BATTLE CARS

SUMMER 2020 EDITION

SUPINFOGAME
RUBIKA

Intermediate Jury - March 2020

Summary & organization

Project overview

1 – Summary & organization

2 – Concept

3 – Battle Royale experience

Gameplay

4 – 3C

5 – Core mechanics & systems

6 – Level design & UX

Art direction

7 – Environment

8 – Vehicle design

9 – FX & sound design

Technical direction

10 – Network & AI

11 – Tools & technologies

Planning

12 – Scope & planning

Core Team

Dylan Fitzpatrick

Vision owner
+ *3C design*
+ *Gameplay prog.*

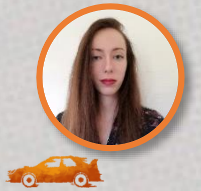
Anthony Rabaux

Lead programmer
+ *Network prog.*
+ *AI prog.*

Léa Galinha

Art director
+ *Vehicle art*
+ *UI art*

Léa Lescuyer



Environment artist
+ *Lighting*
+ *Concepts*

Louis Bayard

Level designer
+ *Micro LD*
+ *Sound design*

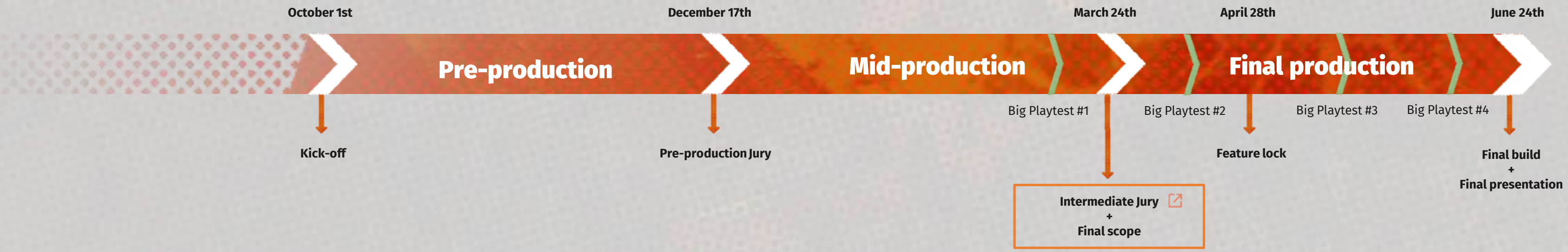
Florian Eschalier

Technical level designer
+ *Macro LD*
+ *Technical support*

Matthias Johan

Producer
+ *Game design*
+ *Presentations*

Global planning



October 1st

December 17th

March 24th

April 28th

June 24th

Pre-production

Mid-production

Final production

Kick-off

Pre-production Jury

Big Playtest #1

Big Playtest #2

Big Playtest #3

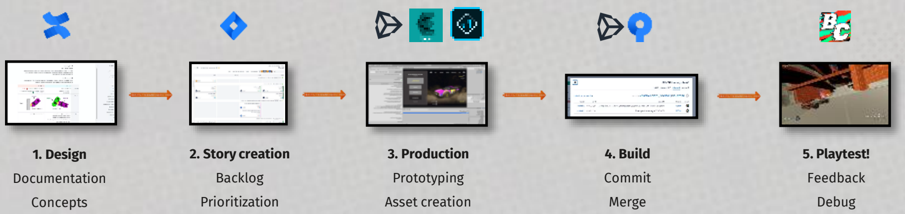
Big Playtest #4

Intermediate Jury
+
Final scope

Feature lock

Final build
+
Final presentation

Feature pipeline



1. Design
Documentation
Concepts

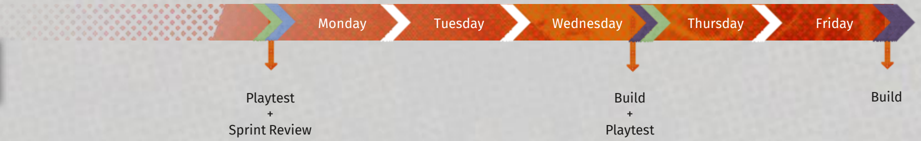
2. Story creation
Backlog
Prioritization

3. Production
Prototyping
Asset creation

4. Build
Commit
Merge

5. Playtest!
Feedback
Debug

Sprint iterative process



Monday

Tuesday

Wednesday

Thursday

Friday

Playtest
+
Sprint Review

Build
+
Playtest

Build

Concept

Vehicular combat

We are targeting **car game enthusiasts**, fond of 2000s arcade franchises, looking for a modern experience!

References: Destruction Derby, Burnout, Need for Speed, SSX...

Genre fusion

Vehicular combat Battle Royale

Battle Royale

We are also targeting **competitive players**, fond of e-sports and Battle Royale games, looking for a gameplay twist!

References: Fortnite, League of Legends, Rocket League...

Player fantasy

« Be more violent and stylish than others. »

Pitch

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

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.

64

Number of players

Up to 64!

