It IS Rocket Science!

The physics and networking of Rocket League

Jared Cone
Lead Gameplay Engineer, Psyonix

GAME DEVELOPERS CONFERENCE® | MARCH 19–23, 2018 | EXPO: MARCH 21–23, 2018 #GDC18

UBM

Agenda

- Physics Engine
- Vehicle Tuning
- Networking

Rocket League Goals

- Fast, responsive vehicles
- Consistent, controllable physics
- Competitive over internet

PHYSICS ENGINE
VEHICLE TUNING
NETWORKING

Physics Engine

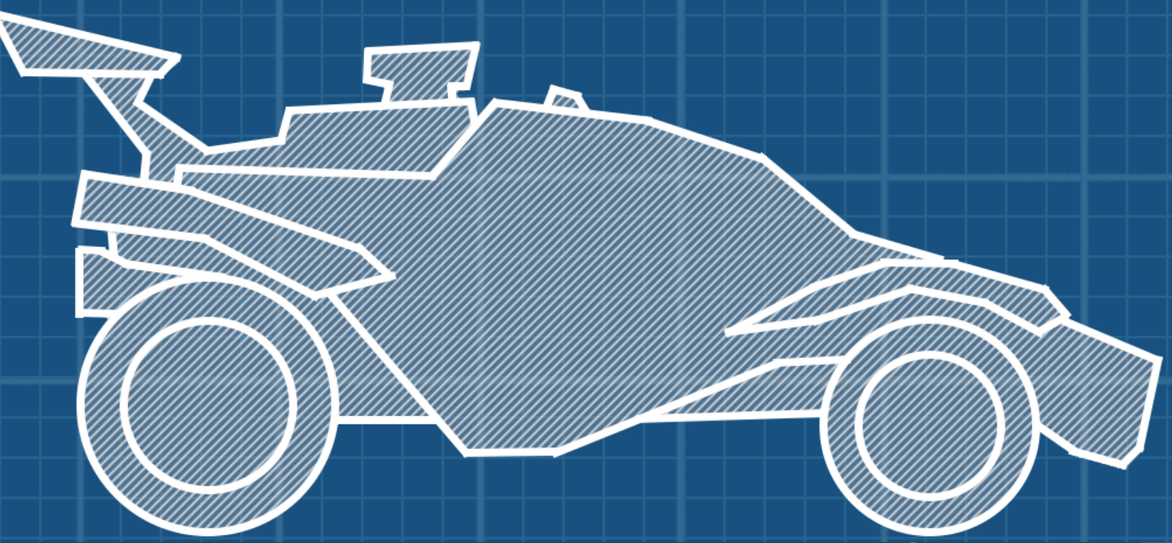
- Bullet Physics Engine
  - Open-source: debuggable, modifiable
  - Fast single-threaded simulation
  - ~1 week integration
- Discrete collision detection
- Fixed tick rate (120hz, 8.33ms)

