

## Rocket league

Objetivos	<ul style="list-style-type: none"><li>• Afianzar los conocimientos adquiridos durante la cursada.</li><li>• Poner en práctica la coordinación de tareas dentro de un grupo de trabajo.</li><li>• Realizar un aplicativo de complejidad media con niveles aceptables de calidad y usabilidad.</li><li>• Aprender tecnologías desconocidas mediante lectura de documentación oficial de estas.</li></ul>
Entregas	<ul style="list-style-type: none"><li>• <b>Entrega obligatoria:</b> clase 16.</li><li>• <b>Entrega con correcciones:</b> clase 14.</li></ul>
Criterios de Evaluación	<ul style="list-style-type: none"><li>• Criterios de ejercicios anteriores.</li><li>• Construcción de un sistema Cliente-Servidor de complejidad media.</li><li>• Empleo de buenas prácticas de programación en C++.</li><li>• Coordinación de trabajo grupal.</li><li>• Planificación y distribución de tareas para cumplir con los plazos de entrega pautados.</li><li>• Cumplimiento de todos los requerimientos técnicos y funcionales.</li><li>• Facilidad de instalación y ejecución del sistema final.</li><li>• Calidad de la documentación técnica y manuales entregados.</li><li>• Buena presentación del trabajo práctico y cumplimiento de las normas de entrega establecidas por la cátedra (revisar criterios en sitio de la materia).</li></ul>

**El trabajo es grupal:** debe ser de autoría completamente del grupo. Cualquier forma de **plagio es inaceptable:** copia de otros trabajos, copias de ejemplos de internet o copias de tus trabajos anteriores de otras materias (self-plagiarism). Si usas material de la cátedra deberás dejar en claro la fuente y dar crédito al autor (a la materia).

# Índice

[Índice](#)

[Introducción](#)

[Descripción](#)

[Autos](#)

[Salto](#)

[Flips](#)

[Turbo](#)

[Disparos](#)

[Flip shot](#)

[Red shot](#)

[Purple shot](#)

[Gold shot](#)

[Partido](#)

[Escenario](#)

[Cámara](#)

[Colisiones](#)

[Gui](#)

[Animaciones](#)

[Sonidos](#)

[Música ambiente](#)

[Configuración](#)

[Aplicaciones Requeridas](#)

[Cliente](#)

[Servidor](#)

[Tests](#)

[Restricciones](#)

[Referencias](#)

# Introducción

Rocket league: sideswipe es un juego para celular que adapta el famoso juego 3D en una estética 2D vista de perfil [1]. El juego 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.

El scope del tp busca realizar un juego simplificado, para computadora. Este será 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].

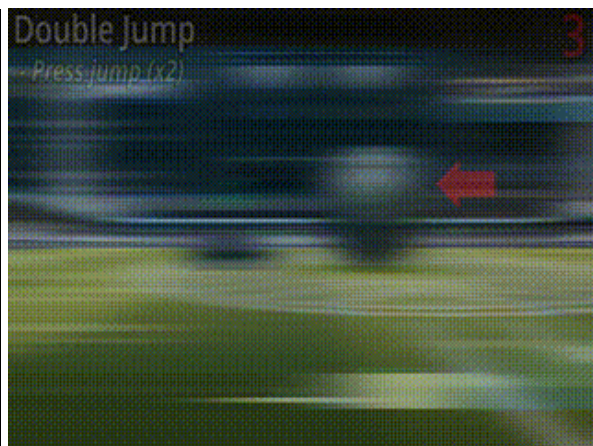
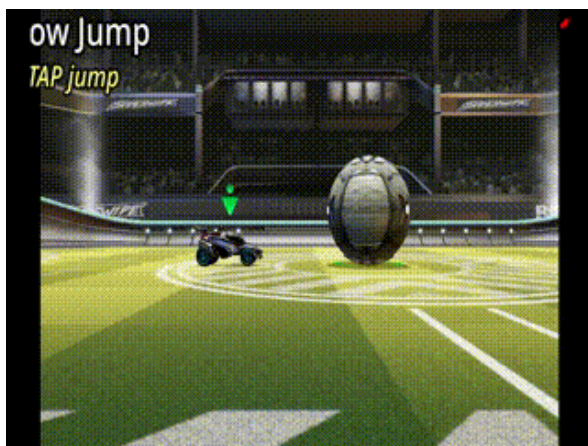
## Descripción

### Autos

El auto es el elemento que podrá ser controlado por el jugador. Podrá moverse hacia los costados (izquierda y derecha). Cuenta con un dispositivo que le permite realizar saltos en el aire para hacer piruetas. Mientras se encuentra en el aire, el auto puede impulsarse una segunda vez para llegar a alturas más alejadas. Cuenta además con un tanque de nitro para tomar altas velocidades. Este segundo impulso puede generar unos disparos especiales en función de en qué parte del auto impacta la pelota (más de eso en la sección [disparos](#))

