



FACULTAD DE INGENIERIA

Universidad de Buenos Aires

Taller de Programación (95.08)

Manual de proyecto

Ejercicio final

1º Cuatrimestre 2018

Integrantes:

NOMBRE	PADRÓN	CARRERA
Tomás Accini	99136	Lic. en Análisis de Sistemas
Joaquín Casal	98280	Lic. en Análisis de Sistemas

Recorte de enunciado

En un principio, el grupo contaba con un tercer integrante: Christian Cucco, quién tuvo que abandonar el grupo. Debido a su salida (la cual se dio el 22 de mayo, es decir, empezada la cuarta semana del proyecto), se realizó un recorte del enunciado del proyecto, de manera que se pudiera mantener la fecha de entrega pactada.

Es por este recorte que el cronograma propuesto quedó de la siguiente manera:

	Alumno 1 Servidor - Modelo	Alumno 2 Cliente - Modelo	Alumno 3 Cliente - Editor
Semana 1 (01/05/2018)	- Draft del modelo (incluyendo lógica del juego y partidas multijugador) - Prueba de concepto con Box2D.	- Mostrar una imagen. - Mostrar una animación. - Mostrar ambas en un lugar fijo o desplazándose por la pantalla (movimiento).	- Draft del cliente y del editor (<i>wireframe</i>). - Determinar a partir del draft qué mensajes se necesitarán en el protocolo de comunicación.
Semana 2 (08/05/2018)	- Modelado del escenario, los gusanos, los disparos y explosiones.	- Dibujado del escenario. - Dibujado de animaciones (gusanos, disparos, explosiones) - Controles por teclado.	- Editor, creación de escenarios.
Semana 3 (15/05/2018)	- Finalización del modelado del juego.	- Finalización de la gráfica del escenario. - Controles por mouse. - Scrolling.	- Finalización del editor. - Carga y guardado de los escenarios (archivos).
Semana 4 (22/05/2018)	- Lógica para la creación de partidas multijugador y múltiples partidas.	- Interfaz gráfica al seleccionar un arma o herramienta.	- Implementación del sistema de sonidos y música.
Semana 5 (29/05/2018)	- Finalización de la implementación multijugador.	- Interfaz para la conexión con el servidor, elegir un nivel, crear/unirse a una partida.	- Creación de escenarios de prueba y preparación de la demo. - Pantallas de victoria y derrota.
Semana 6 (05/06/2018)	- Testing - Correcciones y <i>tuning</i> del Servidor - Documentación	- Testing - Correcciones y <i>tuning</i> del Cliente - Documentación	- Testing - Correcciones y <i>tuning</i> del Editor - Documentación
Preentrega el 12/06/2018			
Semana 7 (12/06/2018)	- Testing y corrección de bugs - Documentación	- Testing y corrección de bugs - Documentación	- Testing y corrección de bugs - Documentación
Semana 8 (19/06/2018)	- Testing - Correcciones sobre Preentrega - Armado del entregable	- Testing - Correcciones sobre Preentrega - Armado del entregable	- Testing - Correcciones sobre Preentrega - Armado del entregable
Entrega el 26/06/2018			

Adicionalmente a lo que se ve en la anterior imagen, se establecieron los siguientes cambios:

- Se eliminan las armas de tipo granada, la bazooka y el bate.
- Quedan los ataques de teleport, dinamita y aéreo.
- Se elimina el feature de múltiples partidas, se va a ejecutar una única partida en el servidor.
- Se elimina el sonido.
- Los sprites animados van a ser los del gusano (descanso, caminar, salto).
- La configuración de las conexiones se hace vía línea de comandos.

División de tareas

La división inicial de tareas se realizó siguiendo la propuesta de la cátedra, es decir, que haya una persona encargada mayormente del servidor, otra del cliente y otra del editor, este último aportando también al cliente.

Fue así que la división inicial de tareas se distribuyó de la siguiente manera:

- **Servidor:** Tomás Accini.
- **Editor:** Joaquín Casal.
- **Cliente:** Christian Cucco.

Debido a la posterior salida de Christian del grupo, hubo que redistribuir tareas, lo cual concluyó en que Joaquín y Tomás se encargaran entre los dos de implementar el cliente que inicialmente iba a hacer Christian.