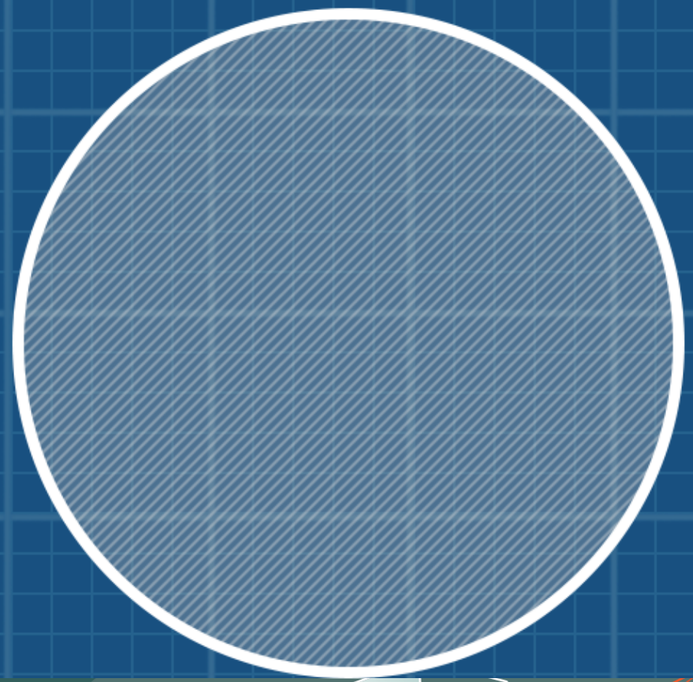

60 hz

Physics Steps

60hz

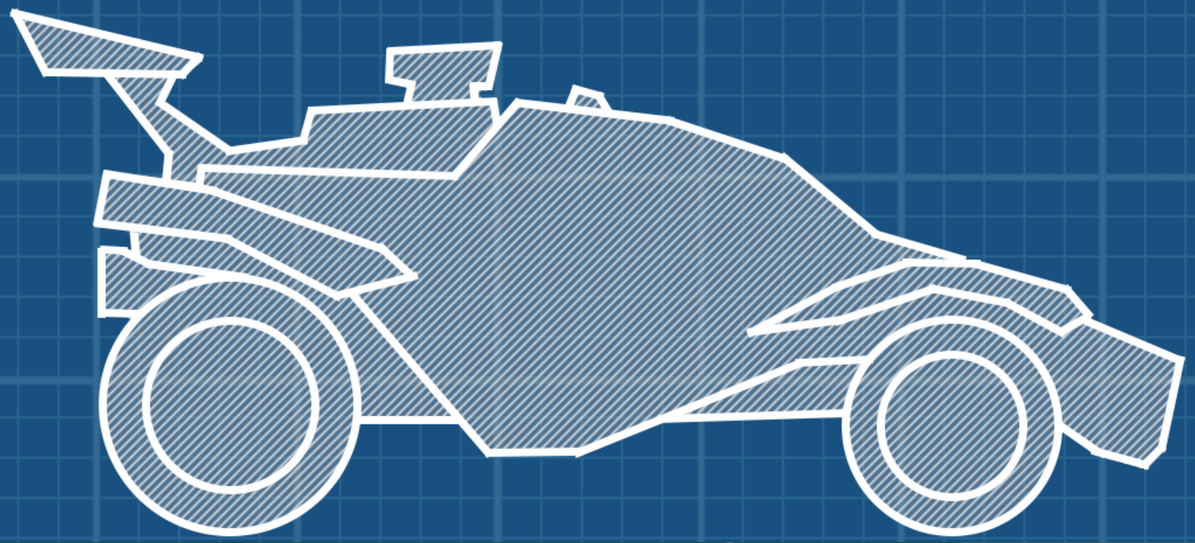
Physics Steps

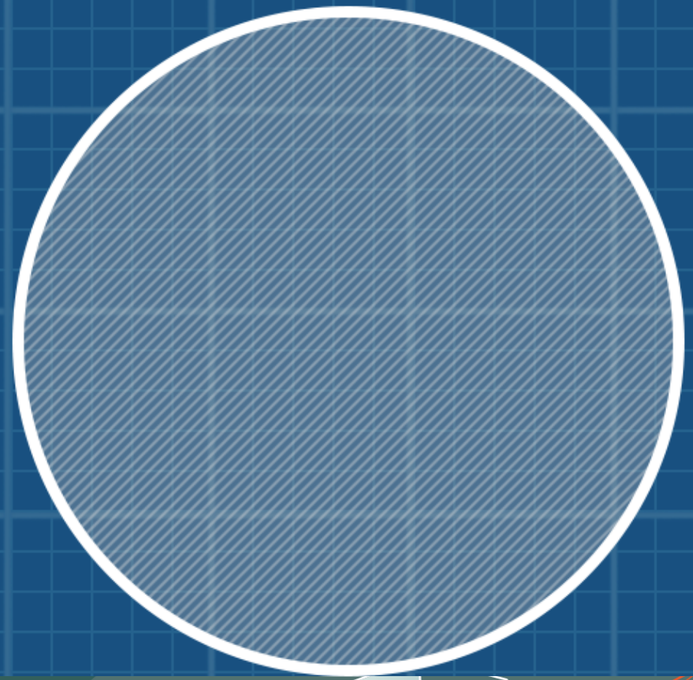




60hz

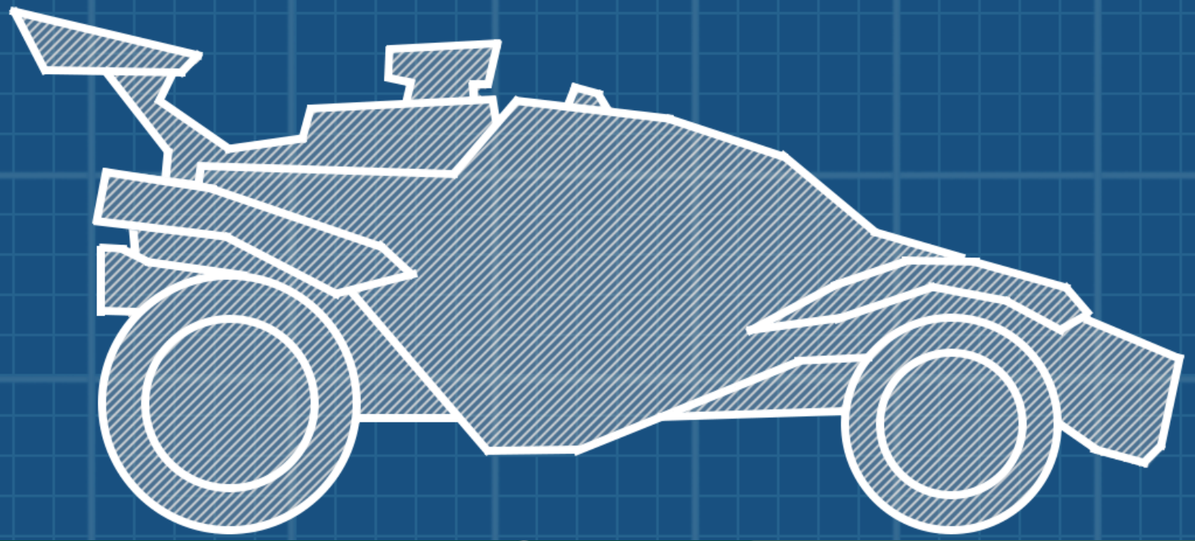
Physics Steps

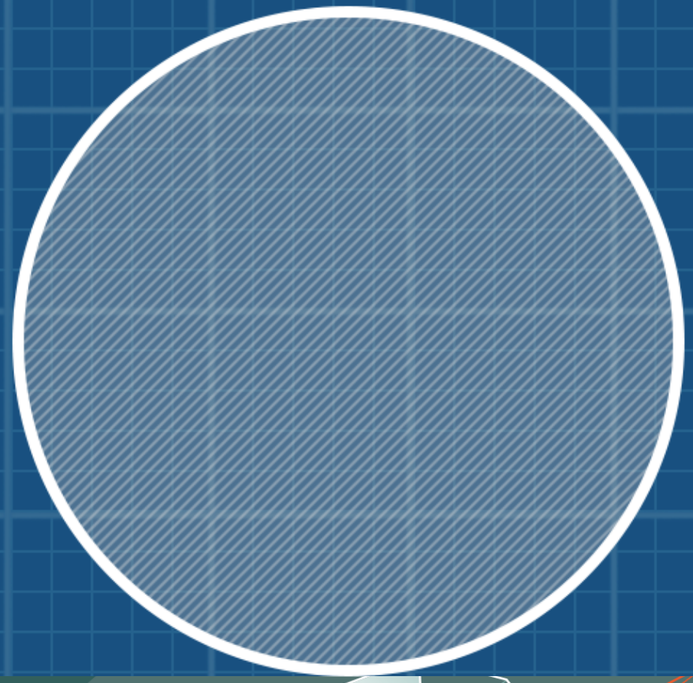




60hz

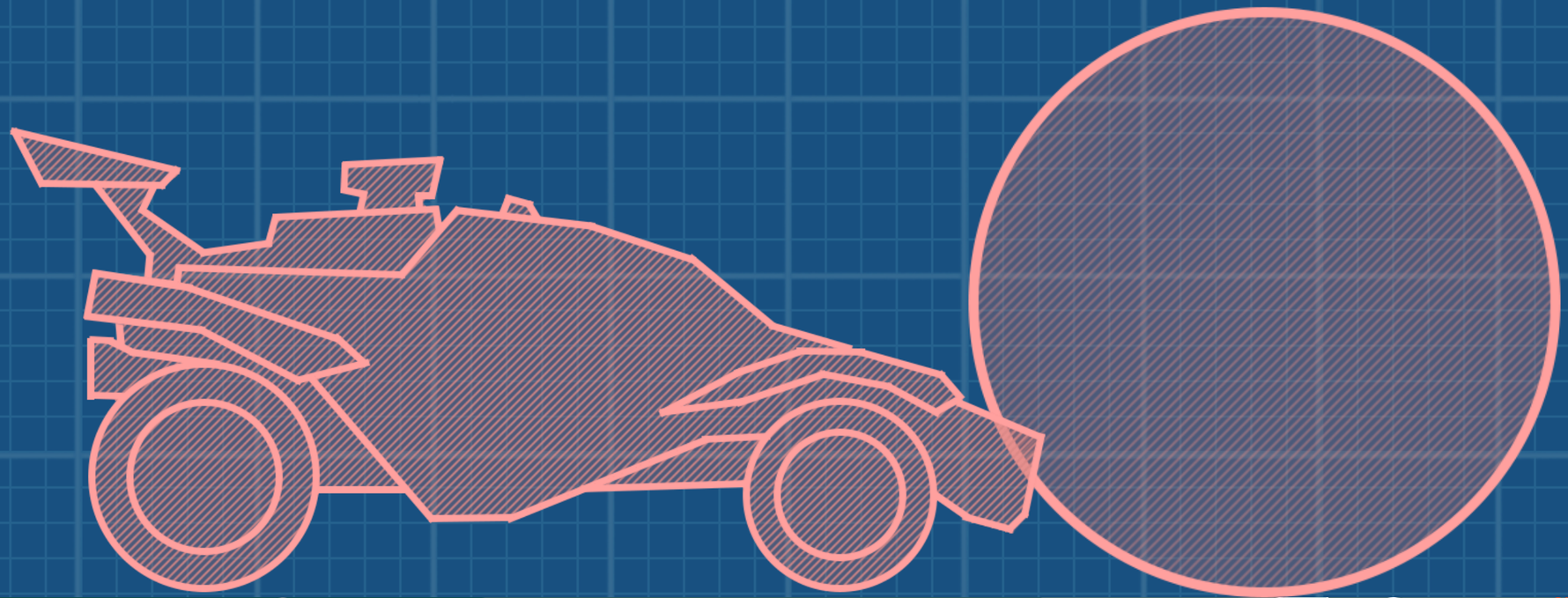
Physics Steps





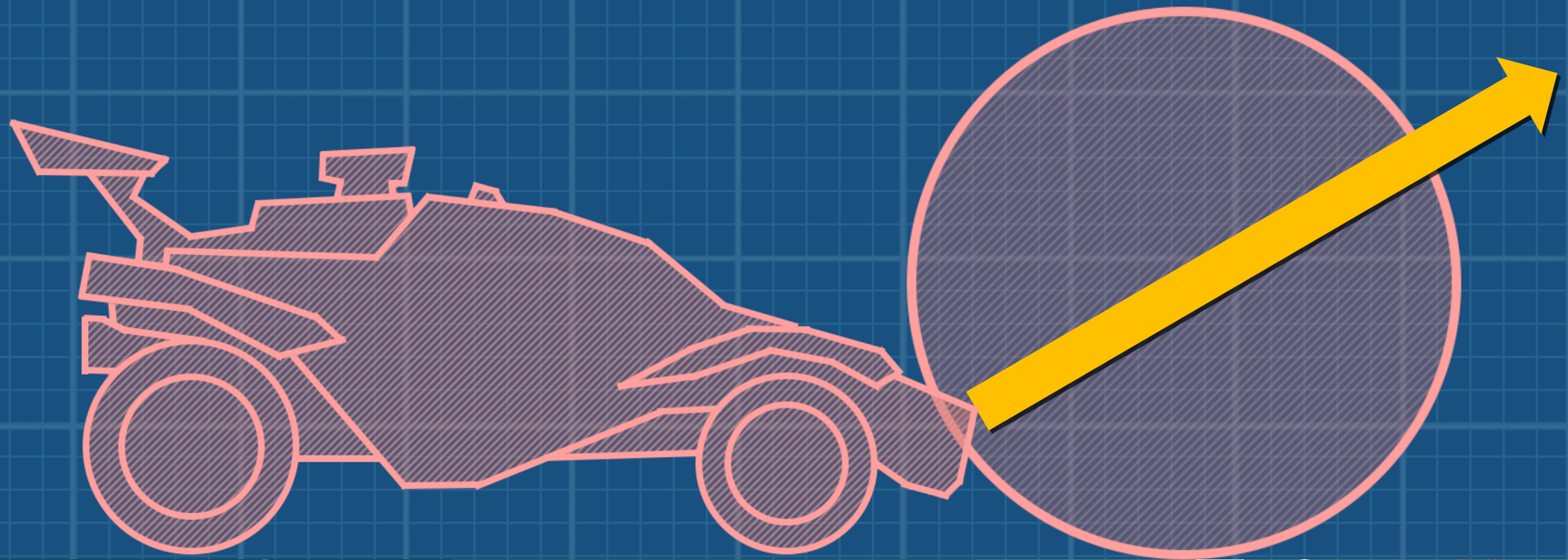
60hz

Physics Steps



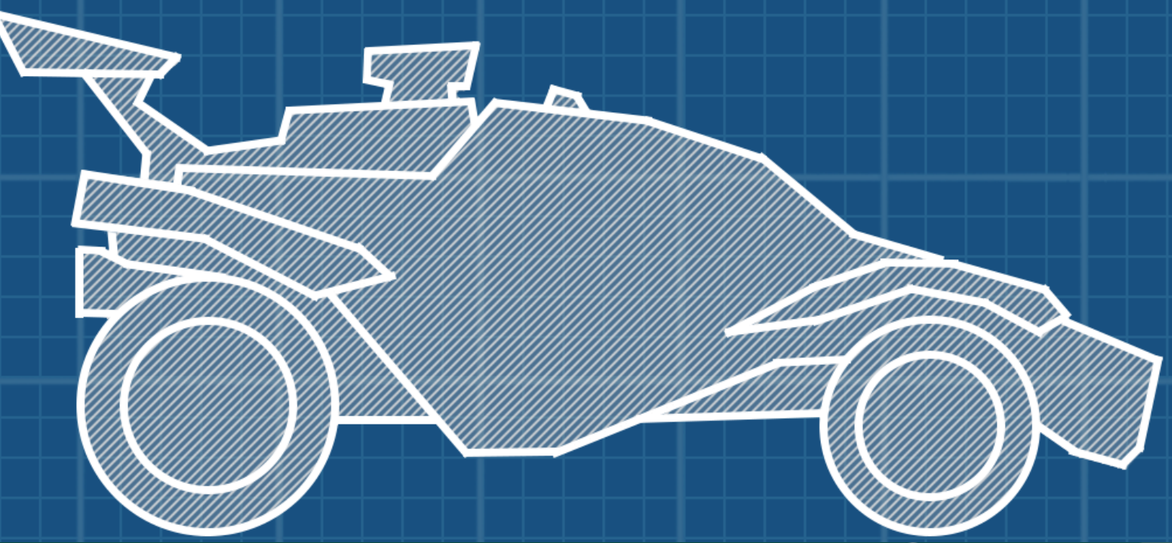
60hz

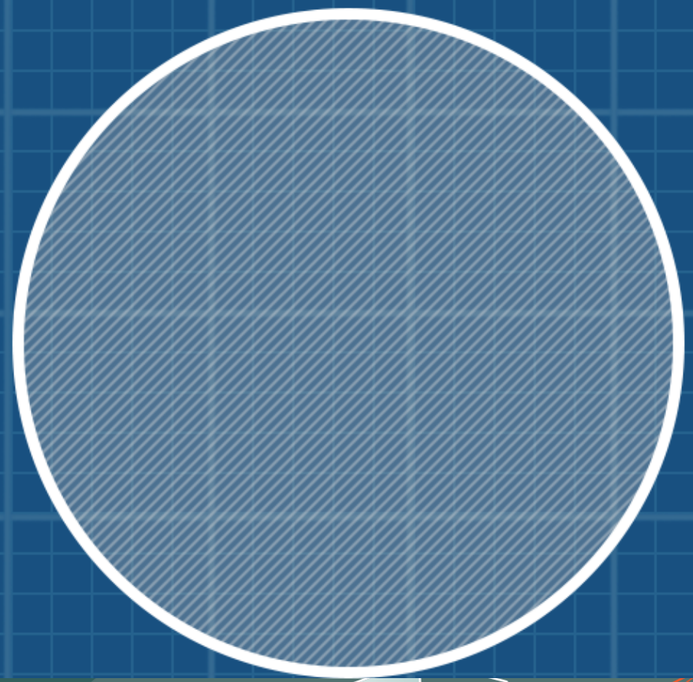
Physics Steps



60hz

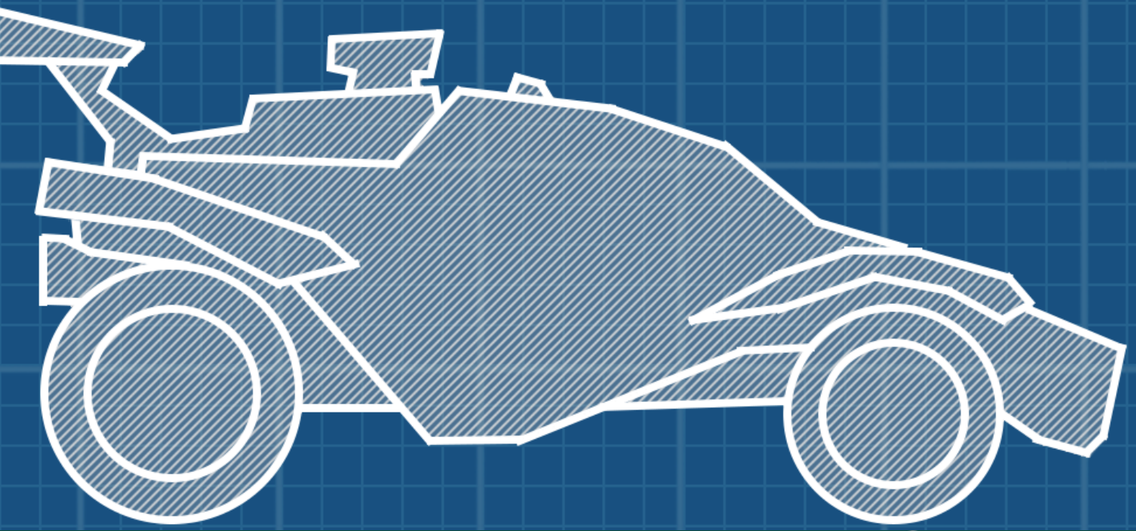
Physics Steps

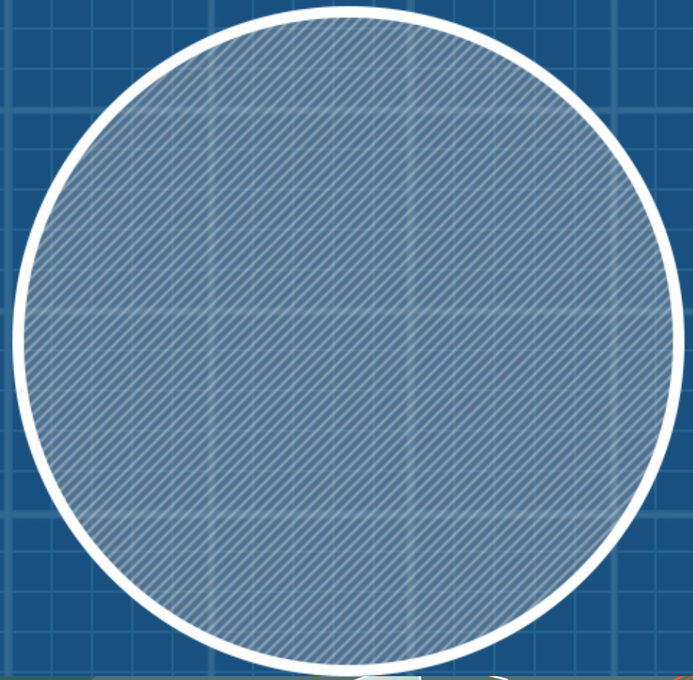




60hz

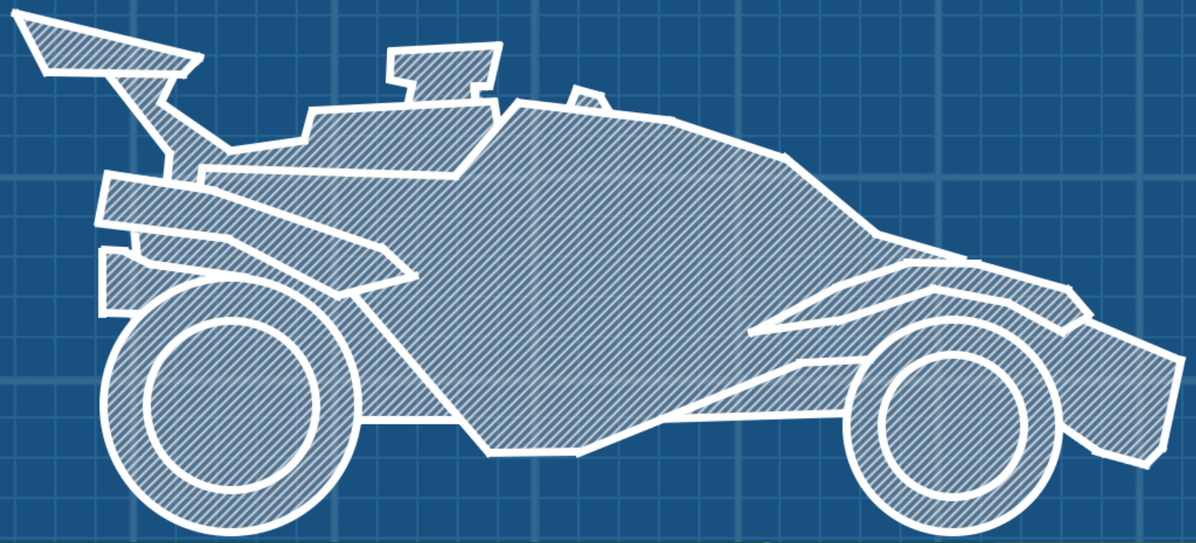
Physics Steps

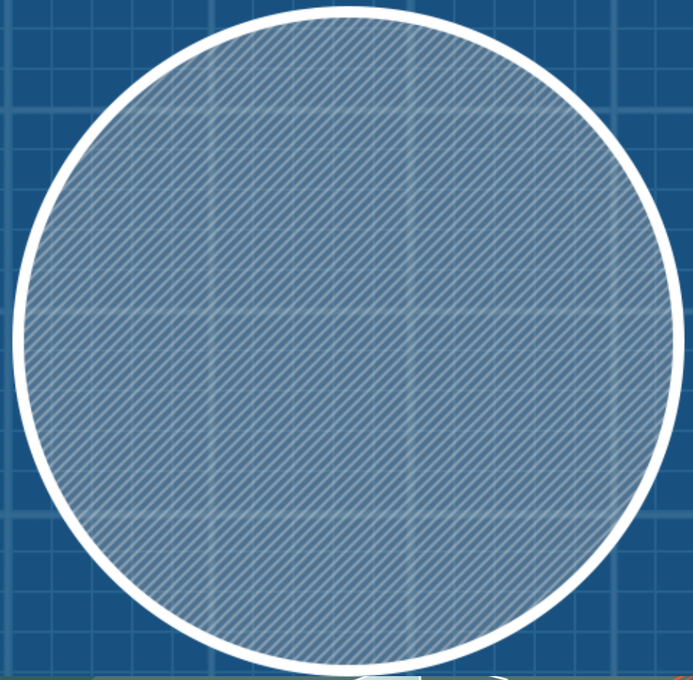




60hz

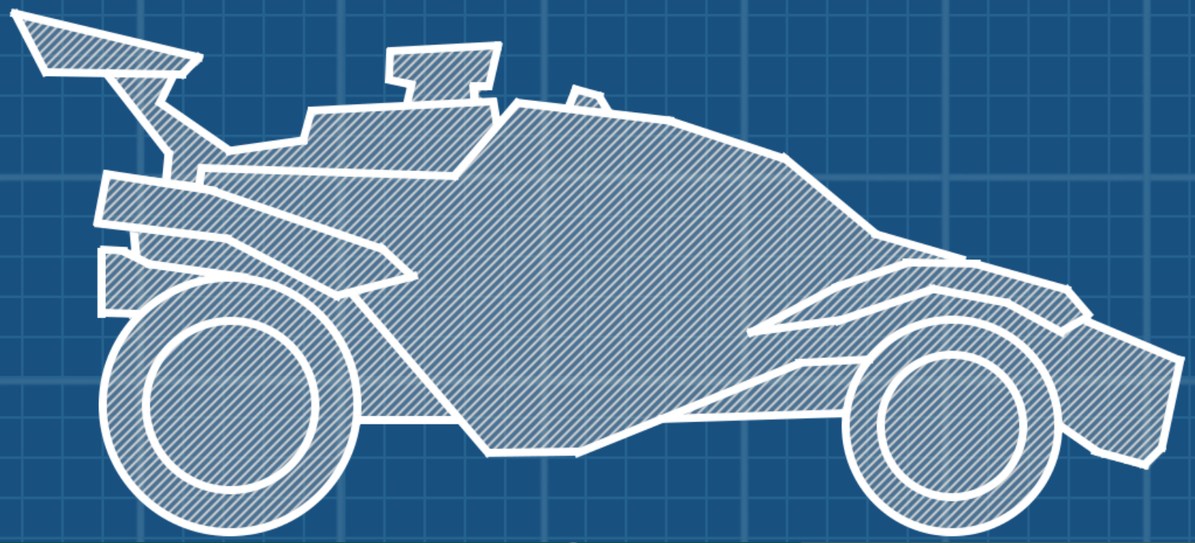
Physics Steps

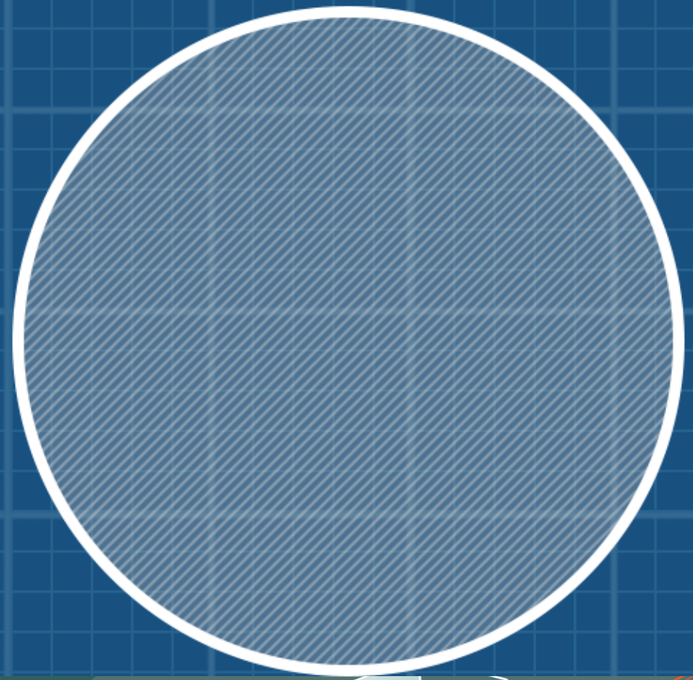




60hz

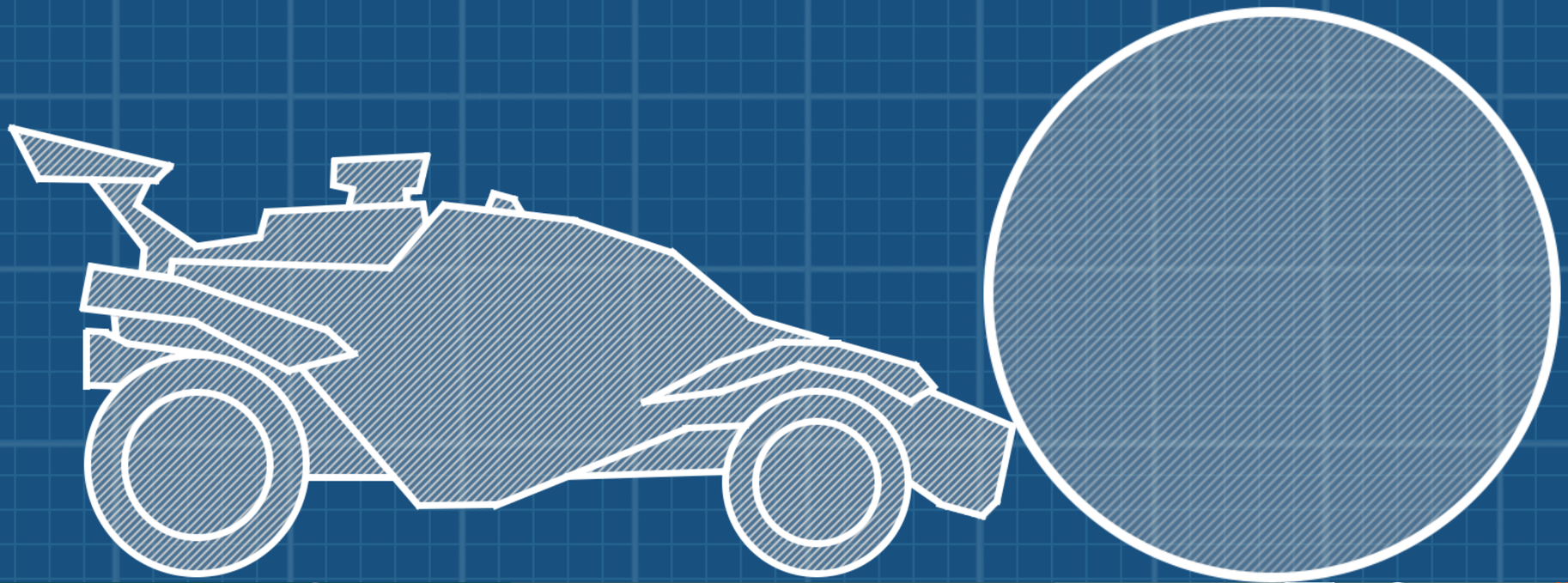
Physics Steps





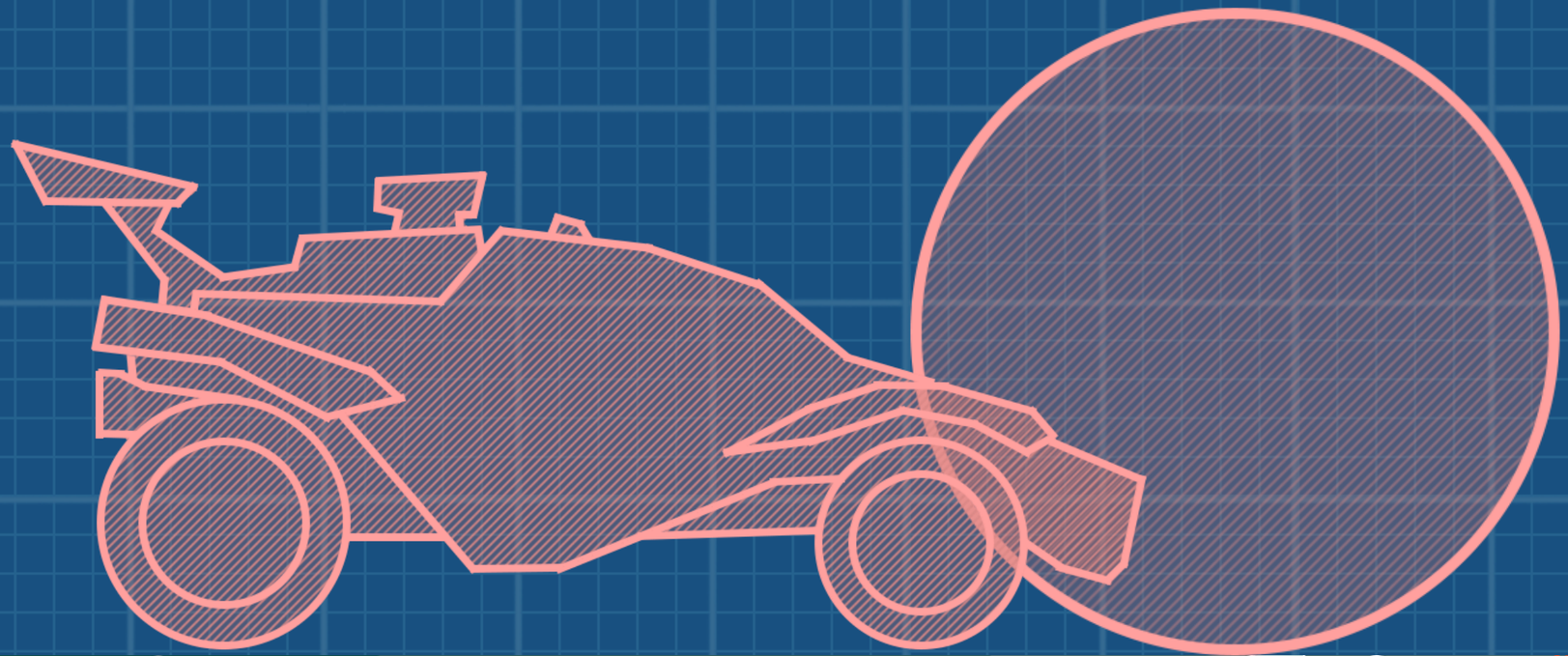
60hz

Physics Steps



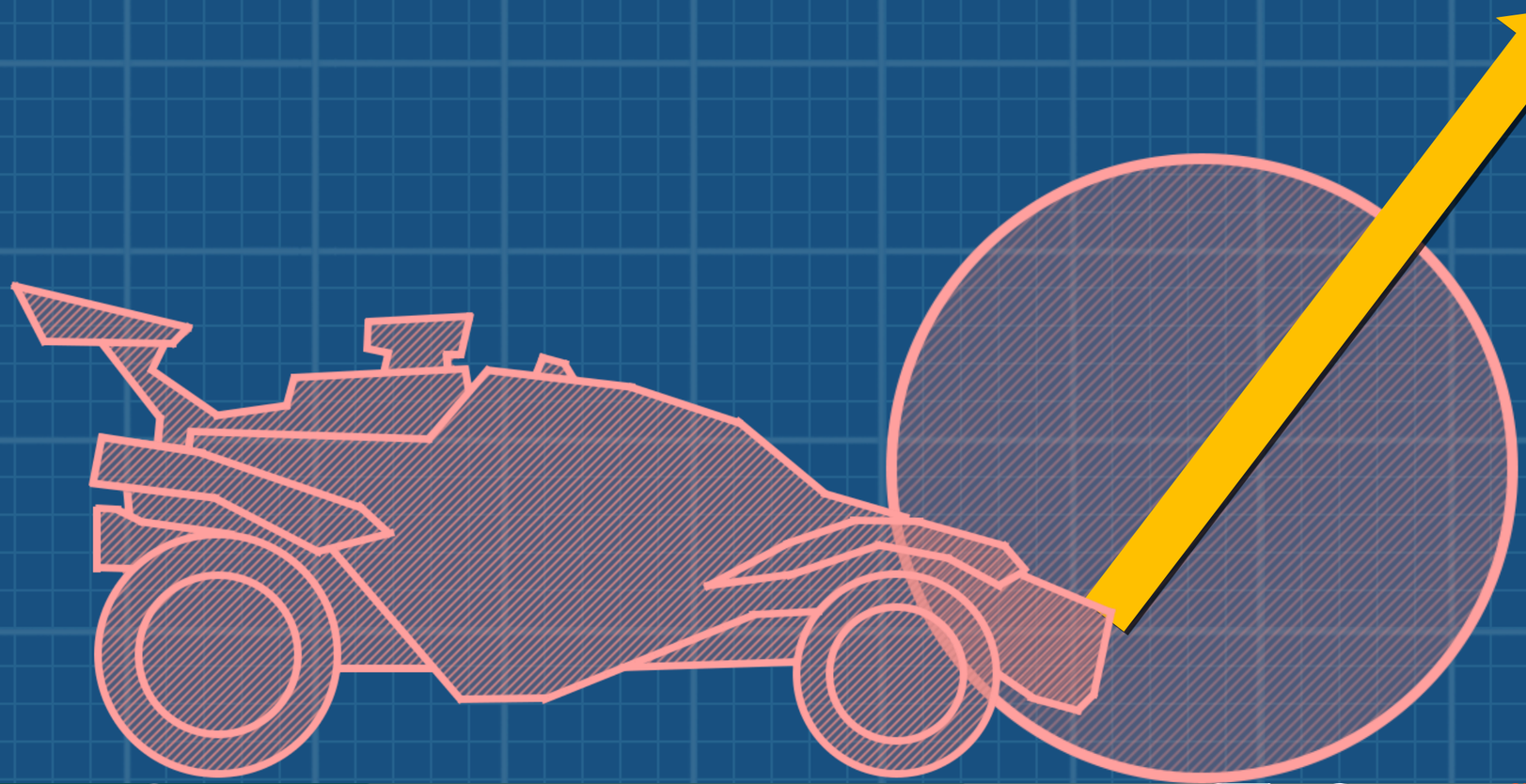
60hz

Physics Steps



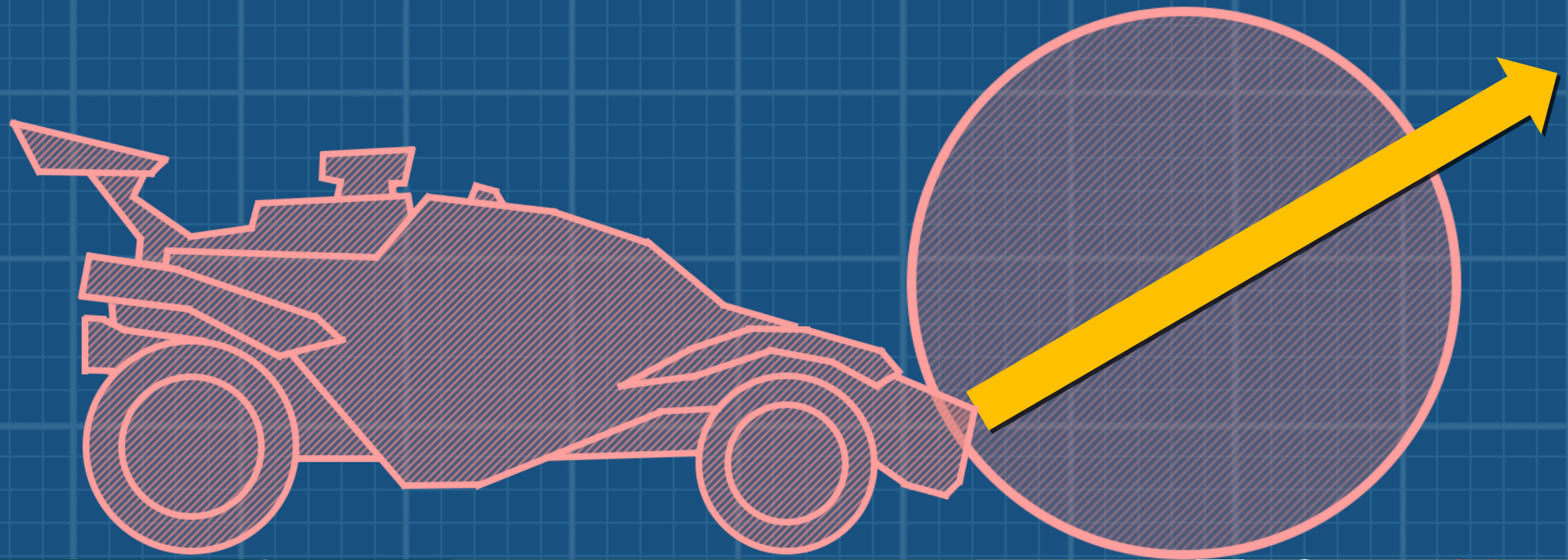
60hz

Physics Steps



60hz

Physics Steps



Physics Framerate

- Slower rate = larger steps = larger penetrations = inconsistent hits
- Higher rate = consistency
- More expensive, especially for network corrections

Physics Engine Summary

- Get control of your physics engine
- Fixed tick rate for consistency
- Higher tick rate for consistency, at a cost

PHYSICS ENGINE
VEHICLE TUNING
NETWORKING

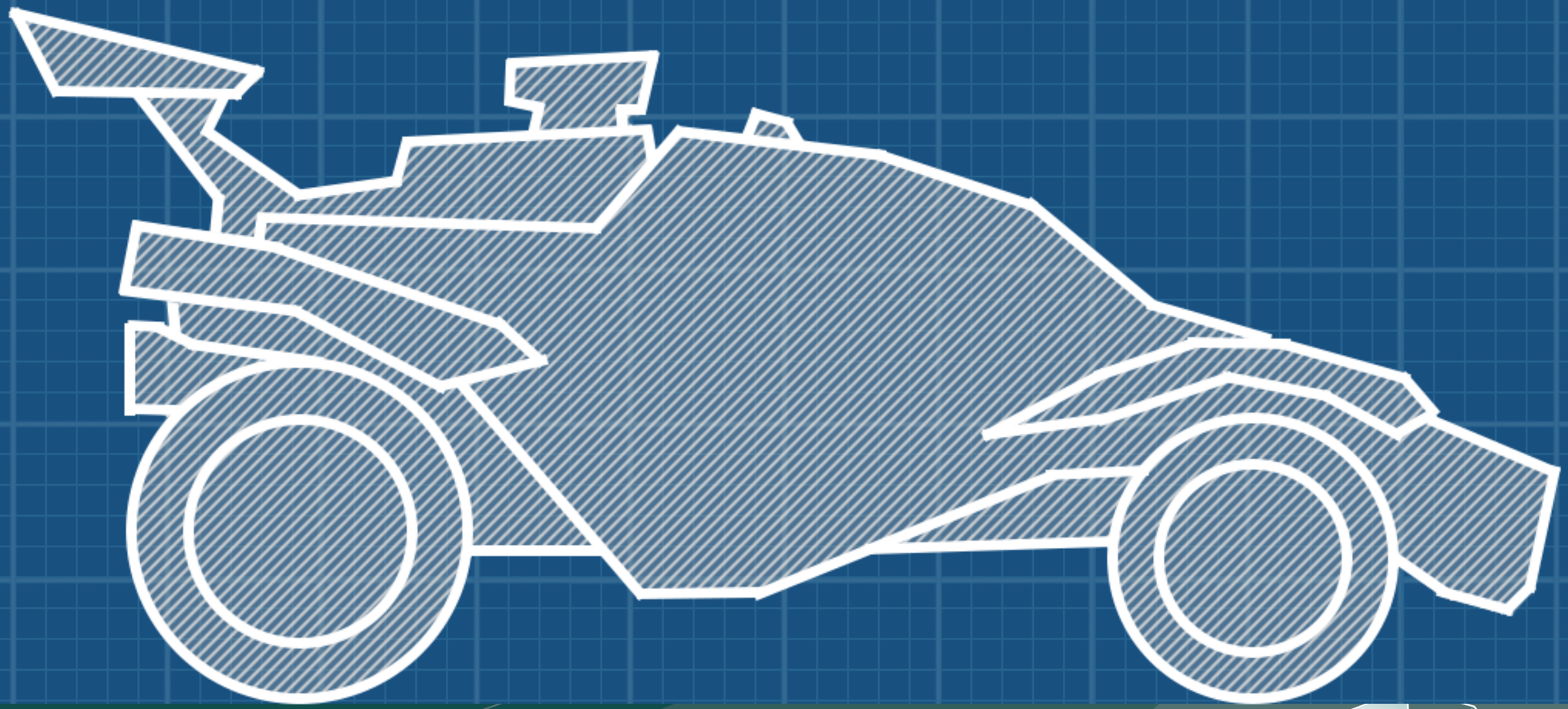
PHYSICS ENGINE
VEHICLE TUNING
NETWORKING

Vehicle Tuning Goals

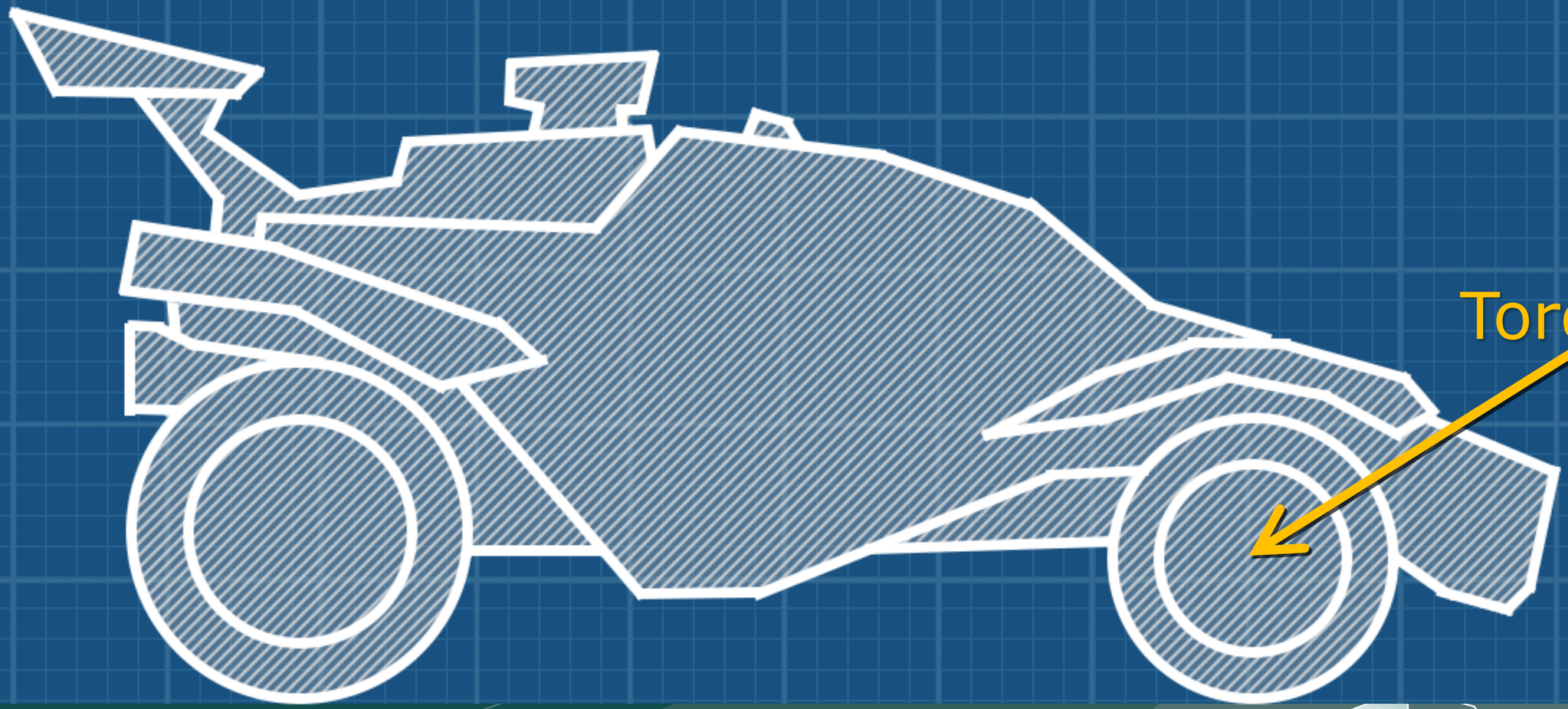
- Fast acceleration and braking
- Sharp steering
- Stable driving, fast recovery

Scenario:
"Faster Acceleration"

