

BATTLE CARS

2020 EDITION

Pre-production - December 2019



Summary



Pre-production key art

Overview

Page 4

Gameplay

Page 10

Art Direction

Page 20

Audio

Page 27

Technical Orientation

Page 30

Organization

Page 34

Conclusion

Page 40

OVERVIEW



Meet the team!



Core team, from left to right

Anthony Rabaux

Lead programmer
Network & AI

Florian Eschaliér

Level Designer
Technical Designer

Louis Bayard

Level Designer
Sound Designer

Dylan Fitzpatrick

Vision owner
3C Designer & Programmer

Léa Lescuyer

Game Artist
Environment & Concept

Matthias Johan

Producer
Game Designer

Léa Galinha

Art Director
Vehicles, FX & UI

We are targeting car game enthusiasts fond of 2000s arcade franchises, such as Need For Speed and Burnout, looking for a new experience!



Genre
 Vehicular combat Battle Royale

64

Number of players
 Up to 64!



Platform
 PC & Consoles



Controller
 Gamepad



Game duration
 5 minutes

« Battle Cars is a **Battle Royale car game** where 64 players ruthlessly fight to the death in an abandoned city. »

A new experience...



Pedal to the Metal

Gotta go fast. Always go full speed ahead, even at the edge of death.



Explosive Brutality

It's always about how hard you can hit'em. The harder the better.



NO Fair Play

Forget about manners, you just need to crush your opponents.

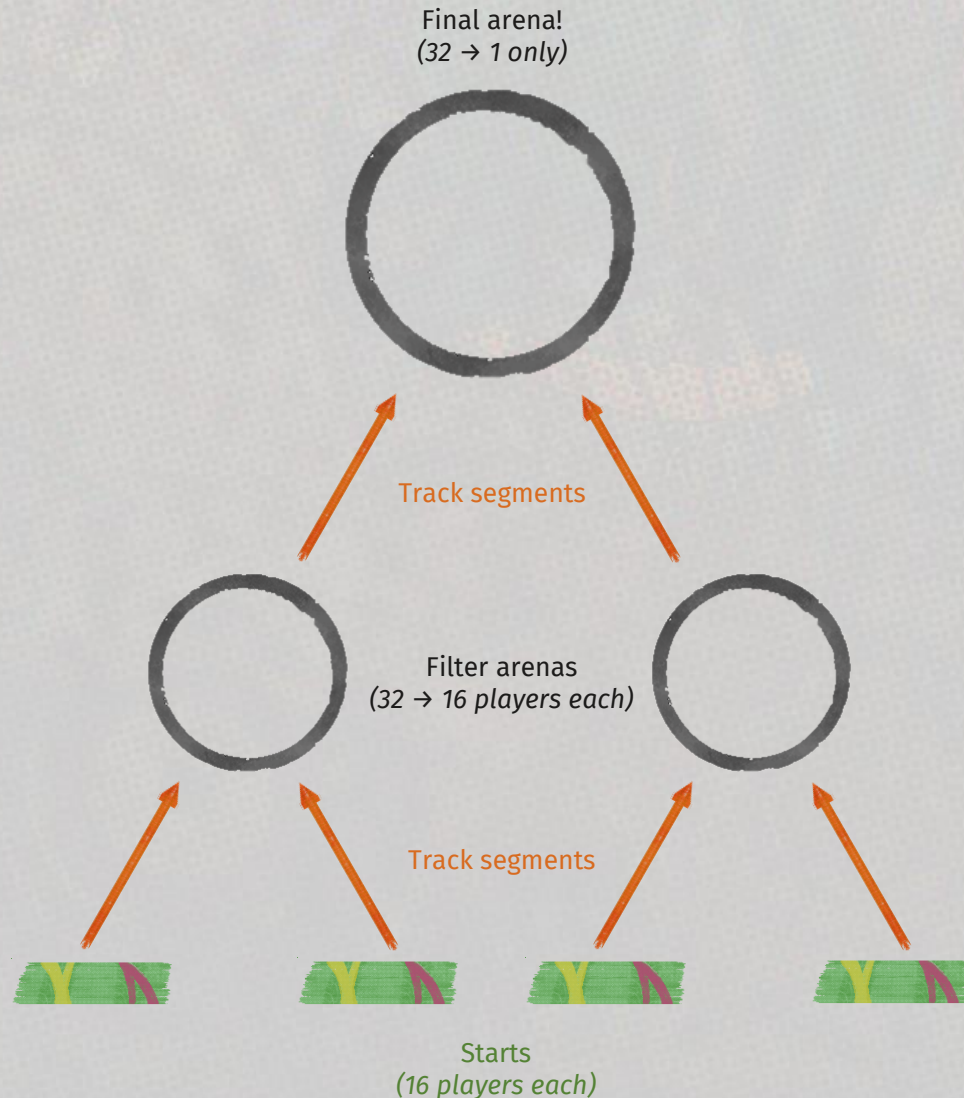


To show off

Destroying feels good. Destroying with style feels amazing.

Our core fantasy is pretty simple:
« *Be more violent and stylish than others.* »

...through a Battle Royale



Game Mode

The game starts with 64 cars. **The goal is to be the last one riding** (one life only). A leaderboard shows the brutality score, which are damage dealt by every player.

Filter arenas

Focus on collisions. **The exit gates won't open unless they are 16 players remaining.** Once the first filter arena exit is opened, here is what happens in the second filter arena: The player with the lowest brutality score is destroyed every 10 seconds, until the total number of car is 16.

Segments

Alternating with arenas, segments focus on high speed challenges. The first car reaching the next arena gets an upgrade. On the other hand, a deadly **shrinking zone** chases all players.

Which Battle Royale key rules are adapted in Battle Cars?

1. The concept of "Last Man Standing" → "Last Car Riding"
2. Players all start from scratch and upgrade their gear by fighting.
3. The well-known "Shrinking Zone".

GAMEPLAY



3C Intentions

Nervous

Very responsive, making players feel like they're not pressing the acceleration triggers hard enough!

Spectacular

There is always something going on screen. Players are left on the edge of their seat.

Intuitive

Thanks to adapted and effective controls, the car is just like an extension of the players' hands.

Burnout Paradise's controller is a great controller reference for Battle Cars, in terms of nervousness and reactivity. For instance, reaching full speed usually takes less than 3 seconds!