### Salto

El auto puede realizar un salto vertical para llegar a mayores alturas y así golpear la pelota antes de que esta toque el suelo. Una vez en el aire, tiene la posibilidad de realizar un segundo salto. Una vez usado este, no podrá volver a usarlo **a menos de que vuelva a tocar el suelo del terreno de juego.**



## Flips

El segundo salto, puede ser usado mientras se direcciona el auto en algún sentido. Si se aprieta el botón de salto y una dirección particular, el auto realizará un flip. Este movimiento es muy útil ya que es un impulso para llegar más rápido a su destino.



## Turbo

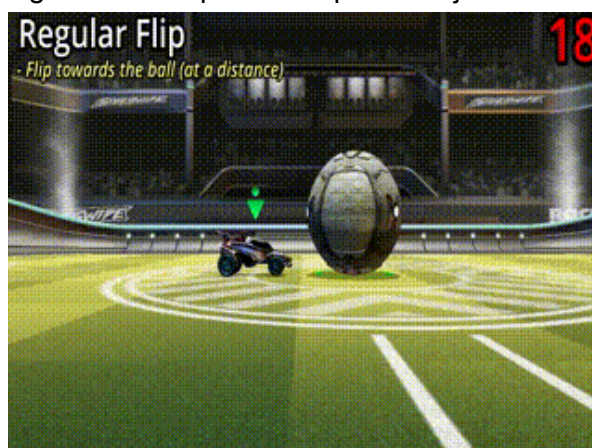
El auto posee un tanque **limitado** de nitrógeno líquido que puede usar para tomar grandes velocidades y llegar antes a golpear la pelota. El tanque se recargará luego de cierto tiempo en el que el **auto se encuentre en el suelo**. Este turbo le sirve además para volar por los cielos si este se encuentra en el aire. La dirección de vuelo estará dado por a donde esté apuntando la trompa del auto.

## Disparos

El auto puede golpear a la pelota simplemente acercadosé a ella a velocidad. Sin embargo, existen algunos disparos especiales si se cumplen ciertas condiciones.

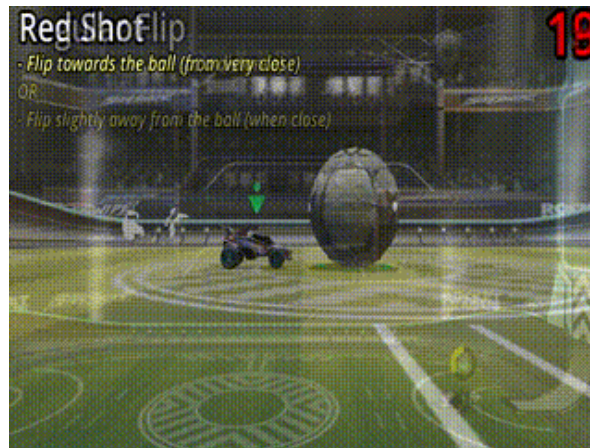
## Flip shot

Este es el disparo mas común, resulta de hacer un flip hacia la pelota a una distancia moderada. Este golpe le genera un impulso a la pelota bajo.



### Red shot

Es el disparo más fuerte. Genera un impulso a la pelota alto. Se realiza haciendo un flip bien pegado a la pelota, para que el impacto a esta se realice con la trompa del auto. El flip hacia adelante también genera que el auto avance en la dirección del golpe, por lo que quedará en una posesión ofensiva para volver a golpear el esférico.



### Purple shot

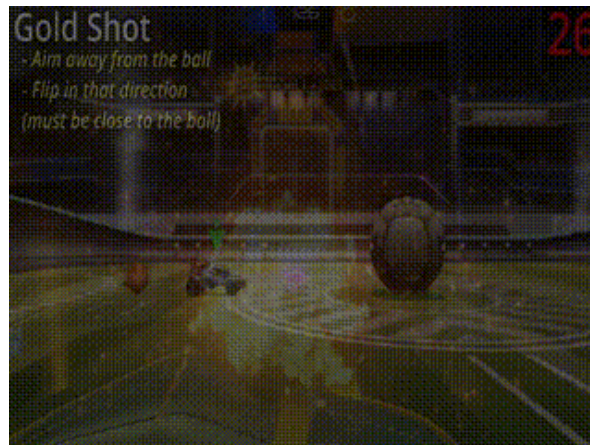
Se realiza golpeando el balón con la carrocería del auto. Se debe apretar el botón de salto para que el impulso de este genere un impulso medio a la caprichosa. Este golpe genera un impulso en la dirección contraria para el auto, lo que resulta útil para estar posicionado para un potencial contra ataque.