Platform

PC & Consoles

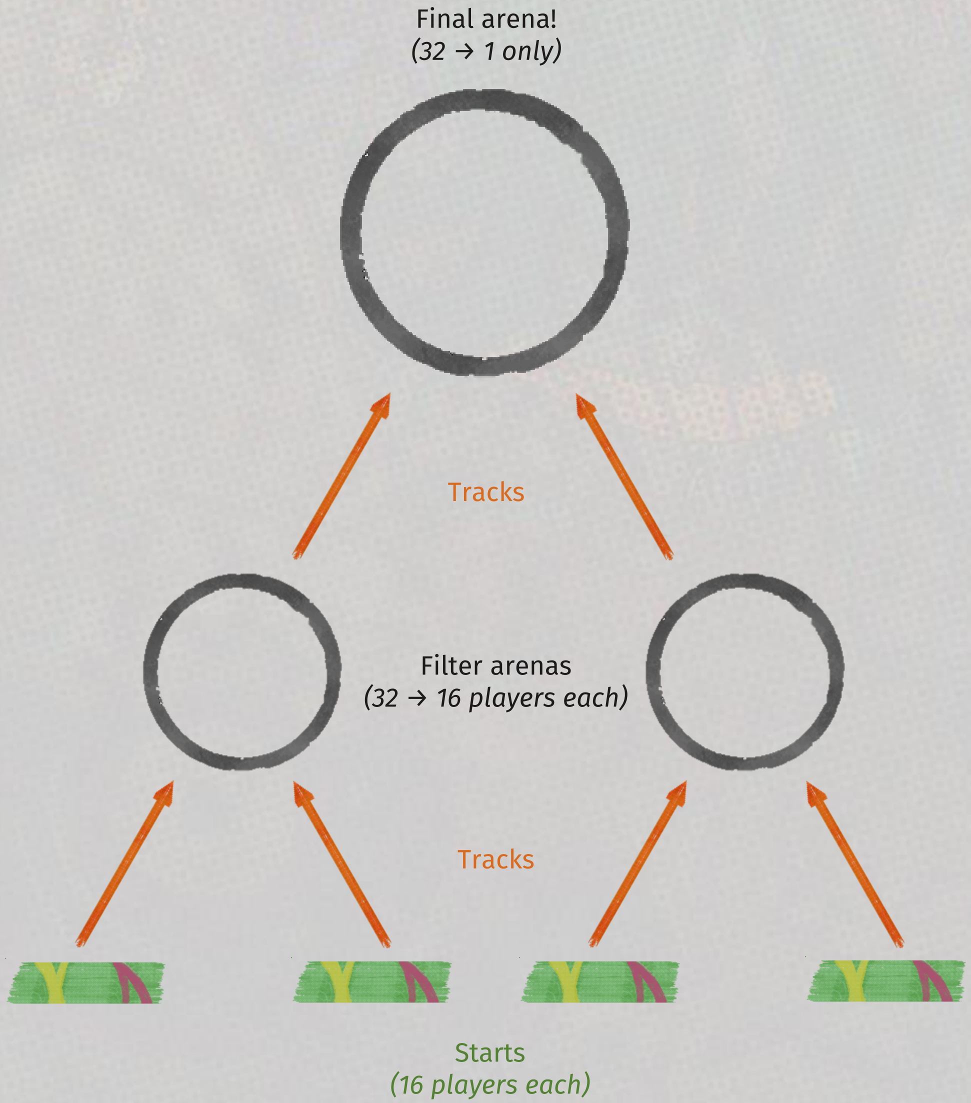
Controller

Gamepad

Game duration

5-10 minutes

Battle Royale experience

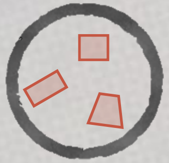


Game Mode

The game starts with 64 cars. **The goal is to be the last one riding (one life only).**
Players can lose by getting their cars too damaged either by other players or shrinking zones.

Arenas

Alternating with tracks, arenas focus on **combat challenges**.
When cars arrive in one of the filter arenas, exit gates won't open unless they are 16 players remaining on both sides.

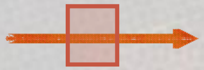


Arena deadzones

As the number of players alive decreases, deadzones cutting the arena into slices appear in a random order, increasing the player density over time. When inside, a player loses car health every second.

Tracks

Alternating with arenas, tracks focus on **speed challenges**.
Destructible objects giving brutality points are spread throughout the track. The first players therefore have a higher chance to destroy them and can gain abilities more quickly.



Track deadzone

Players are chased by a deadzone inside a track, motivating them to reach the arena and preventing campers. When inside, a player loses car health every second.

Which Battle Royale key rules are adapted in Battle Cars?

1. The concept of "Last Man Standing" → The concept of "Last Car Riding"
2. All players start from scratch → Players gain brutality points to upgrade their car and use abilities
3. The famous "Shrinking Zone" → Arena and track "Deadzones"

Nervous

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

Spectacular

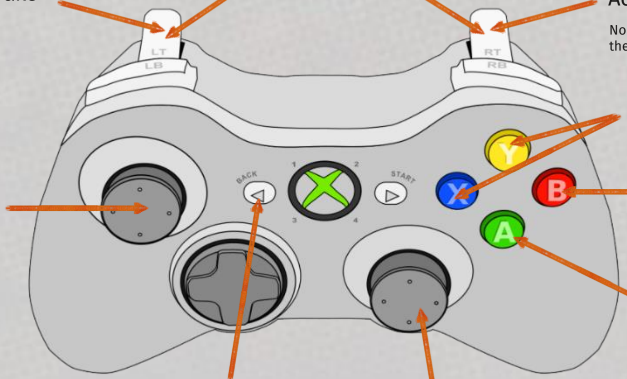
There is always something going on screen. Players are left on the edge of their seat.

Intuitive

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

Control scheme

We discourage players from braking by forcing them to release the right trigger to slow down, while **drifting is easy and highly encouraged**.



Drift

Brake

Accelerate

No actions have been mapped on the shoulder buttons to make the accelerate and brake controls **as accessible as possible**.