Current 3C prototype

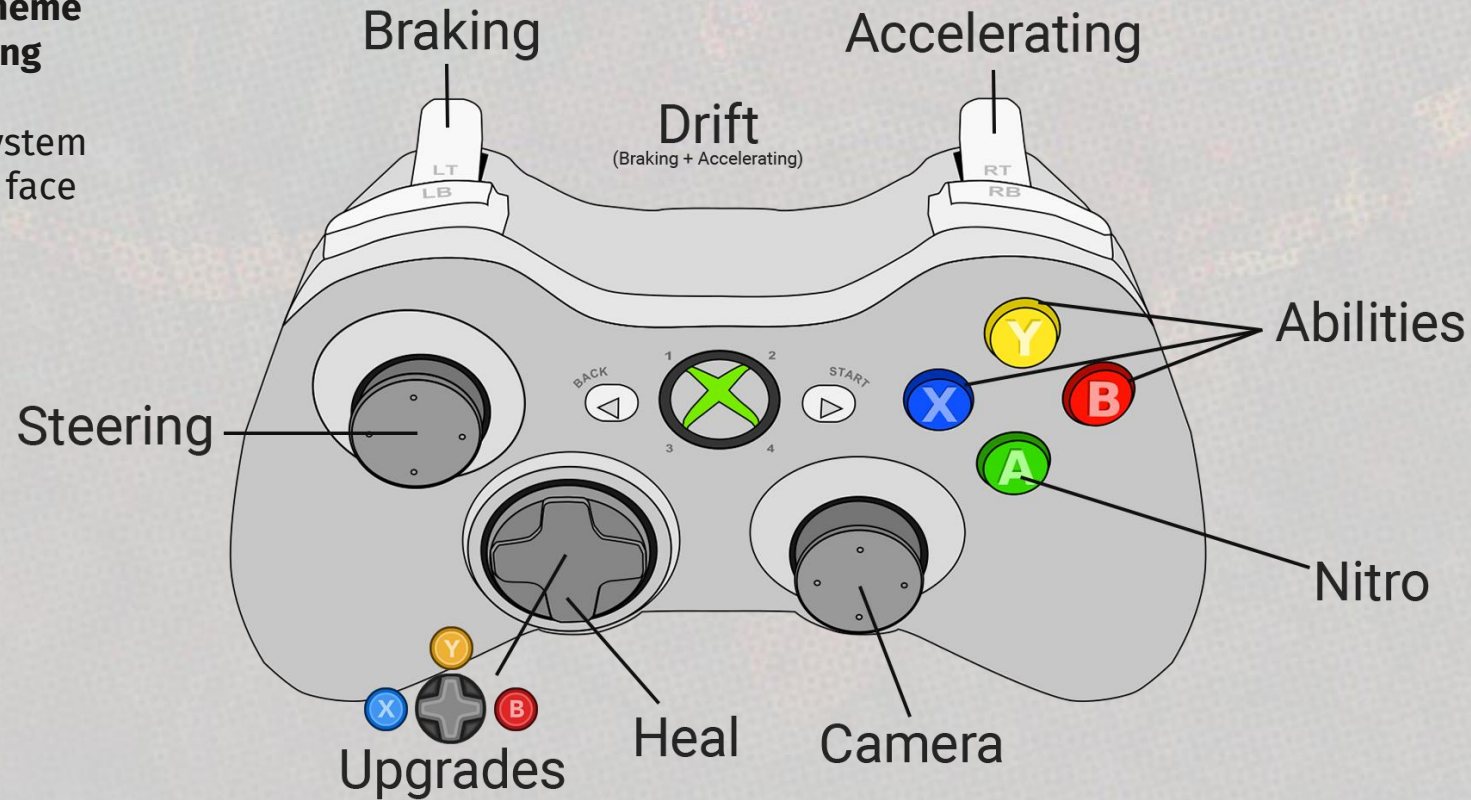


Rocket League's camera is amazing at allowing players to gather information around them. The camera can easily be rotated without disturbing the game's focus.

Controls

We designed our control scheme based on conventional driving inputs.

We also added our ability system with similar patterns on the face buttons and the D-Pad.



We discourage players from braking by forcing them to release the right trigger to slow down. On the other hand, **drifting is heavily rewarded**.

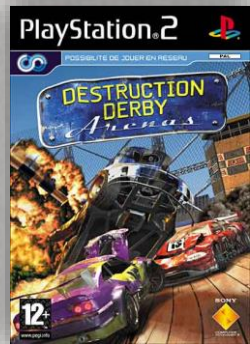
Composition



Collisions

“No gentle push here. Go full speed and aim for the weak points.”

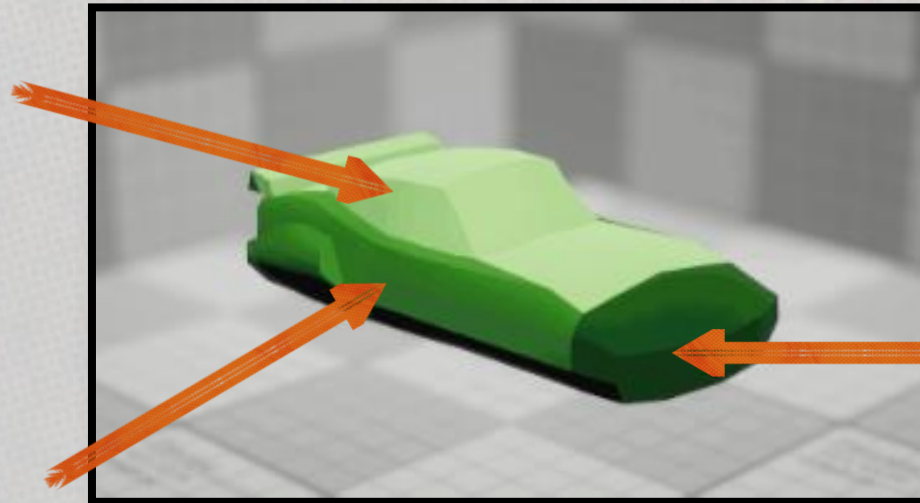
Collisions are the core mechanic of the game. They are at the center of the player's progression as they allow players to upgrade abilities.



Destruction Derby Arenas,
Studio 33 , 2004

Car colliders, showing different spare parts with different strengths.

Weak part
Strength: 0,4



Solid part
Strength: 3,2

Regular part
Strength: 1,8

Power = **Part strength** x **Speed**

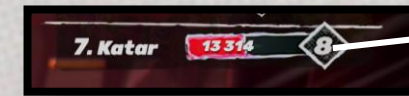
The car with the more power deals damage to the weaker one.

Abilities



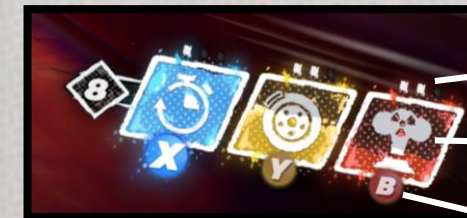
Abilities are directly mapped on X, Y, and B through the customization menu.

Abilities are game changers, either through damage or repositioning.



Brutality level

Players gain brutality points by dealing damages through collisions or abilities. Each level requires a specific amount of brutality points, but grants an ability upgrade.



Ability level

Cooldown

Mapped input



In OnRush, one ability example is a boost dealing damage to nearby enemies.

Tricks

“You think you have enough skill to survive and show off during Battle Break? That’s perfect.”

Players can express their style and boast about their driving skills, thanks to tricks.

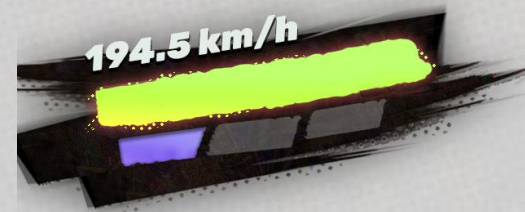


Drifting is one of the most essential tricks.

Players fill up nitro bursts by successfully landing a trick. The better the trick, the greater the amount of nitro. Bursts last for 2 seconds and grant 120% of max speed!