### Gold shot

Este disparo genera un impulso alto a la pelota. Se realiza generando un impulso en la dirección contraria al esférico, cercano a este. Los propulsores del mecanismo de salto del auto generan que el balón salga disparado hacia la dirección contraria. Muy útil pero de difícil ejecución. Un movimiento para expertos.





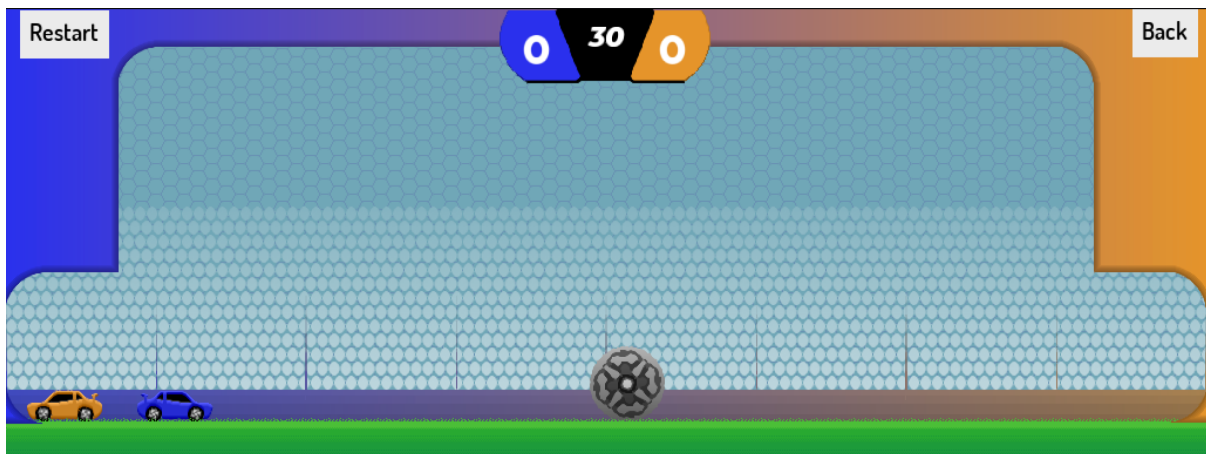
## Partido

La partida durará 3 minutos (configurable, ver sección [configuración](#)). Luego de finalizado, se mostrará un estadística simple con la tabla de goleadores, asistencias y atajadas.

## Escenario

El escenario debe ser un fondo 2D a su gusto que muestre una cancha de fútbol vista de perfil. Puede ser tan complejo y detallado como el juego original o simple como el de la figura 2.





## Cámara

El juego contará con una cámara fija que muestre la cancha en un corte de perfil.

## Repeticiones

Luego de cada gol, el juego mostrará una repetición del gol a modo de highlight.

## Colisiones

Los autos no podrán chocarse entre sí, pero sí pueden interactuar con el terreno de juego (pueden impactar la pelota y pueden andar por las paredes).

## Interfaz gráfica

El juego mostrará durante la partida el tiempo restante de juego y el marcador del partido. Para el auto del jugador que lo controla, además se mostrará una barra indicando el nitrógeno restante que puede usar.

## Animaciones

El juego no debe mostrar imágenes estáticas sino pequeñas animaciones para darle mayor realismo:

- El movimiento de los autos: cuando se giran en otra dirección, y el humo que desprende al activar el nitrógeno.
- La pelota girará cuando esta se encuentre en movimiento.
- Las explosiones cuando se realiza un gol.

Se dejan como sugerencias algunos sprites, aunque si el desarrollador lo desea puede

elegir lo que les parezca mejor [3].

## Sonidos

Como todo juego se debe reproducir sonidos para darle realismo a los eventos y acciones que suceden[4]:

- Cuando hay un salto.
- Cuando se utiliza el nitro.
- Cuando se realiza un gol
- Cuando se golpea la pelota

Si la cantidad de eventos que suceden es muy grande, algunos sonidos pueden ser evitados para no saturar al jugador con tanta información.

## Música ambiente

El juego debe reproducir una música ambiente, con un volumen relativamente bajo.

## Configuración

Todos los atributos del juego (velocidad, altura de salto, fuerza gravitatoria, elasticidad de la pelota, etc) y cualquier otro parámetro deben ser configurables por archivo.

*Es importante que todos los parámetros sean configurables: permite que se ajusten para tener un juego más balanceado y divertido a la vez que le permite a los docentes realizar pruebas.*

## Aplicaciones Requeridas

### Cliente

Se deberá implementar un cliente gráfico para que el usuario pueda conectarse al servidor, crear o unirse a una partida y poder jugar.

