

# Manual del Proyecto

Taller I Veiga - 2do Cuatrimestre 2017

Ayudante: Martín Di Paola

## Integrantes

- 96455 Matin Queija
- 96618 Damián Cassinotti
- 97131 Juan Pablo Rombolá

## Enunciado

El trabajo práctico de este cuatrimestre consistió en implementar una variante del clásico género Tower Defense, en donde varios enemigos emergen desde uno o varios portales y se mueven por caminos predefinidos hacia su destino y es el objetivo de los jugadores evitar que lleguen a destino.

Para tal fin, los jugadores deben cooperar entre sí y colocar torres de defensa que atacan automáticamente a los enemigos, así como también lanzar hechizos contra los enemigos directamente.

Los enemigos aparecen de a hordas cada cierto tiempo, dependiendo de cada nivel. Luego de haber acabado con todas las hordas, los jugadores logran la victoria; si al menos un enemigo logra llegar a destino los jugadores pierden.

## División de Tareas y Evolución del Proyecto

Para poder llevar a cabo el proyecto nos dividimos las tareas entre los tres miembros del grupo como se propone en el enunciado:

	<b>Alumno 1 Servidor - Modelo (Martín)</b>	<b>Alumno 2 Cliente - Modelo (Juan Pablo)</b>	<b>Alumno 3 Cliente - Editor (Damián)</b>
<b>Semana 1</b> (03/10/2017)	Draft del modelo (incluyendo lógica del juego, chat y partidas multijugador)	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> ).
<b>Semana 2</b> (10/10/2017)	Modelado del escenario, las hordas, las torres y los disparos.	Dibujado del escenario con proyección isométrica. Dibujado de animaciones (portales, torres, enemigos, disparos, explosiones)	Editor, creación de escenarios definiendo qué ambiente usar, y cómo son las hordas del nivel.

<b>Semana 3</b> (17/10/2017)	Finalización del modelado del juego.	Interfaz gráfica al seleccionar una torre (mostrar los detalles). Interfaz para el upgrade. Scrolling.	Permitirle al usuario dibujar sus propios escenarios: los terrenos, los caminos, los portales.
<b>Semana 4</b> (24/10/2017)	Lógica para la creación de partidas multijugador y múltiples partidas.	Interfaz para el lanzamiento de hechizos. Interfaz para el chat y la señal temporal.	Finalización del editor.
<b>Semana 5</b> (31/10/2017)	Soporte para la sala de chat. Finalización de la implementación multijugador.	Interfaz para la conexión con el servidor, elegir un nivel, crear/unirse a una partida. Pantallas de victoria y derrota.	Implementación del sistema de sonidos. Carga y descarga de niveles.
<b>Semana 6</b> (07/11/2017) <b>Preentrega</b>	- 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
<b>Preentrega el 14/11/2017</b>			
<b>Semana 7</b> (14/11/2017)	- Correcciones sobre Preentrega - Testing y corrección de bugs - Documentación	- Correcciones sobre Preentrega - Testing y corrección de bugs - Documentación	- Correcciones sobre Preentrega - Testing y corrección de bugs - Documentación
<b>Semana 8</b> (21/11/2017) <b>Entrega</b>	- Testing - Documentación - Armado del entregable	- Testing - Documentación - Armado del entregable	- Testing - Documentación - Armado del entregable
<b>Entrega el 28/11/2017</b>			

A pesar de los esfuerzos realizados, no pudimos seguir el cronograma como se esperaba. Lo que pasó fue que nos fuimos retrasando un poco cada semana, al punto de ir desplazando cada vez más lo que serían las Semanas 6, 7 y 8.

Igualmente, para el día de la Preentrega pudimos presentar una versión jugable del trabajo, obviamente, con menos features.

## Inconvenientes encontrados

Tuvimos varios problemas a lo largo del proyecto. Uno de ellos fue encontrar la manera de correr GTK para el Lobby y SDL para el juego. Optamos por usar las dos librerías porque SDL permite tener control de todo lo que se muestra por pantalla a niveles muy detallistas,

sumado a que sirve para mostrar animaciones, mientras que GTK está orientado a GUI, es decir a paneles y botones, que poco nos iban a servir para generar una buena experiencia de usuario para el jugador durante el juego. Pudimos haber hecho todo en SDL, pero justamente para la parte del Lobby se necesitan muchas herramientas de GUI que SDL no trae consigo e implementarlas correctamente nos iba a llevar mucho tiempo.

Aparte de eso cada uno se trabó o tuvo problemas en algún momento con su parte. Por ejemplo, con la vista costó mucho hacer el algoritmo de movimiento de las unidades para que se viera fluido y soportase los distintos mensajes del servidor.

## Análisis de puntos pendientes

Por suerte, en esta sección podemos decir que no nos quedaron puntos pendientes. Es más, pudimos agregar features que no se pedían como por ejemplo que el juego y el editor se puedan usar con touchscreen, tener zoom en el mapa y tener quickcast para los hechizos y botones de colocar torre.

Un tema con el que no quedamos tan conformes es el uso de memoria del lado del cliente. Esto se debe a que la librería SDL pierde memoria de solo iniciarla, por lo que mucho no pudimos hacer al respecto. Por ejemplo, con solo abrir y cerrar el editor (sin realizar ninguna acción), ya vemos un leak de aproximadamente 40MB.

## Herramientas

Las herramientas que utilizamos a lo largo del proyecto fueron:

- CLion (IDE) para C++
- Github para control de versiones y repositorio
- Valgrind para encontrar errores de memoria
- Photoshop y GIMP para todos y cada uno de los gráficos que se ven en el proyecto

## Conclusiones

En conclusión, fue un trabajo muy laborioso, agradable y entretenido de hacer, la elección de TP fue realmente acertada. Nos esforzamos por hacer un juego que nos gustara a nosotros, sabiendo que lo primero es aprobar, pero siempre teniendo presente que el resultado final fuese uno con el que estuviéramos satisfechos.