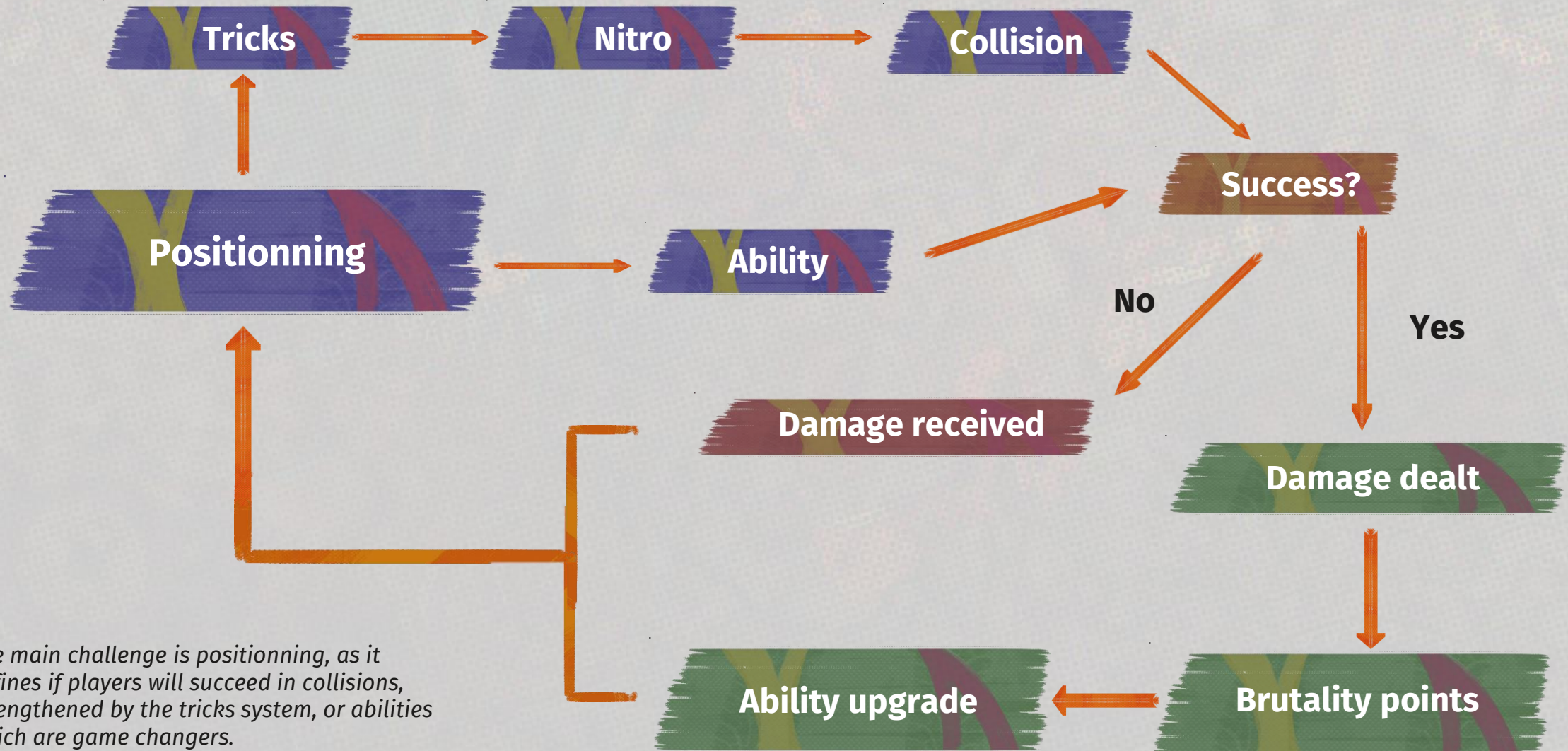
Some of our tricks:

- Drift (accelerating and braking at the same time)
- Air time (as long as the car is in the air)
- Near miss (every time the car dodges an obstacle)
- Donuts (making a 360° turn while drifting)
- Barrel rolls (spinning during air time)



Players can also trade one nitro burst for additional health.

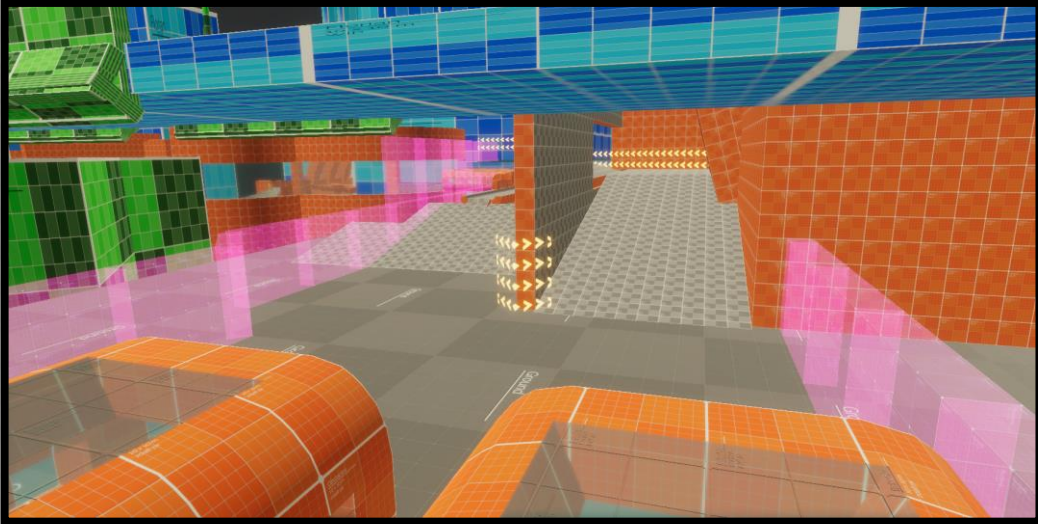
Core loop



The main challenge is positioning, as it defines if players will succeed in collisions, strengthened by the tricks system, or abilities which are game changers.

Level Design intentions

Segments



- Build up nitro
- Multiple layers
- Quick
- Aggressive
- Less lethality



Skate parks inspired us to create ramps to gain speed and credible boundaries.



Arenas



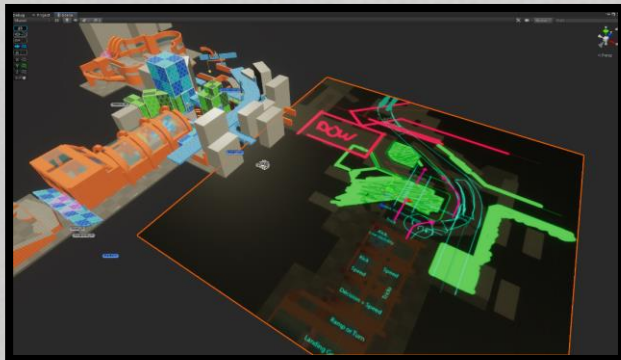
- Closed space
- No direction
- Enough room for speed
- Enforce collisions
- Gather information from other players



Level Design pipeline

1. Rough Gameplay Layout

2D & 3D Layout to establish the intentions



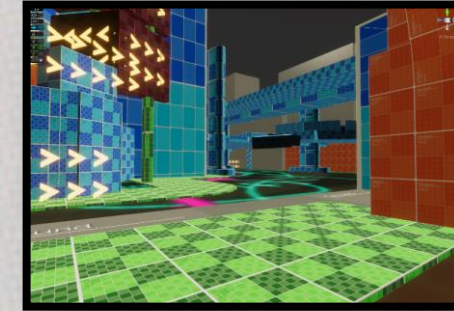
2. References gathering



Gathering and taking inspiration from architecture, games, sports on reference boards



3. Layout Phase



Layout phase using existing generic draft models and making placeholders for custom ones, listed in an asset list.