Use assigned ability

Heal

All mechanics which are not linked to the car basic movement have been **mapped on the face buttons by order of frequency**, going from A to Y.

Boost

Steer

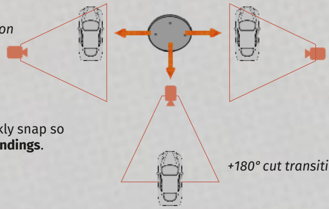
Player can also honk by pressing the left stick!

Display full leaderboard

Competitive players can access the full leaderboard to compare all game mode stats.

Rotate camera

Cameras left, right, and back angles quickly snap so that the player **quickly analyze its surroundings**.



Composition & UI

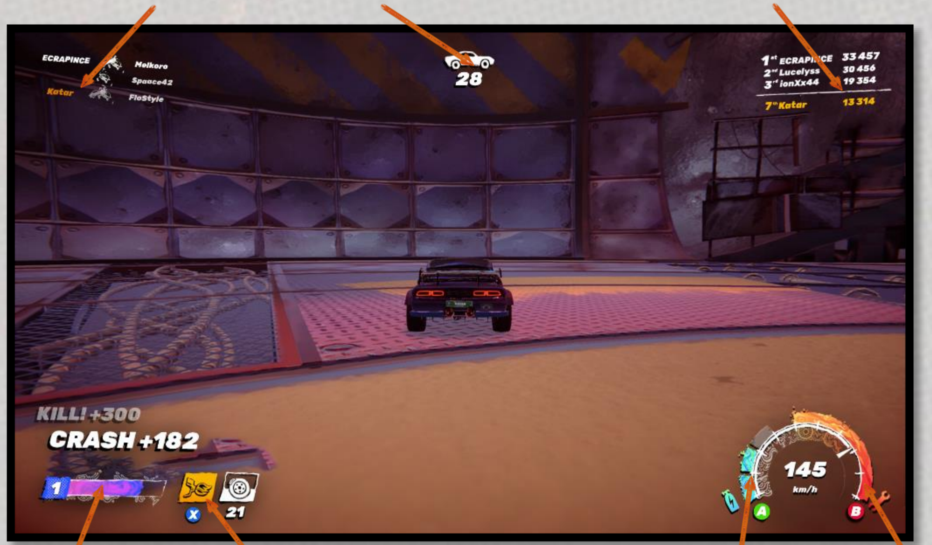
Displayed at all times on the top-center part of the screen, it's the **most crucial information** regarding the game state.

We realized through playtests that **players love seeing and comparing who are the top players**, even though the information is not essential.

Killfeed

Cars remaining

Brutality points leaderboard



Brutality level

Keywords indicating how the player gained brutality points quickly appear and disappear, giving this 2000s arcade feel, inspired by the SSX series trick system.

Abilities

The assigned **button only appears once the ability cooldown is over**, encouraging the player to use the ability thanks to affordance.

Nitro bursts

Car health

The speed and health indicators have been moved to the down-right corner of the screen to respect **car game conventions**.

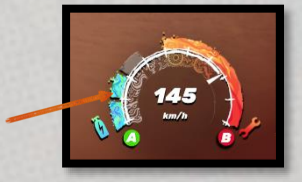
Core mechanics & systems

Tricks

Tricks are the main way to go faster and heal, and the main system representing the **« Show-off » pillar**, heavily challenging on controls.

Some iconic tricks:

- Drift
- Air time
- Near miss (dodging nearby obstacles)
- Barrel rolls (rolling the car when in the air)



Performing a trick adds a nitro burst.

There are **2 options to consume a nitro burst**:

A

Boost: *increasing the car's speed by 120% for 1 second straight.*

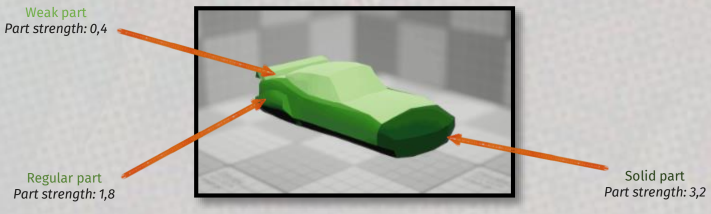
B

Heal: *recovering 10% of the car's total health.*

A **nitro boost** increases the player speed. Therefore, making tricks allows the player to go faster and **increase the player's power during collisions**.

Collisions

Collisions are the main way to deal damage in Battle Cars, and the main system representing the **« Explosive brutality » pillar**.



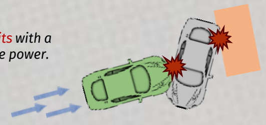
Car's collider, showing different spare parts with different strengths.

Power = Part strength x Speed

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

A successful collision is when the player is the one dealing damage to the other. In that case**, the player gains as many brutality points as the damage dealt**.

Green car is faster and hits with a stronger part. It has more power.



Green car is the one succeeding in the collision. It deals even more damage because Grey car also has a 2nd collision with an obstacle.
It earns even more brutality points!

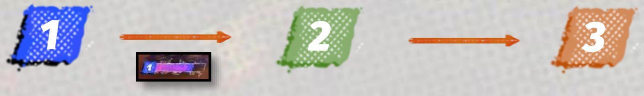
The "No fair play" pillar is not directly represented by a mechanic. It is represented by **rewards encouraging a "No fair play" behavior** such as depicted above.

Abilities

Abilities are the main way to value the **player's progression** throughout the race. They are **game changers**.

