus.com Reference. 1983-2013. Disponible en <http://www.cplusplus.com/reference>. [Consultado el 10 de Junio de 2013].
- [9] De Berg, M., Cheong, O., Van Kreveld, M., & Overmars, M., “*Computational Geometry: Algorithms and Applications*”, Springer Editorial, Third Edition, 2008.
- [10] Salomon, D., “*Curves and Surfaces for Computer Graphics*”, Springer Editorial, 2006.
- [11] Taylor & Francis Group, “*OpenGL Insights*”, CRC Press, 2012.
- [12] Rost, R., “*OpenGL Shading Language*”, Second Edition, 2006.
- [13] McKesson, J., “*Learning Modern 3D Graphics Programming*”, 2012.