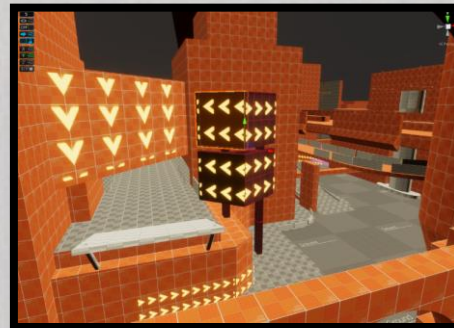


4. Gameplay & Rhythm

After the first initial Layout, we often make a photoshop pass, to visualize important views and plan the rythm and player flow in the level.

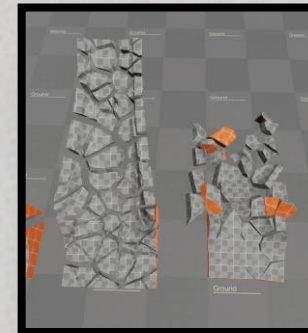
5. Clean World Building

Usage of instanced kits for maintenance. Creating new kits as needed in 3dsMax with custom scripts to spend less time mapping and exporting these assets. Metrics pass...



6. Destruction Phase

Quickly create our destructible physics assets directly in-engine using our custom made Houdini Tool.



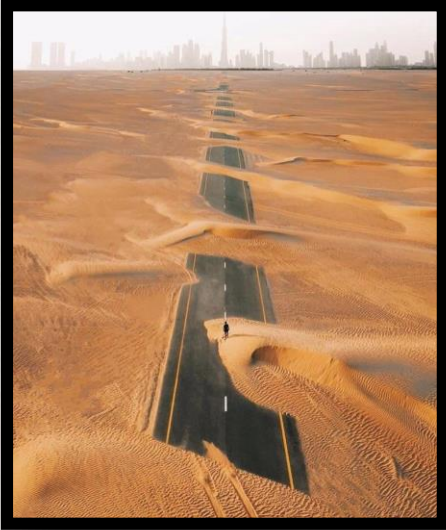
7. Testing!



Publish in the modifications for testing and iterations. Quickly available to the team in the next build.

ART





Abandoned city

Battle Cars is set in a **city** located on cheap land. The city was a brand new project **in construction** funded by private investors. The unique architecture is a mix of both modern **Western buildings and Brutalism**. Unfortunately, investors didn't have enough funds to finish the project, causing the **city to be fully abandoned**.

Underground competition

Ten years after, the city was granted a second heartbeat when racers from all parts of the world organized **extreme underground races** in this deserted playground. “**Battle Cars**”, an event mixing the Battle Royale concept with racing was born. Because the climatic conditions were so warm and dry inside the city, racers only squatted buildings for a couple of weeks per season, known as the “Battle Break”. It was during this time that a seasonally Battle Cars was set.



Streaming

The event started gaining popularity **online**, the goal being to make each season more remarkable than the previous one with major circuit changes in both racing tracks and arenas, always taking advantage of the original city. **Screens, drones and cameras were installed** to stream the event and Battle Cars became widely known in the underground scene!



Pedal to the metal



Wipeout Omega Collection,
Creative Vault Studios, 2017

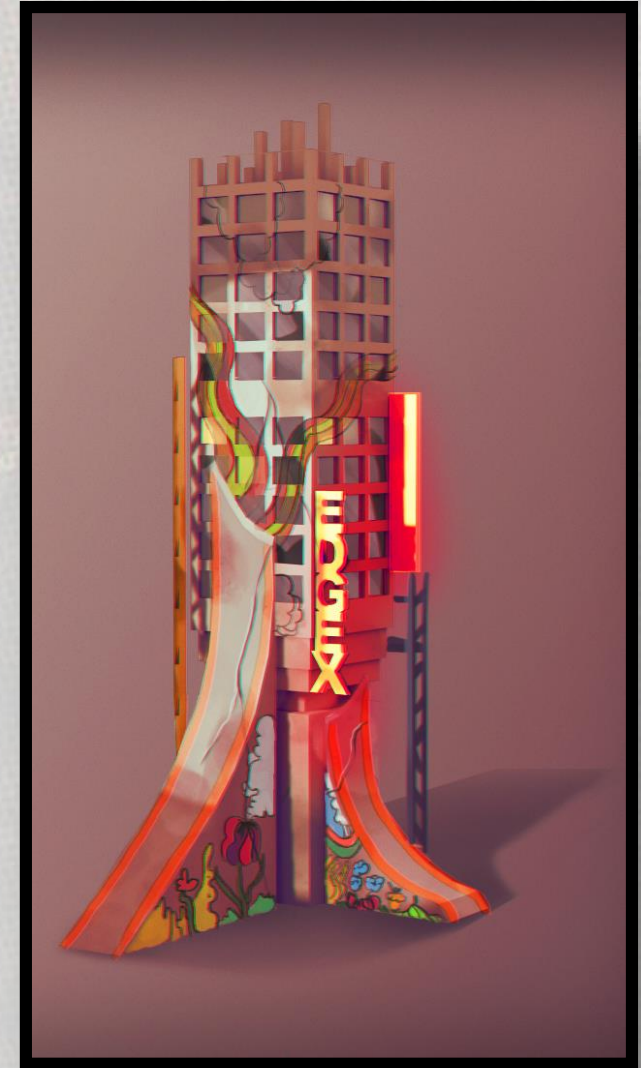
In Battle cars, we enhance the **feeling of speed** by disposing objects at **even distances** (public lights, trash cans, etc...) on the sides of the road. **Paintings on the ground** follow the track.

The camera also has a major impact on our game feel, using **motion blur** to **motivate the player to go faster**.

We also use the **car's design and trails** to give more satisfaction to players. The whole environment was also built around this idea, as we intend to make it appealing enough for players to drive in and discover the city.



Battle Cars pre-production scene

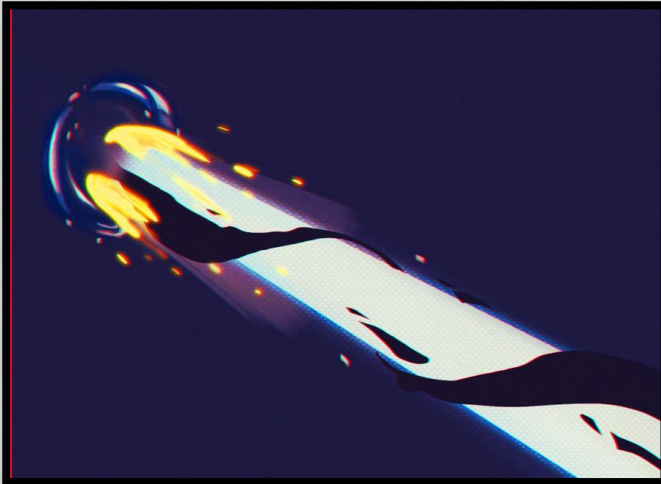


Battle Cars concept art: Brutalist building

Explosive brutality



Battle Cars concept art: Fire and Nitro FX



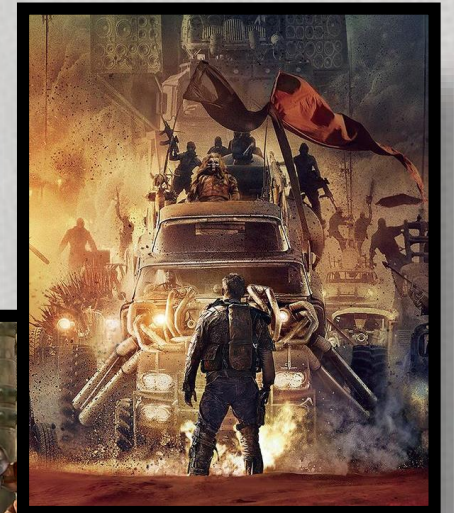
Visual Effects

Our VFX are exaggerated, just like in comic books. They display flash colors, use graphic patterns and chromatic aberrations. They also have a traditional animation rendering, with a low framerate effect.