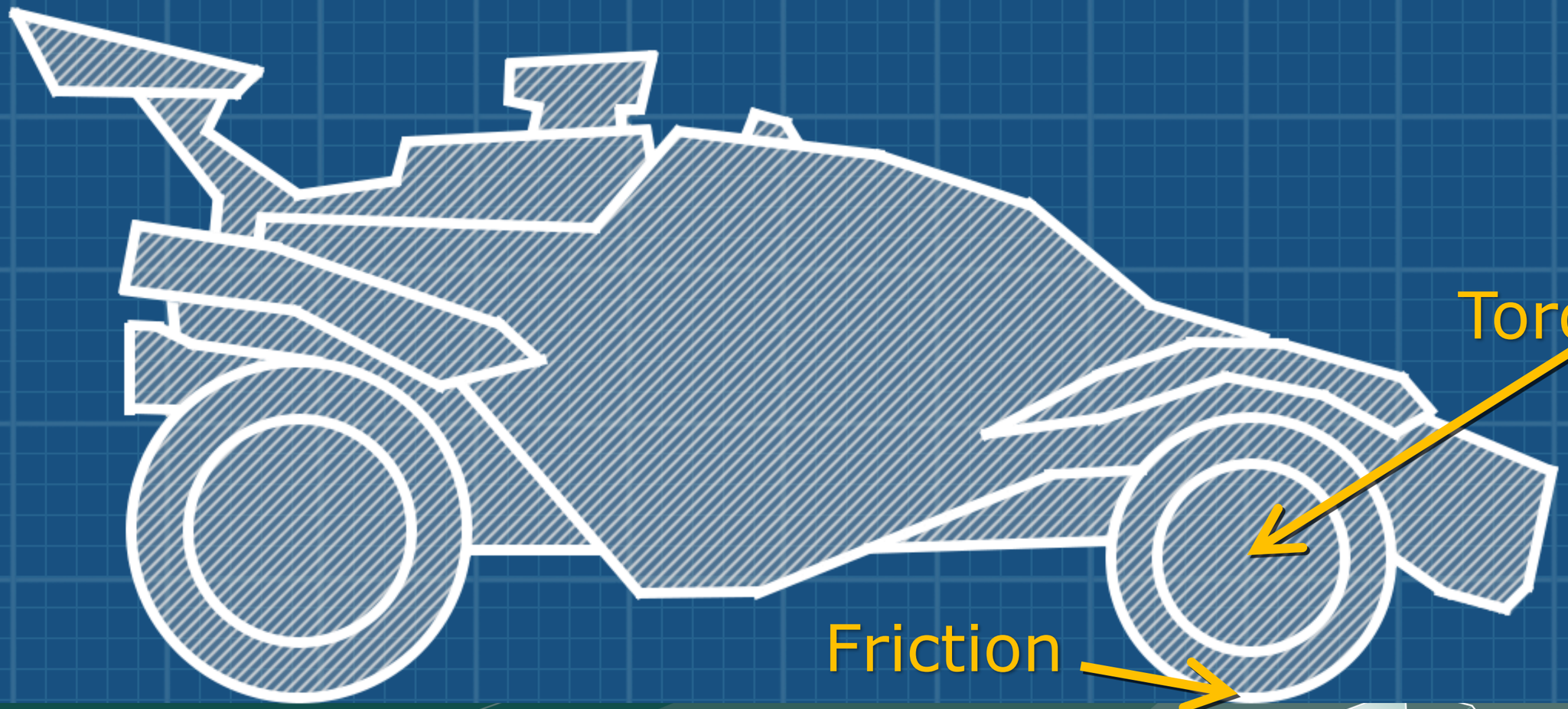
Acceleration



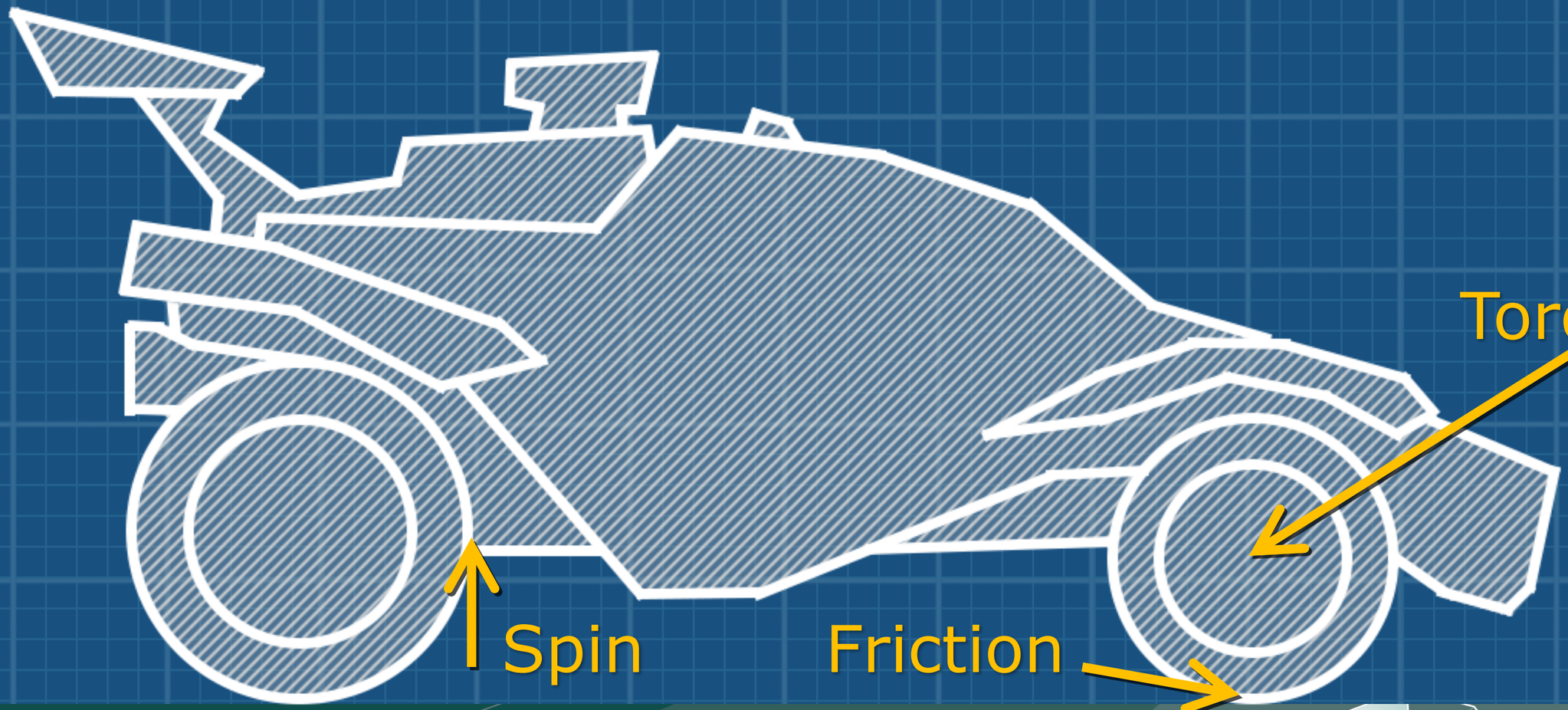
Acceleration

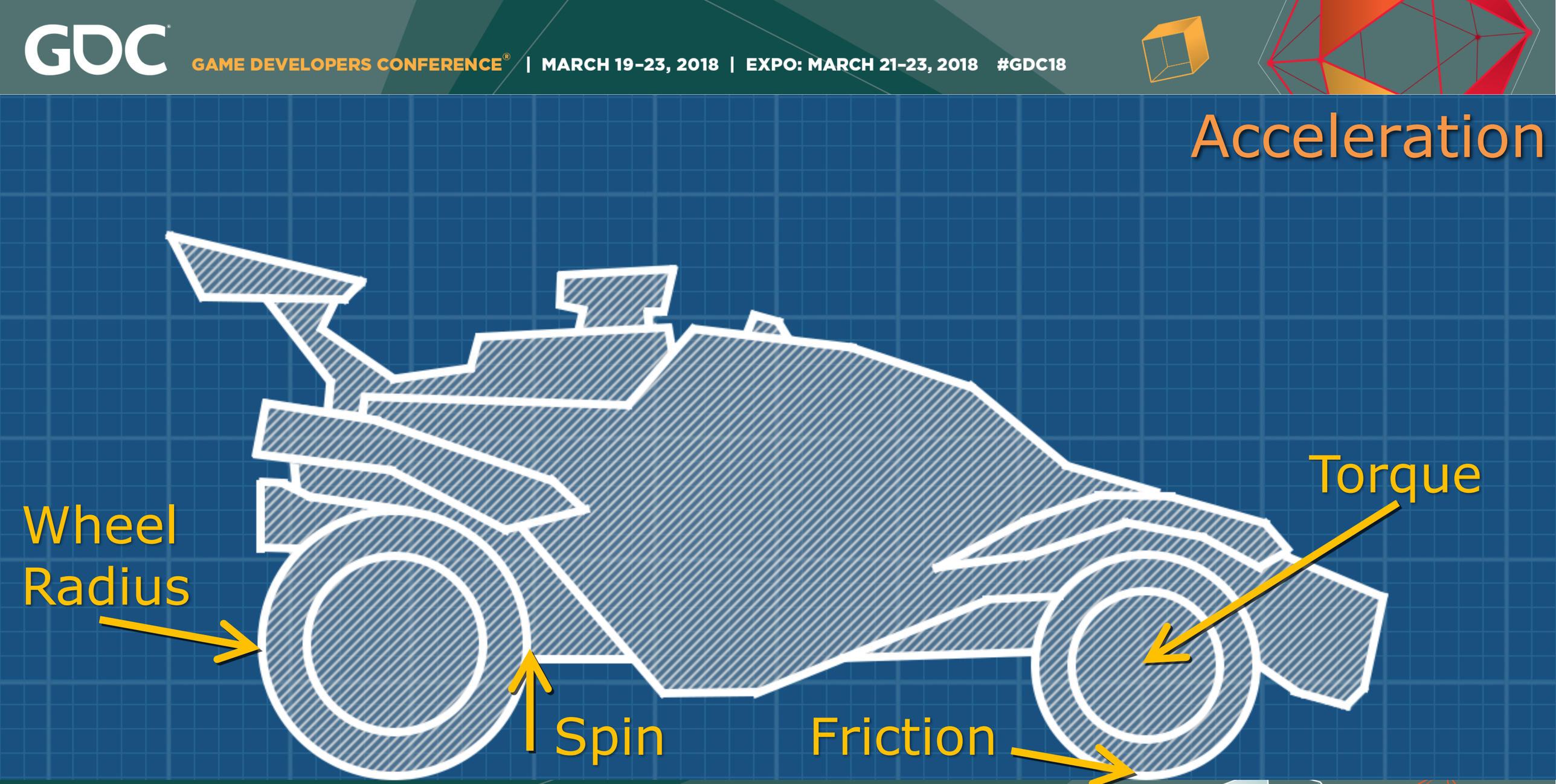


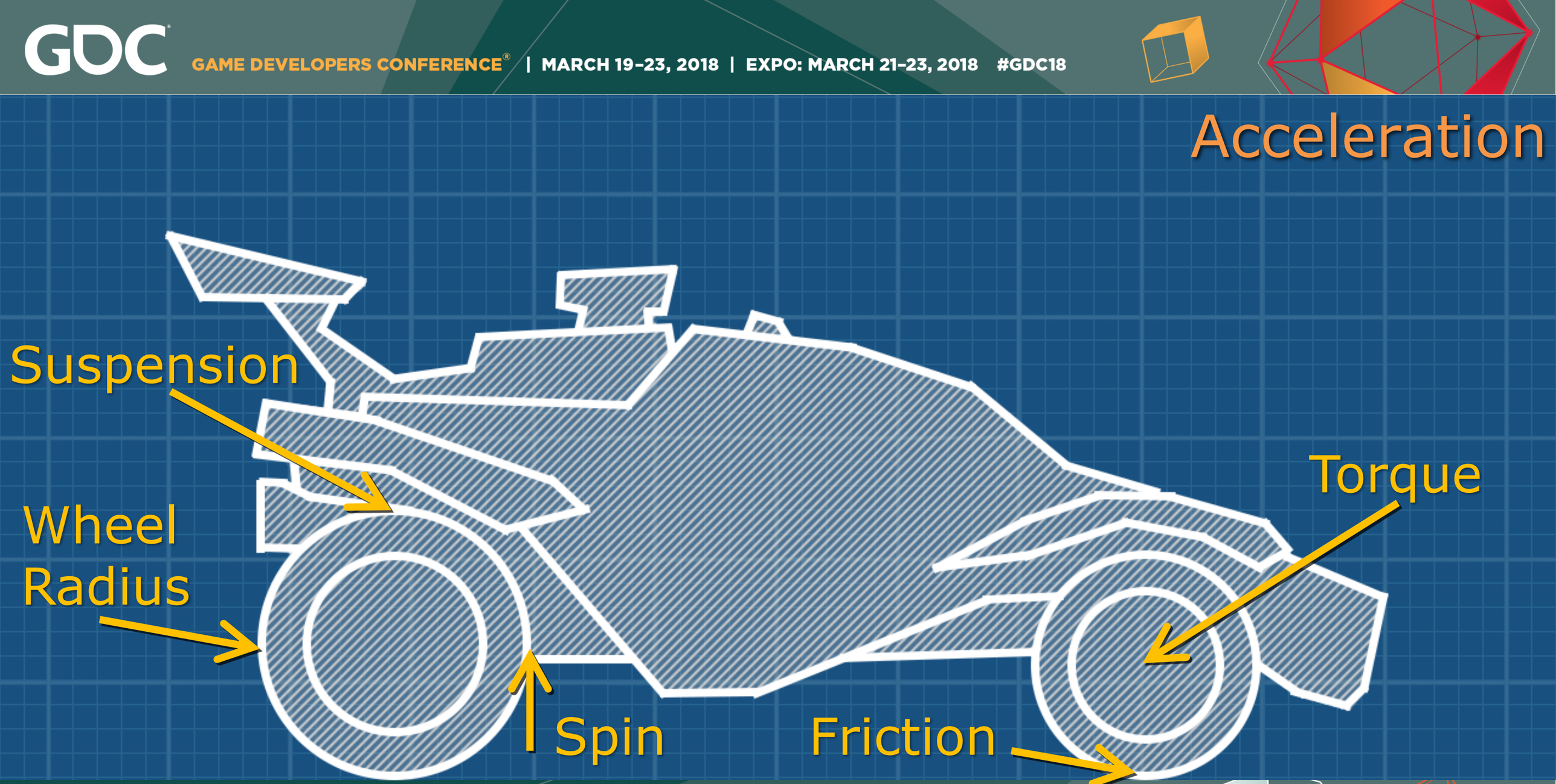
Acceleration



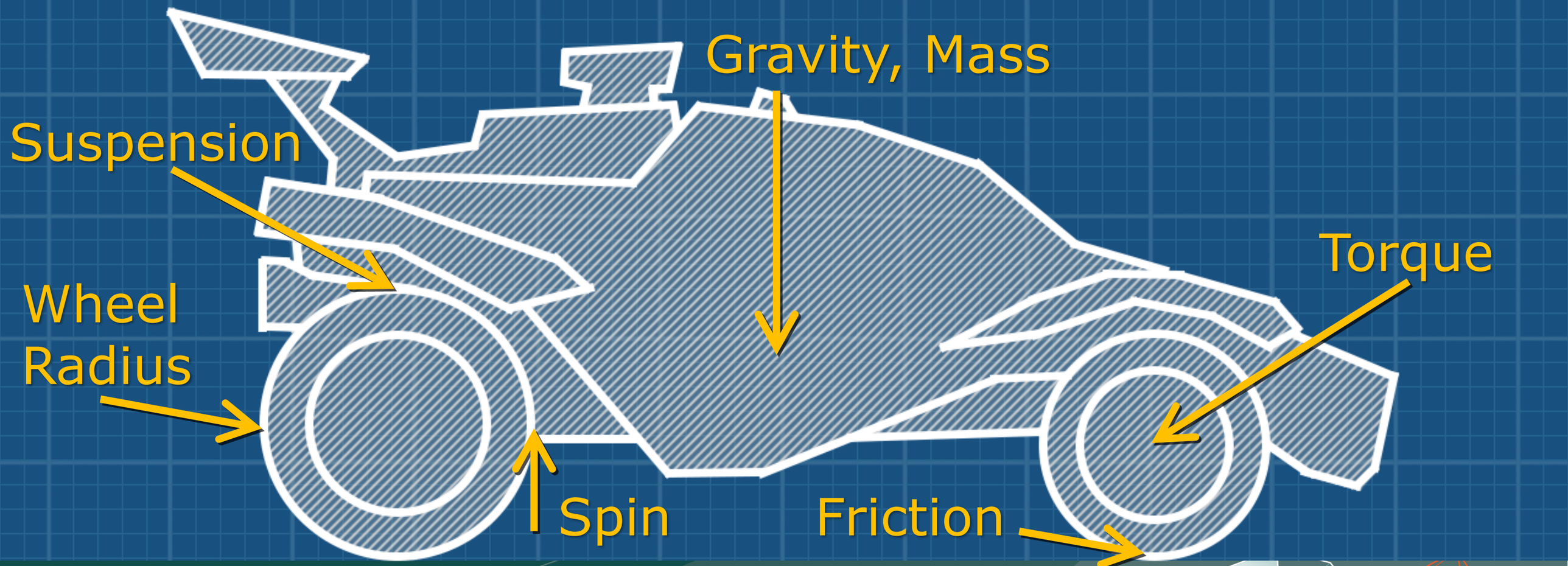
Acceleration



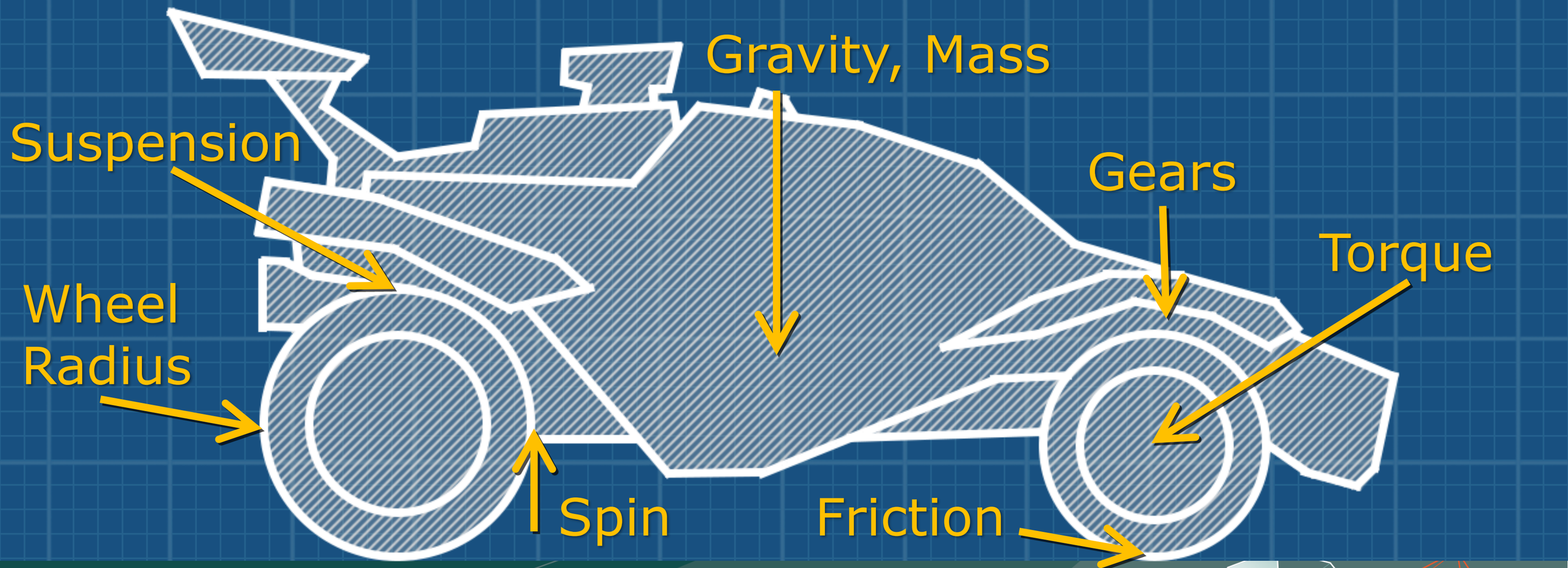

Acceleration
Torque
Wheel Radius
Spin
Friction

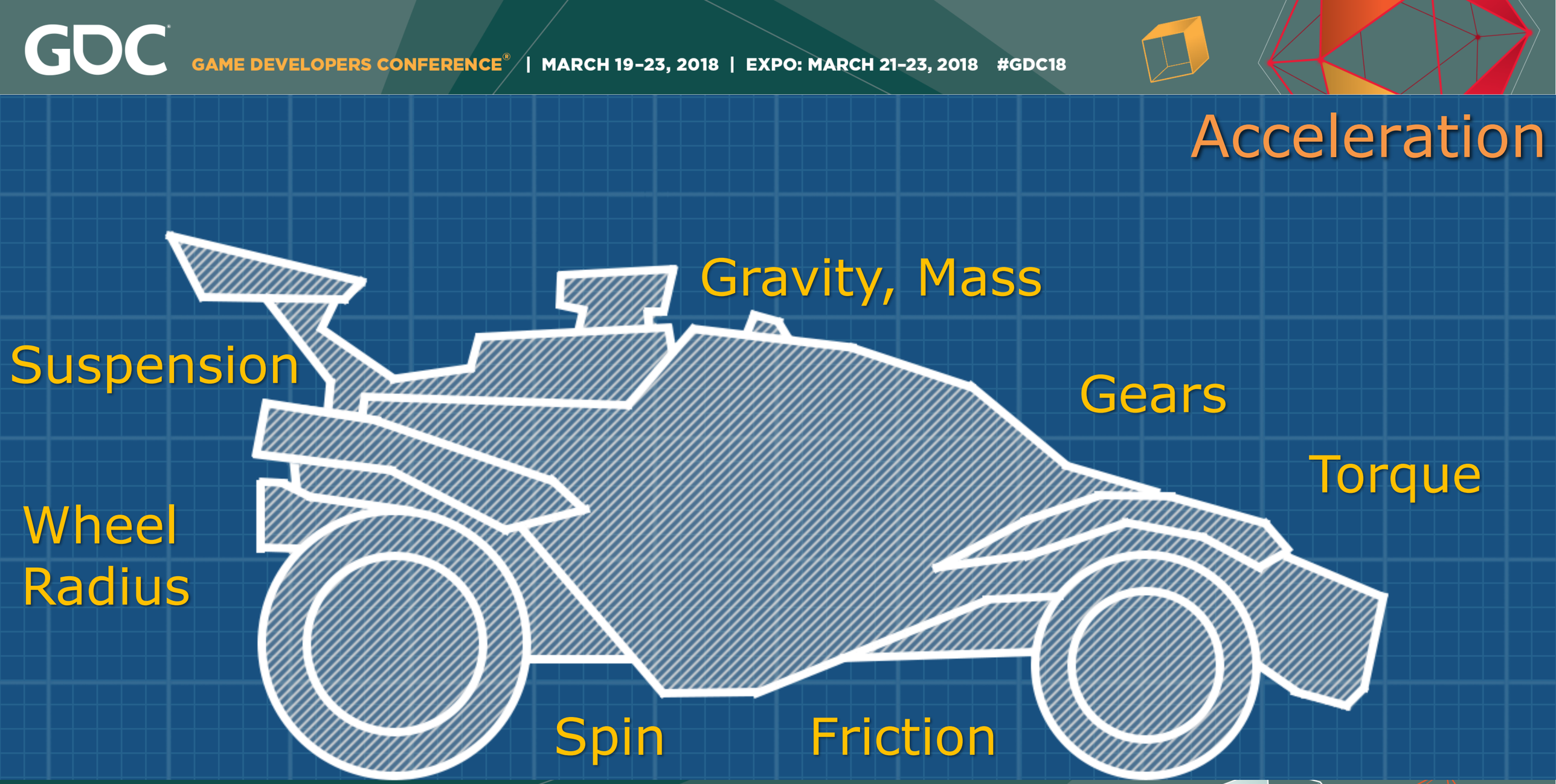

Acceleration
Suspension
Wheel Radius
Torque
Spin
Friction

Acceleration



Acceleration




Acceleration
Gravity, Mass
Suspension
Gears
Torque
Wheel Radius
Spin
Friction

Reduce Complexity

- Transmission
  - Maybe don’t have one
  - Use force/accel curve instead

Reduce Complexity

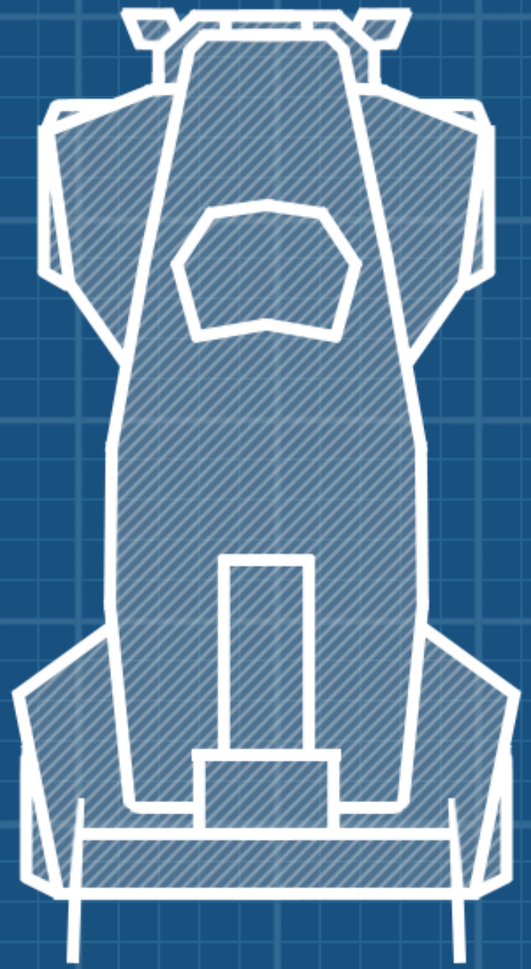
- Mass
  - Keep constant
  - Ignore when applying forces

Reduce Complexity

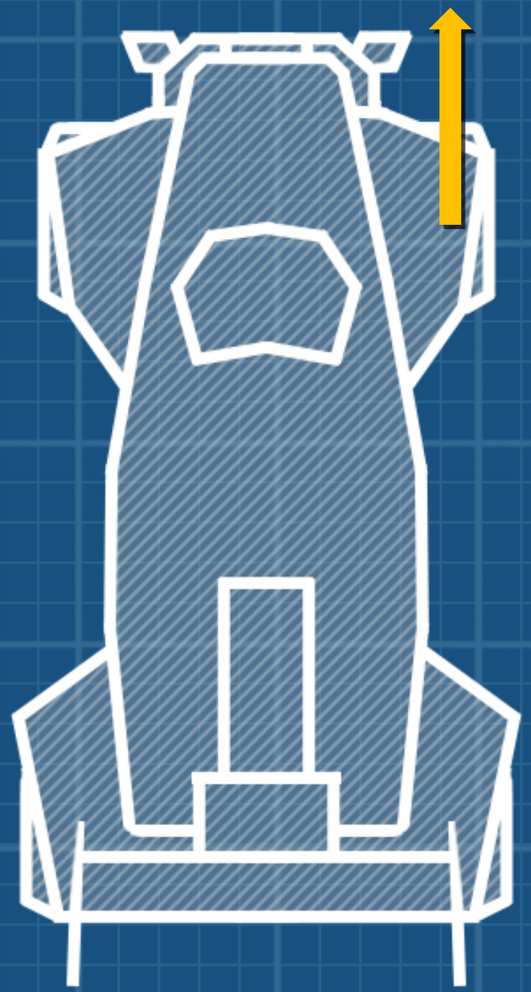
- Tire Friction
  - Can live without longitudinal friction
  - Simplify lateral friction

Simple Friction

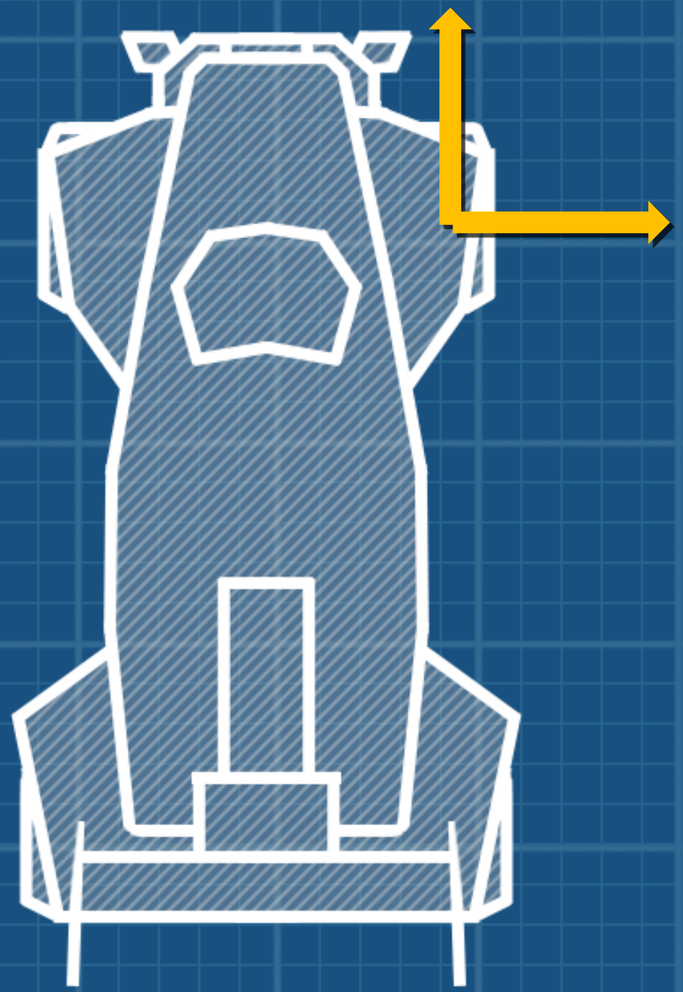
Simple Friction



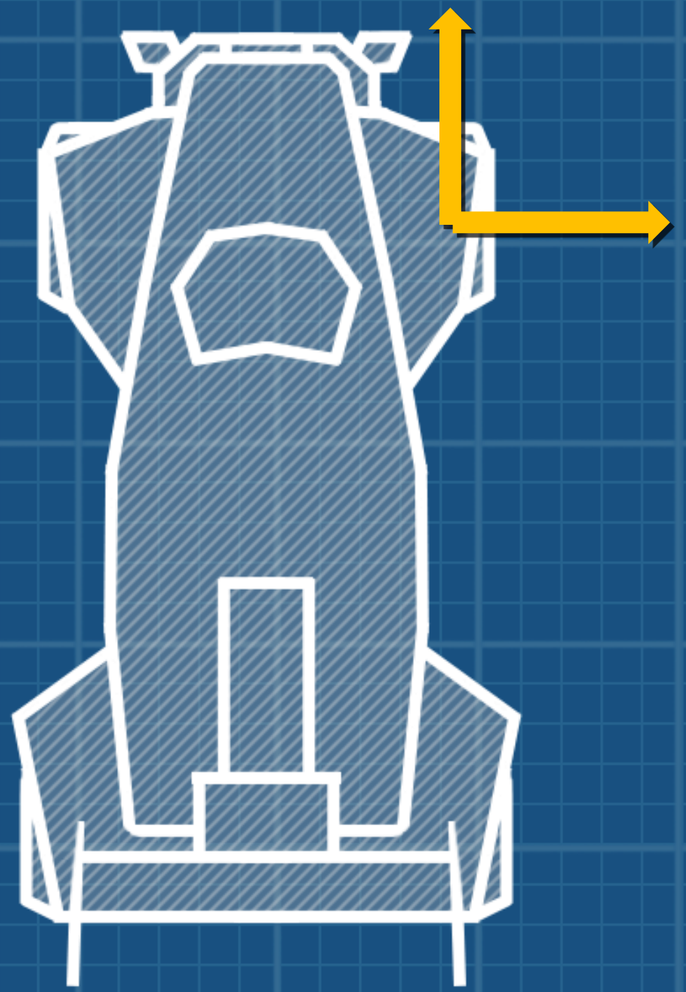
Simple Friction



Simple Friction

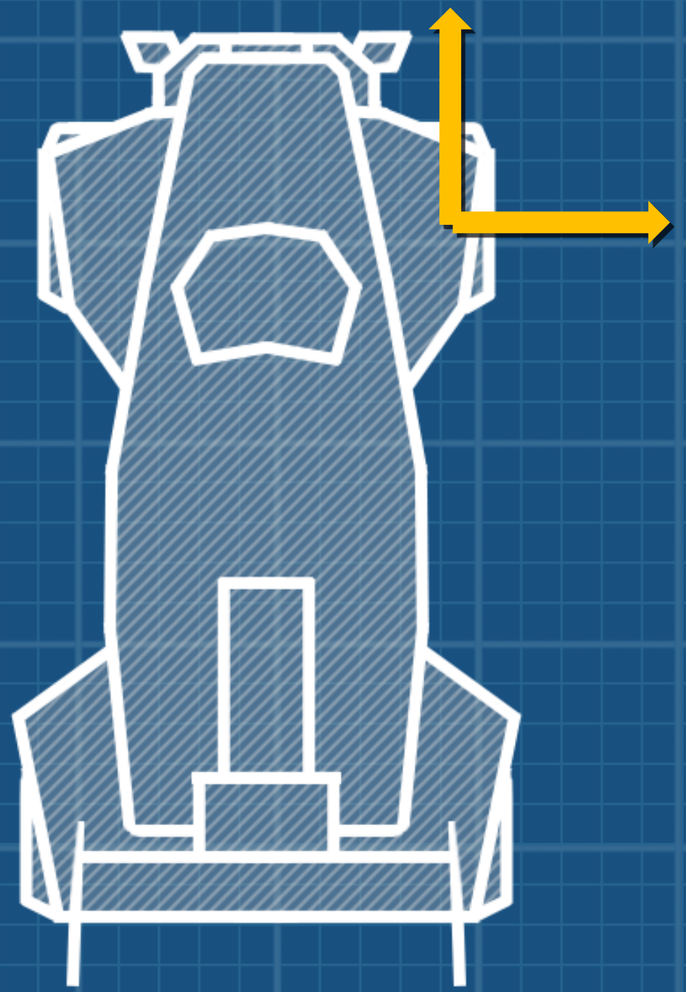


Simple Friction



Ratio =
SideSpeed / (SideSpeed + ForwardSpeed)

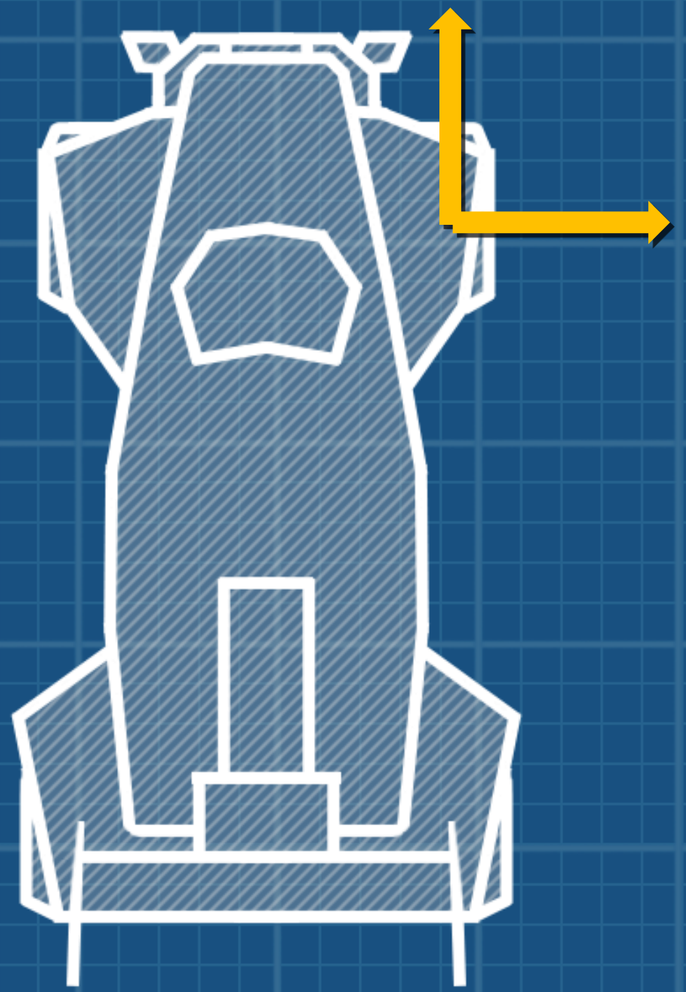
Simple Friction



Ratio =
SideSpeed / (SideSpeed + ForwardSpeed)

SlideFriction = Curve(Ratio)

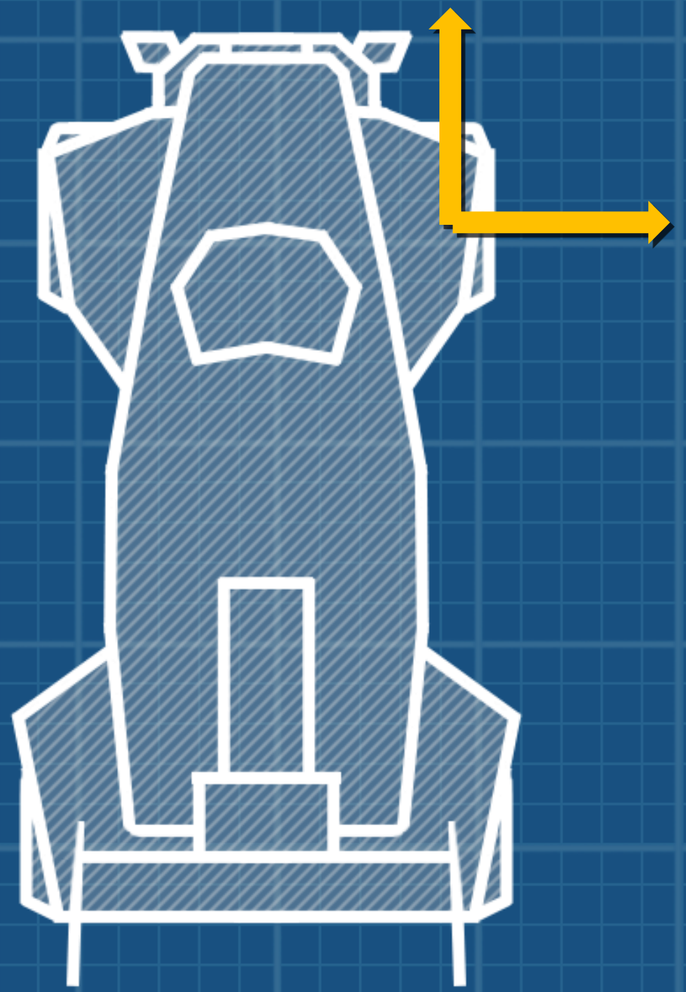
Simple Friction



Ratio =
SideSpeed / (SideSpeed + ForwardSpeed)

SlideFriction = Curve(Ratio)
GroundFriction = Curve(GroundNormal.Z)

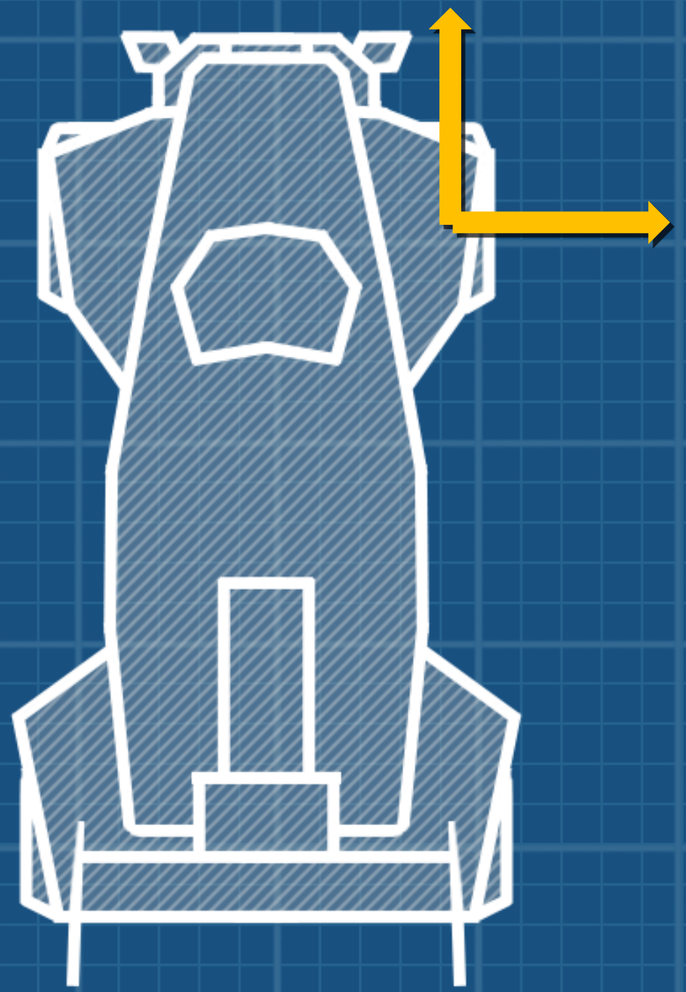
Simple Friction



Ratio =
SideSpeed / (SideSpeed + ForwardSpeed)

SlideFriction = Curve(Ratio)
GroundFriction = Curve(GroundNormal.Z)
Friction = SlideFriction * GroundFriction

Simple Friction



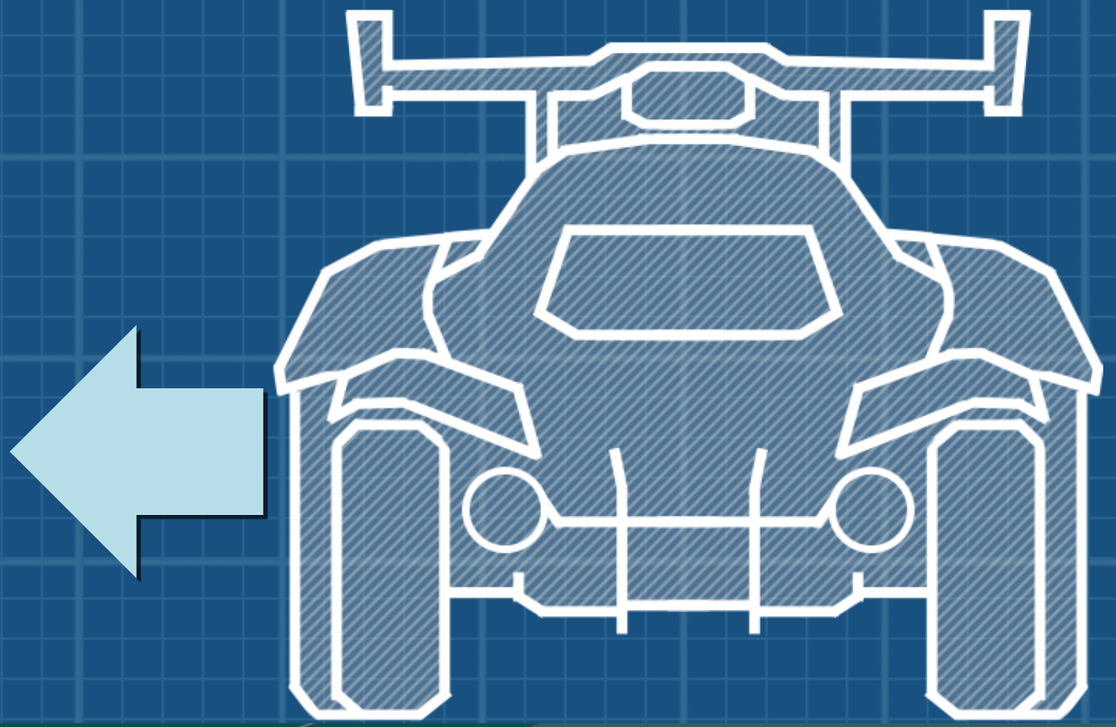
Ratio =
SideSpeed / (SideSpeed + ForwardSpeed)

SlideFriction = Curve(Ratio)
GroundFriction = Curve(GroundNormal.Z)
Friction = SlideFriction * GroundFriction