Evolución del proyecto

Cronograma propuesto

	Alumno 1 Servidor - Modelo	Alumno 2 Cliente - Modelo	Alumno 3 Cliente - Editor
Semana 1 (01/05/2018)	- Draft del modelo (incluyendo lógica del juego y partidas multijugador) - Prueba de concepto con Box2D.	- Mostrar una imagen. - Mostrar una animación. - Mostrar ambas en un lugar fijo o desplazándose por la pantalla (movimiento).	- Draft del cliente y del editor (<i>wireframe</i>). - Determinar a partir del draft qué mensajes se necesitarán en el protocolo de comunicación.
Semana 2 (08/05/2018)	- Modelado del escenario, los gusanos, los disparos y explosiones.	- Dibujado del escenario. - Dibujado de animaciones (gusanos, disparos, explosiones) - Controles por teclado.	- Editor, creación de escenarios.
Semana 3 (15/05/2018)	- Finalización del modelado del juego.	- Finalización de la gráfica del escenario. - Controles por mouse. - Scrolling.	- Finalización del editor. - Carga y guardado de los escenarios (archivos).
Semana 4 (22/05/2018)	- Lógica para la creación de partidas multijugador y múltiples partidas.	- Interfaz gráfica al seleccionar un arma o herramienta.	- Implementación del sistema de sonidos y música.
Semana 5 (29/05/2018)	- Finalización de la implementación multijugador.	- Interfaz para la conexión con el servidor, elegir un nivel, crear/unirse a una partida.	- Creación de escenarios de prueba y preparación de la demo. - Pantallas de victoria y derrota.
Semana 6 (05/06/2018)	- Testing - Correcciones y <i>tuning</i> del Servidor - Documentación	- Testing - Correcciones y <i>tuning</i> del Cliente - Documentación	- Testing - Correcciones y <i>tuning</i> del Editor - Documentación
Preentrega el 12/06/2018			
Semana 7 (12/06/2018)	- Testing y corrección de bugs - Documentación	- Testing y corrección de bugs - Documentación	- Testing y corrección de bugs - Documentación
Semana 8 (19/06/2018)	- Testing - Correcciones sobre Preentrega - Armado del entregable	- Testing - Correcciones sobre Preentrega - Armado del entregable	- Testing - Correcciones sobre Preentrega - Armado del entregable
Entrega el 26/06/2018			

Para mostrar un cronograma más fiel a la realidad del desarrollo del proyecto, se decidió realizar un cuadro parecido al del cronograma propuesto, pero dividiendo las tareas en servidor, cliente y editor en lugar de la separación por desarrolladores.

Cronograma real

	Servidor	Cliente	Editor
Semana 1 (01/05)	Instalación e investigación de herramientas necesarias para el desarrollo (Box2D).	Instalación e investigación de herramientas necesarias para el desarrollo (SDL).	Instalación e investigación de herramientas necesarias para el desarrollo (QT).
Semana 2 (08/05)	Desarrollo del movimiento de los gusanos en Box2D	Instalación e investigación de herramientas necesarias para el desarrollo (SDL).	Prueba de concepto, cargando fondo de pantalla e implementando actions.
Semana 3 (15/05)	Modelado de los gusanos, jugadores, turnos y escenario	Prueba de concepto, mostrando una imagen y fondo de pantalla.	Implementación de drag & drop.
Semana 4 (22/05)	Modelados de las armas, explosiones y vida de los gusanos	-	Implementación lógica de posicionamiento de gusanos y vigas.
Semana 5 (29/05)	Comunicación entre cliente y servidor, sockets y threads	-	Desarrollo tamaño de escenario y conversión de pixels a metros.
Semana 6 (05/06)	Documentación y testing	Desarrollo del modelo y lógica de comunicación. Mínima muestra de imágenes.	Guardado y carga de mapas. Edición de tamaño de escenario.
Preentrega (12/06)			
Semana 7 (12/06)	Resolver bugs y agregar detalles de jugabilidad. Envío de nuevos mensajes al cliente	Muestra de imágenes. Resolver inconvenientes con memoria.	Corrección de bugs. Ajuste de la vista para que coincida con el cliente.
Semana 8 (19/06)	Resolver bugs y agregar detalles de jugabilidad. Envío de nuevos mensajes al cliente. Documentación. Revisión general del código.	Agregado de textos, imágenes y animaciones. Documentación. Revisión general del código.	Ajuste de la vista para que coincida con el cliente. Documentación. Revisión general del código.
Entrega (26/06)			