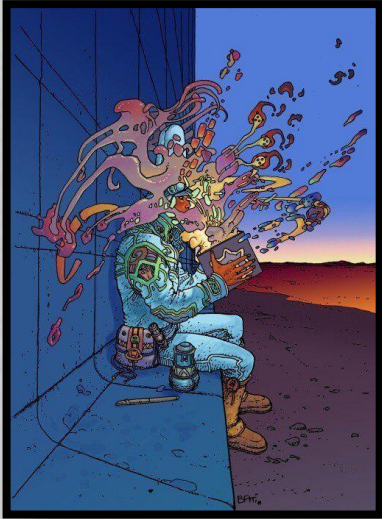
We also intend for Battle Cars to be brutal and explosive. We use specific design and colors: Battle Cars is **rusty** and **dirty**. It is a **huge mess** composed of **savage silhouettes**.

It is also a game with a lot of visual feedback, thanks to its **colored VFX and shaders**.

There is clearly a comedy vibe in the game, just like in **gladiator fights** during Antiquity. The camera emphasizes on **brutal actions**: screen shake, slow motion.



No fair play



Battle Cars has an **anarchist** and **vandalistic** identity: racers use **tags and destroyed objects** to mark their territories, resulting in a **huge mess**.

The intention is to remind that players are **alone** against everything (and everyone else). It is a very **hostile** environment.

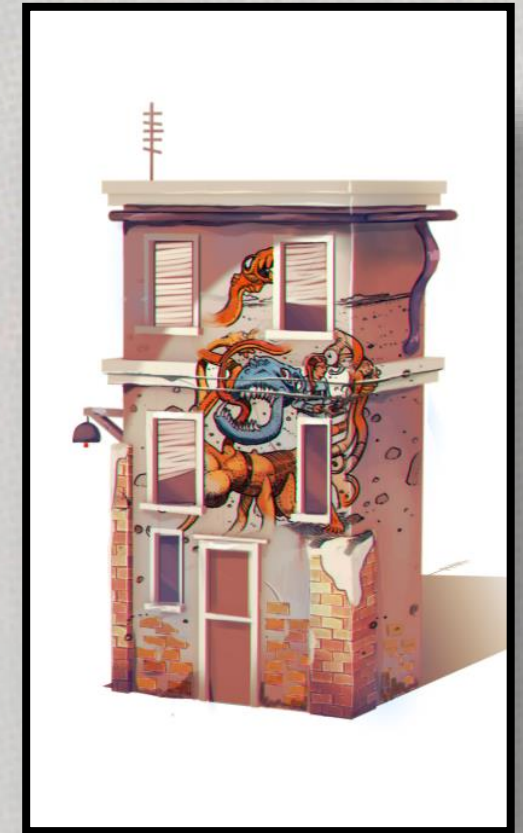
Racers also have nothing to lose: they are here to show-off. The city is **fragile, breakable, almost living**. Fires start out of nowhere, buildings are falling into pieces. It is **unstable**.

Street Art

Our street art style is composed of big and clear illustrations, giving each building a unique and coherent feel. Directly inspired by Moebius' drawings, it is quite colorful and crazy, guiding players throughout the race.



Graffiti sketches used as decals on buildings



Battle Cars concept art: abandoned building and street art

To show off



Battle Cars is a game where players have to **show off**. Player's abilities **are huge and impressive**. Level design elements are visually enhanced by **explosions and colorful lights**.

Players also have the possibility to give their car an exaggerated personality: there are a lot of **eccentric customizations**.

Brutalism helps us giving the whole environment a **purposely overdone mood**. It is very **colorful**, even if the environment has to follow a strict color palette in order to stand out from the players' cars.



Battle cars pre-production scene



Brutalism

Brutalism is an architectural style using solid blocs and geometric shapes. These huge concrete buildings are smooth enough to give players the desire to drive on them. They also have a unique aspect, giving them a "show-off" mood.

Vehicle design



Our Battle Car was designed for battle. It has many **armored pieces**. Its **broken shapes** give it a crafted aspect, just like if it was **completely made by the player for the player**.

The intention is not to make it perfect and clean, as the player needs to sacrifice some spare parts without any remorse.



Battle car 3D model

The spare parts on the car leave enough room for customization.

Graffiti are also used to enhance the street and underground vibe.



Battle car concept art



Its **sharp edges** and **curved silhouette** make the car optimized for racing and **high-speed collisions**.

AUDIO



Sound Design



Intentions

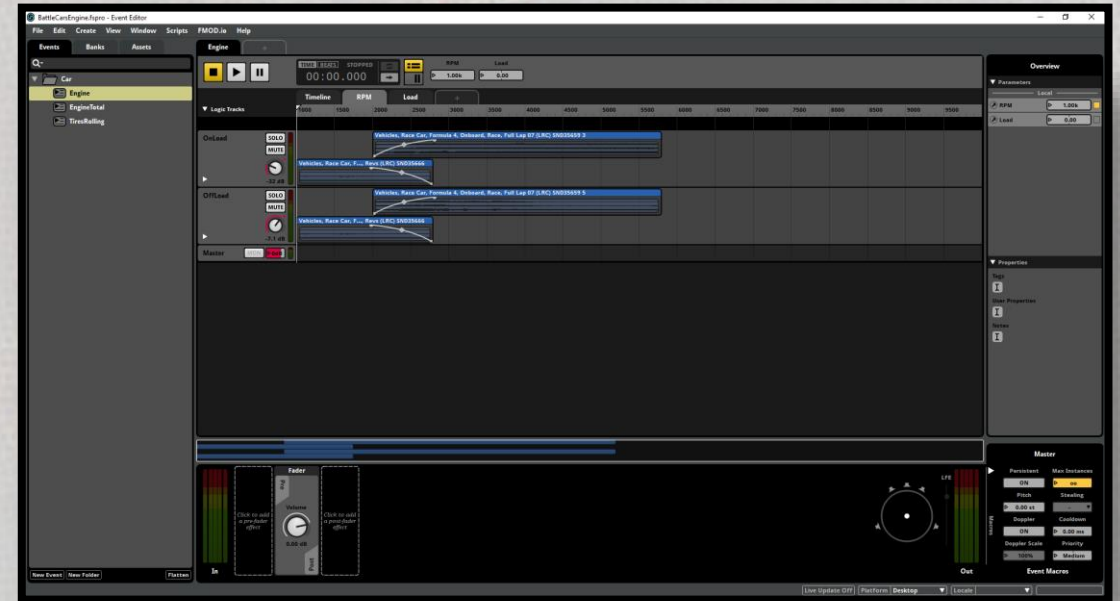
In Battle Cars, our sound is hyper-realistic, in order to give credibility to the impacts between cars. We also want to have some intersections between music and sound to keep the flow and pushing players forward.

For example, there is high pass filter on the music while players are in the air to strengthen the sensation of flight.

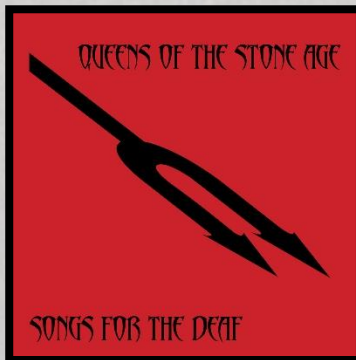
FMOD & car engines

FMOD is our sound engine, which we can integrate to Unity. It comes with built-in parameters for vehicles sound.

