

# ***Rocket league***

## Manual de Proyecto

[95.08] Taller de Programación 1

Curso Veiga

Segundo cuatrimestre de 2022

Alumno	Padrón	Mail
Amaya , Santos Emanuel	96891	samaya@fi.uba.ar
Romero Gonzales, Alan Xochtiel	108316	alromero@fi.uba.ar
Bragantini, Franco Julian	97190	fbragantini@fi.uba.ar

# Índice

<b>Índice</b>	<b>1</b>
<b>2.División de Tareas</b>	<b>1</b>
<b>3.Evolución</b>	<b>1</b>
<b>4.Complicaciones</b>	<b>2</b>
<b>5.Herramientas</b>	<b>2</b>

## 1.Enunciado

El presente trabajo consiste en implementar una variante 2D del juego Rocket league, mismo consiste en un partido de fútbol, en donde los jugadores son autos que pueden hacer distintos movimientos para poder empujar una pelota dentro del arco contrario.

Será implementado en 2D en un modo multijugador en línea, donde cada jugador podrá controlar uno de los autos del equipo.

El juego tendrá una simulación física para la trayectoria de la pelota, los autos y los tiros especiales usando el framework Box2D.

## 2.División de Tareas

Iniciamos viendo que es lo que cada integrante tenía para ofrecer al team, dejando la primera semana para preparación de ideas, aquí se remarcó que Santos Amaya tenía un bosquejo bastante importante que fue la base del diseño para el Servidor.

Quedando a partir de esa semana dividir el soporte de física brindado por Box2D a cargo de Alan Romero Gonzalez, y el cliente con SQL2pp a cargo de Franco Bragantini.

## 3.Evolución

(18/10 a 25/10)Primera semana revisar enunciado y generar ideas.

(25/10 al 8/11)siguiente 2 semanas Amaya Santos por parte del Server fue acoplar el tp2, tapando lo del menu y pisando mucha funcionalidad para implementar nueva, mientras los demás chicos experimentaban con sus librerías y preparaban su protocolo, mientras tanto se implementó un SnapShot como encargado de contener el mensaje de Server al cliente.

(8/11 al 15/11 ) Semana 4(semmana antes de la entrega) corríamos el server y el cliente harcodeando bastantes cosas teníamos visual de juego y motor de físicas acoplado al Server. pero limitados a mover y saltar sin pelota activa, y con desfases en mensajes.

(15/11 al 22/11) Semana 5(semmana de entrega) llegamos a tener un juego fluido pero sin ningún feature de saltos cubiertos y por el apuro de correr, mezclar los tp no respetamos RAII y no liberamos memoria.

(22/11 al 29/11)Semana 6 no avanzamos mucho en general de ningun lado ya que (no es que nos aliamos para no codear) sino que pasaron cosas (FIUBA COSAS y cosas personales en una caso)

(29/11 al 6/12) Semana intensa iniciamos el viernes a la tarde faltaban todos los features de tiros, no teníamos el soporte para hacerlos con el protocolo actual, cambiamos el protocolo el domingo, mientras se terminaba de cerrar leaks en el server se inyectaba en el motor de físicas los sensores, se refactorizar todo el cliente modularizados, para el lunes antes de la entrega poder probar los tiros especiales; finalizando el martes con meter el fin de juego con estadísticas.

## 4.Complicaciones

Protocolo en el Cliente cero escalable sin duda, modificado el 3 días antes de la entrega.Tuvimos inconvenientes con el protocolo ya que del lado del server a veces llenabamos mal el SnapShot o del lado del cliente casteaba mal, no se sabía y tal vez luego era un tema de leer más un byte.

Tampoco pensamos en los errores de leaks, nos teníamos seguros al implementar RAII hasta 7 días antes de la entrega final.

## 5.Herramientas

Por otro lado, se utilizó CxxTest como librería para generar pruebas y testear el código en la medida que fuera necesario. integrada exitosamente el último día con el test a SnapShot.

Valgrind fue la principal herramienta para la detección y tratamiento de errores, sobre todo en aquellas operaciones donde se contaban con varios hilos de ejecución corriendo al mismo tiempo. Claramente se comenzó a usar 3 días antes de la entrega. Antes se usó GDB pero su uso era complicado, también por que no supimos integrarlos al VSCode.

## 8.Conclusiones

Fue un tp interesante, extenso nos confiamos por un buen inicio, pero faltaba camino por recorrer y nos sigue faltando.

Tuvimos un desarrollo a la par donde cada vez que nos trabamos, preguntamos al grupo si veía el error o si tenía idea mejor para implementar solamente dando las ideas, acciones y requerimientos que tengo cumplir sin entrar al 100% la comprensión de código ajeno, y el caso de complicación por mala respuesta buscábamos el problema en conjunto, ya al correr en simultáneo y tener tantas dependencias tanto en el Server como en el Client es fácil equivocarse.

Por parte de el uso de las librerías externas estamos contento con el uso que les dimos y las integraciones que hicimos.