They can either:

- Improve the player's positioning
- Deal damage
- Both

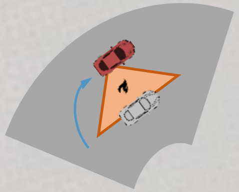


All players start at level 1 with no abilities. When getting enough brutality points, players level up and can add one of their 2 abilities. The maximum level is 3.

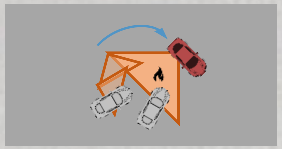
They are **assigned the X or Y button prior to the game**, inside the customization menu. When activated, an ability **cools down** before it can be activated again. X Y

Example: when activated, the **fireslide** creates flames (zone damage) when drifting.

Situation 1: *red car uses fireslide while overtaking another car in a turn.*



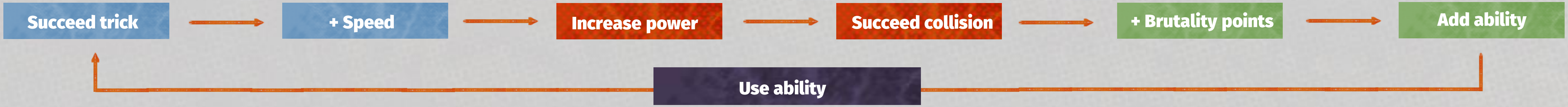
Situation 2: *red car uses fireslide to circle and deal damage to two oncoming cars in an arena.*



Every ability is versatile, and **can be used in different situations** such as depicted above.

Perfect gameplay loop

This loop sums up the 3 systems and represents a perfect scenario. Here is the full core loop.



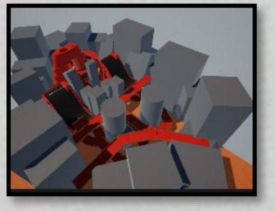
Succeed trick → + Speed → Increase power → Succeed collision → + Brutality points → Add ability

Use ability

Level design & UX

Macro Level Design

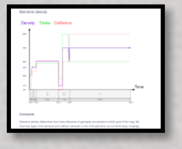
City



Starts

The city map, is designed and modeled by Florian, while Louis designs and builds the micro elements on the different tracks and arenas.

A **huge challenge we faced was redesigning a start** allowing 16 players to ride at the same time. We solved the issue by creating road dividers.



Map flow

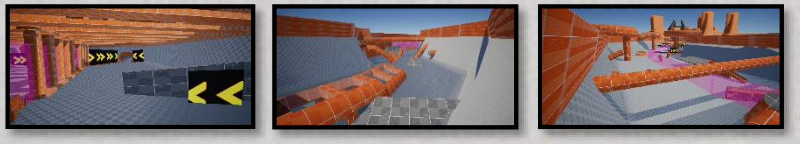
We carefully designed curves of several aspects of the level design (element density, lighting, speed, spacing...) to create a contrast between the tracks and the arenas and better control the game's flow.

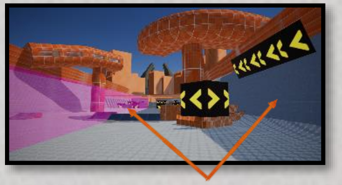
Tracks

Tracks are the main way to **represent the "Pedal to the Metal" pillar** and keep the player's controller at a high speed.

Target dynamics

- **Multiple layers and choices**
- Build up the nitro with elements favorising tricks
- Less lethality





Throughout the track, player has different choices. For example here, he can either go left and take the kick to perform an air time trick and gain a nitro burst, or go right to be in a better position for the next turn. Moments with wider spaces where the player can think about the next choice also create potential collisions!

Arenas

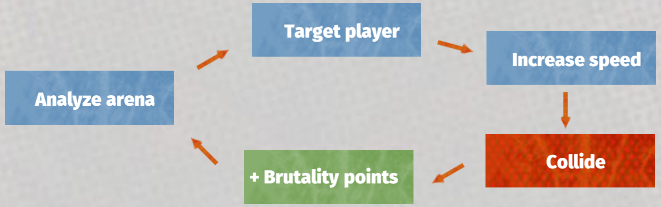
Arenas are the pinnacle of the **"Explosive brutality" pillar** and players are encouraged to have a "No fair play" behavior.

Target dynamics

- **Enforce collisions**
- Close space, no direction
- Gather information frome the other players

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

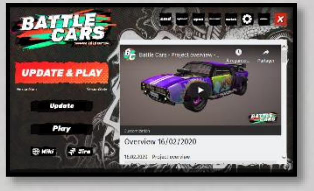
Micro combat loop



Player story

We designed an ideal player story lasting not more than 15 minutes for the final jury. This method allowed us to get a much bigger control over Battle Cars' experience and flow, even within the menu.

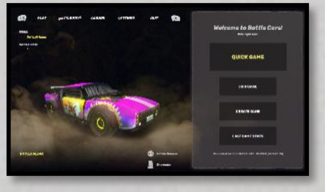
1. Launcher



Overview 16/02/2020

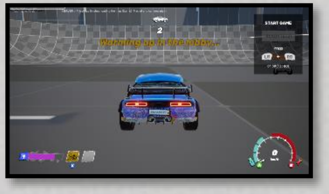
The build version is updated and launched automatically. The player can also get news on the project.

2. Menu



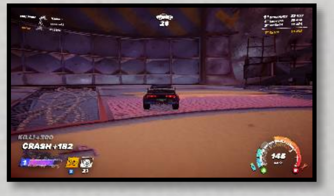
The player gets a clear overview of all features and may launch a game in 1 button press!