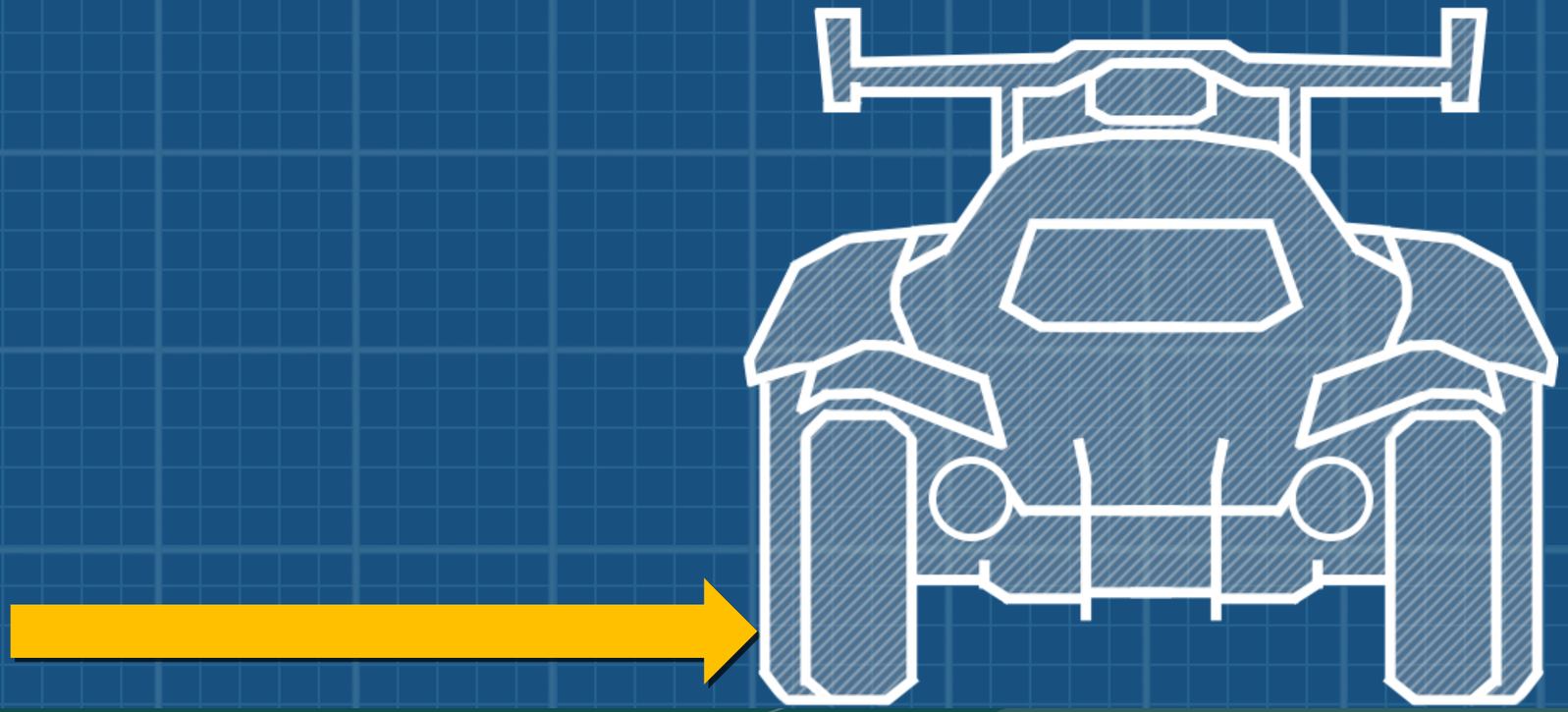
Impulse = Constraint * Friction

Friction Location

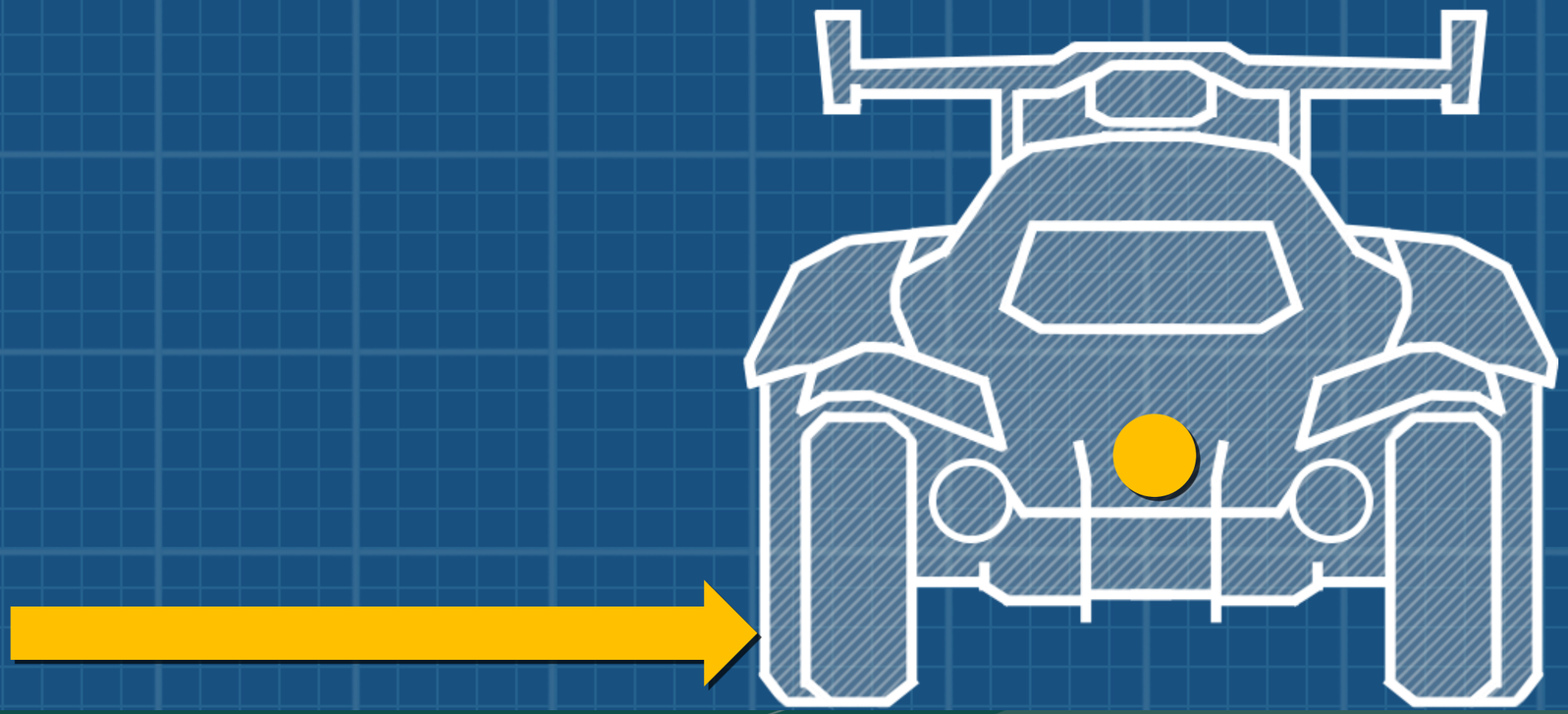
Friction Location



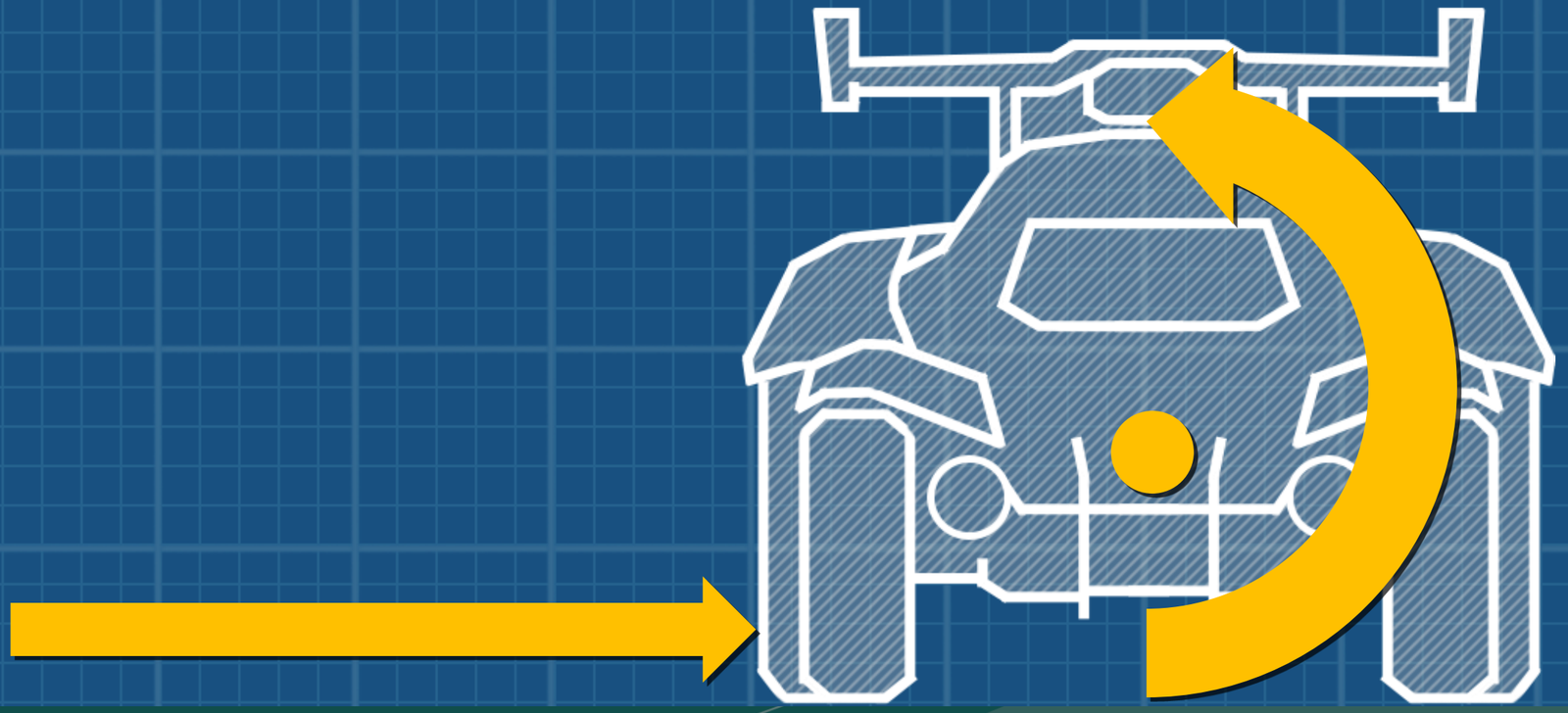
Friction Location



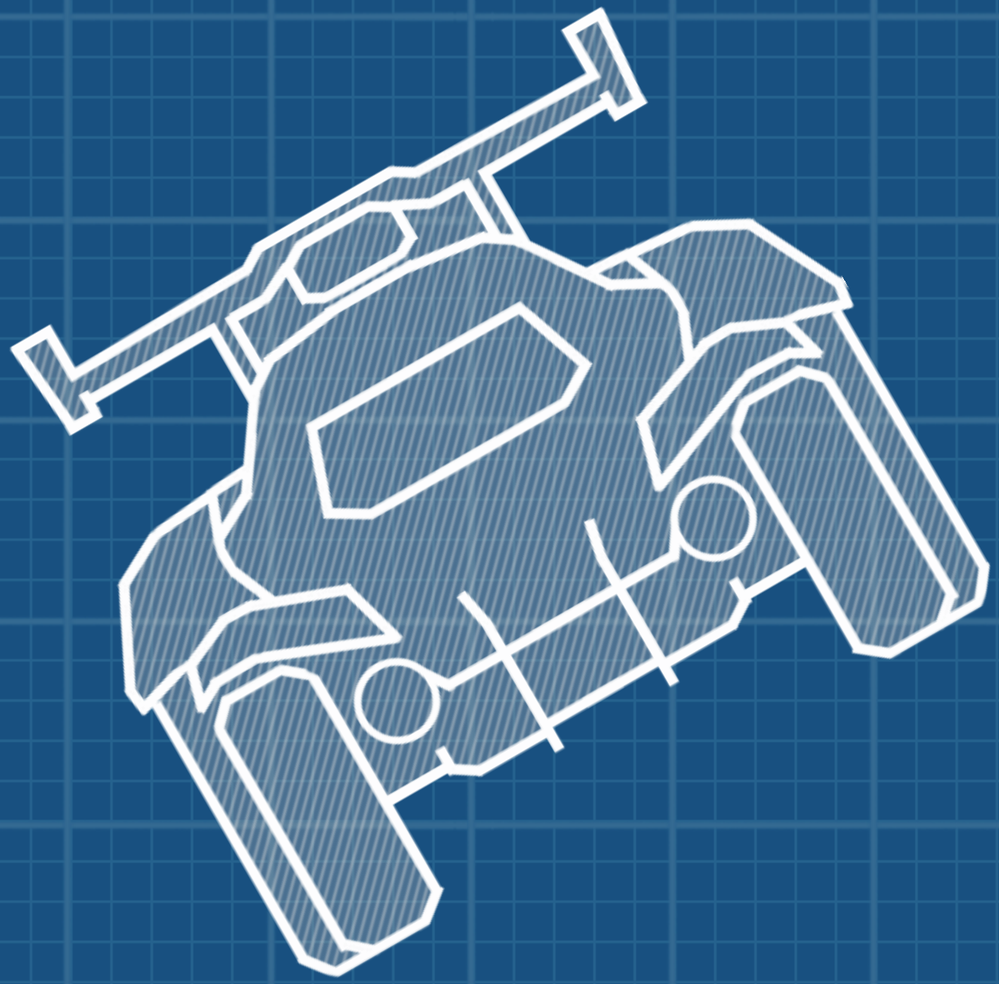
Friction Location

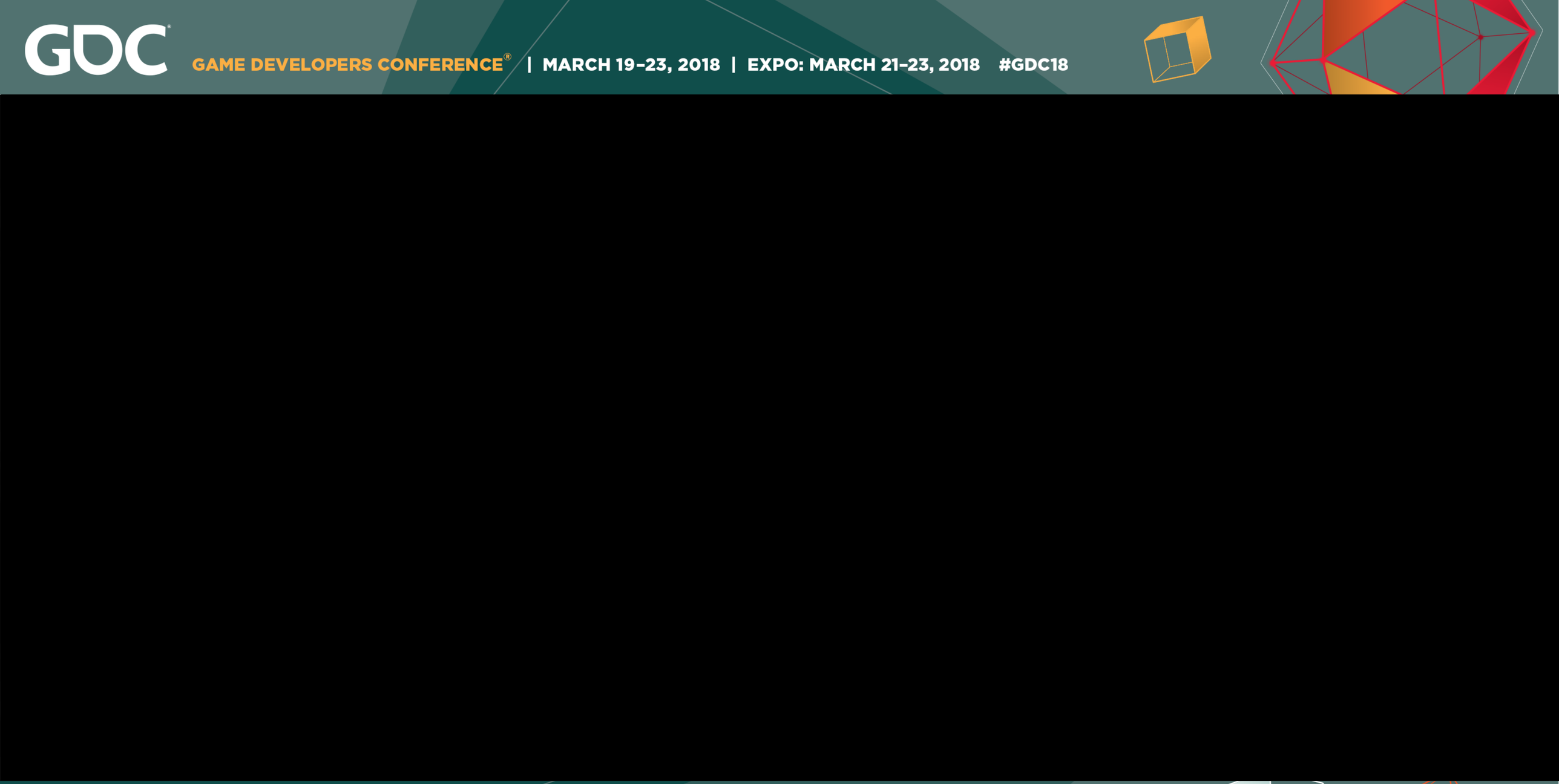


Friction Location



Friction Location

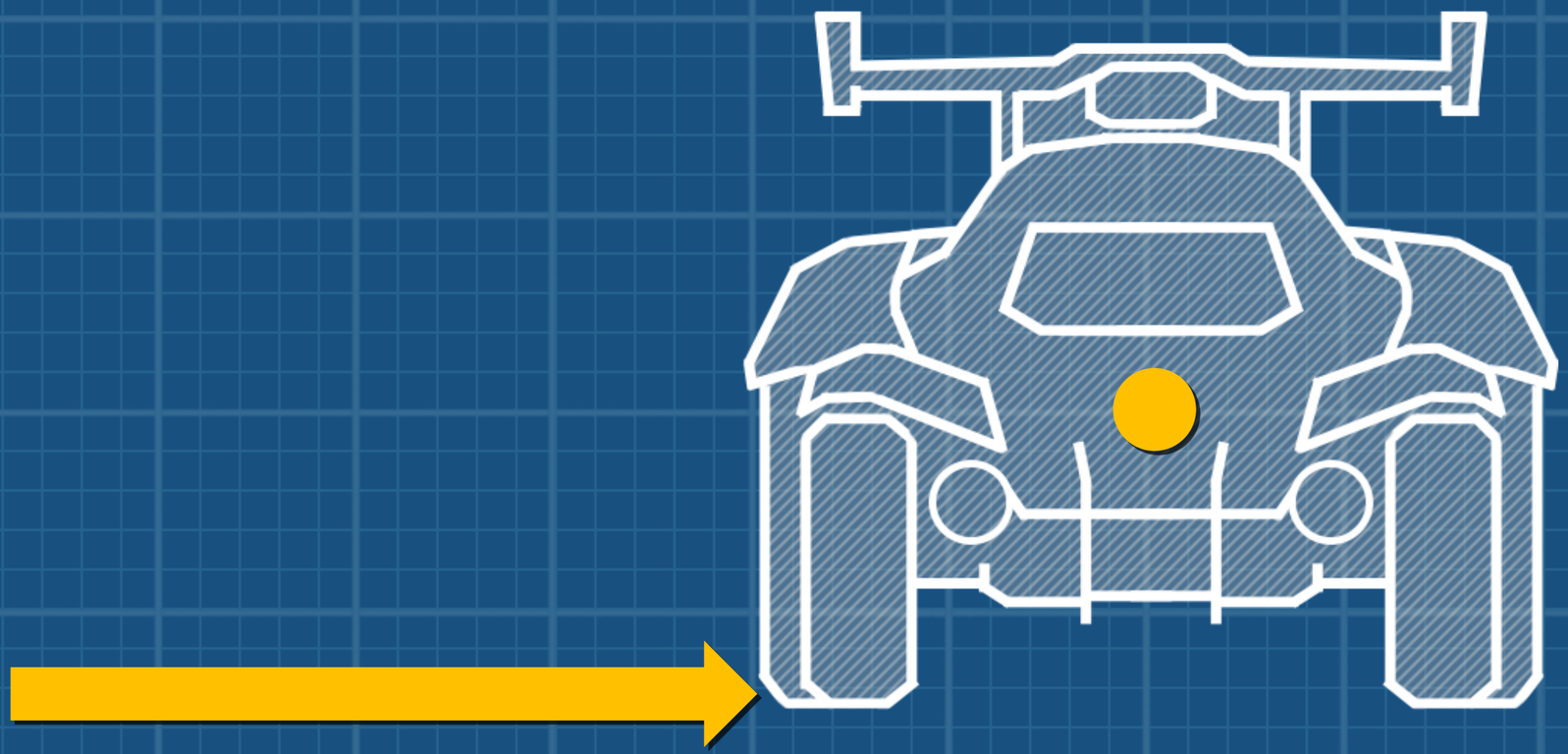




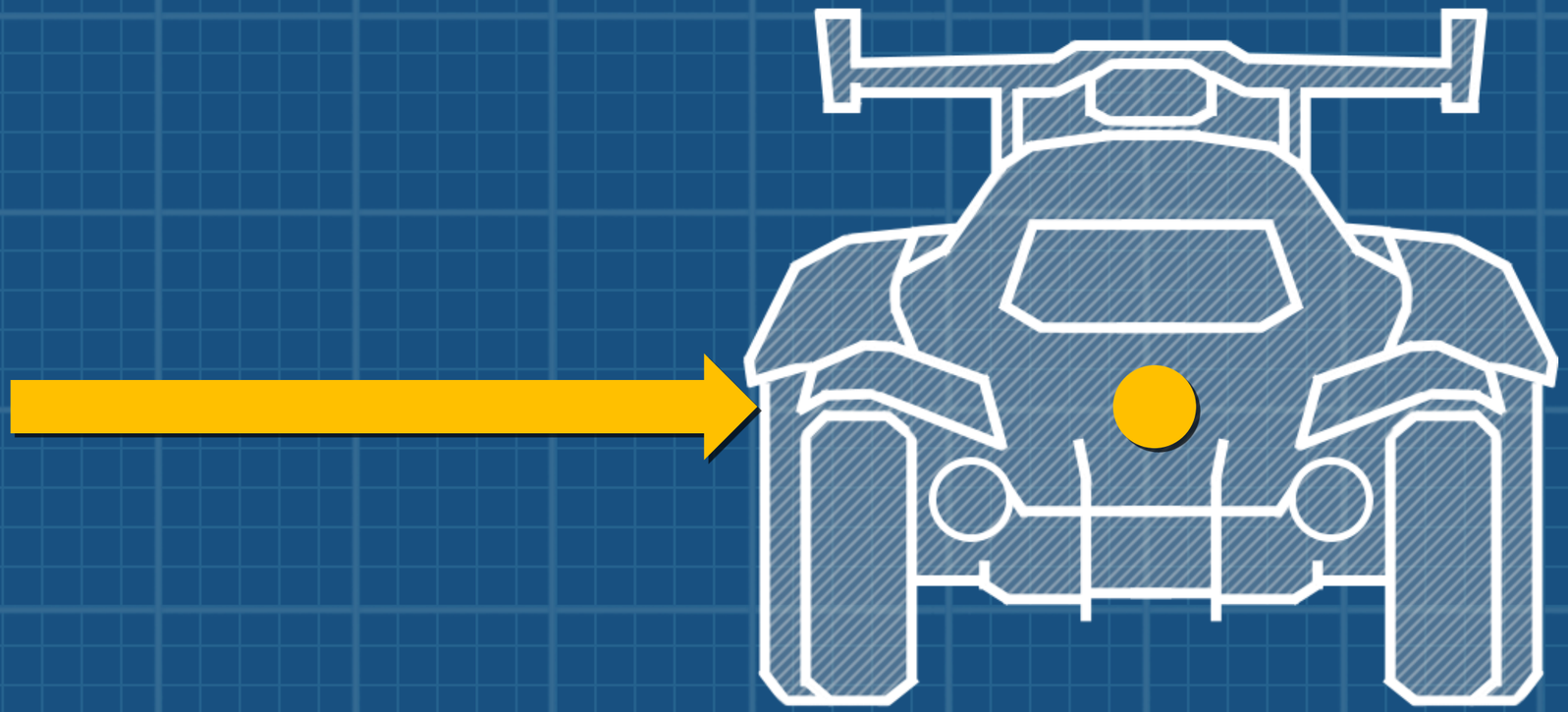
Possible Workarounds

- Lower friction
- Limit steer angles
- “Stay upright” constraint
- Apply forces at CM height

Forces at CM

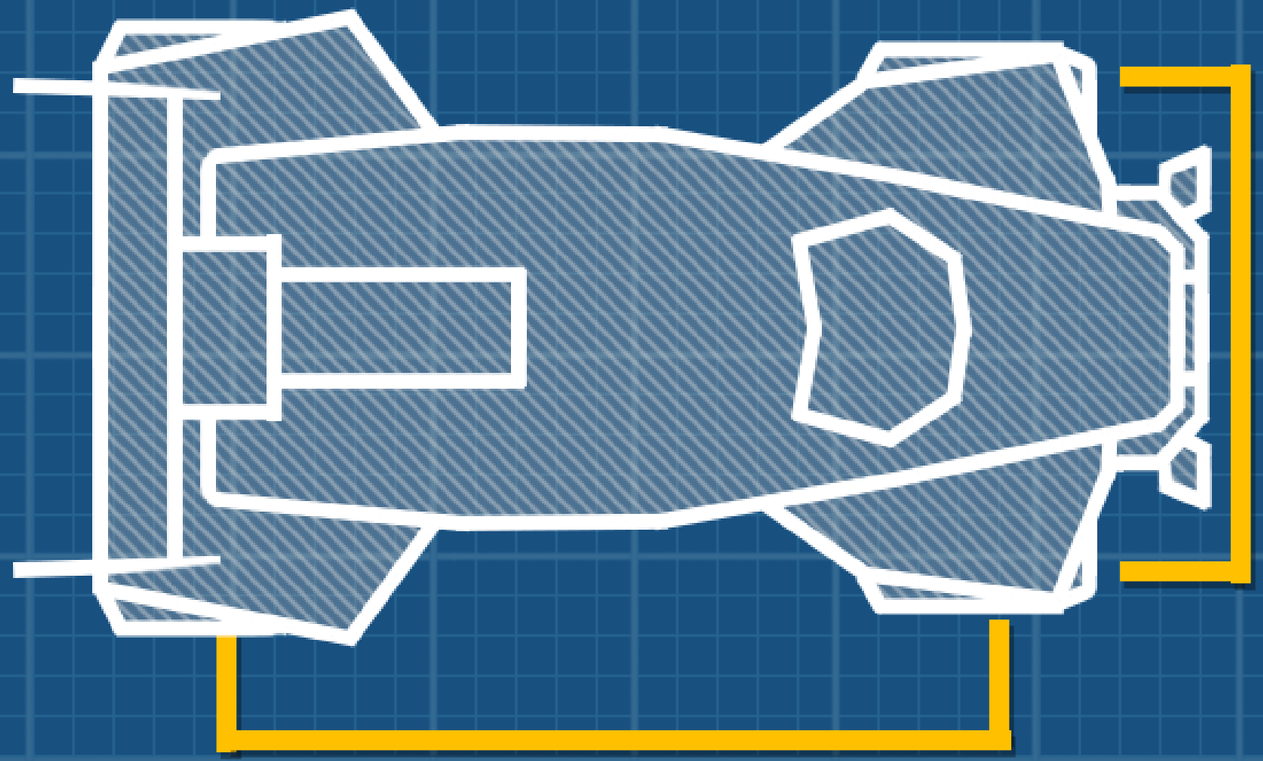


Forces at CM



Wheel Positions

Wheel Positions



Wheel Positions

Wheel Positions +2cm

Wheel Positions

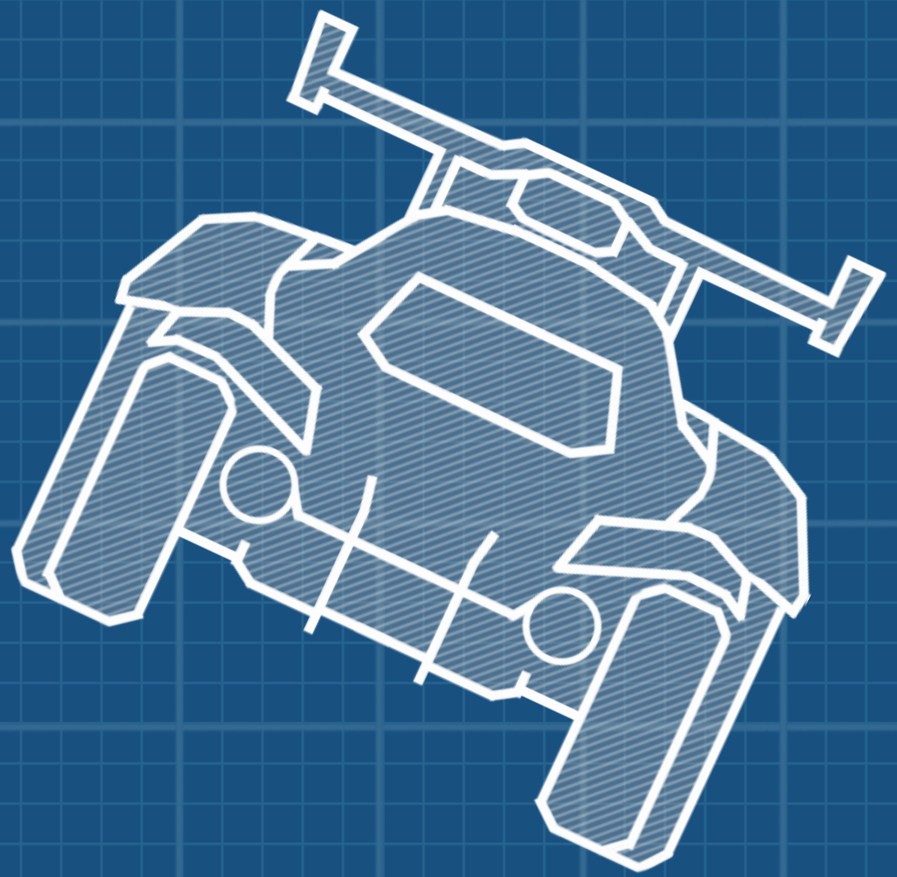
- Originally defined by artists
- Big effect on handling
- Long iteration times
- Transitioned to “preset” system

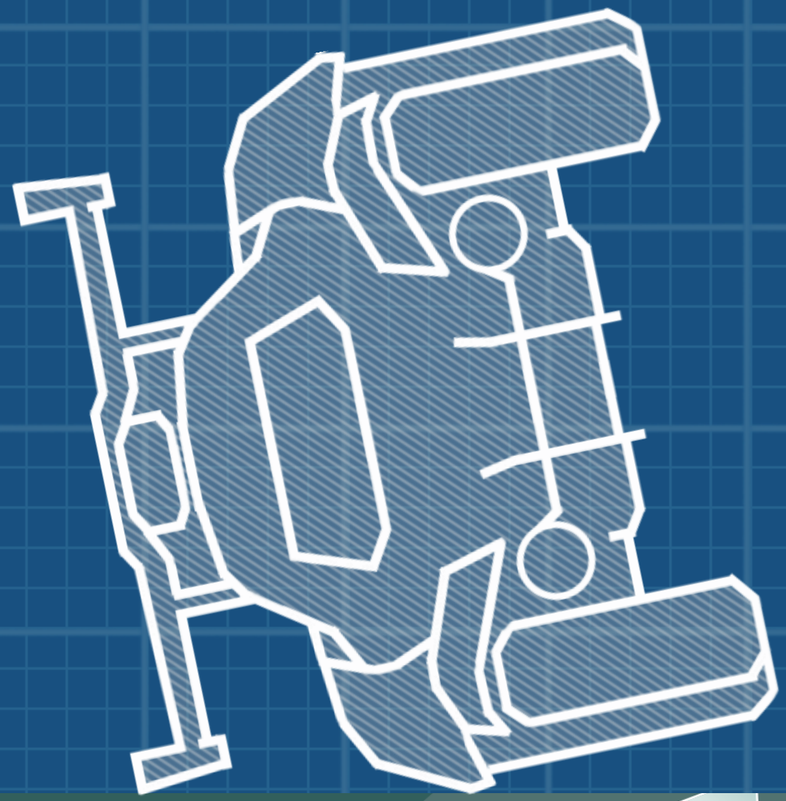
Physics Presets

- Divorce physics setup from visuals
- Collision box size & translation
- Wheel positions, radii
- Many vehicles, few presets
- Faster iteration
- Physics and visuals don’t match