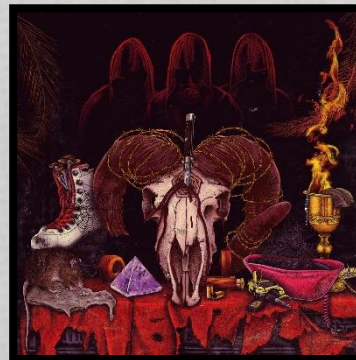
For example, the car acceleration sound is linked to the revolution per minutes (RPM).



Our intention with music is to make it as overexcited as our game is violent.
Our main references are Punk, Rock, and Metal.



Queens of the Stone Age,
Songs for the Deaf,
2002



Carpenter Brut,
Trilogy,
2015



Shaka Ponk
The Black Pixel Ape,
2014

TECHNICAL ORIENTATION





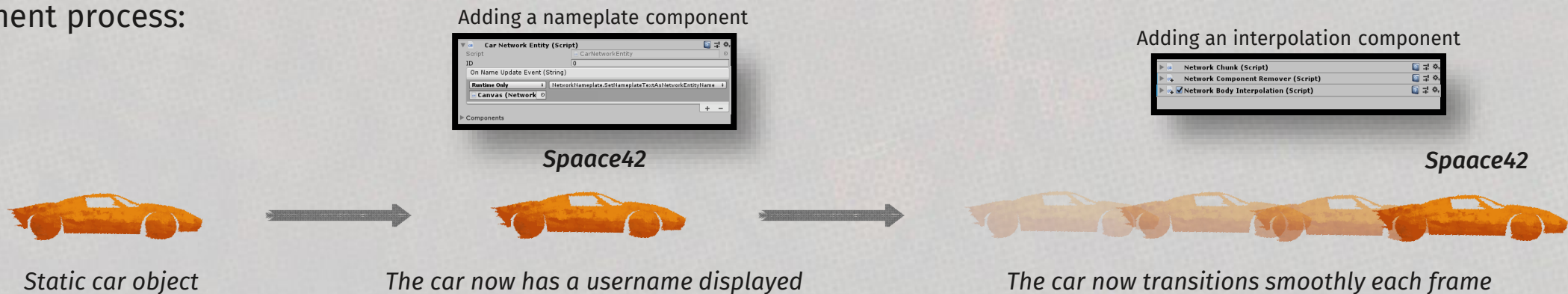
Engine: Unity3D

Since making a functional network is our biggest challenge, it is also what defined what engine Battle Cars was going to be prototyped on:

1. First of all, because our network programmer was very familiar with Unity's coding language and architecture.
2. Secondly, because it's also much faster to build a C# library with Unity3D from scratch than optimizing the default replication system from Unreal Engine 4.

We developed our own network overlay based on the Lidgren framework, allowing us to write and read packets, reusable in other Unity projects.

Component process:



The network code is modular, allowing us to add or remove features quickly, reducing regression!

Because our gameplay heavily relies on physics, we took advantage of the pre-production phase to try new technologies, and determine if we could implement them in Battle Cars' development.

ECS

The Entity Component System is an official framework, oriented on data management and optimization.

Stage	Count	Time	Perf. Factor
Before DOTS Implementation	2 000	9 ms	1x
ECS	2 000	1 ms	9x
ECS + Job System	2 000	0.2 ms	35x
ECS + Jobs + Burst Compiler	20 000	0.04 ms	2250x

Perf. In a third-person zombie shooter

Pros:

Network performance gains, modular architecture, easier tool possibilities.

Cons:

Still in beta, entity variable not tweakable on runtime, and missing documentation.

Bullet

Bullet is a physics engine simulating collision detection, soft and rigid body dynamics.



Psyonix uses a custom Bullet-based engine for Rocket League's physics

Pros:

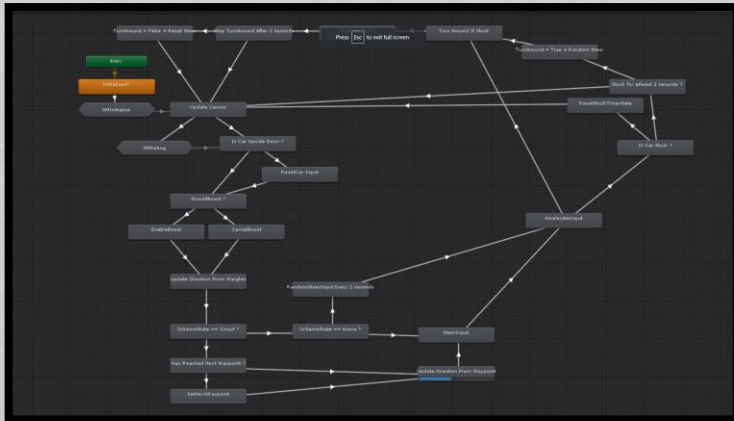
Deterministic physics

Cons:

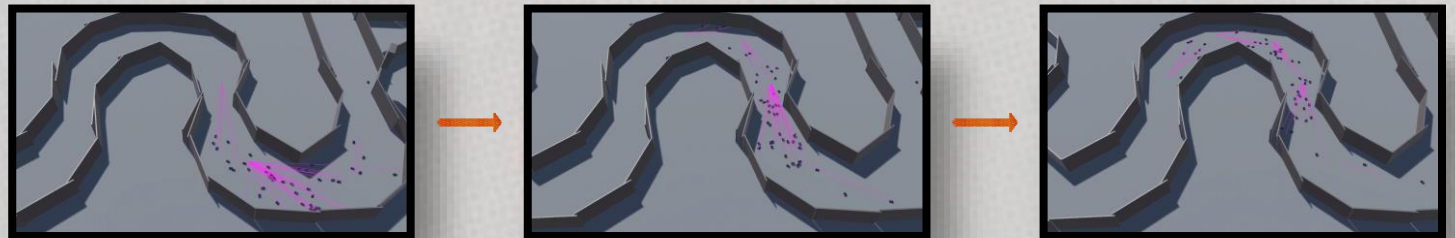
No control over physics update, and missing documentation.

Although both ECS and Bullet offer powerful advantages, **we decided to stick to traditional MonoBehavior with basic physics**, giving us perfectly acceptable results and allowing us fast controller iterations.

Even though our goal is to fill servers with only real players, it may happen that clients are missing to fill the needed 64 car slots. In that case, **Als automatically fill up the missing slots**. For example, if there are only 40 clients, 24 Als join the server.



Behavior tree made with Unity's animator.