3. Lobby



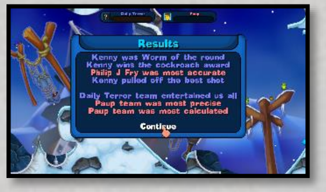
Player waits for other players to join the server, can warm-up and get used to the controls.

4. Game



Player plays the game, it's the pick of the experience! Player potentially dies and switches to spectate mode before the game ends.

5. Stats



Results

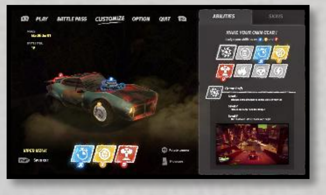
Kenny was Worm of the round
Kenny wins the cockroach award
Philip J Fry was most accurate
Kenny pulled off the best shot

Daily Terror team entertained us all
Paup team was most precise
Paup team was most calculated

Player compares stats with other players and potentially gets new vinyls to customize the car.

We realized players love comparing themselves to each other! Reference: the Worms series.

6. Customization

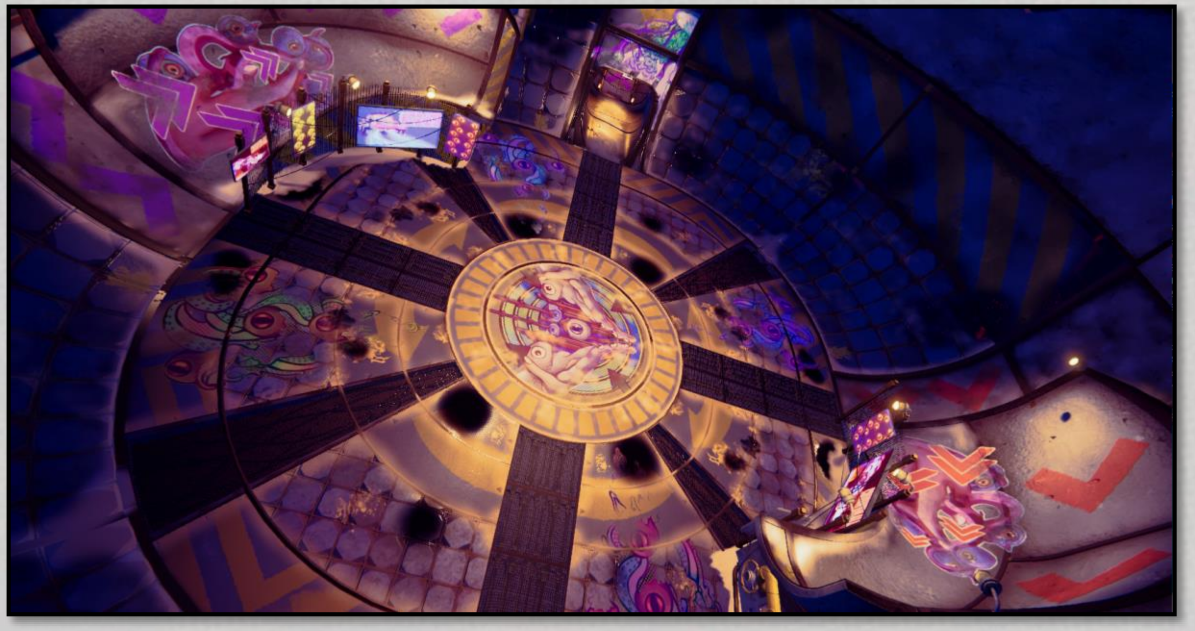


Player customizes car with new abilities and assigns new abilities to better fit a strategy.

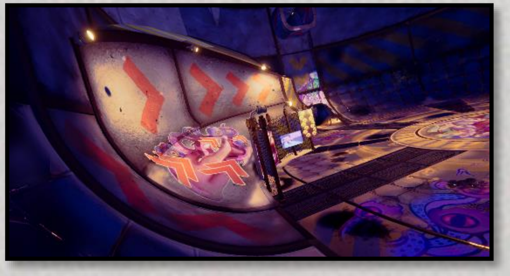
Environment

Focus: the final arena

We aimed for a fully reskinned final arena for the intermediate jury, level design, level building, and lighting included, allowing us to get a target render for the other scene and check how the art direction is seen from the controller's camera.



Graffiti signs

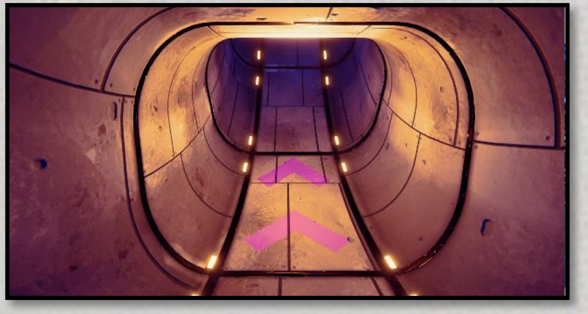


Arrows and graffiti are used as decals on the different level design elements to increase their affordance and motivate the player to use them.

Since the arena is symmetrical and we're targeting competitive players, the same asset has a different color palette based on its position.

Graffiti also allows us to immerse the player into the undeground vibe of the Battle Cars event!

Lighting



Lighting plays a big role in the level readability and emphases on the important level design elements, by targeting the entries and exits of the most important chunks.

Composition

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. For example here, the graffiti motivates player to collide at the center of the arena.