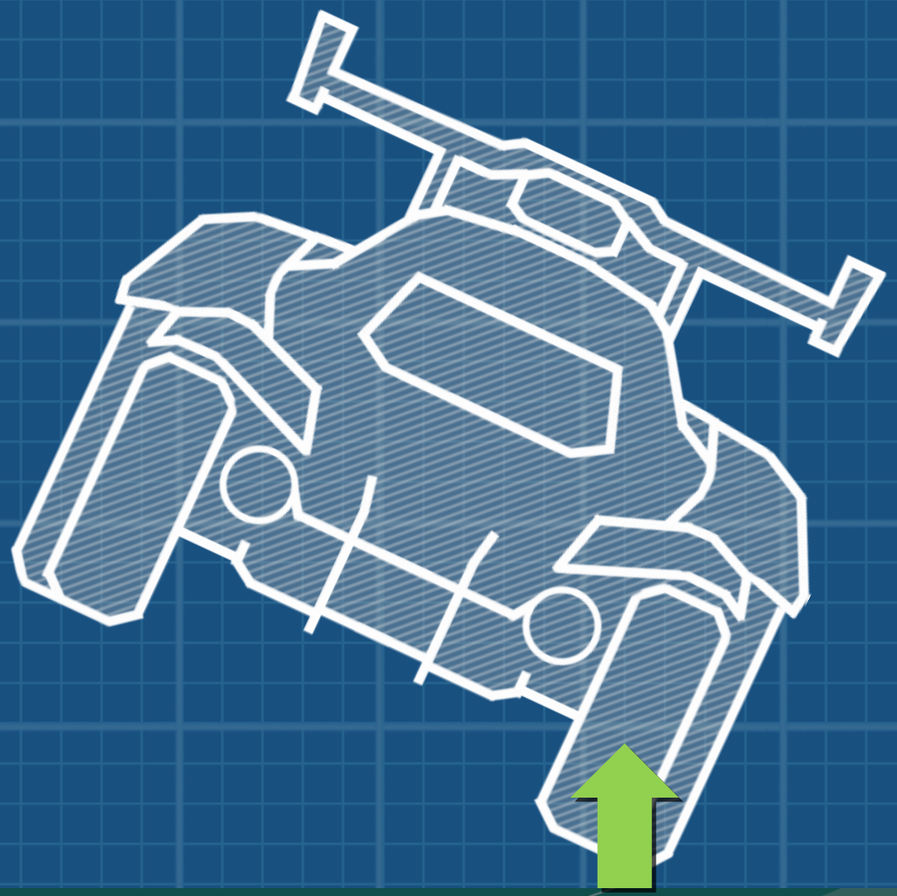
Stability Forces

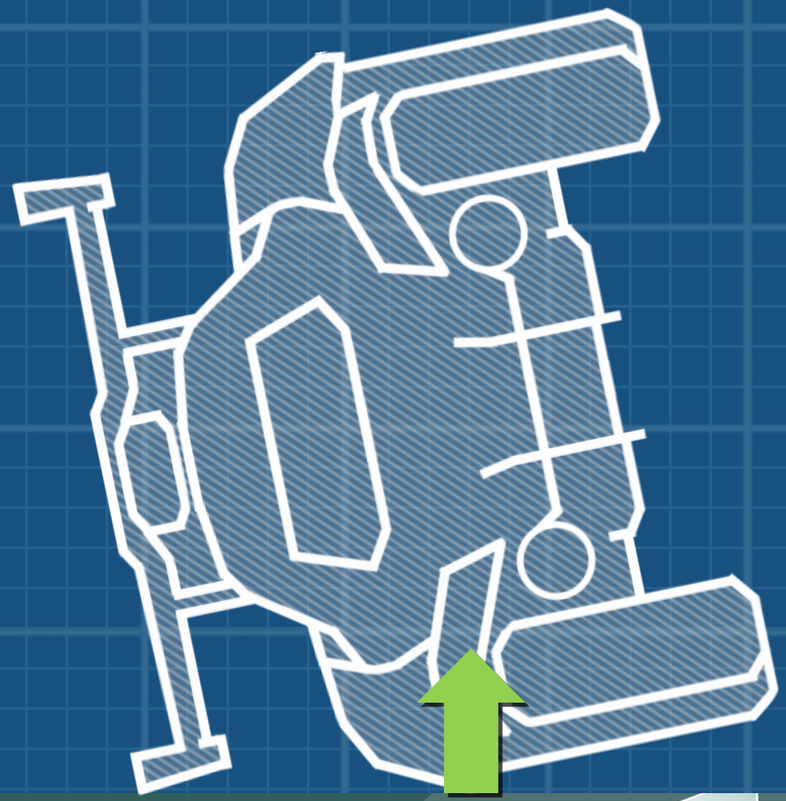
Stability Forces



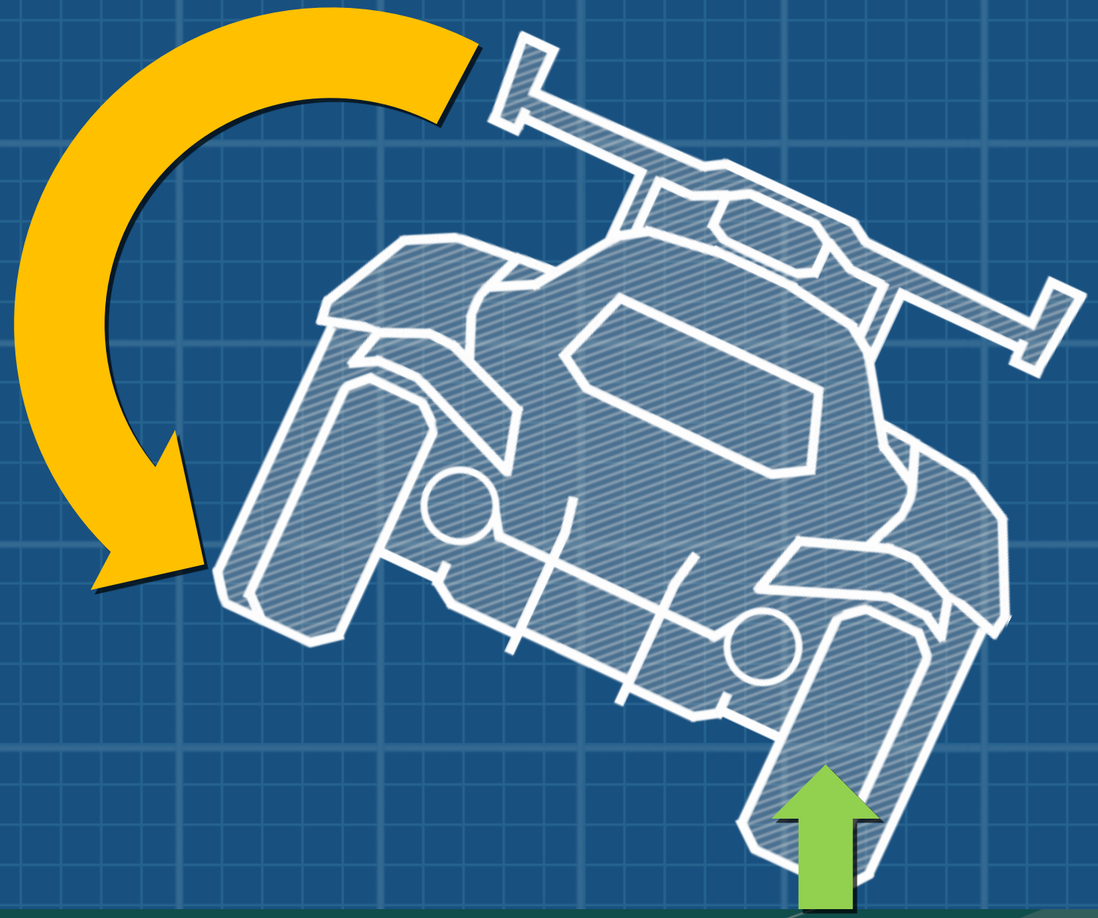


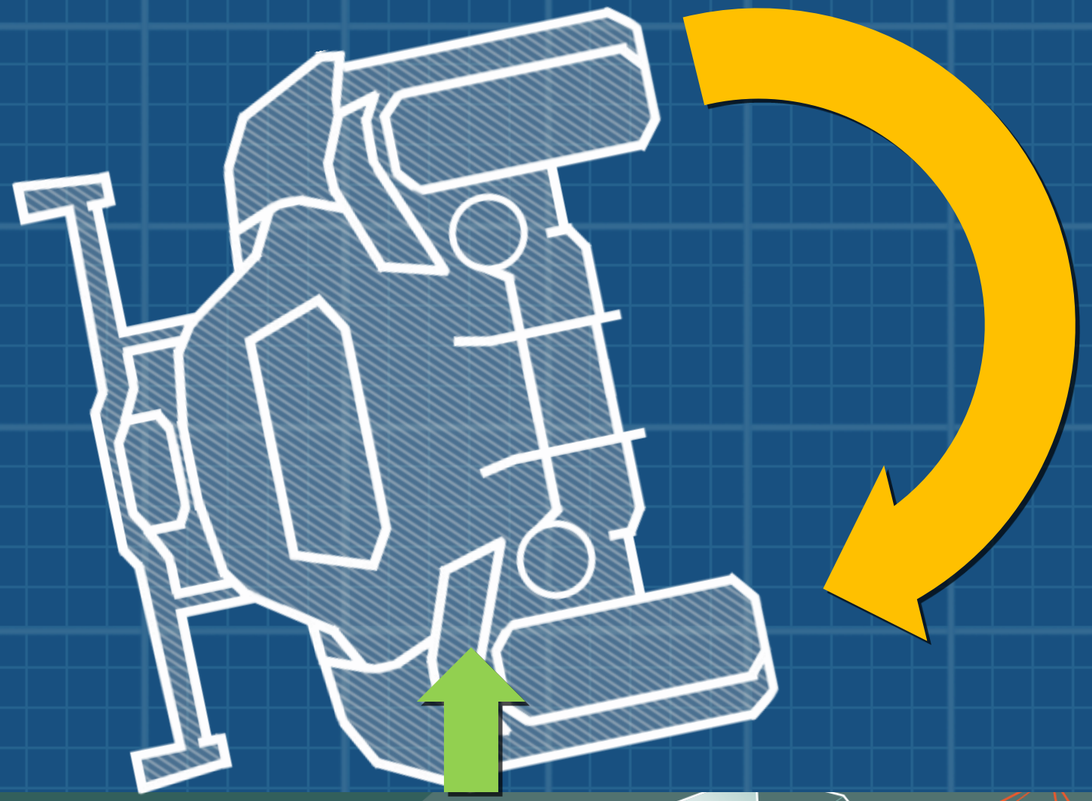
Stability Forces



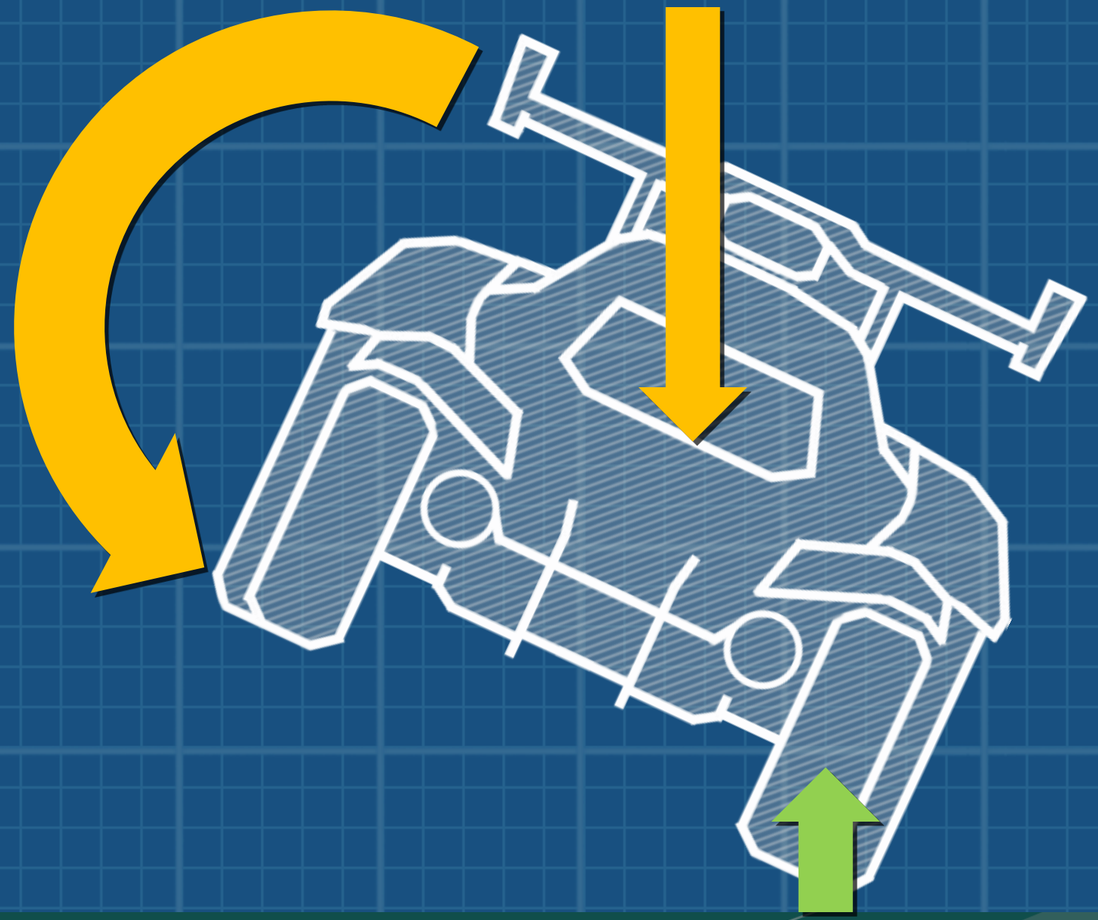


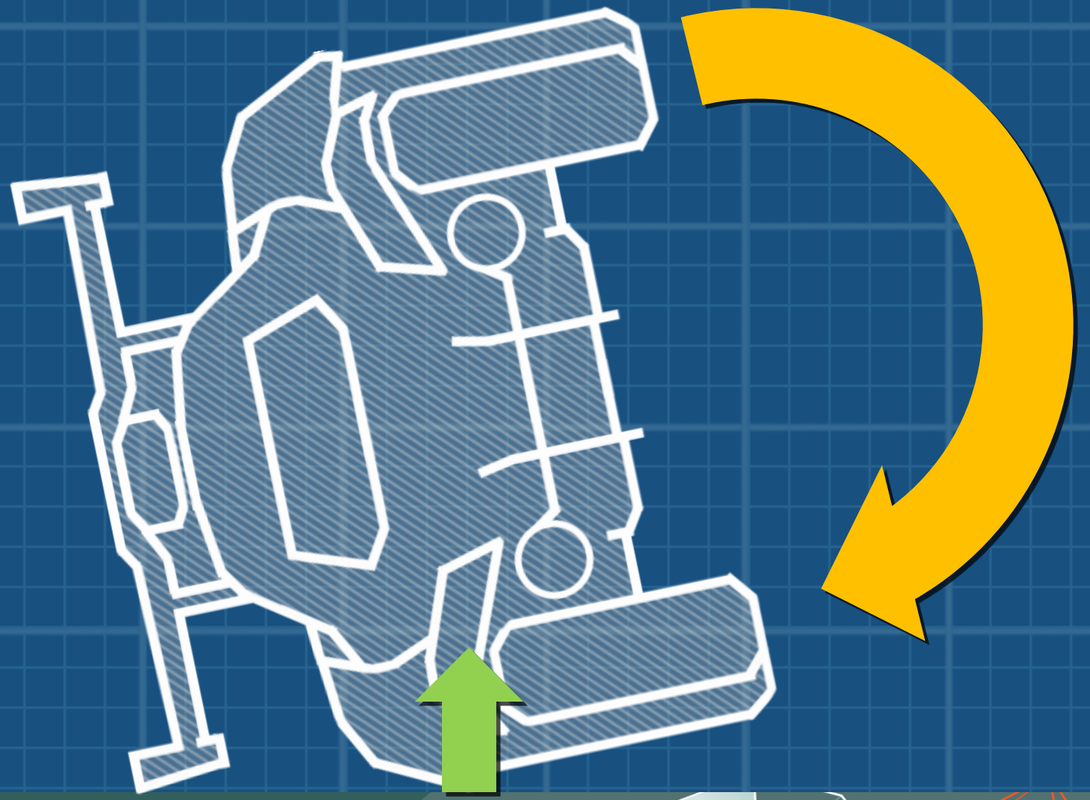
Stability Forces



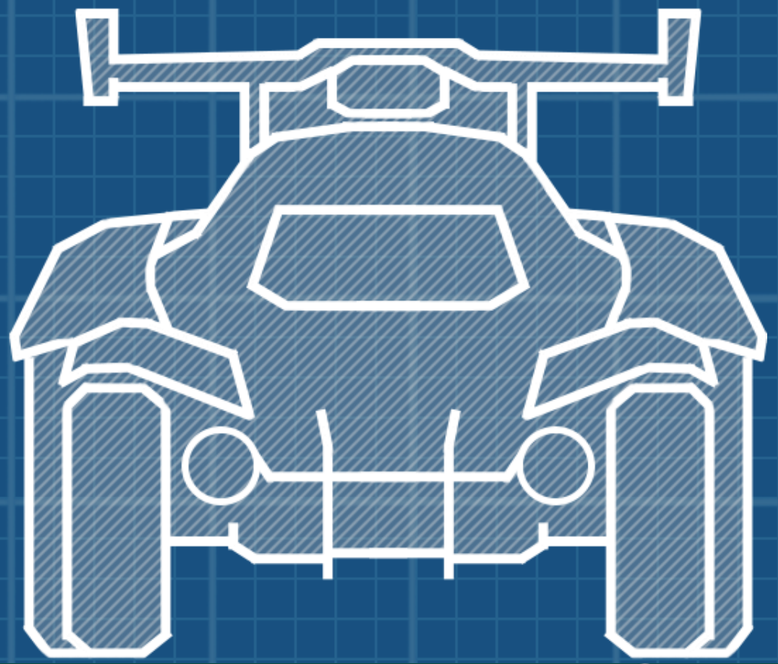


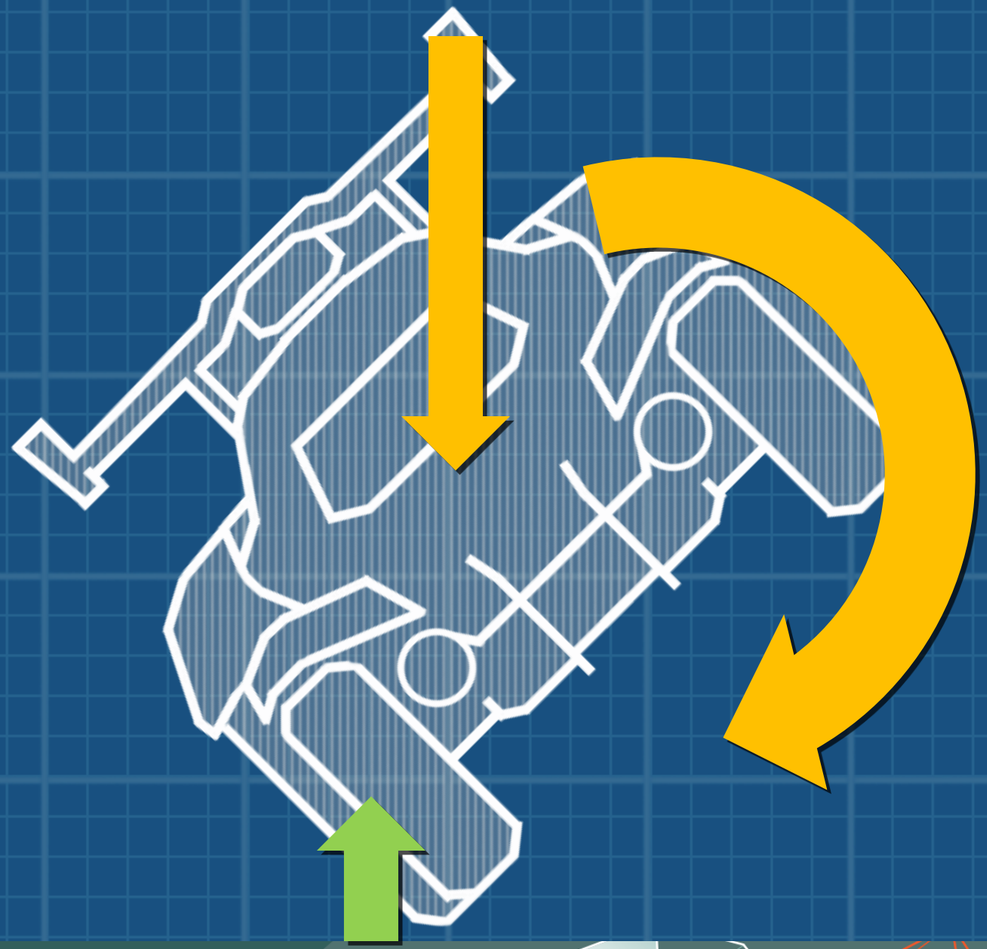
Stability Forces



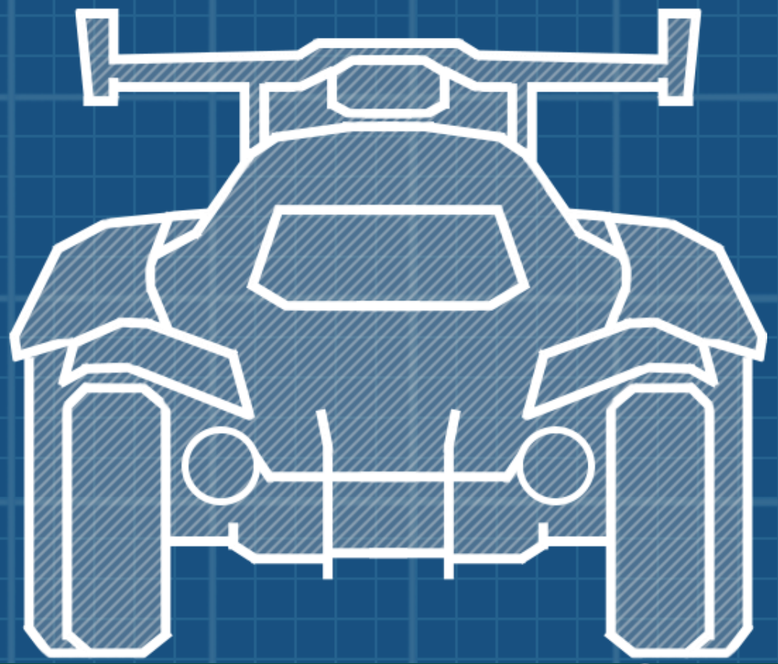


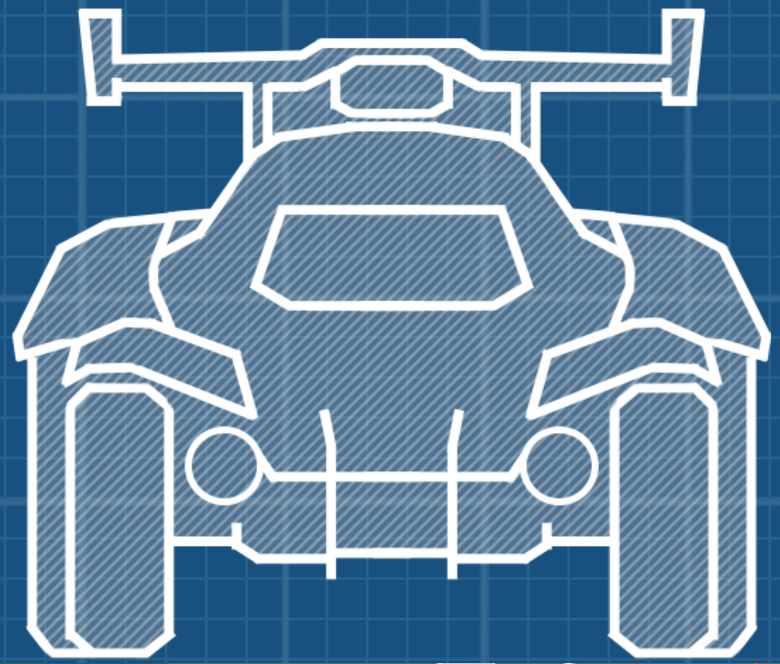
Stability Forces





Stability Forces





Vehicle Tuning Summary

- Start simple or reduce complexity
- Fake realism with visuals
- Separate vehicle setup from visuals

PHYSICS ENGINE
VEHICLE TUNING
NETWORKING

PHYSICS ENGINE
VEHICLE TUNING
NETWORKING

Rocket League Challenges

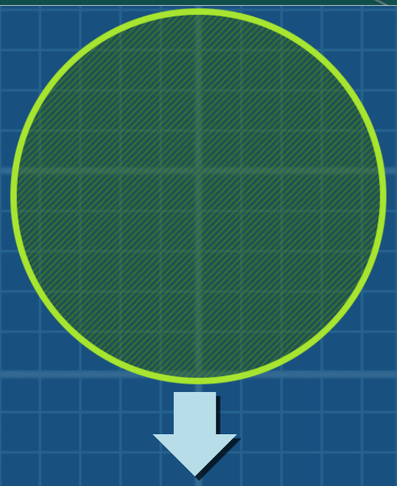
- Input delay is not an option
- Client prediction for rigid-body vehicles
- Server can’t wait for client input
- Collision with moving objects
- 100% server authoritative

Wait for client input?

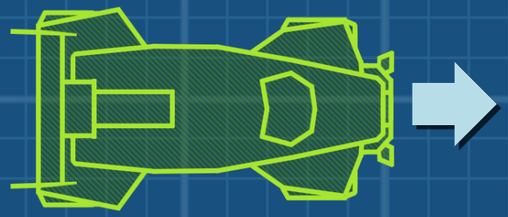
- Player inputs to server suffer jitter, loss
- To compensate, server waits for input before running physics
- Not good for rigid-body simulation
- Can result in de-sync when hitting moving object

Wait For Client Input

Client

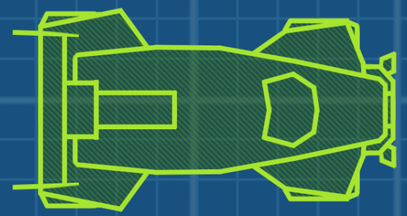


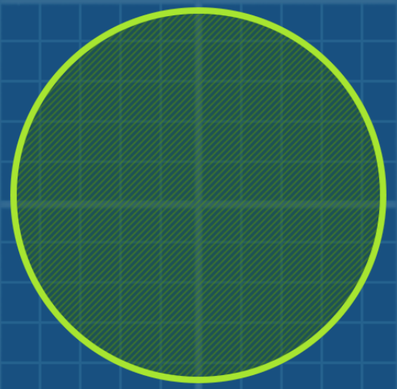
Wait For Client Input



Client

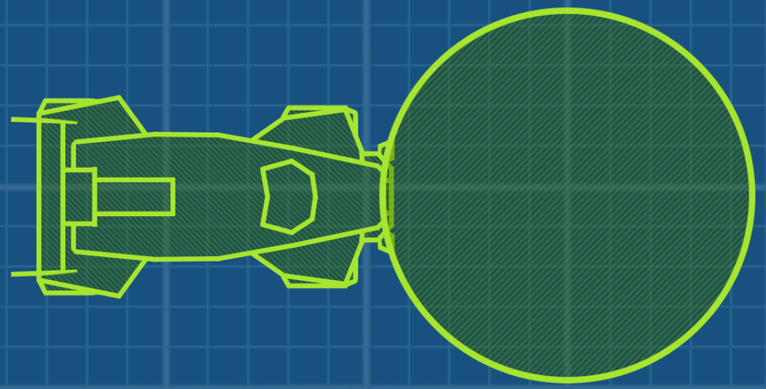
Wait For Client Input





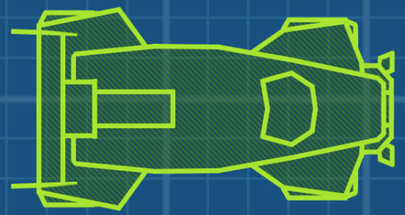
Client

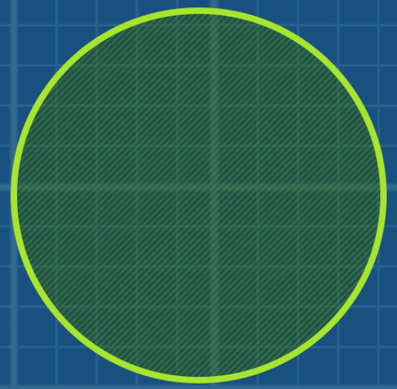
Wait For Client Input



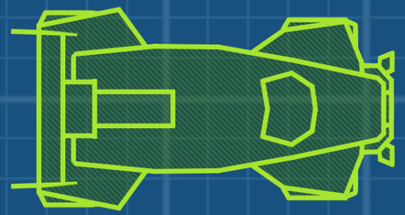
Client

Wait For Client Input

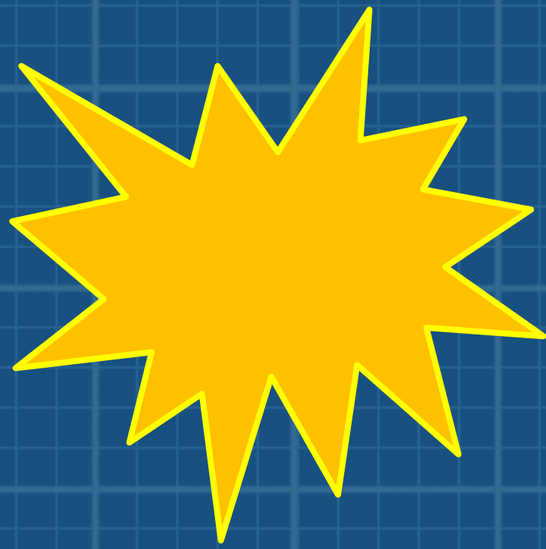




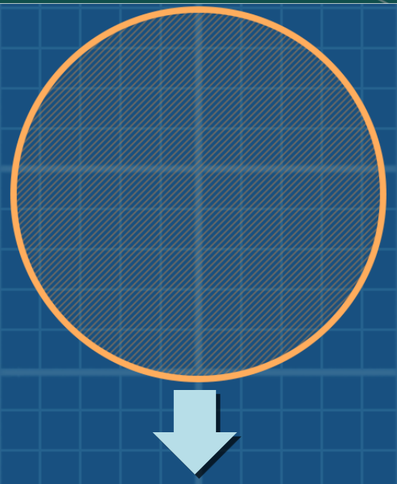
Client



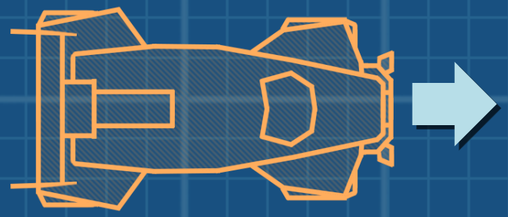
Wait For Client Input



Server

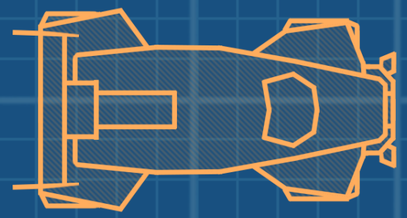


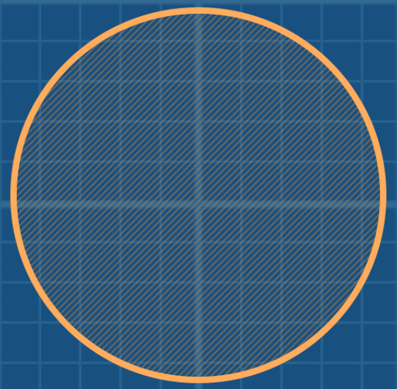
Wait For Client Input



Server

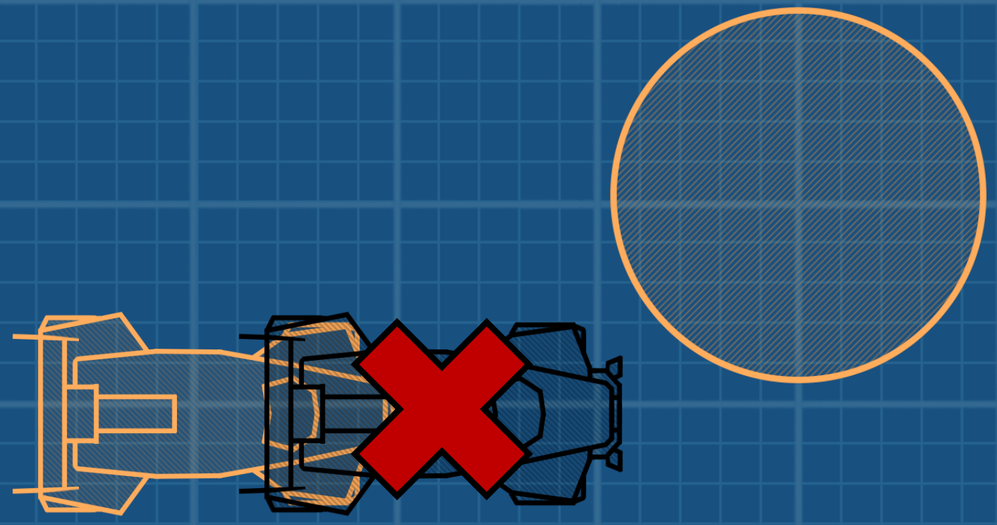
Wait For Client Input





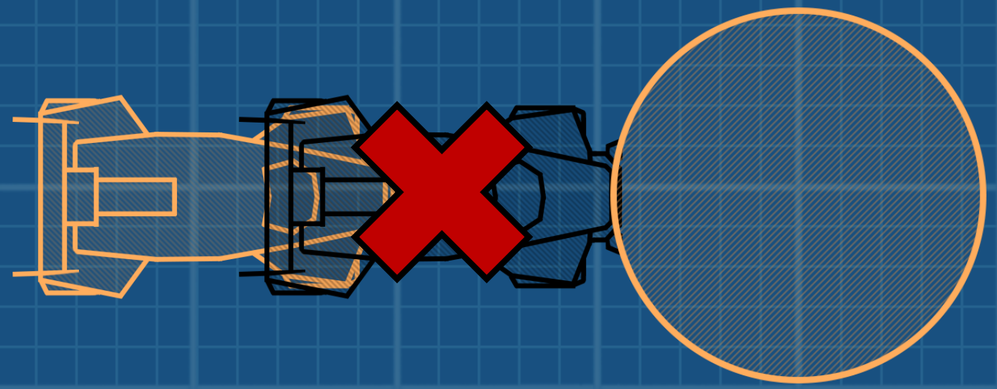
Server

Wait For Client Input



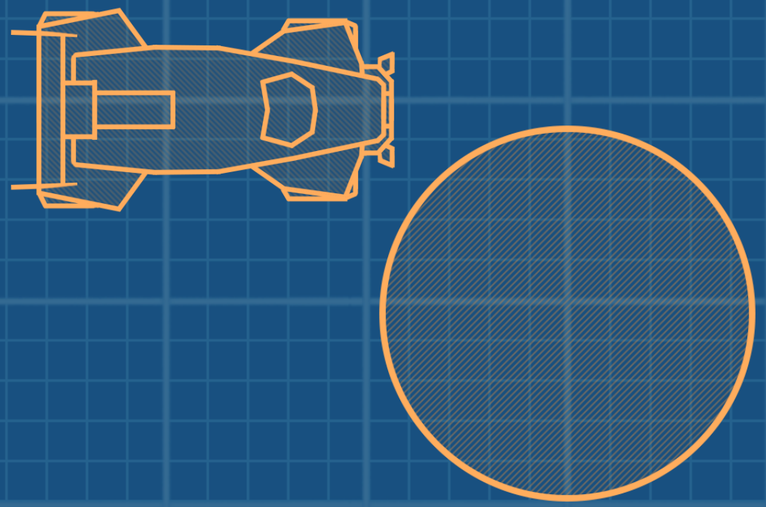
Server

Wait For Client Input



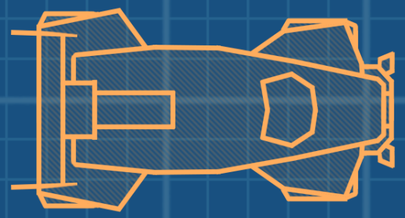
Server

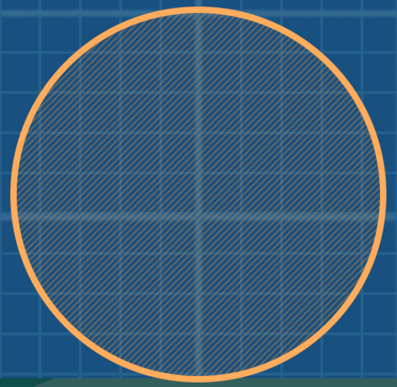
Wait For Client Input



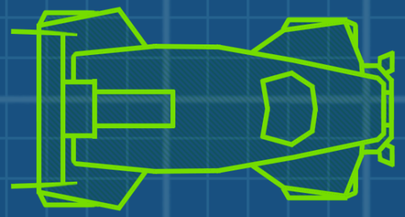
Server

Wait For Client Input

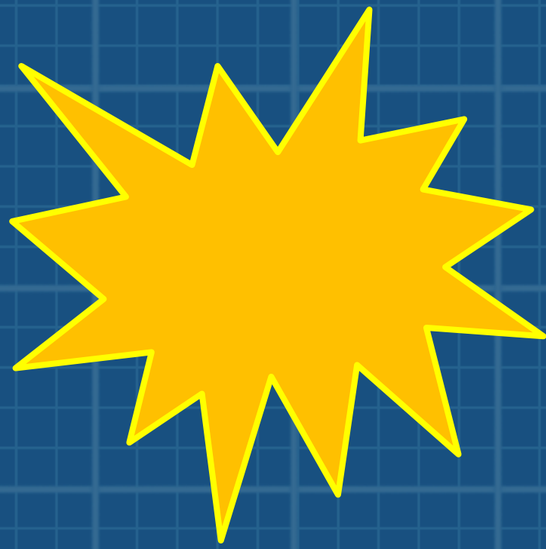




Client

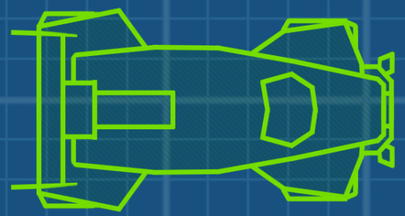


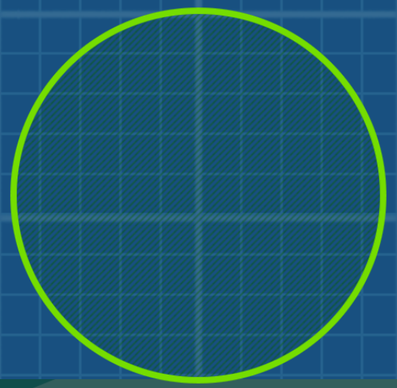
Wait For Client Input



Client

Wait For Client Input



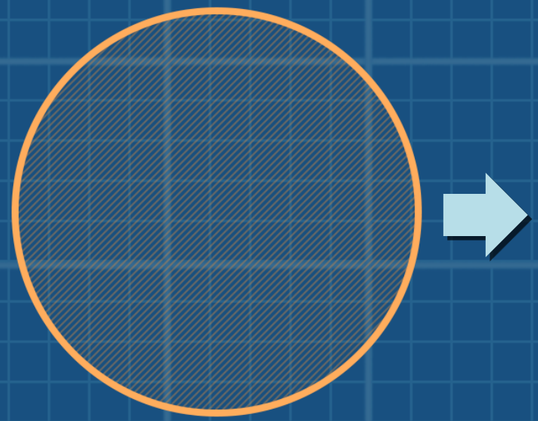


Hit Moving Object?

- Client predicts his vehicle
- Server authoritative ball
- Client interpolates ball
- Client's vehicle and the ball exist in two different timelines

Hit Moving Object

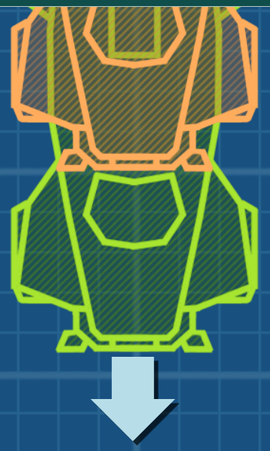
Server

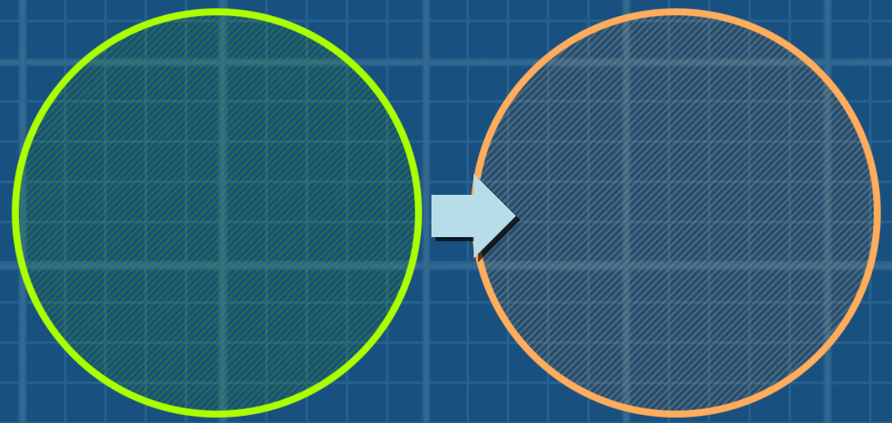


Hit Moving Object

Hit Moving Object

Server
Client

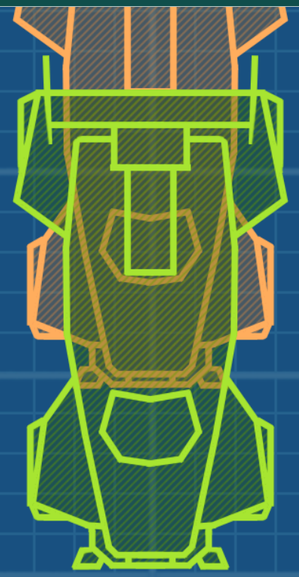


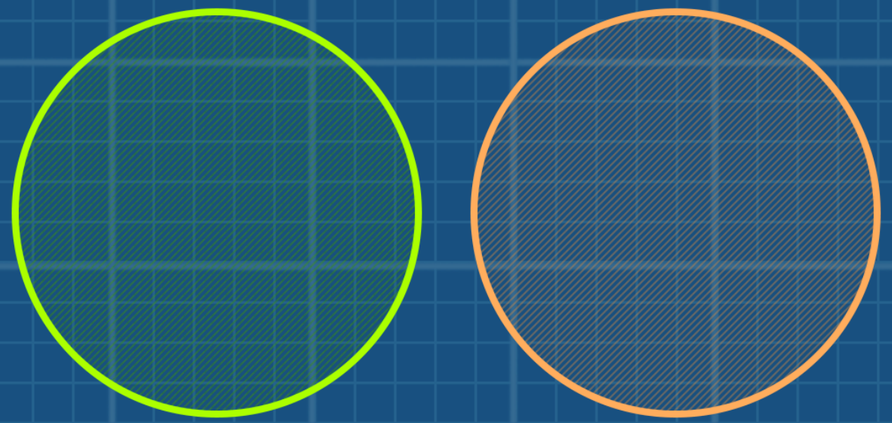


Hit Moving Object

Server

Client

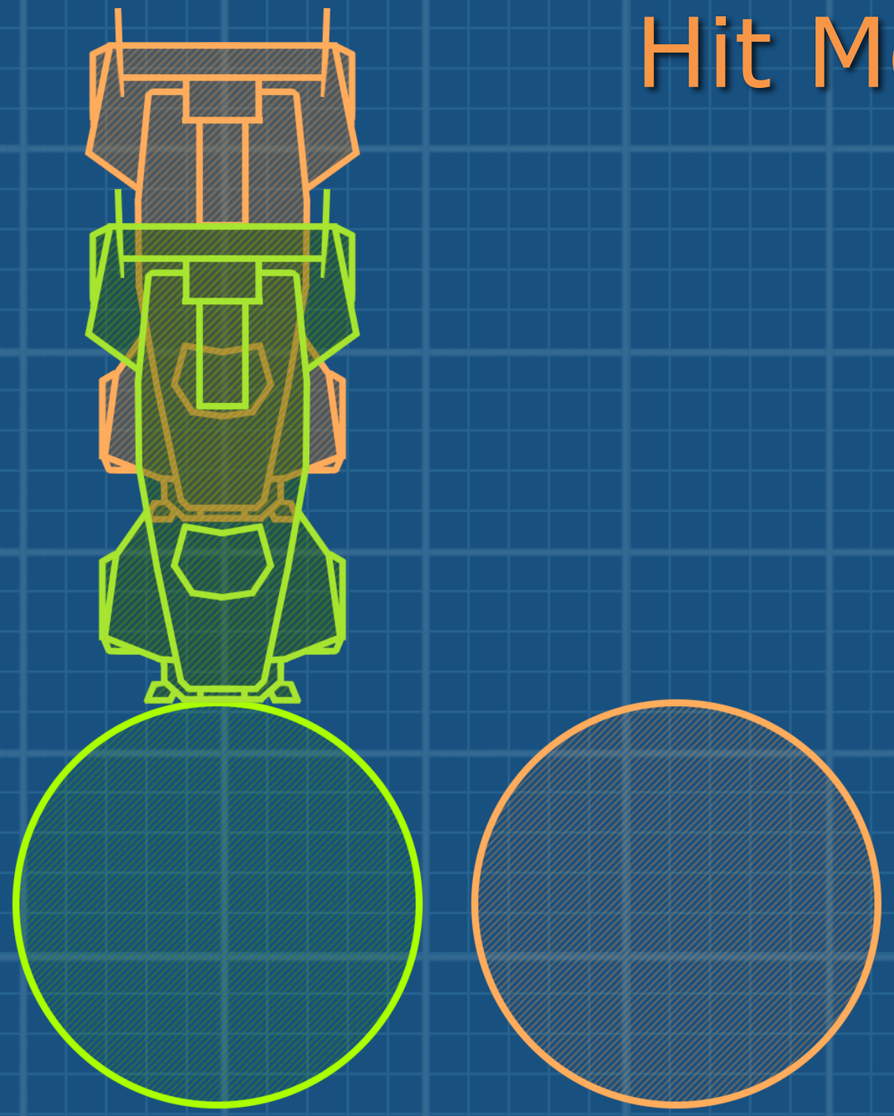




Hit Moving Object

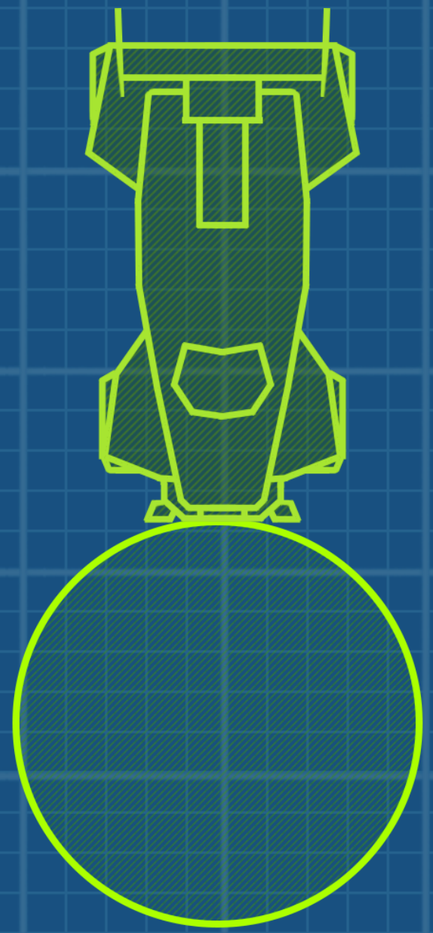
Server

Client



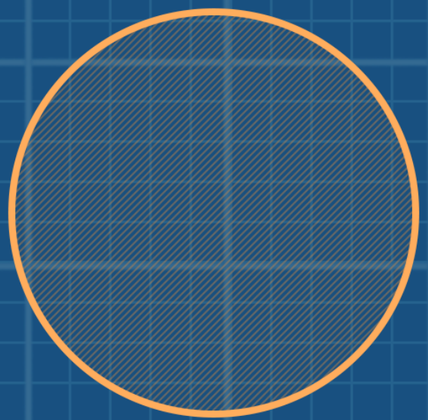
Hit Moving Object

Client



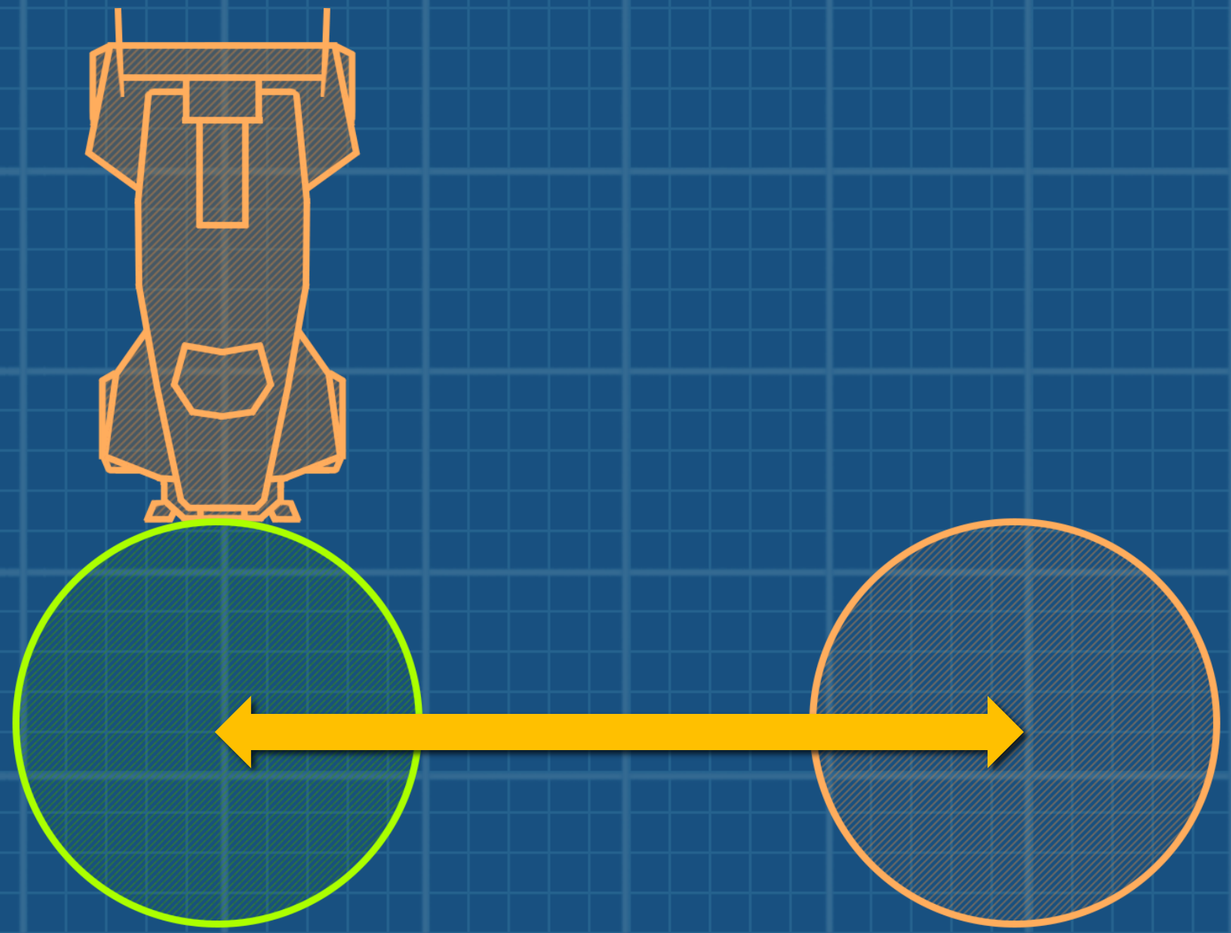
Server

Hit Moving Object



Server

Hit Moving Object



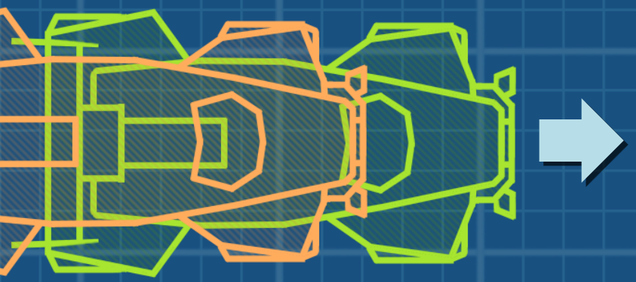
Lag Compensation?