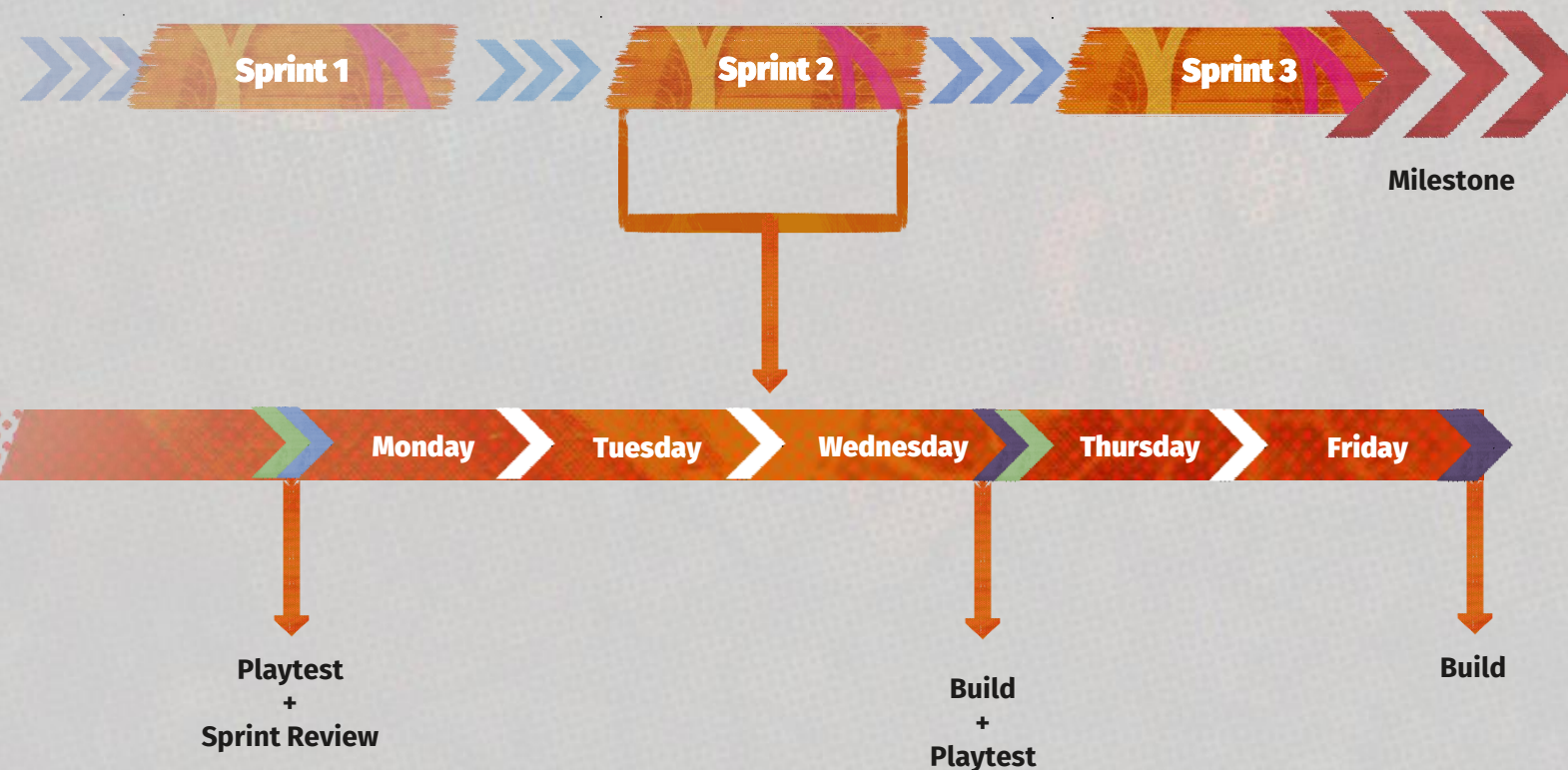
Als using a ray cast system to move through the level.

As shown above, we're currently using state machines. However, this has proven to be very limited if we want to challenge players. **We're therefore willing to develop Als using Machine Learning in the future.** The AI parameters will then be tweakable by Game Designers to simulate player strategies.

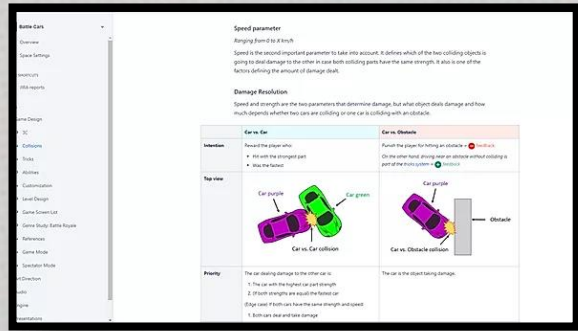
ORGANIZATION

Process framework

Our process framework is based on the SCRUM method. Sprints last 1 week in order to allow quick iterations. Processes such as stories, standup meetings, and sprint reviews are implemented in order to make the team more efficient.

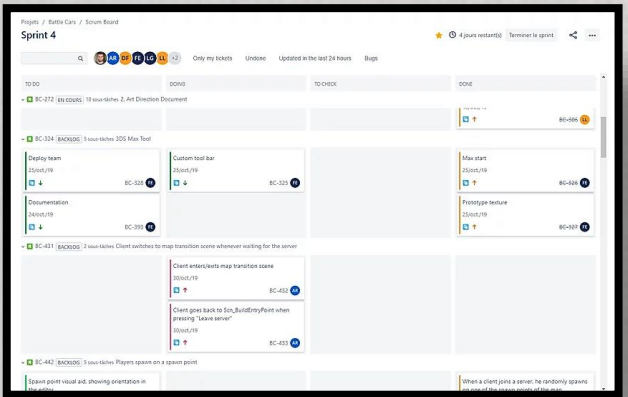


Feature pipeline

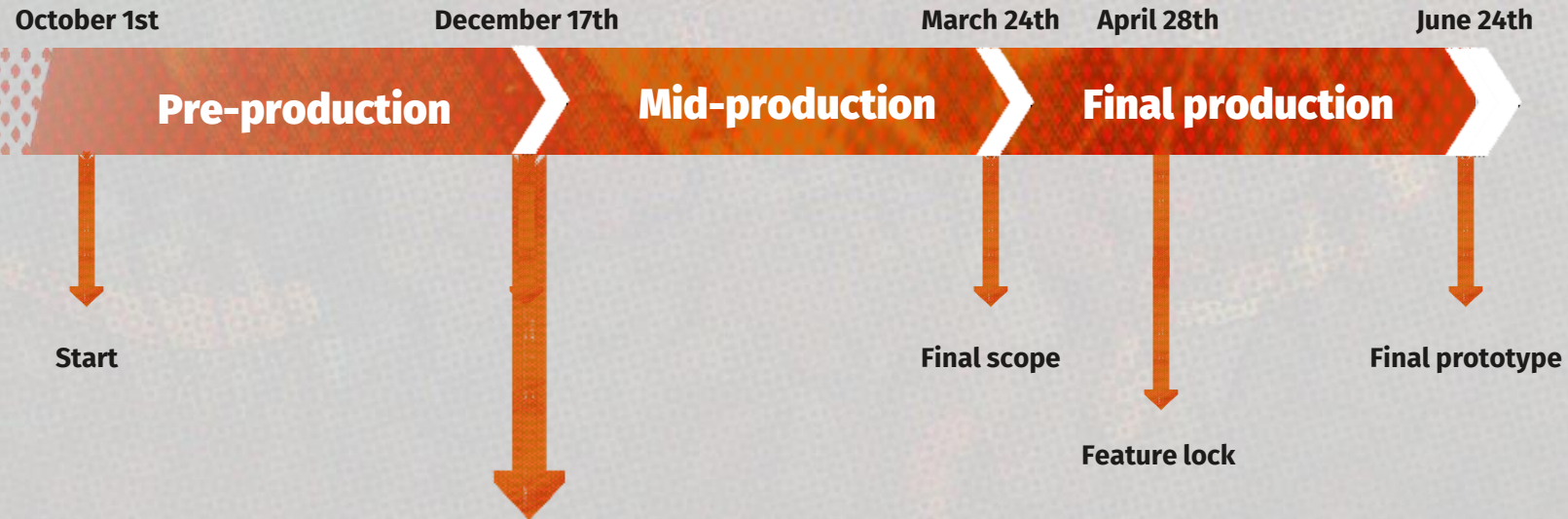


1. A feature is first designed and documented, going from the intentions (matching our core pillars) all the way to technical documents for programmers.