### Servidor

Se deberá implementar un servidor con soporte de múltiples partidas en simultáneo. Deberá poder indicarle a los clientes que se conecta qué partidas ya están creadas y están disponibles para que el usuario pueda unirse a alguna de ellas.

Al momento de iniciar la partida el servidor deberá asignar de forma aleatoria a cada



jugador su equipo correspondiente. Este equipo será identificado con un color particular.

## Tests

Tanto el cliente como el servidor deben contar con una suite de tests unitarios para asegurar un estándar de calidad de la entrega. De forma obligatoria se deberá contar con:

- El protocolo de comunicación completamente testeado (100%)
- Lógica de manejo de partidas parcialmente testeada (60%)

## Restricciones

La siguiente es una lista de restricciones técnicas exigidas:

1. El sistema se debe realizar en C++17 utilizando librerías *gtkmm*, *SDL* y/o *qt*.
2. Los archivos de configuración deben ser almacenados en formato YAML. A tal fin, y con el objetivo de minimizar tiempos y posibles errores, se permiten distintas librerías externas (consultar sitio de la cátedra). No está permitido utilizar una implementación propia de lectura y escritura de YAML.
3. Para la simulación de la física del juego se debe usar el framework Box2D [2].
4. Es condición necesaria para la aprobación del trabajo práctico la entrega de la documentación mínima exigida (consultar sitio de la cátedra). Es importante recordar que cualquier elemento faltante o de dudosa calidad pone en riesgo la aprobación del ejercicio.
5. Para la test suite, se sugiere la utilización de alguna librería del estilo xUnit [7]. Si bien existen varias librerías disponibles en lenguaje C++, se recomienda optar por Gtest[8], CxxTest [9] o CppUnit [10] o Catch2 [11].

## Referencias

[1] Rocket league: [https://es.wikipedia.org/wiki/Rocket\\_League](https://es.wikipedia.org/wiki/Rocket_League)

[2] Box2D: <https://box2d.org/documentation/>

[3] Sprites:

Explosiones:

[https://www.kindpng.com/imgv/iiiTmRh\\_explosionfull-magic-explosion-sprite-sheet-hd-png-download/](https://www.kindpng.com/imgv/iiiTmRh_explosionfull-magic-explosion-sprite-sheet-hd-png-download/)

[https://www.kindpng.com/imgv/hRRiim\\_drawn-explosions-sprite-transparent-explosion-sprite-sheet-hd/](https://www.kindpng.com/imgv/hRRiim_drawn-explosions-sprite-transparent-explosion-sprite-sheet-hd/)

[https://www.kindpng.com/imgv/ioxxRRi\\_smoke-explosion-sprite-sheet-hd-png-download/](https://www.kindpng.com/imgv/ioxxRRi_smoke-explosion-sprite-sheet-hd-png-download/)

[https://www.kindpng.com/imgv/Tobwom\\_smoke-sprite-sheet-transparent-hd-png-download/](https://www.kindpng.com/imgv/Tobwom_smoke-sprite-sheet-transparent-hd-png-download/)

Autos:

[https://www.sprisers-resource.com/genesis\\_32x\\_scd/ayrtonsennassupermonacogpii/sheet/163292/](https://www.sprisers-resource.com/genesis_32x_scd/ayrtonsennassupermonacogpii/sheet/163292/)

[https://www.sprisers-resource.com/game\\_boy\\_gbc/vrallychampionshipeditionedition99/sheet/60856/](https://www.sprisers-resource.com/game_boy_gbc/vrallychampionshipeditionedition99/sheet/60856/)

[https://www.sprisers-resource.com/game\\_boy\\_advance/simpsonsroadrage/sheet/168209/](https://www.sprisers-resource.com/game_boy_advance/simpsonsroadrage/sheet/168209/)

[https://www.sprisers-resource.com/game\\_boy\\_gbc/vrallychampionshipeditionedition99/sheet/60880/](https://www.sprisers-resource.com/game_boy_gbc/vrallychampionshipeditionedition99/sheet/60880/)

[4] Sonidos:

<https://www.youtube.com/watch?v=gxfyQnd-0ew>

<https://www.youtube.com/watch?v=qWy6xrgBxic&list=PLEJLxXa8kpYc4wT2GK4mo80G0reLybgGV&index=29>

[5] YAML: <https://es.wikipedia.org/wiki/YAML>

[7] Frameworks XUnit: <http://en.wikipedia.org/wiki/XUnit>

[8] Gtest: <https://github.com/google/googletest>

[9] CxxTest: <http://cxxtest.com/>

[10] CppUnit: [http://sourceforge.net/apps/mediawiki/cppunit/index.php?title=Main\\_Page](http://sourceforge.net/apps/mediawiki/cppunit/index.php?title=Main_Page)

[11] Catch2: <https://github.com/catchorg/Catch2>