Universe

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.

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.

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!

Key elements

Street art

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

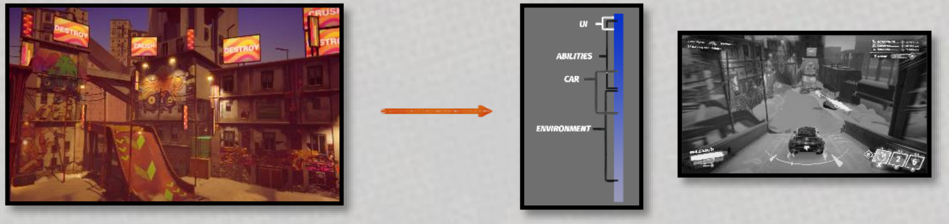
Building concept, pre-production phase.

Sensation of speed

*Objects are disposed at even distances (public lights, trash cans...) on the sides of the road. **Paintings on the ground** follow the track.*

Reference: Wipeout Omega Collection, Creative Vault Studios, 2017

Readability



A huge fear during pre-production was the level readability. We therefore made a lot of research and created a document prioritizing all the visual elements.

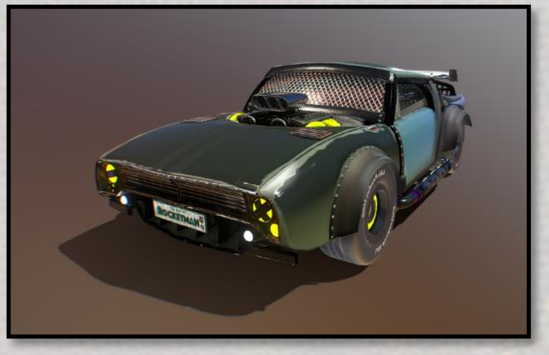
Vehicle art

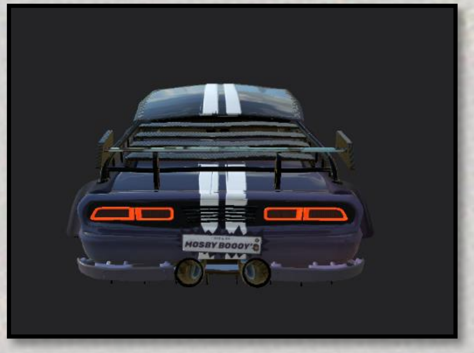
Design

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.”

Sharp edges and curved silhouette make it **optimized for racing and high-speed collisions**. The intention is not to make it perfect and clean, as the player needs to sacrifice some spare parts without any remorse.

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





Customization system

Step 1: colors

All players can have their **own identity** without altering the art direction.

At any time in the game, and without any prerequisite, the car color can be changed, among a defined color palette.

Material tests. Without further customization, the car looks cleaner but also less aggressive and adapted to a defined playstyle.

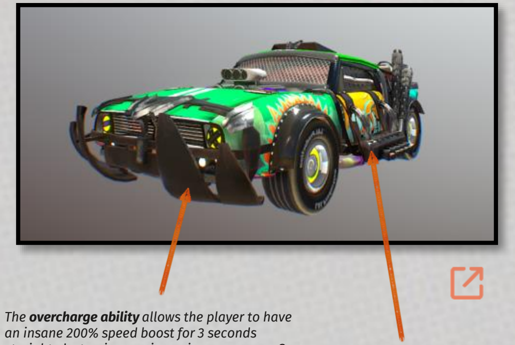
Step 2: vinyls

Our intention is to **reward players** for their achievements without giving them a gameplay advantage.

Every time the player earns an achievement during the stat screen, a new vinyl linked to the achievement is obtained. The car can then be customized so that the player visually shows-off his past victories to the others.

Vinyl rewarding the “reach 300 km/h” achievement, represented by a blue falcon. This vinyl was designed by Marianne Fourmanoit, outsourcer.

Step 3: abilities



*The **overcharge ability** allows the player to have an insane 200% speed boost for 3 seconds straight, destroying any incoming enemy car. On the other hand, the car steering is blocked!*

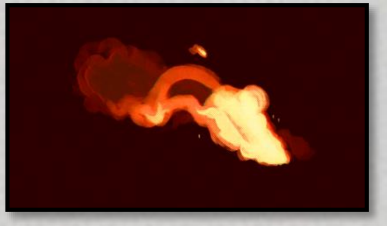
*The **fireslide ability** allows the player to trigger flames on the side of its vehicles while drifting, dealing a huge zone damage to any enemy car in contact with those flames.*

Every ability is represented by one or several 3D models grafted on the car, **easily identifiable** by other players from all sides, and giving a huge sensation of evolution to the owner!

FX & sound design

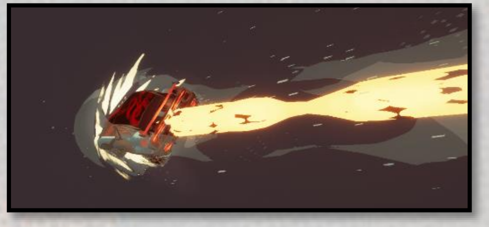
VFX

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.