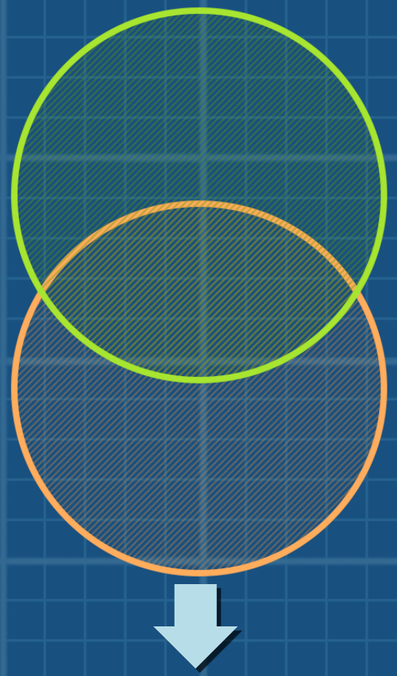
- Take idea from FPS lag compensation
  - Client predicts shot, server confirms
- How to apply to Rocket League?
  - Client predicts hitting the ball
  - Server performs lag compensation, confirms hit, updates ball trajectory

Lag Compensation

Lag Compensation

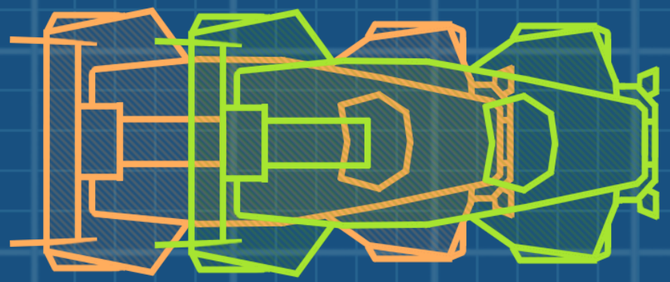
Server
200ms Client

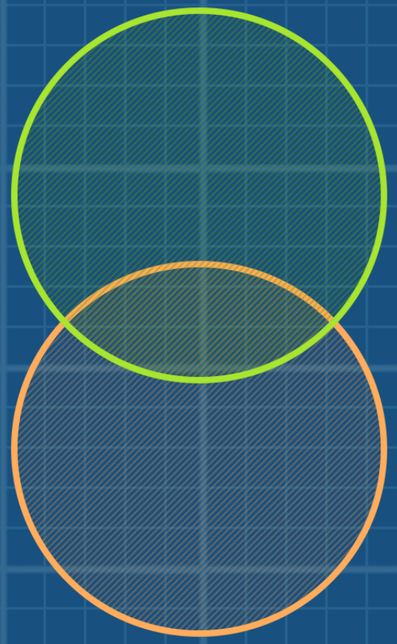




Lag Compensation

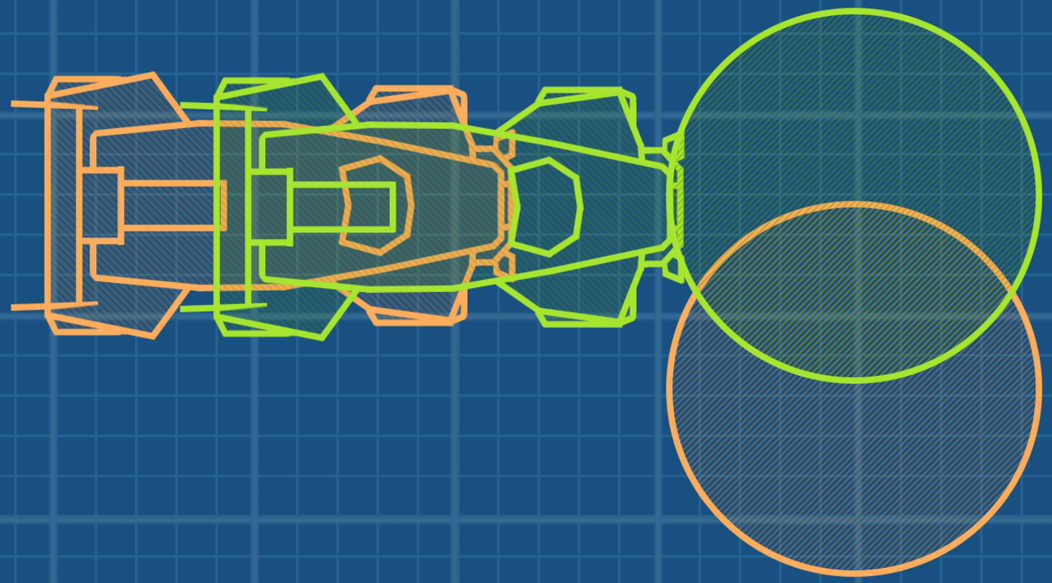
Server
200ms Client





Lag Compensation

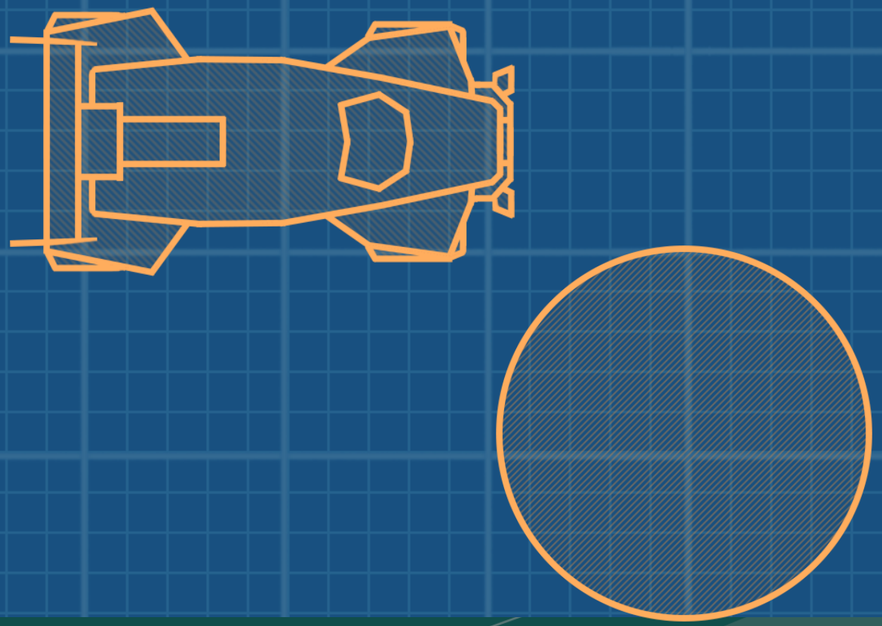
Server
200ms Client



Lag Compensation

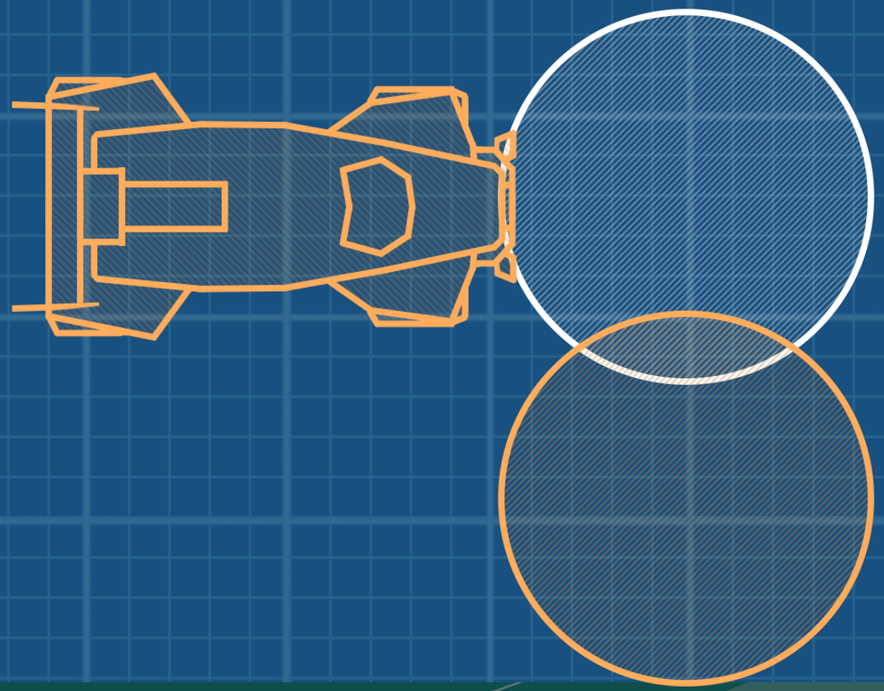
Server

200ms Client



Lag Compensation

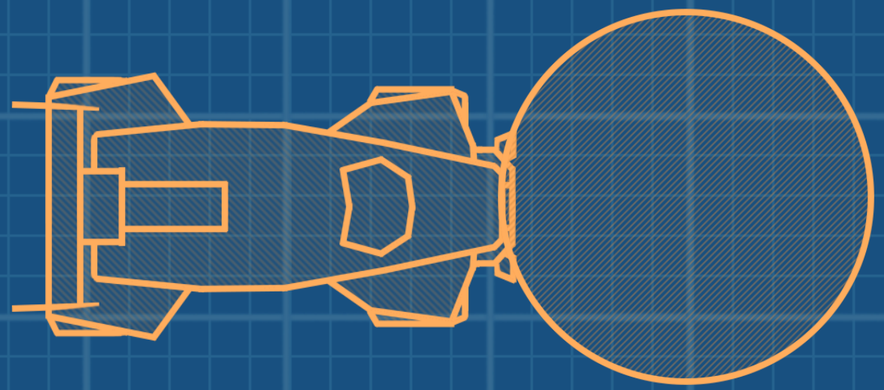
Server
200ms Client



Lag Compensation

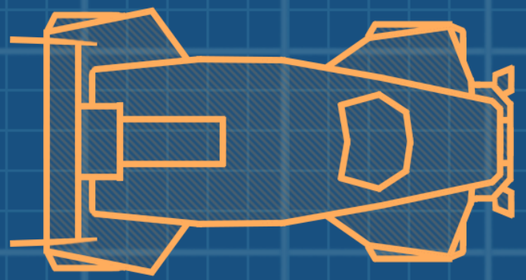
Server

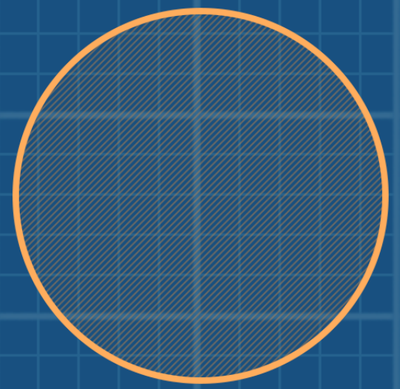
200ms Client



Lag Compensation

Server
200ms Client

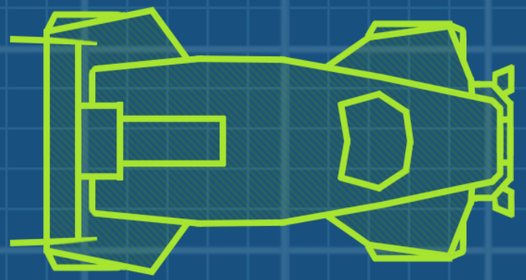


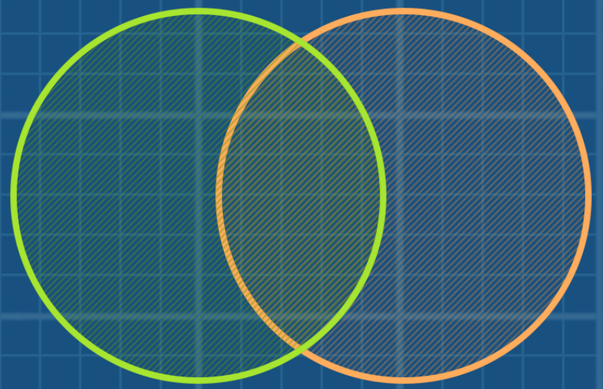


Lag Compensation

Server

200ms Client

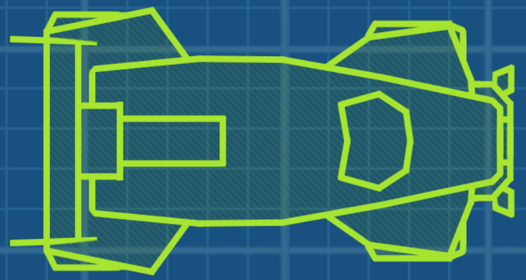


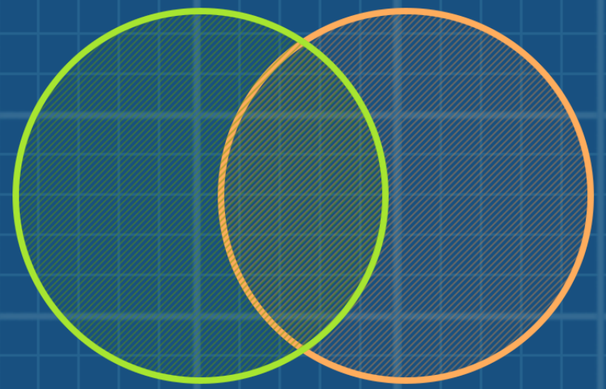


Lag Compensation

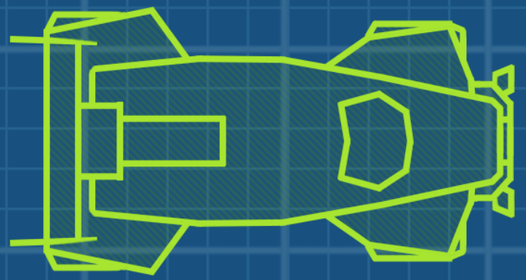
Server

200ms Client

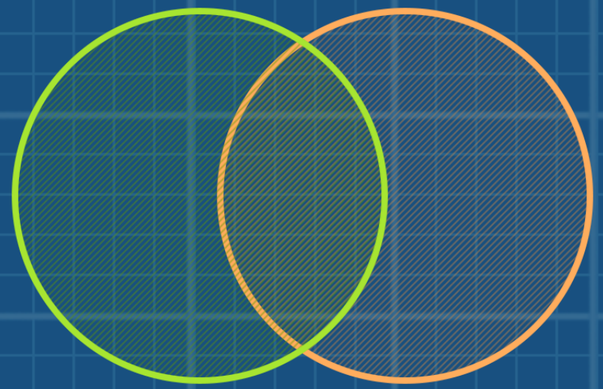




Server
200ms Client

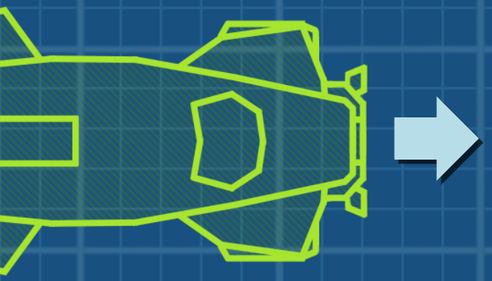


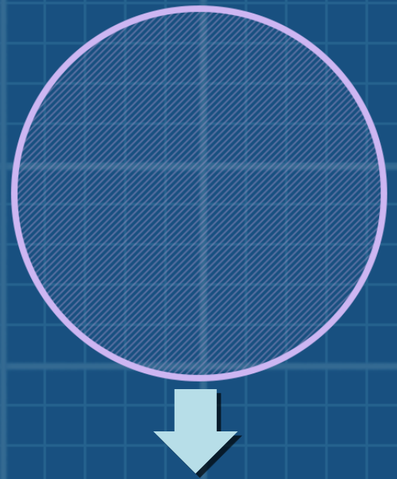
Lag Compensation

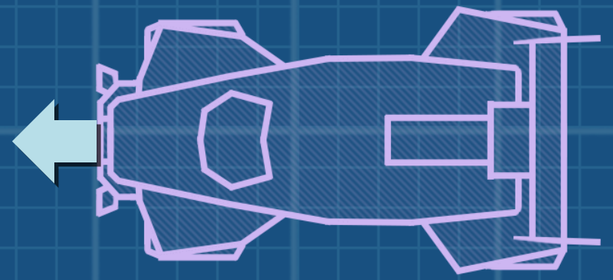


10ms Client

Lag Compensation

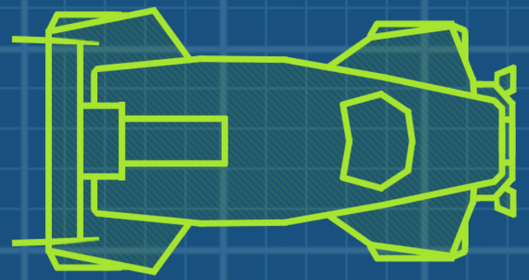


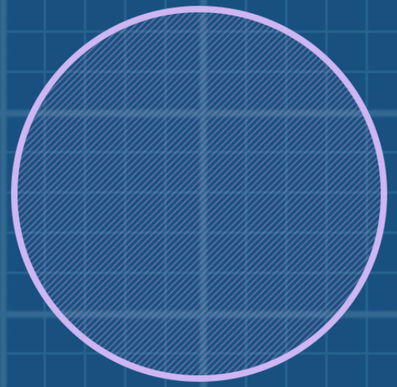


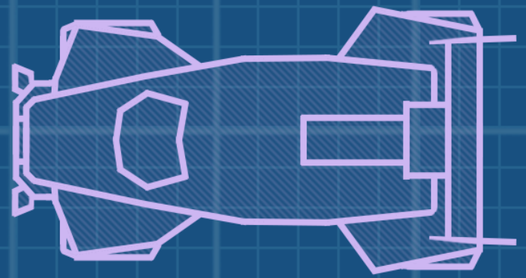


10ms Client

Lag Compensation

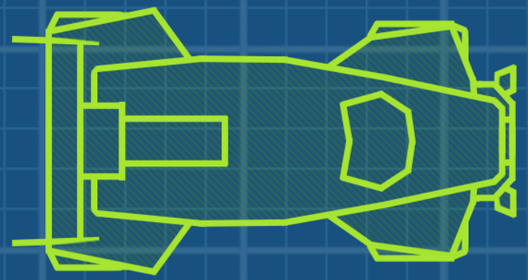


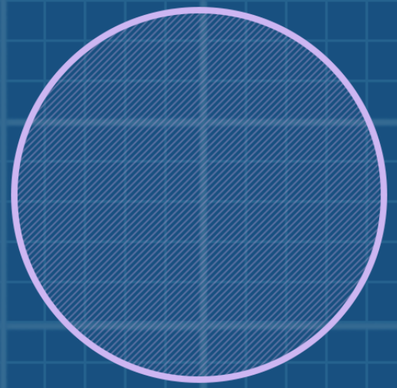


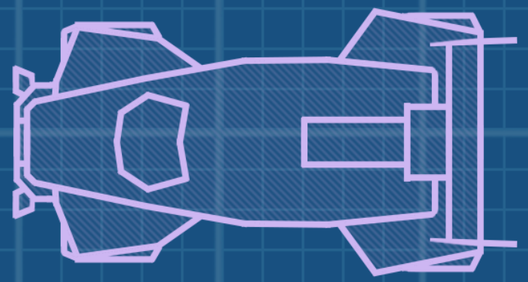


10ms Client

Lag Compensation

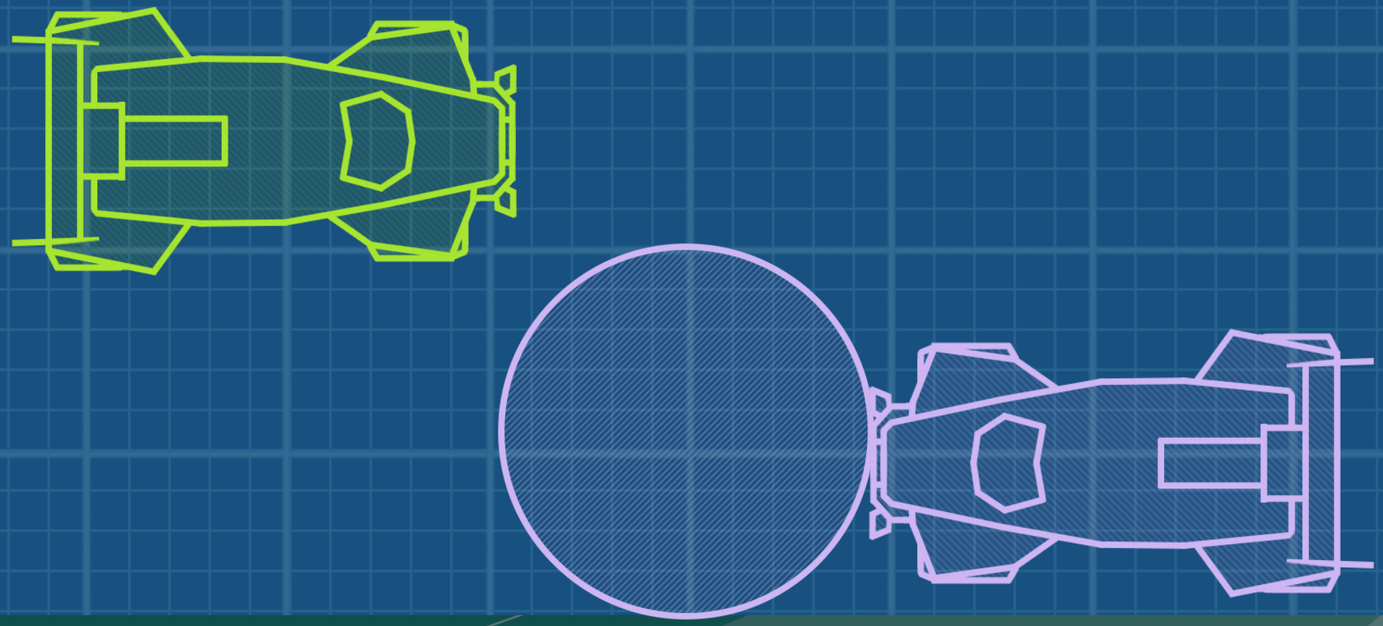






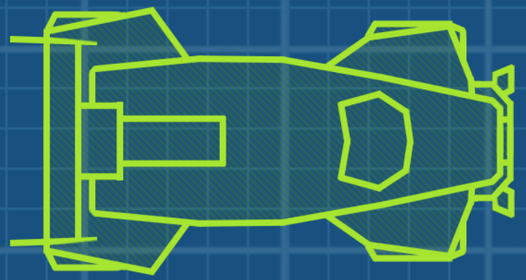
10ms Client

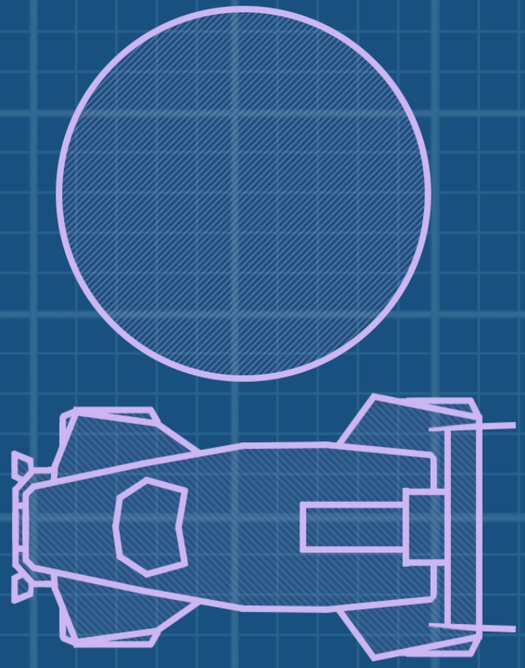
Lag Compensation



10ms Client

Lag Compensation





Rocket League Networking

- Server buffers player inputs
- Client predicts everything

NETWORKING

- INPUT BUFFER
- PREDICT EVERYTHING

Input Buffer

- Client sends input for every physics frame
- Client inputs do not arrive at constant rate
- Server buffers client input
- No need to pause for input
- Eliminates some cheats (speed, jitter)
- Increases average latency

Input Buffer Requirements

- Try to avoid empty buffer (runs physics using previous player input)
- Try to avoid large buffer (adds latency)
- How to grow or shrink buffer?

Upstream Throttle

- Server can tell client to run faster or slower
- Buffer low? Client runs extra physics frames
- Buffer full? Client runs fewer physics frames
- ‘Overwatch’ Gameplay Architecture and Netcode GDCVault.com

Downstream Throttle

- Server consumes 0, 1, or 2 inputs per frame
- Buffer low? Use 1 input for 2 frames
- Buffer full? Consume 2 inputs for 1 frame
- Effective but with minor desyncs