Inconvenientes encontrados

El principal inconveniente que nos encontramos fue el aprendizaje de herramientas completamente nuevas (como lo son las bibliotecas para interfaz gráfica utilizadas), utilizadas en un paradigma también nuevo (como lo es la programación orientada a eventos), con todo lo que esto conlleva. Esto sumado al escaso tiempo con el que se disponía fueron los mayores desafíos a superar durante el desarrollo. Adicionalmente, replantear la organización del proyecto a la mitad de este debido a la salida de uno de los integrantes actuó como un agravante de lo mencionado anteriormente.

Por otro lado, habiendo desarrollado gran parte del cliente, nos encontramos con que este utilizaba demasiada memoria, haciendo que la ejecución del mismo se detenga. Este problema está relacionado con lo mencionado en el párrafo anterior de tener que aprender una herramienta nueva durante el desarrollo, ya que costó bastante trabajo identificar en donde se producía el error.

Algo que tuvimos que tener muy presente a la hora de la integración de las aplicaciones fue el sistema de coordenadas que utiliza cada una de las bibliotecas utilizadas en cada aplicación. Sucede que Qt y SDL (las herramientas de interfaz de usuario) utilizan un sistema de coordenadas distinto al de Box2D (motor de física), por lo que se tuvieron que realizar conversiones para la comunicación entre sistemas.

Otra complicación que tuvimos, que puede sonar trivial comparado con las otras, fue con los sprites del juego. Se tuvo que modificarlos manualmente para poder utilizarlos debido a que no se adaptaban del todo a las necesidades que se tenían, lo cual implicó una inversión de tiempo no planeada.

Análisis de puntos pendientes

Consideramos que para que el producto final sea atractivo sería necesario implementar todos los recortes que se realizaron, ya que estos recortes derivaron en que la interfaz no sea muy amigable para el usuario final. Por ejemplo, la creación y unión de partidas tanto del servidor como del cliente sufre de esto. Una vez iniciado el juego, este no tiene sonidos, lo cual deteriora la experiencia del jugador; además de la baja cantidad de armas que se pueden utilizar.

Por otro lado, sería mejor que el cliente contara con más animaciones de las que tiene, de manera que la experiencia parezca más real.

Por último, sería una gran idea automatizar las pruebas de integración entre cliente y servidor, para aumentar considerablemente la velocidad de desarrollo y reducir la incertidumbre al introducir nuevos cambios.

Herramientas

Control de versiones

Para facilitar el desarrollo en equipo y el control las distintas versiones que fue teniendo el proyecto, se utilizó **git**, que es un sistema de control de versiones descentralizado. Particularmente, lo utilizamos a través del sitio web *github*, en donde creamos un repositorio en donde pusimos el código de todo el proyecto.

Editor de interfaz gráfica

Solo se utilizó un editor de interfaz para el desarrollo del editor. Para esto, se utilizó Qt Designer, que viene integrado con Qt Creator, el IDE de Qt, la biblioteca de uso general de interfaces gráficas.

Testing suite

Se utilizó **cxx** para realizar pruebas del servidor. Con ellas se probó todo el modelo desarrollado y la correcta integración y funcionamiento de Box2D.

Editor de texto

Se utilizaron Visual Studio Code y Atom como editores de texto para el desarrollo. También se utilizó el IDE Qt Creator para el editor, ya que resultaba cómodo por tener integradas las herramientas para el desarrollo con Qt.

Debug

Para verificar el uso correcto de memoria por parte de los distintos programas se utilizó el mem check de Valgrind.

Conclusiones

Como conclusión, nos llevamos una experiencia de cómo es trabajar con un cliente con fechas de entrega fijas, así como también de cómo aprender herramientas nuevas durante el desarrollo, como lo son las bibliotecas de interfaz gráfica utilizadas, el motor de física y el paradigma de programación orientada a eventos.

Como último comentario, resulta una pena que se haya puesto tanto esfuerzo por parte de los integrantes en un producto que le faltan cosas (mencionadas en el apartado *Análisis de puntos pendientes*) para ser un producto distribuible, aunque sea a baja escala.