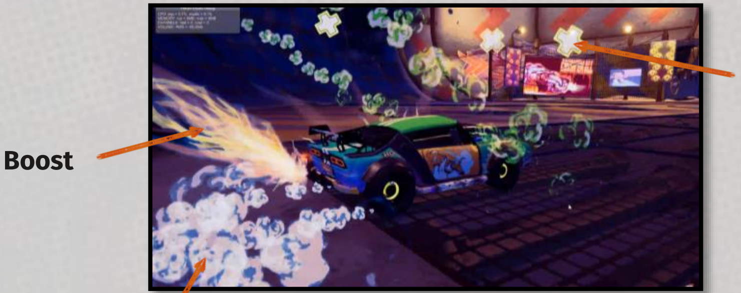


Fire concept art

Nitro concept art



VFX pre-production scene

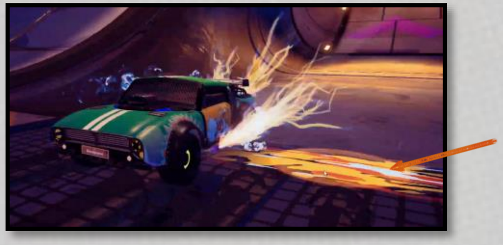


Heal

Boost

Drift

Click on the images to read the VFX video.



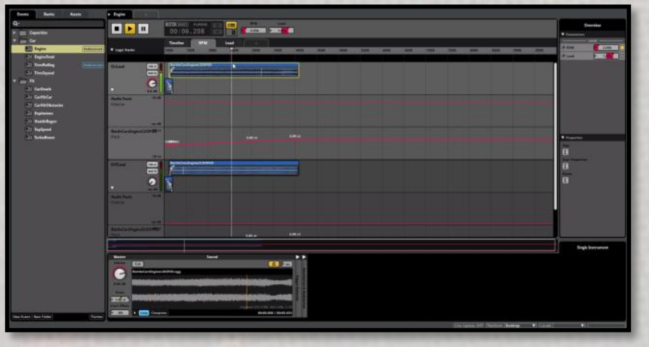
Fireslide ability

Thanks to our outsourcer Louis Houyez, we managed to **prototype and integrate placeholder VFX to add game feel**. Léa Galinha, our art director, will rework on the textures and particles tweaking during the final production phase.

SFX

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

We just integrated FMOD to the Unity project:



Click on the image to read the SFX video.

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).

Music

Listen to our **demo music** in our project overview video (0:02 and 2:05)

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

Feature example: a high pass filter on the music while a player is in the air to strengthen the sensation of flight.

Network & AI

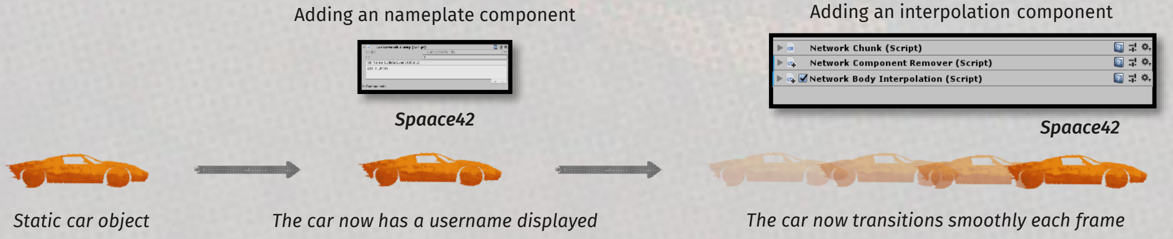
Engine: Unity3D

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 and architecture
2. Secondly, because it's also much faster to build C# library with Unity3D from scratch than optimizing the default replication system from Unreal.

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

Component process



Adding an nameplate component

Adding an interpolation component

Network Chunk (Script)
Network Component Remover (Script)
☑ Network Body Interpolation (Script)

Static car object

The car now has a username displayed

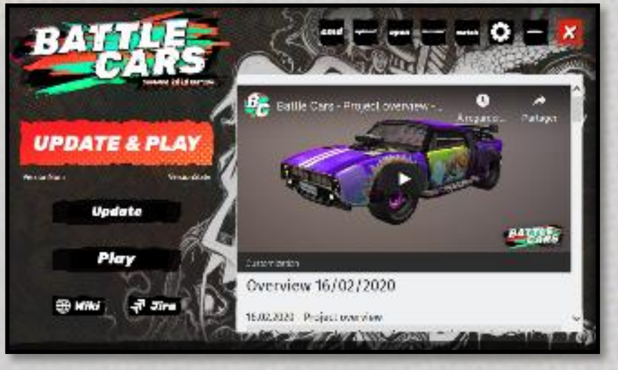
The car now transitions smoothly each frame

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

Launcher

Developing a launcher made us gain a huge amount of time. Some of its benefits:

- Quick updates for both developers and players, for debugging and large-scale playtests.
- A changelog displaying the build's latests changes
- A tool allowing a quick management of the matchmaking and server system



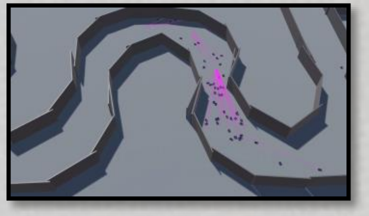
Overview 16/02/2020

16.02.2020 · Project overview

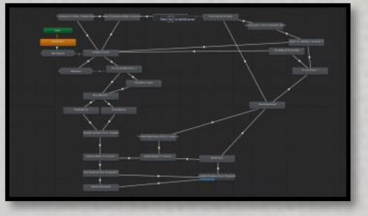
AI cars

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, **AIs automatically fill up the missing slots**. *For example, if there are only 40 clients, 24 AIs join the server*. The AI parameters will then be tweakable by Game Designers to simulate player strategies.