NETWORKING

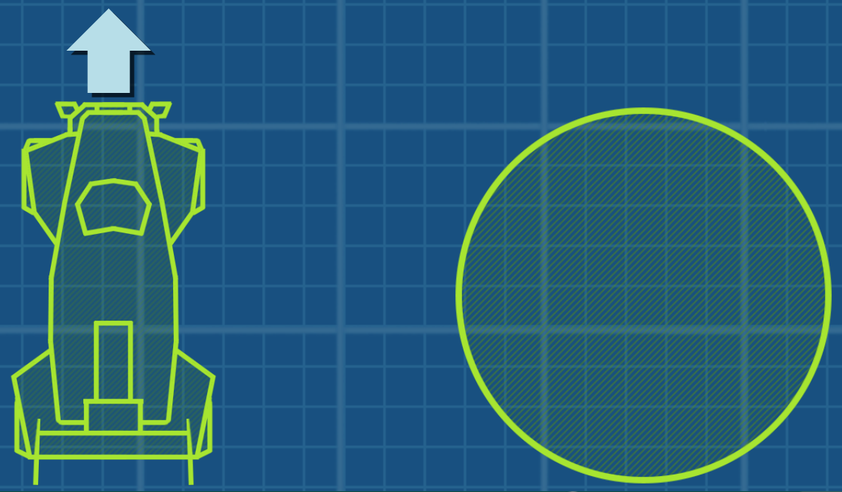
- INPUT BUFFER
- PREDICT EVERYTHING

NETWORKING

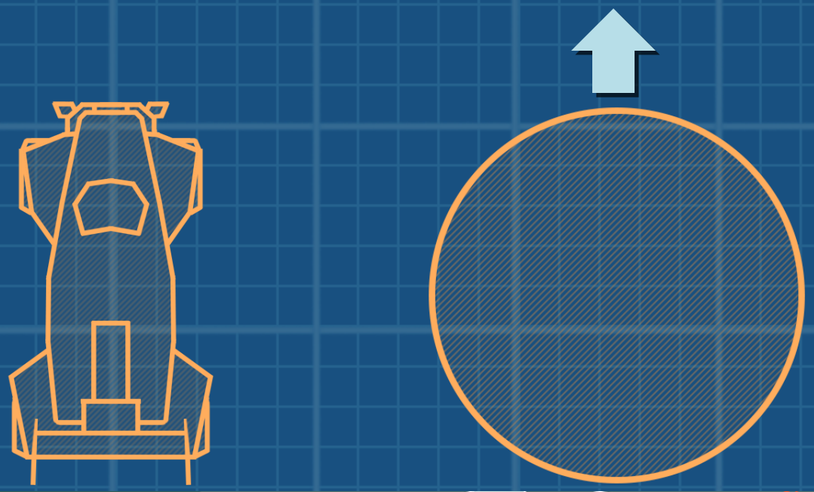
- INPUT BUFFER
- PREDICT EVERYTHING

Prediction & Correction

Client

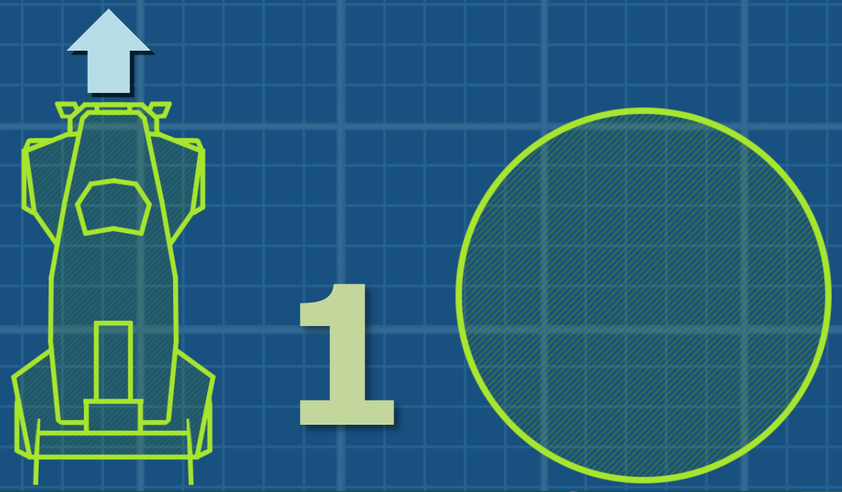


Server

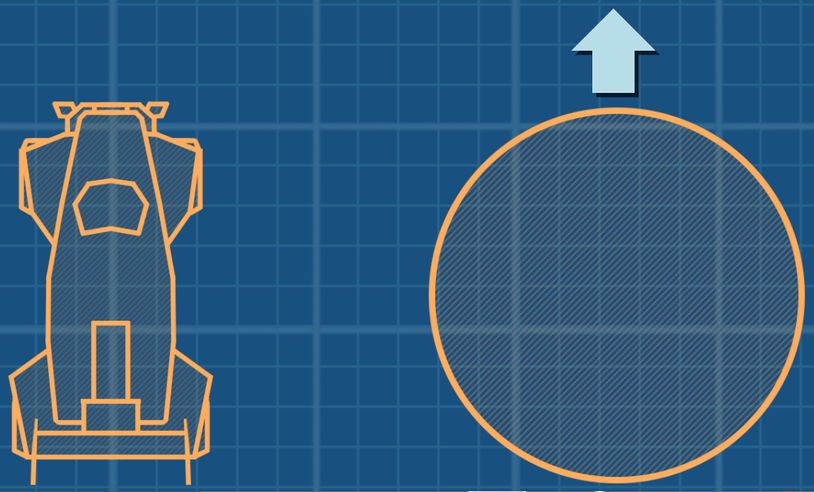


Client

Record input, frame #



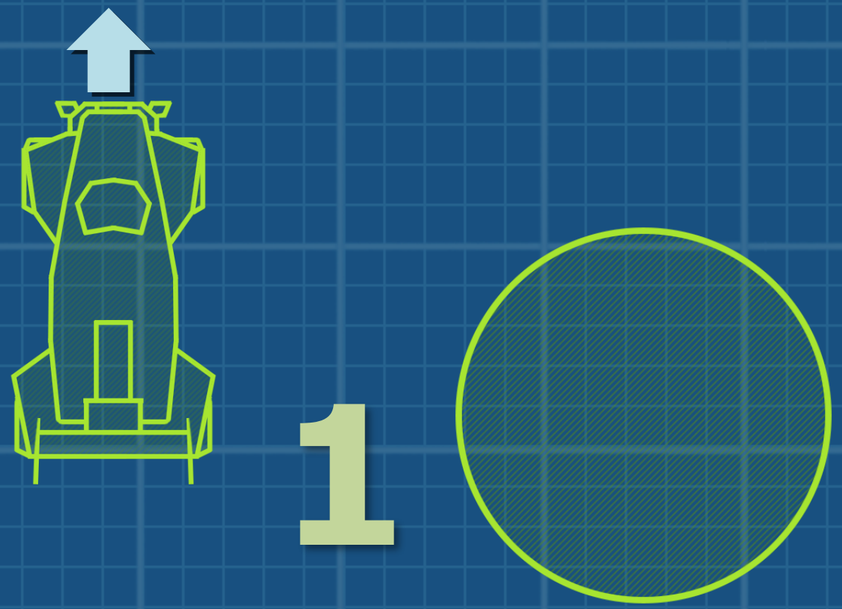
Server

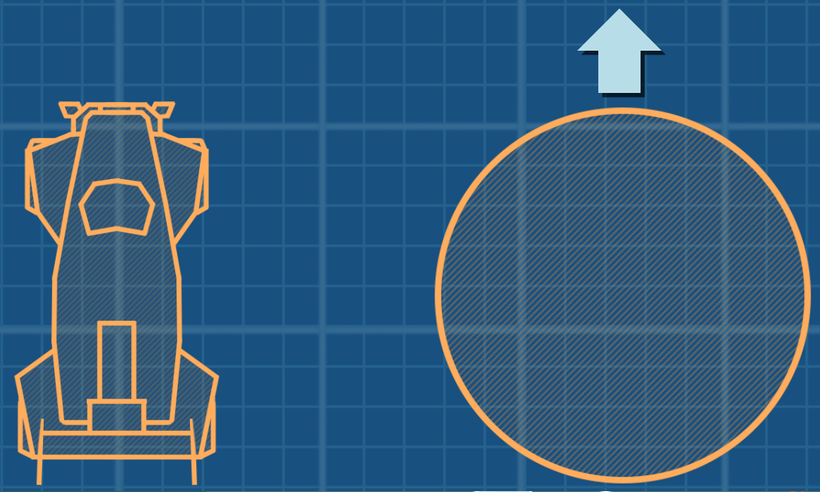


Client

Server

Run physics

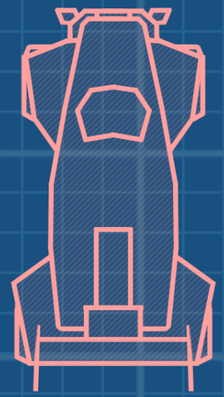




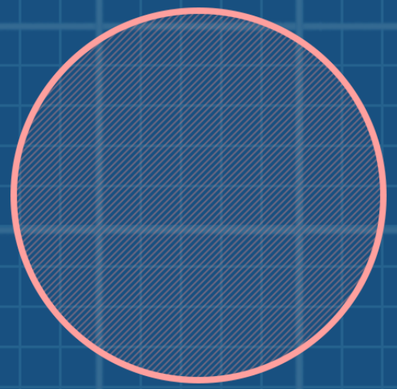
Client

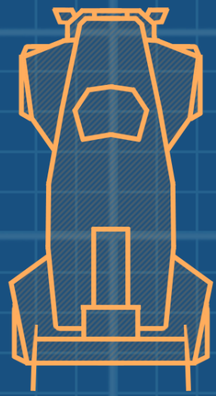
Server

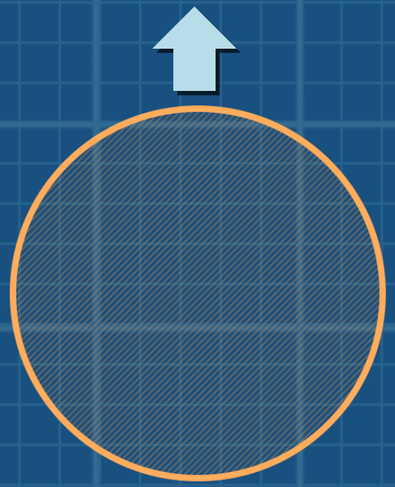
Record history



1



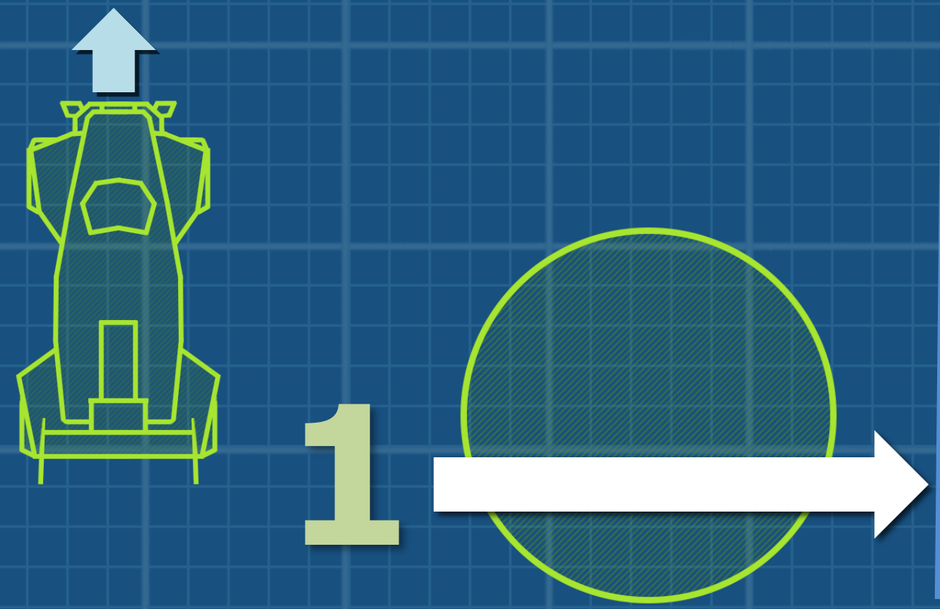


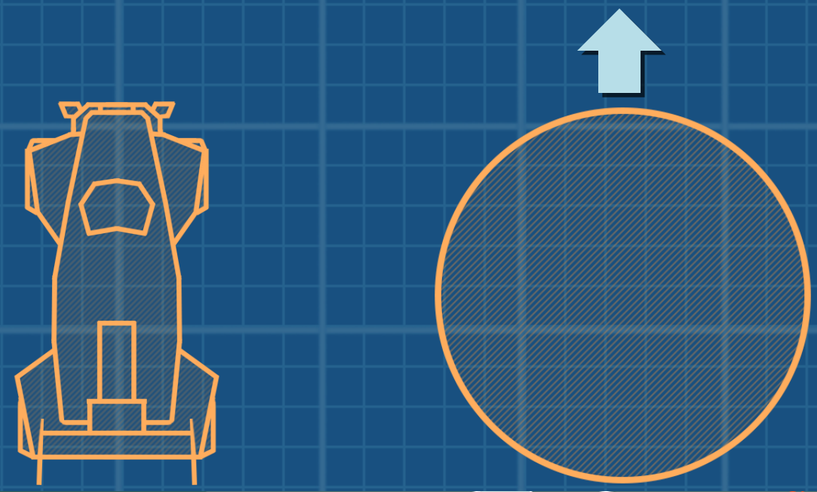


Client

Server

Send input, frame #

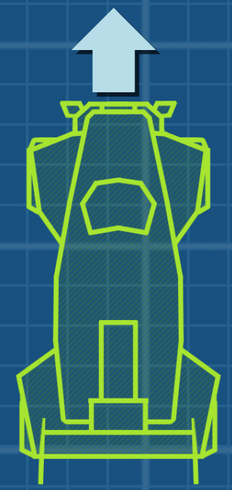


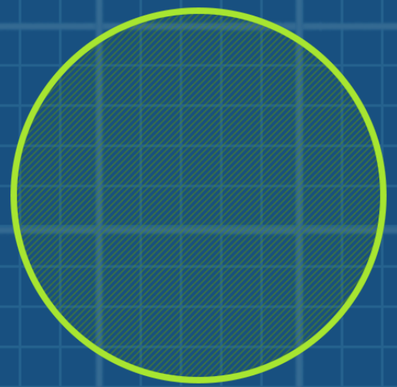


Client

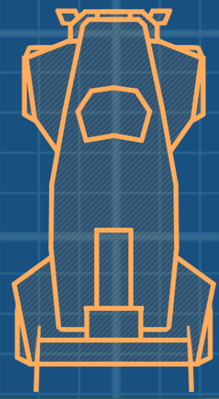
Server

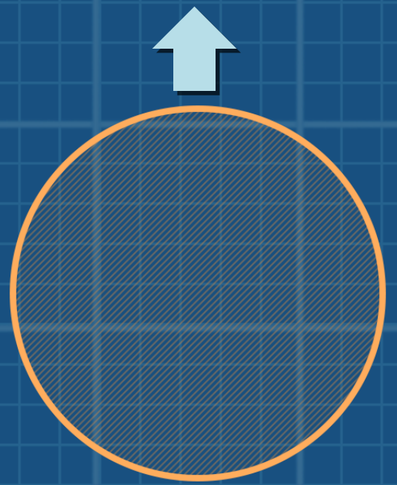
Send input, frame #





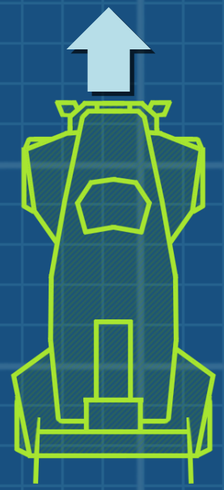
1

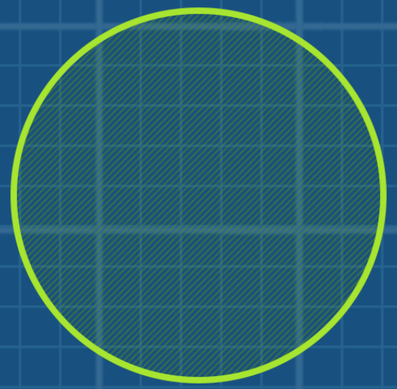




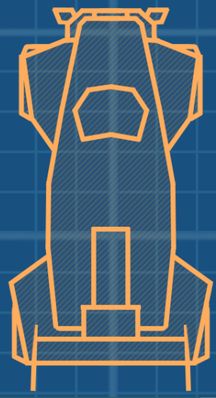
Client

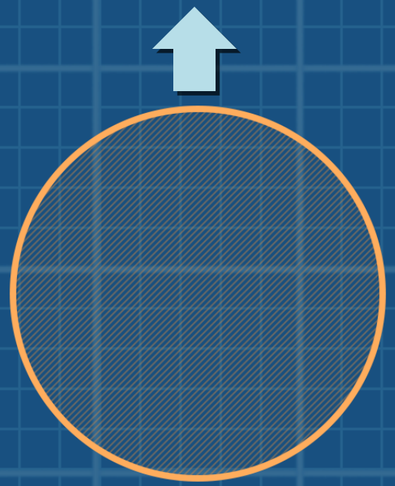
Server





2 0 1

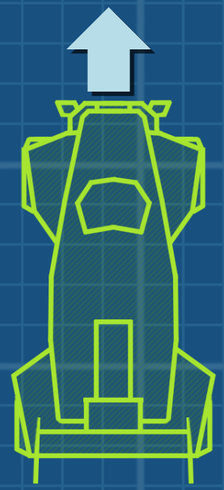




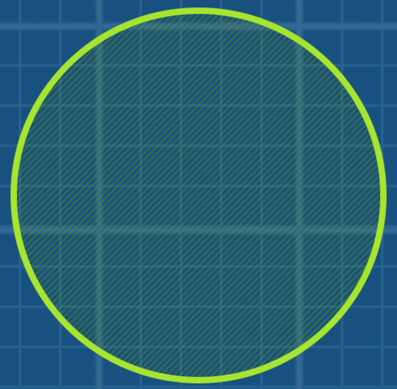
Client

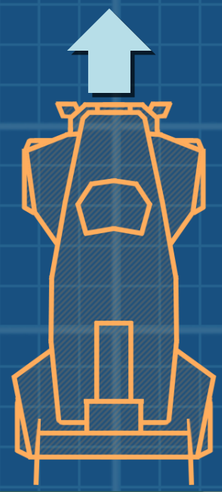
Server

Read input, run physics

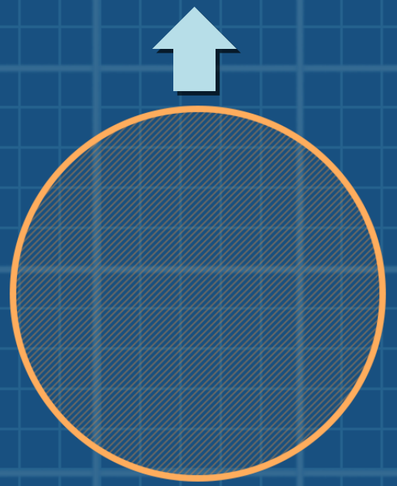


2

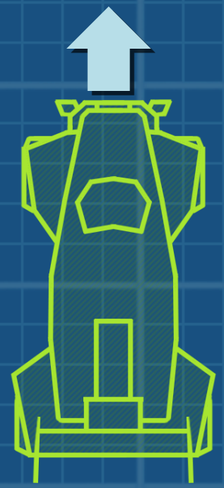




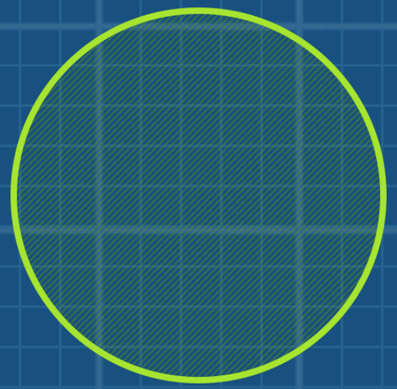
1



Client

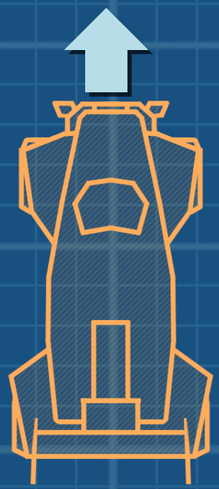


3

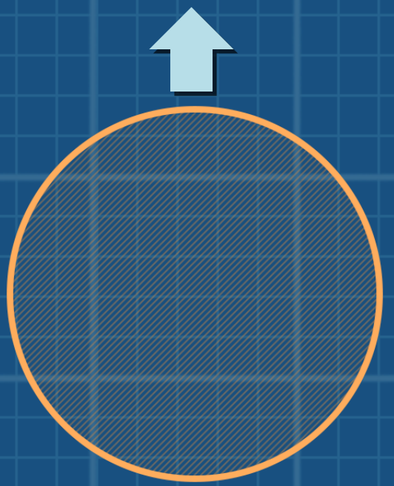


Server

Read input, run physics



1

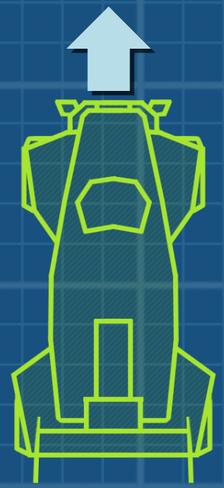


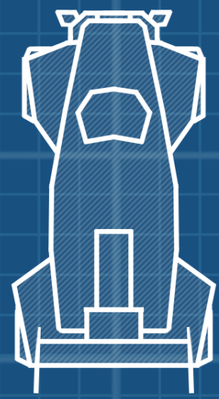
Client

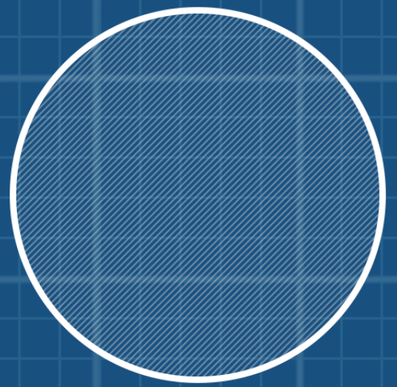
Server

Send client frame #, physics state

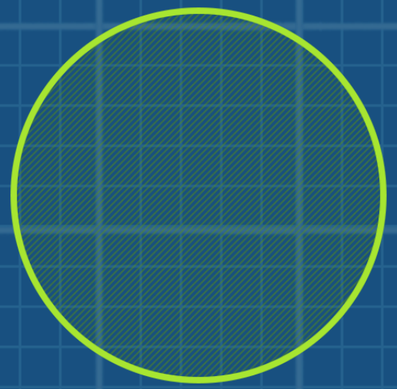
1





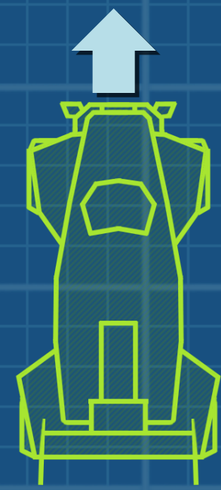


3

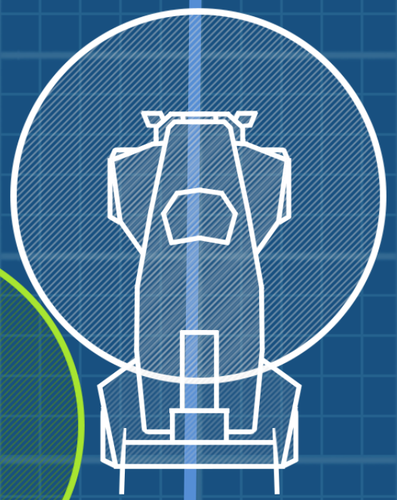


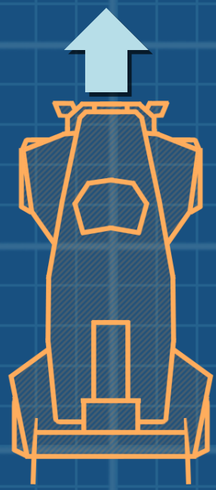
Client

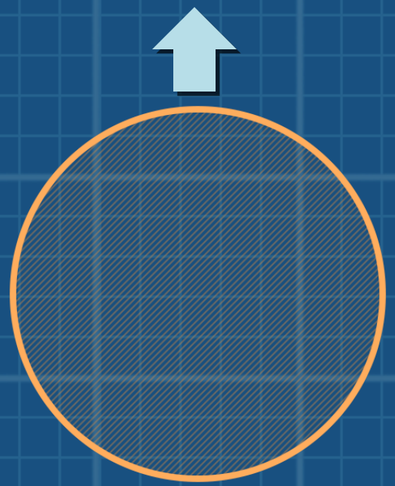
Server



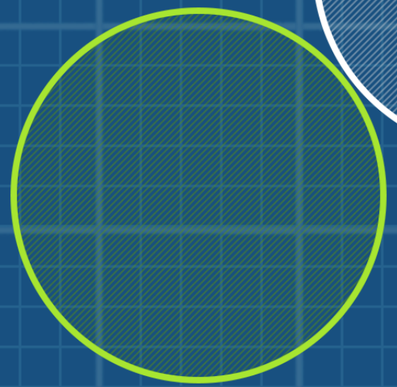
1

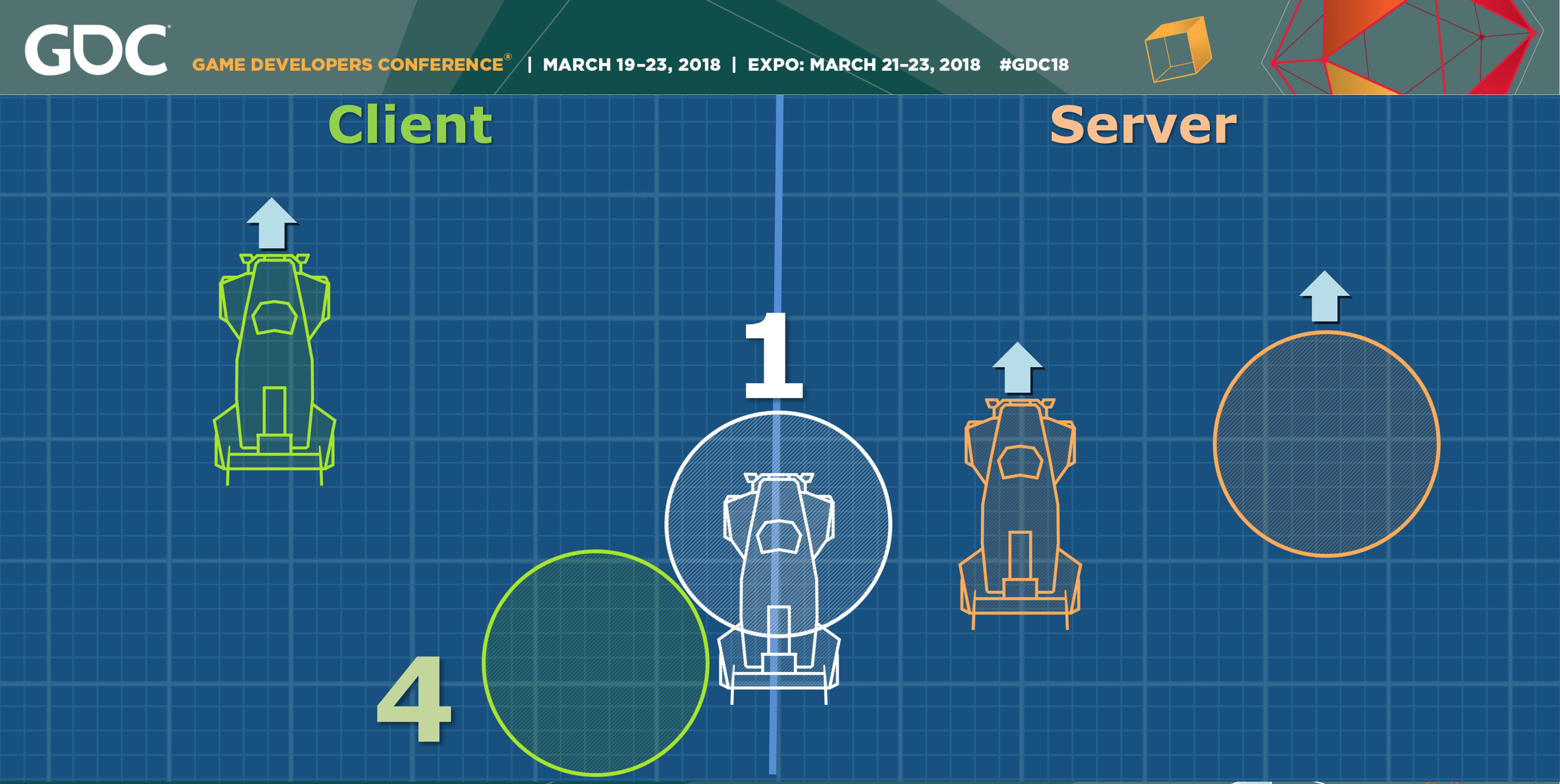






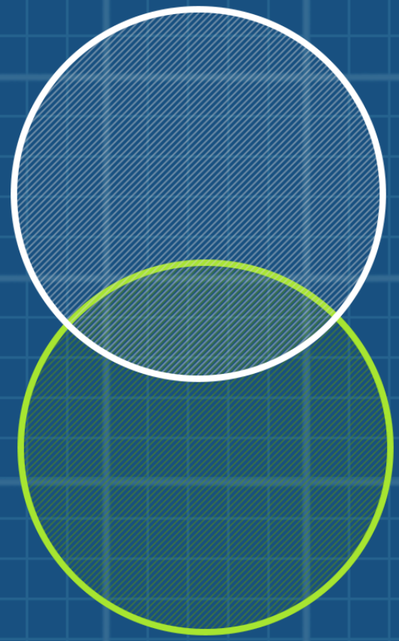
3




Client
Server
1
4

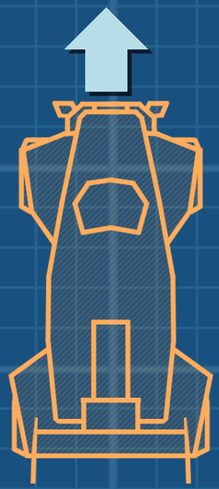
Client

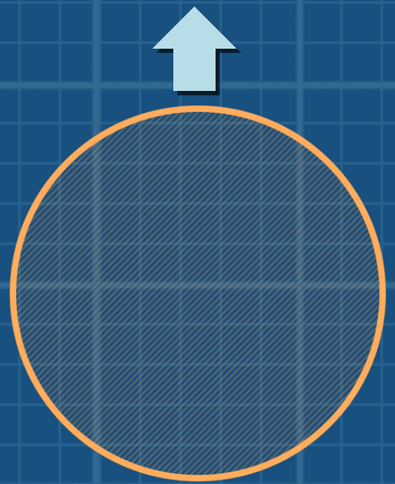
1



4

Server

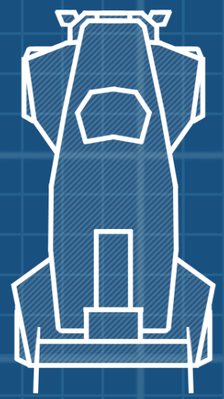


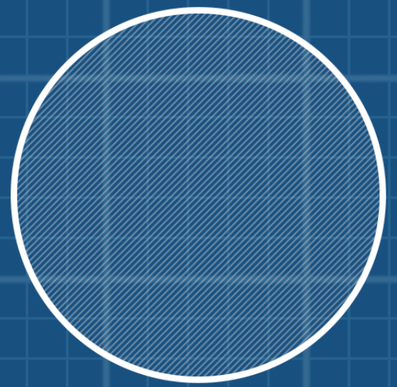


Client

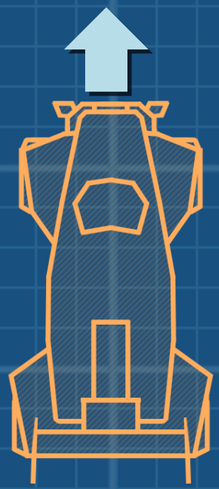
Compare server physics to recorded history

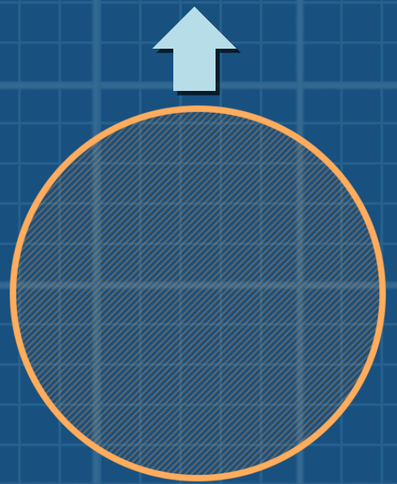
1





Server

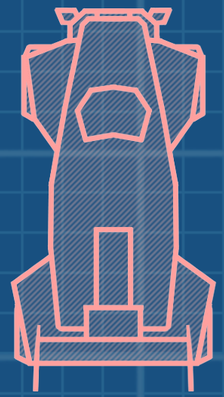




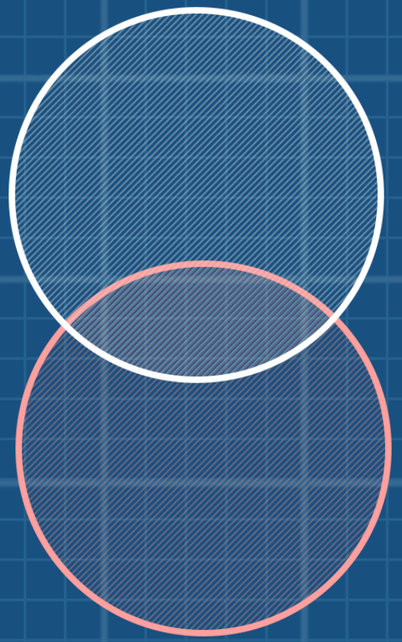
Client

Compare server physics to recorded history

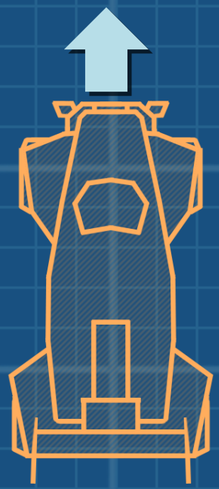
1

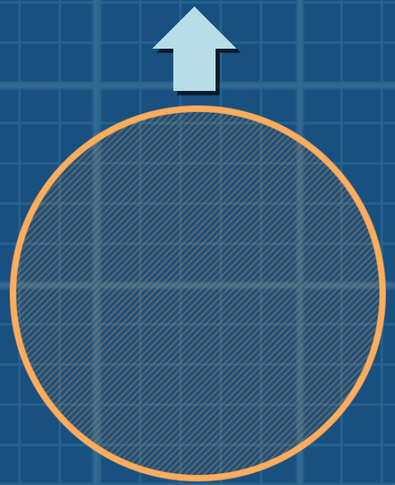


1



Server



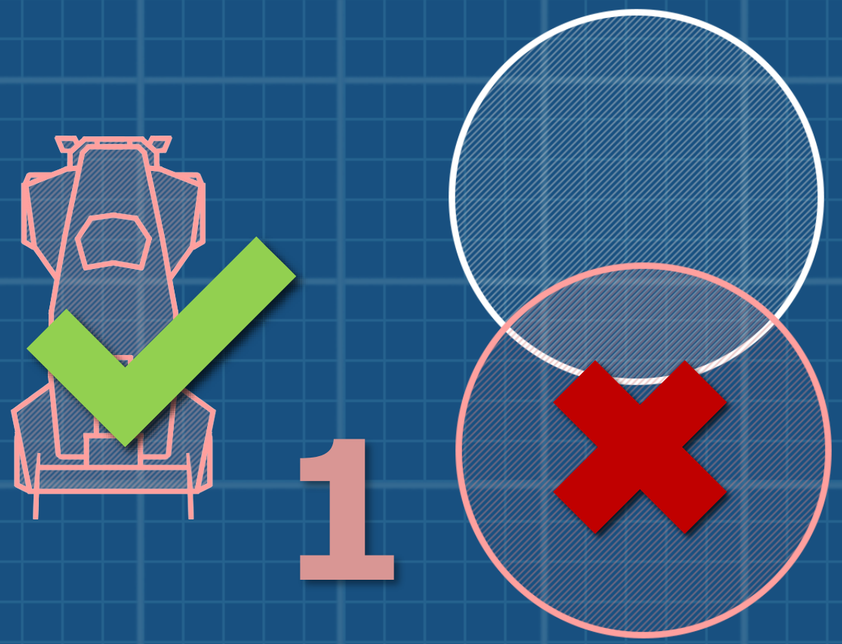


Client

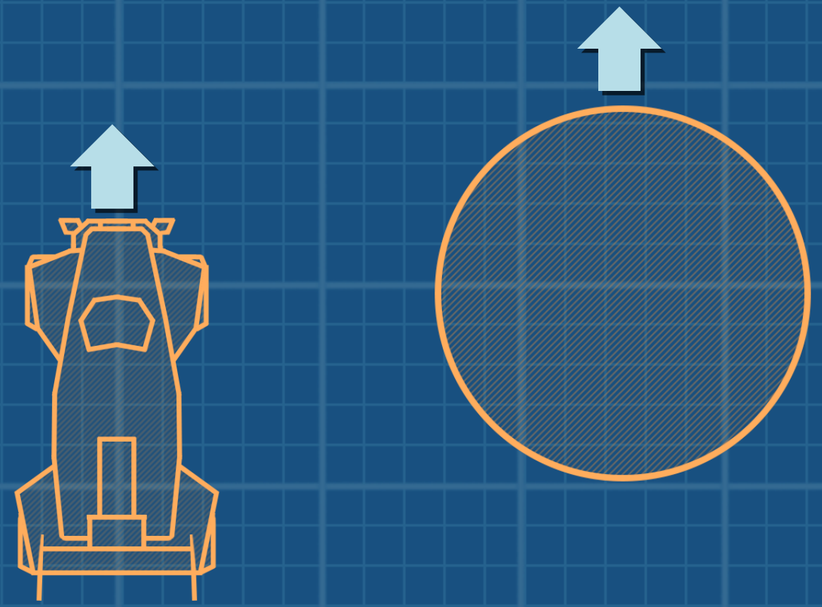
Server

Large difference requires correction

1

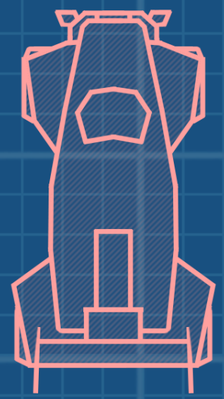


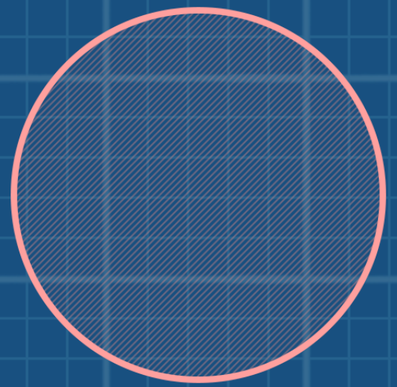
1



Client

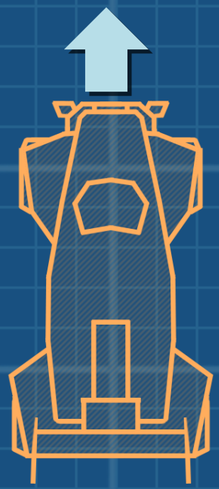
Update history to new data

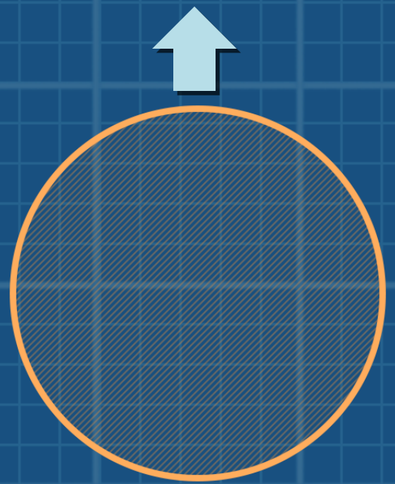




1

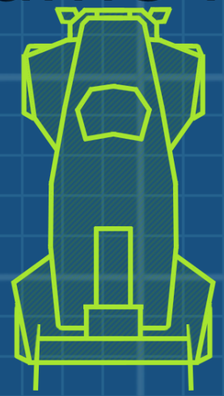
Server



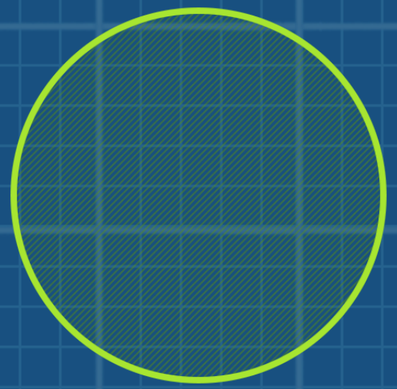


Client

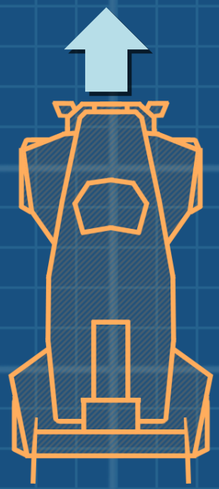
Revert all physics actors to that frame in history

