

# Manual de Proyecto

CS2D

[7542] Taller de Programacion I  
Primer cuatrimestre 2021

Alumno	Padrón
Santiago Pablo Fernandez Caruso	105267
Federico Elias	96105
Mateo Capón Blanquer	104258

## Índice

<b>1. Integrantes - División de tareas</b>	<b>2</b>
<b>2. Evolución del proyecto</b>	<b>2</b>
<b>3. Inconvenientes encontrados</b>	<b>2</b>
3.1. Tiempo para realizar el proyecto . . . . .	2
3.2. Dificultades con el diseño y la experiencia de usuario . . . . .	2
3.3. Eficiencia . . . . .	3
<b>4. Análisis de puntos pendientes</b>	<b>3</b>
4.1. Resolución de pantalla configurable . . . . .	3
4.2. Animaciones y detalles de renderizacion . . . . .	3
4.3. Mas indicadores . . . . .	3
4.4. Velocidad de las Personitas . . . . .	3
<b>5. Herramientas</b>	<b>3</b>
<b>6. Conclusiones</b>	<b>4</b>

## 1. Integrantes - División de tareas

- **Capón Blanquer, Mateo:** Servidor y protocolo de comunicación.
- **Elias, Federico:** Editor, Initiator, FinalPhase, YAML Parser para cargar los mapas en el servidor, parte de las armas en el servidor, musica y sonidos del lado del cliente.
- **Fernandez Caruso, Santiago Pablo:** Parte grafica y logica del cliente (etapa juego), parte del protocolo de comunicacion.

## 2. Evolución del proyecto

- Semana 1: Primer modelo del editor, configuración del Cmake, bosquejos e ideas sobre la conexión cliente-servidor y protocolo.
- Semana 2: Archivos de configuración YAML, lógica de las armas, avances en la conexión, incluimos Box2d.
- Semana 3: Inicio de desarrollo del Login, creamos clases necesarias para usar componentes de SDL, implementamos los primeros eventos y arreglamos bugs en la conexión, se comienza a implementar la lógica de los movimientos y los ataques, avances en el editor, renderizacion de mapas del lado del cliente.
- Semana 4: Manejo de eventos de sdl, renderizacion de armas, lógica de compra de armas, el jugador rota y se mueve, se implementa el hud, lógica de grab/drop weapon, implementamos la cámara.
- Semana 5: Implementación de la fase inicial y la fase final, corrección de errores, agregamos sonidos y animaciones, lógica de ataques terminada.
- Semana 6: Fase inicial y final terminada, se implementa el stencil, arreglo de bugs.
- Semana 7: Arreglo de errores y funcionalidades que faltaban, realizamos la documentación y revisión general del código.

## 3. Inconvenientes encontrados

### 3.1. Tiempo para realizar el proyecto

A pesar de que en lineas generales sabíamos casi siempre como desarrollar el proyecto, fue muy difícil manejar los tiempos. Mas aun, considerando que las primeras semanas debimos recuperar el TP3 y hacer otros TPs y rendir exámenes de otras materias.

Tuvimos que decidir priorizar ciertas tareas por sobre otras. Desde ya, nos hubiese gustado documentar todo el código con PRE y POST condiciones; tener diseños más robustos; hacer tests; y mejorar muchos aspectos gráficos. Sin embargo, decidimos centrarnos en cumplir con todos los features pedidos.

### 3.2. Dificultades con el diseño y la experiencia de usuario

Al no haber tenido nunca materias de diseño ni de ux fue difícil adaptarnos al desarrollo de la parte gráfica, es por esto que las diferentes aplicaciones del proyecto (InitialPhase, FinalPhase, Initiator, Editor) son relativamente básicas en términos de aspecto y usabilidad.

### 3.3. Eficiencia

En un principio no confiábamos en la rápido que puede llegar a trabajar la cpu. Nos complicamos demasiado en intentar hacer un modelo muy eficiente, cuando en todas las iteraciones del gameloop se pueden mandar unos pocos bytes de mensajes sin sobrecargar la cpu.

## 4. Análisis de puntos pendientes

Si bien el trabajo cumple con todos los features que se plantean en el enunciado, durante el desarrollo surgieron ideas y optimizaciones que, de ser implementadas, ofrecerían una mejor experiencia de usuario. Algunas son:

### 4.1. Resolución de pantalla configurable

Si bien el usuario puede modificar el ancho y alto de la pantalla en el archivo de configuración del cliente falta implementar una transformación que reescale los tiles y sprites de manera que sin importar que resolución elija el usuario todos vean la misma porción de mapa.

### 4.2. Animaciones y detalles de renderizacion

Nos hubiera gustado mostrar una animación mostrando desde donde proviene el ataque al ser atacados. También hay detalles para corregir en la renderizacion del arma del player ya que no se ve donde debería.

### 4.3. Mas indicadores

No llegamos por falta de tiempo pero hubiera estado bueno mostrar mas indicadores cuando ocurren errores, mensajes mas precisos/ Tambien mostrar mas información sobre los usuarios, por ejemplo: los nombres, quien mata a quien, etc.

### 4.4. Velocidad de las Personitas

La realidad es que con poco tiempo de desarrollo podríamos hacer la velocidad configurable, sin embargo esto no es posible ahora, dado que los objetos se están moviendo a razón de 100 metros por segundo en Box2D. Lo que deberíamos corregir es simplemente cambiarle el metros a otra unidad de medida adecuada.

## 5. Herramientas

Se listan a continuación las distintas herramientas que se utilizaron en el desarrollo del proyecto:

- Git
- QtCreator
- Overleaf
- Gdb
- Valgrind
- Cmake
- SDL2

## 6. Conclusiones

Debemos reconocer que aprendimos muchísimo en el desarrollo del presente proyecto. Nos topamos con las dificultades diarias de un programador: aprender a trabajar en equipo, lidiar con los tiempos de entregas, tener que decidir prioridades, aprender nuevas herramientas por nuestra cuenta sin que sea enseñado por un profesor (YAML, Box2D, y algunas funciones de SDL y QT), entre otras.

Fue muy interesante aprender sobre las conexiones cliente servidor, y el uso de hilos.