Due to timing and scope changes, **we abandoned the idea of making a machine learning navigation for the moment**.



Our current AIs in engine using raycasts to detect the oncoming obstacles and decide if they either go left or right.



Behavior tree using Unity's animator

Tools & technologies

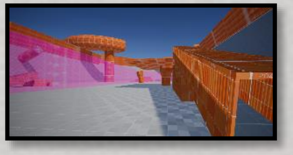
Spectator mode

Losing doesn't disconnect you directly from a game. It instead switches to the spectator mode, allowing you to continue to observe the game.

This feature is quiet useful for the team in developing the presentation tool, the trailer, the menu camera, as well as reinforcing a streamable-friendly experience for e-sport players.

Spectators can alternate between **3 different types of cameras**:

1. Free fly (player can fly and roam around like around)
2. Pursuit (the player follows another player's car, like if it was his own)
3. Cinematic (automatically aims and positions itself)



Current free fly camera.

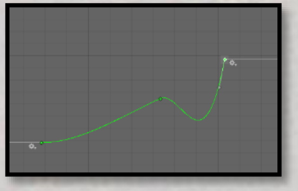
The player can also directly take a screenshot of their current angle.

Presentation tool

Our goal for the final jury is to make an in-game presentation. We **fully finished** the development and even made a tutorial for our presentation tool!

Slides can alternate between **3 different types of cameras**:

1. Controller camera (the car controller with all its features)
2. Spectator camera (similar to the spectator mode controller)
3. Fixed camera



The car is blending between two positions (one clean, one broken) with a morph-target based system.

Mesh deformation

We made several research and on the topic, and listed all possibilities to deform the mesh. Even though not essential for the gameplay, deforming car meshes during collisions is a great addition to gamefeel.

The best mesh deformation for Battle Cars, in terms of both scope and computing power is a **mesh blending system**.

The car is blending between two positions (one clean, one broken) with a morph-target based system.

Heatmap

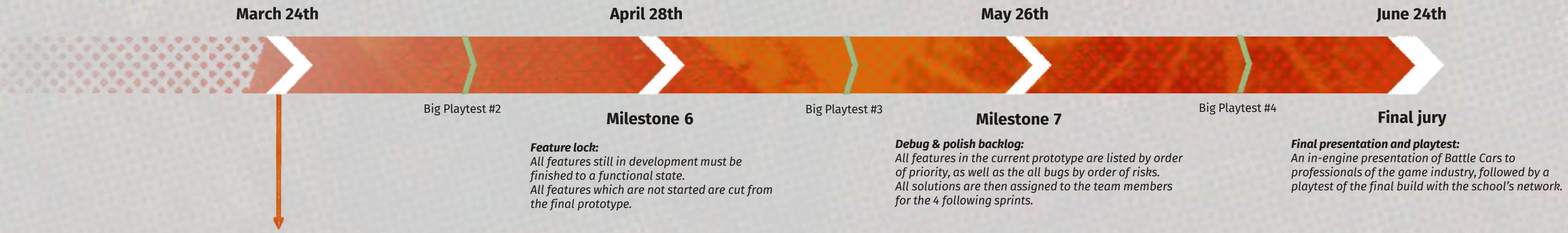
We're planning on developing an heatmap system to **better understand where in the map specific events occurred**. For example, where do players take the most damage, where do players boost etc...

We'll develop a JSON structure and use both our AI tests and our Big Playtests (open playtest with 40+ players) to gain enough data to analyze.

Our current AIs in engine using raycasts to detect the oncoming obstacles and decide if they either go left or right.

Scope & planning

Retro planning



March 24th

April 28th

May 26th

June 24th

Big Playtest #2

Milestone 6

Feature lock:
All features still in development must be finished to a functional state.
All features which are not started are cut from the final prototype.

Big Playtest #3

Milestone 7

Debug & polish backlog:
All features in the current prototype are listed by order of priority, as well as the all bugs by order of risks.
All solutions are then assigned to the team members for the 4 following sprints.

Big Playtest #4

Final jury

Final presentation and playtest:
An in-engine presentation of Battle Cars to professionals of the game industry, followed by a playtest of the final build with the school's network.

Mid-production retrospective

The team decided to directly focus on the main weaknesses spotted during the pre-production phase:

1. We made a **3rd and final iteration of the 3C** to solve the stability issues and make game feel tweaking easier for the future.
2. One of our biggest fears was the readability, the level designers and artists worked hand to hand to create a **readability chart**, fitting with the art direction.
3. We went through a lot of tests and suggestions by trainers, to finally find out a **working pipeline for our level designers and level artists**.
4. Thanks to the **Big Playtests** (open playtests gathering 40+ players), we managed to highly improve our network, and get a huge amount of player data.

Scope tricks and cuts

We had to make sacrifices to make sure we deliver the best possible experience at the right time. Here are some measures we already took to reduce the production scope:

- The **first tracks and filter arenas are reversed clones** of each other.
- **Moving obstacles are cut**. They are not essential even though they add a bigger layer of interactivity.
- **Abilities** have been simplified: they **don't evolve anymore**, also making the system more accessible.

Final jury scope

Must have

- Polished 3C experience
- 4 abilities and 3 tricks
- The final arena, the final track, the starts

Should have

- Customization system
- Spectate mode
- Intermediate arenas and tracks

Nice to have

- Extra game modes (ex: capture the flag)
- 6 abilities and 5 tricks
- AI navigation with machine learning

Thank you for reading!

We remain hopeful and wish to stay productive even when going through the Covid-19 recent events.