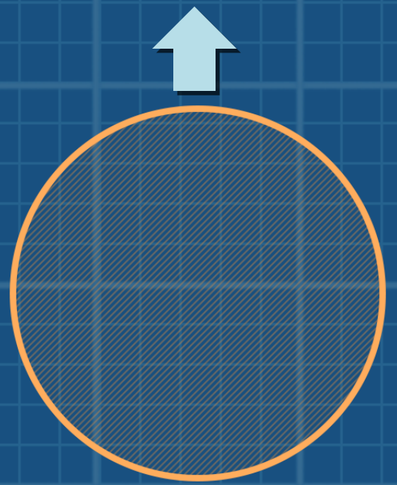


4



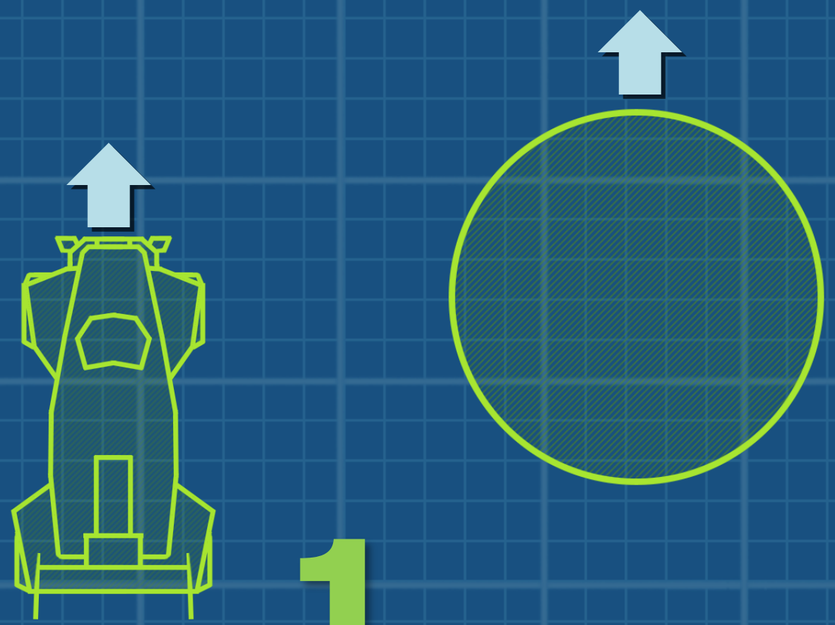
Server





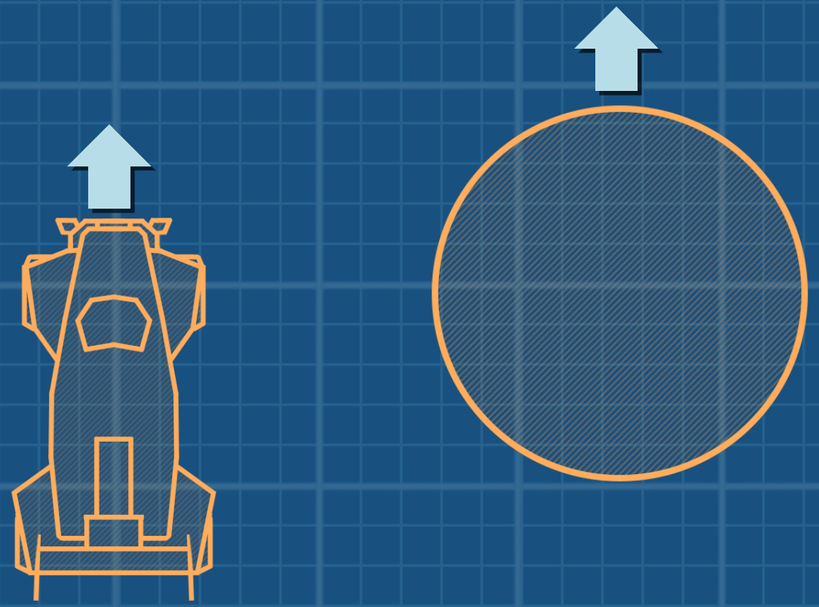
Client

Revert all physics actors to that frame in history



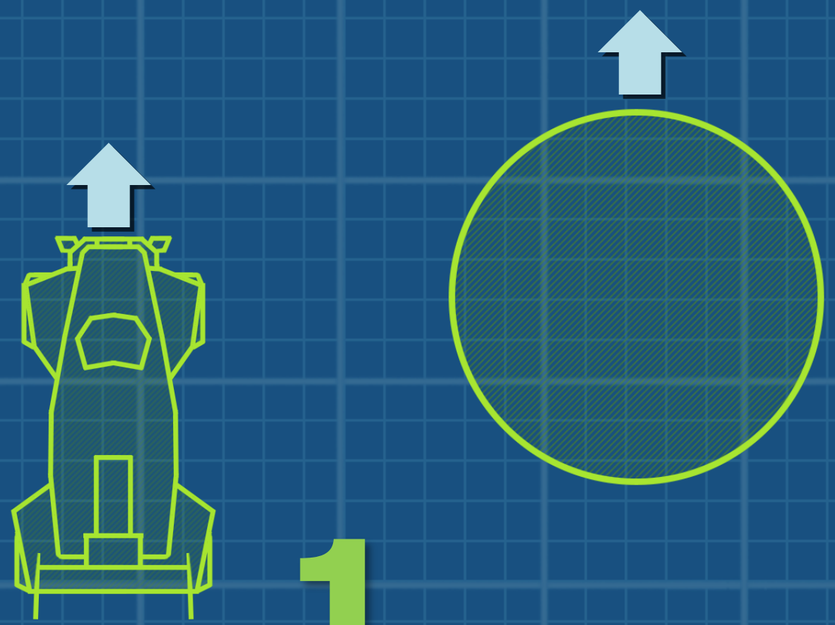
1

Server



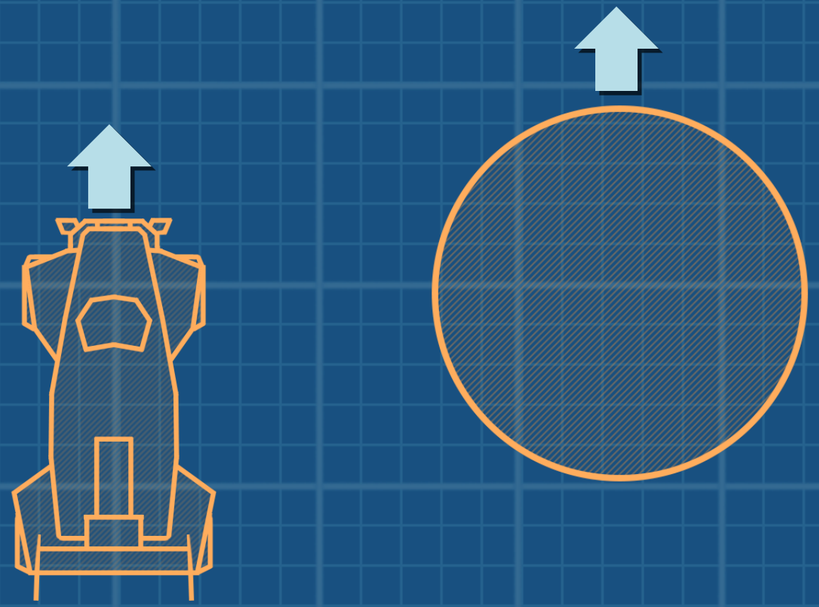
Client

Run multiple physics frames to catch up



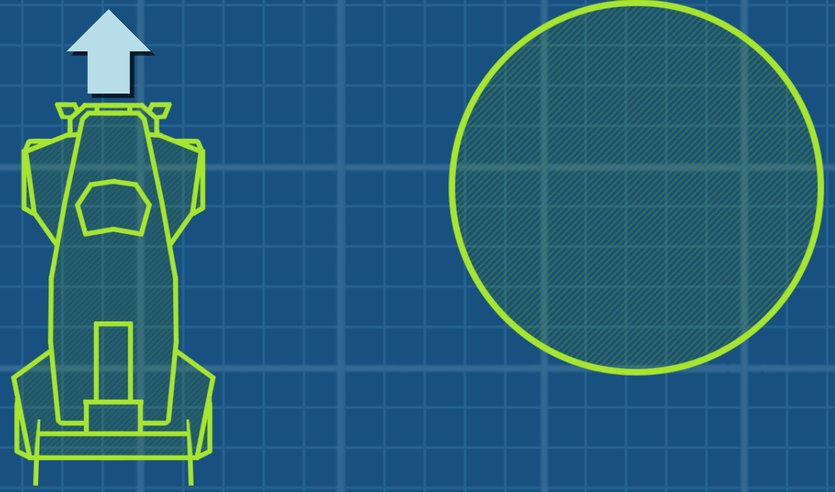
1

Server



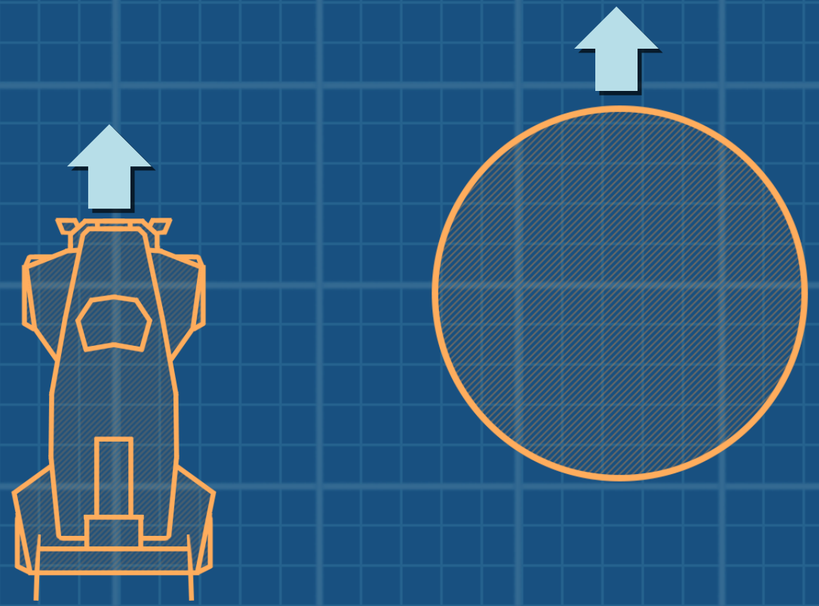
Client

Run multiple physics frames to catch up



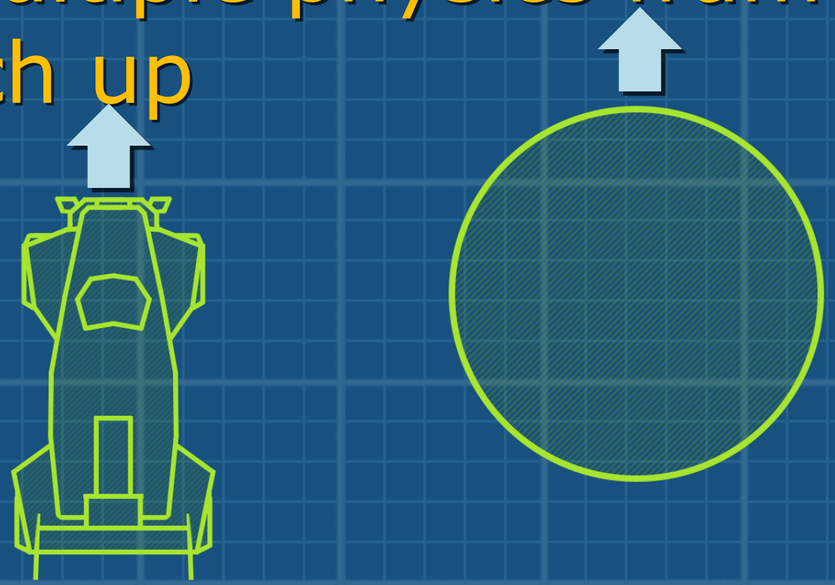
2

Server



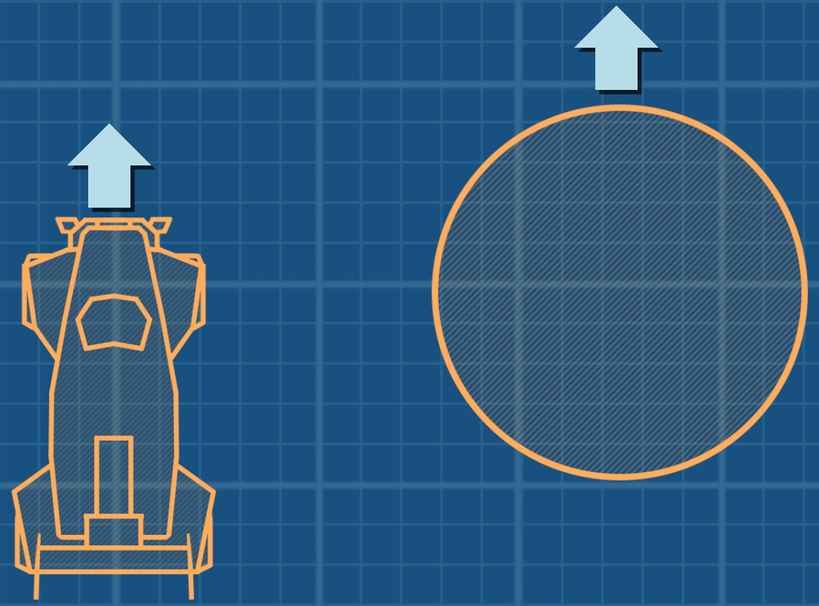
Client

Run multiple physics frames to catch up

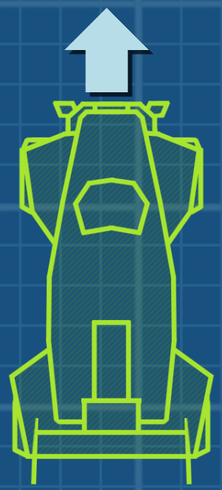


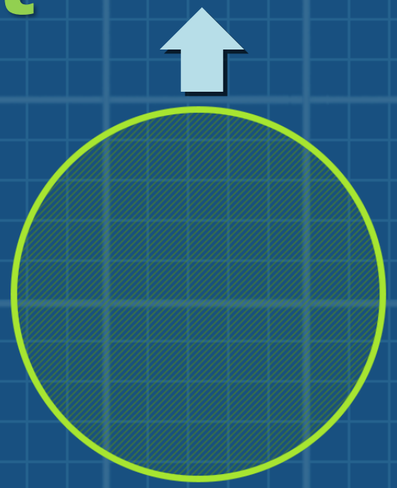
3

Server

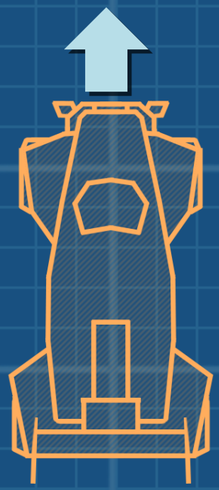


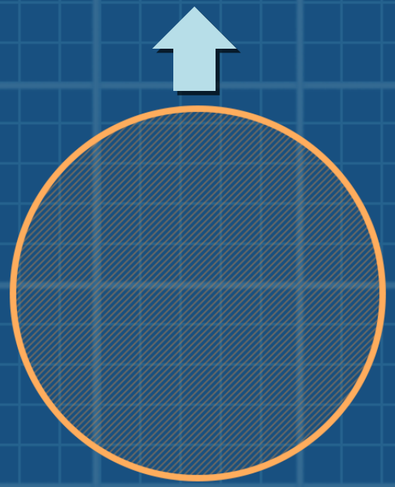
Client





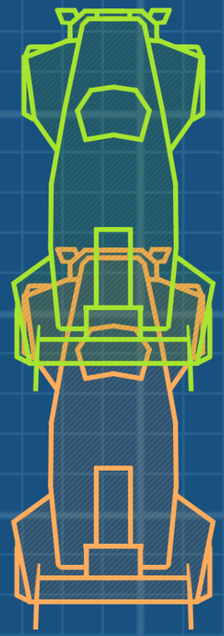
Server

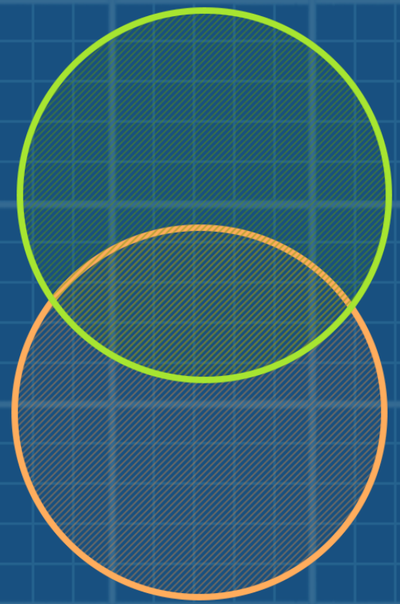




4

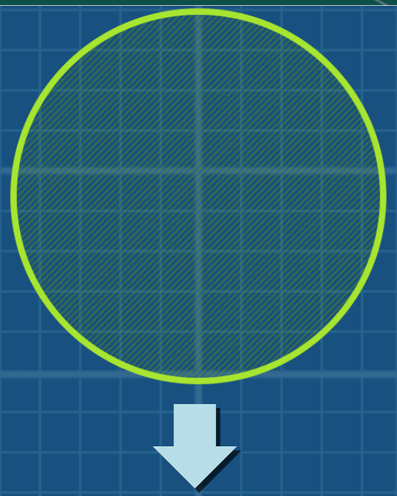
Client
Server

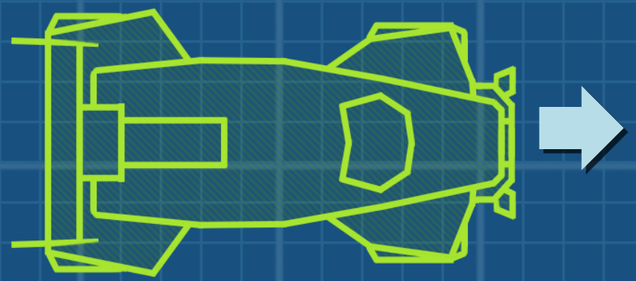




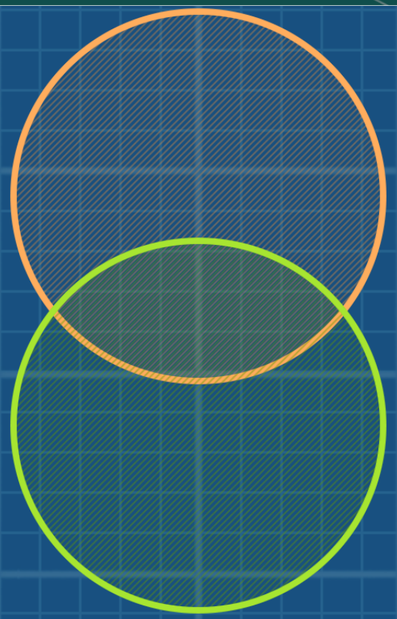
Hitting the Ball

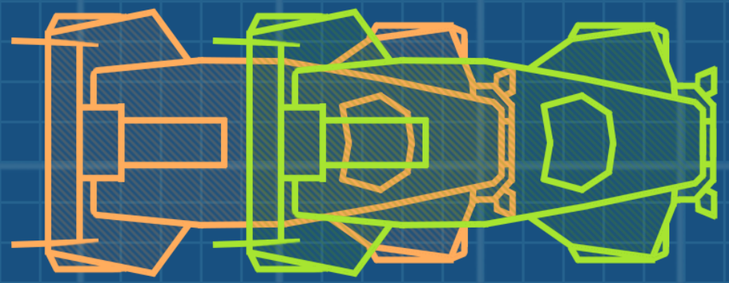
Client



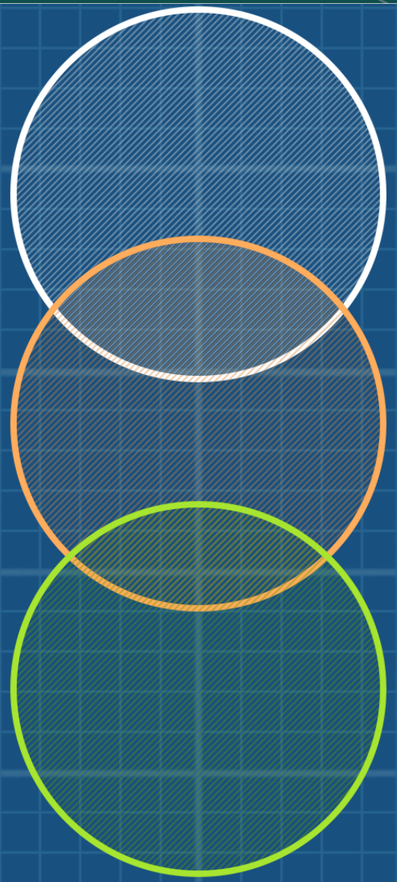


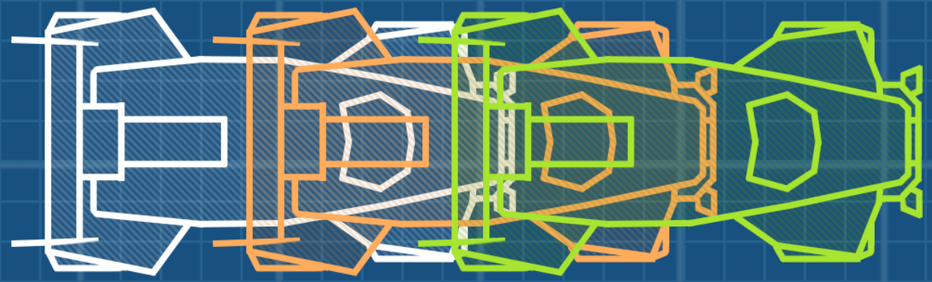
Client
Server



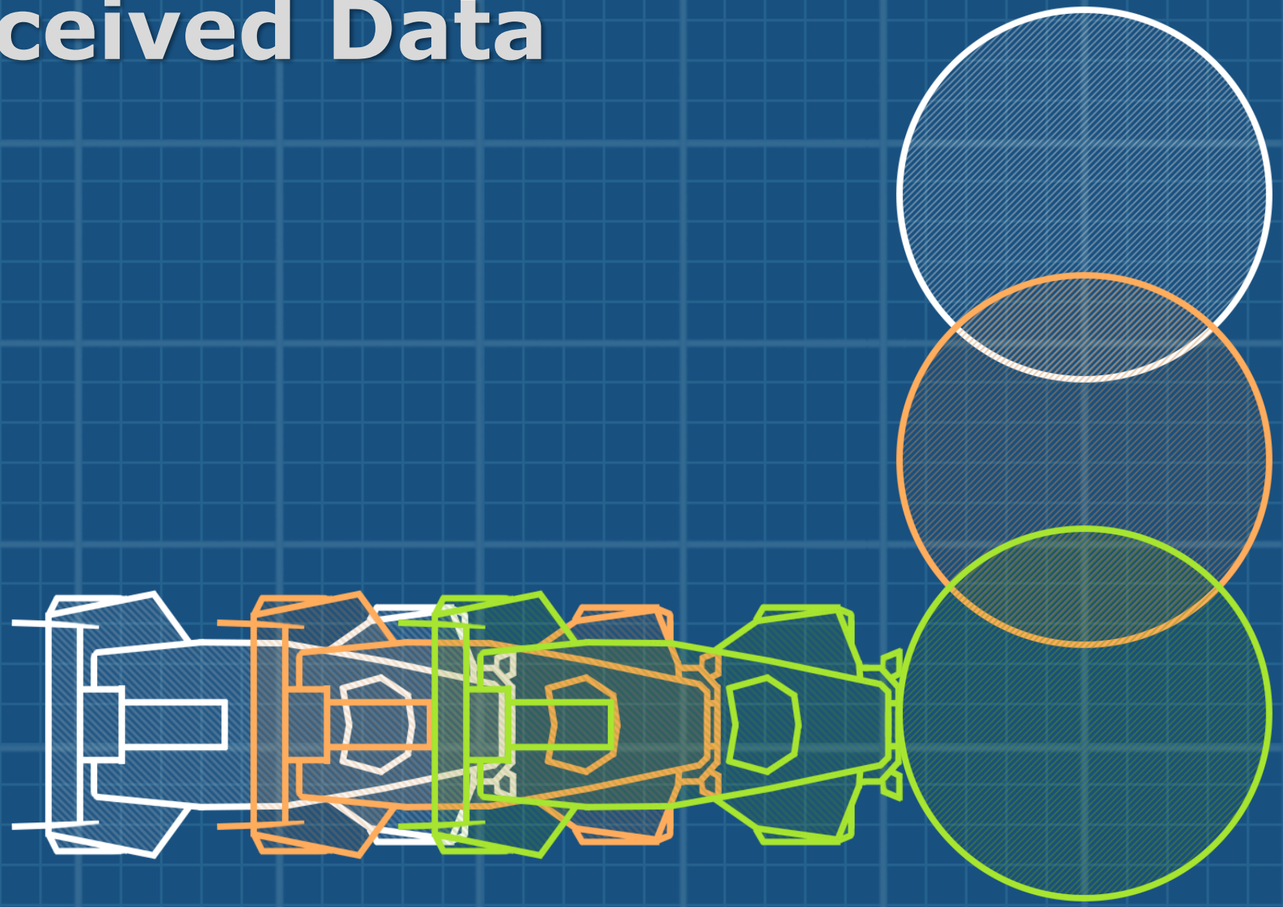


Client
Server
Received Data

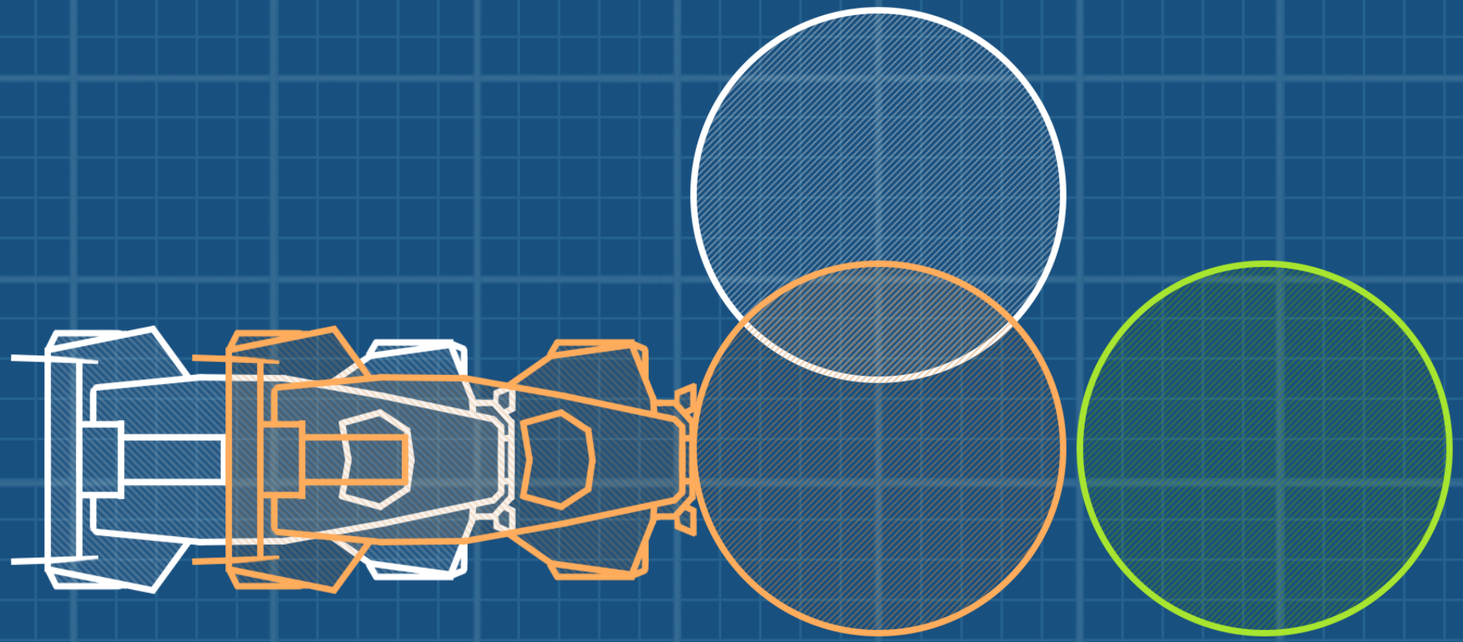




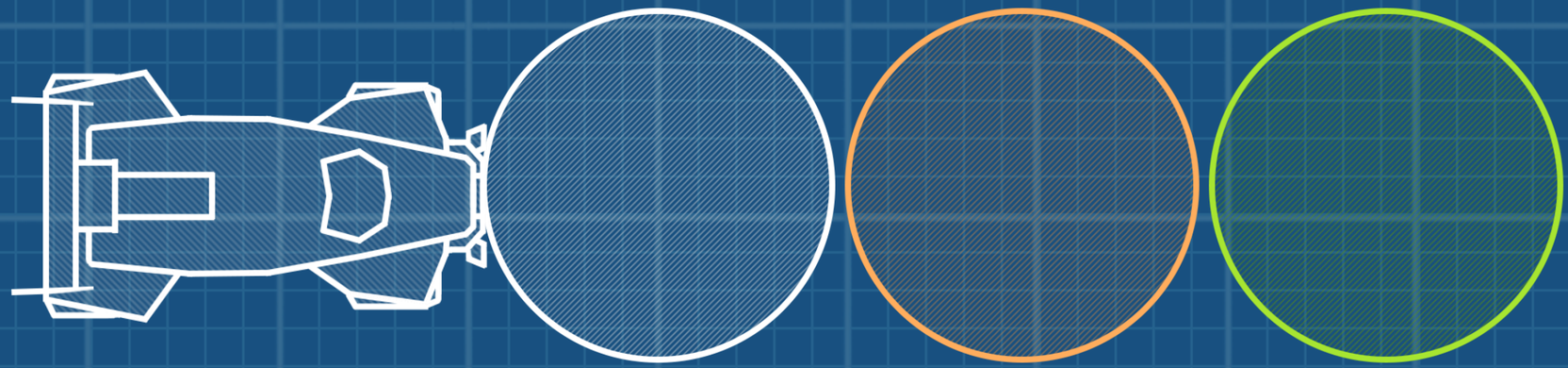
Client
Server
Received Data



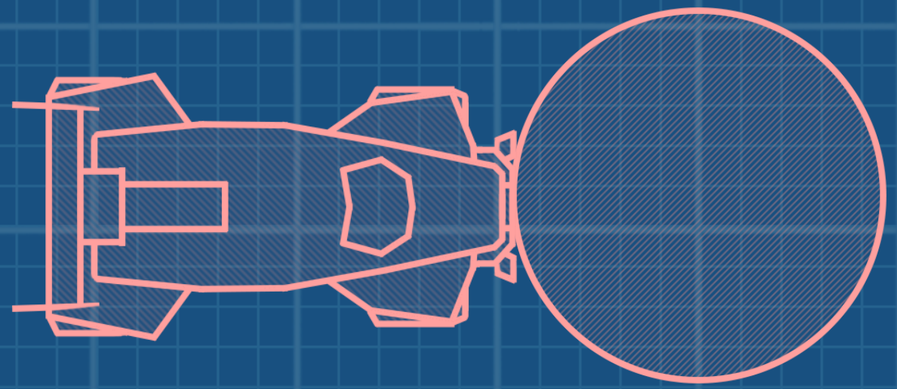
Client
Server
Received Data



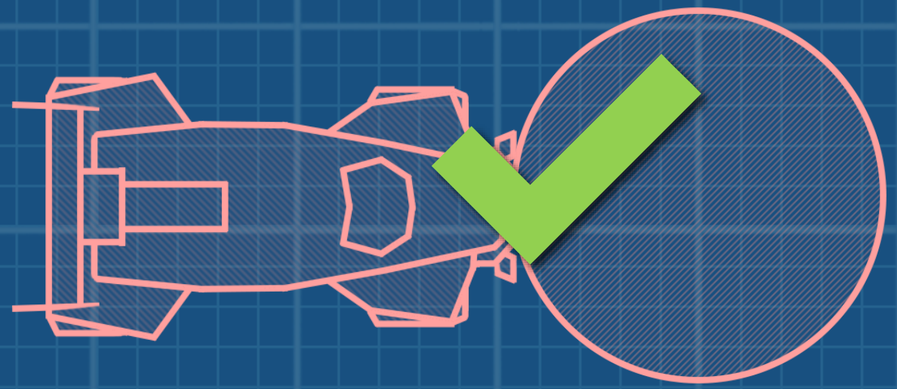
Client
Server
Received Data



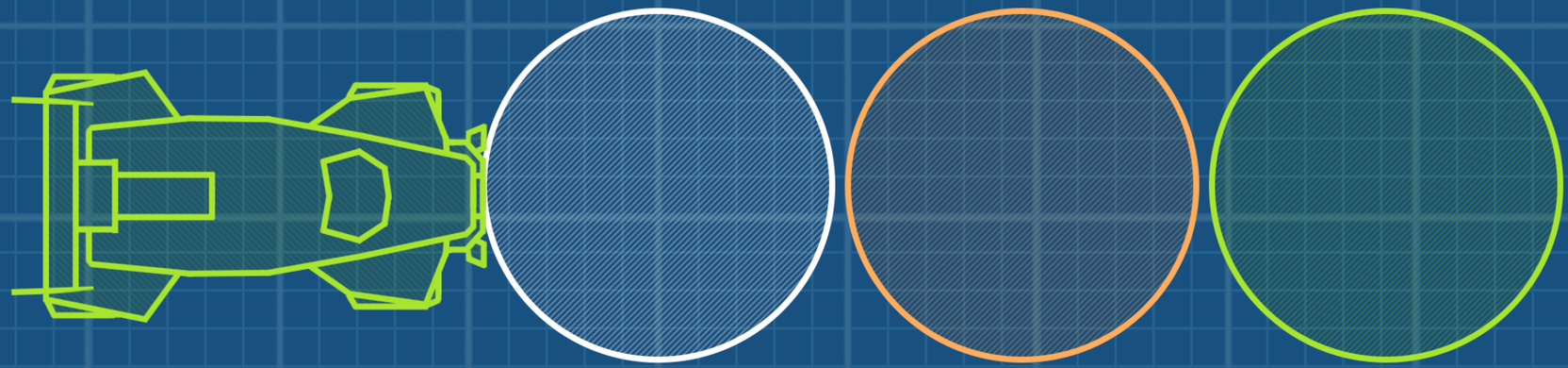
Client
Server
Received Data
History



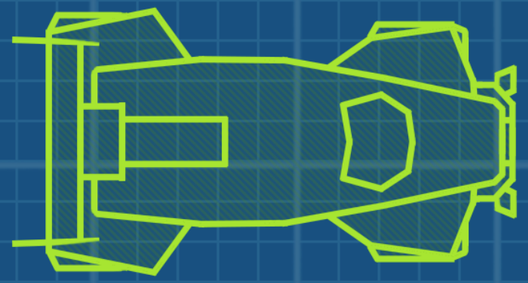
Client
Server
Received Data
History

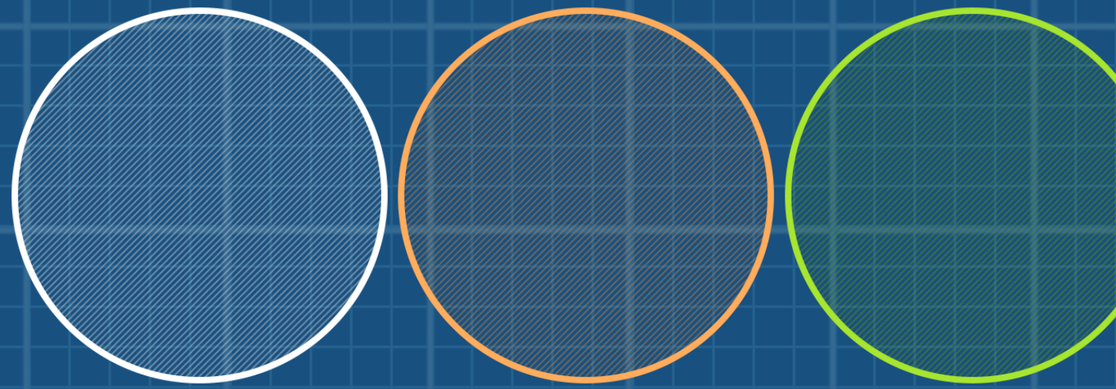


Client
Server
Received Data



Client
Server
Received Data





Predict Everything

- Client drives to where the ball is going to be
- Works well with ball (predictable)
- Not as well with cars (unpredictable)
- No server-side lag compensation.
- Expensive corrections
  - 200ms ping, 120hz = 24 correction frames

Networking Results

- No input delay
- High-ping clients don’t ruin game
- Reliably hit moving objects
- 100% server authority

SUMMARY

Summary

- Reducing complexity can help you reach your design goals
- You do have a choice of physics engine
- Networking is still hard
- Fixed tick rate and client input buffers FTW

Resources

- Fix Your Timestep GafferOnGames.com
- ‘Overwatch’ Gameplay Architecture and Netcode GDCVault.com
- Client-Side Prediction and Server Reconciliation gabrielgambetta.com
- Client Side Prediction and Server Reconciliation gamasutra.com
- Rocket Science youtube.com

Thank you!