2. The feature is then created on JIRA. Its Confluence page is linked, and stories are attributed to team members.



3. Commits are linked to JIRA stories. Allowing anyone to quickly check the current state of a task through JIRA. Uploaded features are then tested during the next week's playtest.



Pre-production retrospective

Results were highly satisfying for the most part:

- Our network is fonctionnal.
- The new structure of Level Design, making Battle Royale and vehicular combat meet, is effective.
- Pipelines for every feature, whether ability, trick or anything else, was defined.
- But most importantly, a solid creative direction has been set, allowing everybody to have a clear vision of the project.

Team strengths



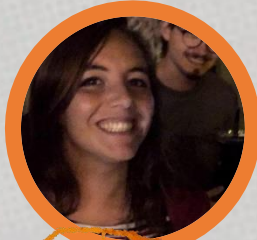
Dylan Fitzpatrick

Vision owner, 3D designer and programmer
Setting the game's creative direction with a lot of passion and pride!



Léa Lescuyer

Game Artist, environment and concept
Specialized in making things instantly looking prettier.



Léa Galinha

Art Direction, vehicle, UI, FX
Being very polyvalent and taking all feedback into account.



Florian Eschalier

Level Designer, and Technical Designer
Very effective problem solver.



Matthias Johan

Producer, and Game Designer
Emphasis on documentation and organization.



Anthony Rabaux

Lead programmer, network and AI
Always finishing tasks on time and making new propositions!



Louis Bayard

Level Designer
Sound Designer
High focus on intentions. Always working with a good will.

Must have

- Polished 3C experience
- 3 arenas and 6 segments
- 3 abilities and tricks

Should have

- Ability customization
- Destructible objects
- 6 abilities and tricks

Nice to have

- Spectator Mode
- Capture the flag mode
- Cosmetic customization

Outsourcing needs

In case the team wants to deliver a more polished experience:

- Regular playtesters, testing the network and all new implemented features together.
- 2D Artists, in order to add more graffiti decals to the City of Battle Cars!

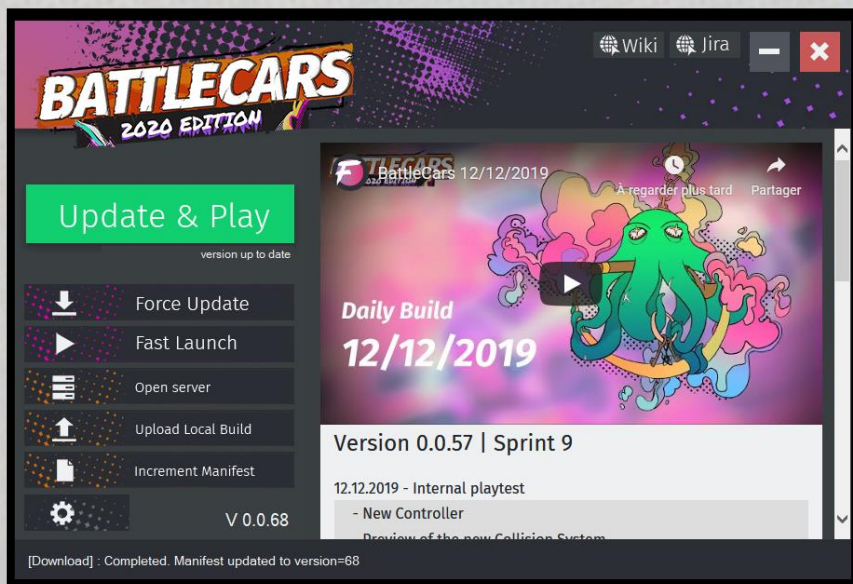
BATTLE CARS

2020 EDITION

Conclusion

Bonus features

Launcher



This feature was just finished!

- Quick updates for both the developers and the players
- A changelog displaying the build's latest changes
- A tool allowing quick deployment for playtests

Spectator Mode



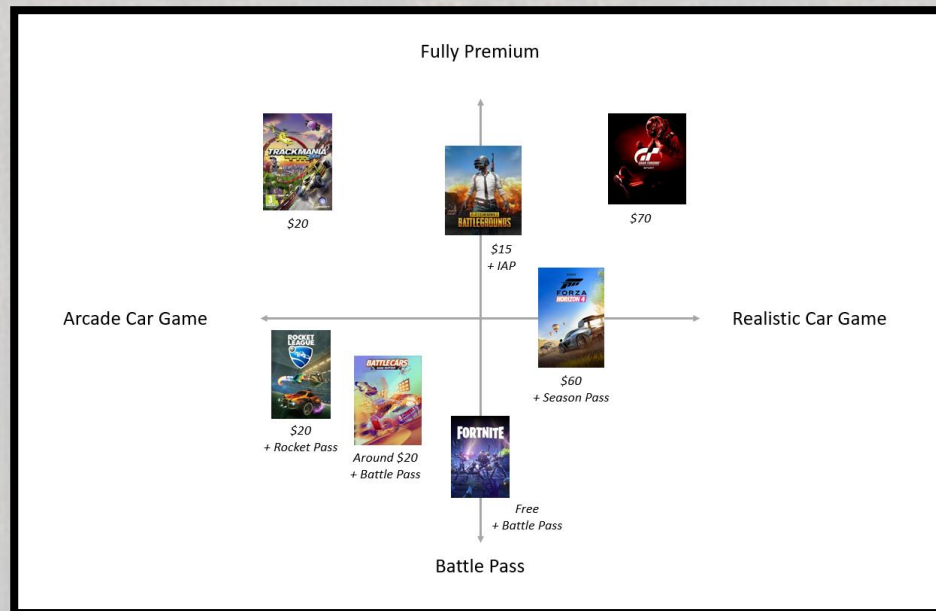
This feature is currently in development (credits: League of Legends)

- A highly streamable game
- A tool allowing quick analysis for level designers

Business model

Targeting 2000s arcade game enthusiasts, Battle Cars could be priced for ~ \$15 on both PC and consoles.
For an additional ~ \$10, players could obtain a **Battle Pass**, proposing quests that when fulfilled, offer no gameplay advantages but rather cosmetics such as skins, engine sounds, icons .

Bullet



Our business model is very close to Rocket League's.

Battle Pass



The Battle Pass system from Battle Royales even inspired Rocket League's Rocket Pass.

Old-school game feel

A very nervous and instinctive car controller with simple but effective mechanics such as collisions and tricks!

A chaotic environment

Both through a disorganized and colorful art direction, and hyper-realistic sounds!

The fusion of two genres

The old-school arcade car game feel meeting the popular Battle Royal experience!



Dylan Fitzpatrick

Technical Game Designer
dylanfitzpatrick.net



Léa Galinha

3D Artist
artstation.com/ecrapince



Léa Lescuyer

Environment Artist
artstation.com/lealescuyer



Anthony Rabaux

Programmer
anthonyrabaux.free.fr



Louis Bayard

Level Designer
louisbayard.eu



Florian Eschaliér

Technical Level Designer
eschalierflorian.com



Matthias Johan

Game Designer
matthiasjohan.com