



x

Sommaire

Introduction

Team 3

Game Design

Pillars 6
Researches 7
References 8-9
3Cs 10
Guardiola's loop 11
OCR 12
MDA 13
Gamefeel 14
Rules 15
Features 16
Sign and feedbacks 17

Level Design

Tutorial 20
LD Study 21-22
Arena Building 23

Game Programming

Basics 26
Ball State 27
Payer State 28
Multi Scene 29

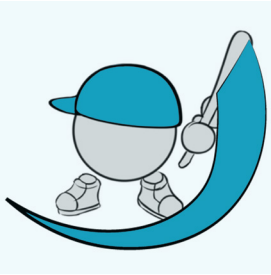
Visuals

Intentions 32
Character 33
Environement 34
UI 35

Sound Design

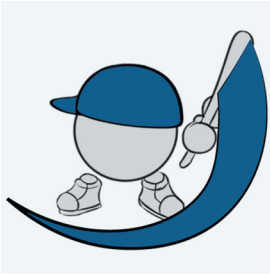
Intentions 38
Reference 39-40
Sound List 41
Process 42
Music 43

Team



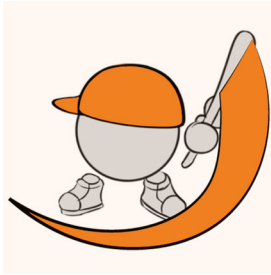
Aranda Ivan

Project Lead
GD Lead
GP



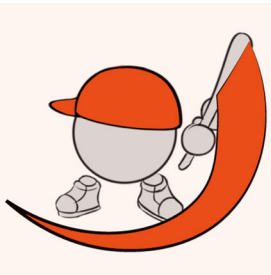
Baptiste
Moreau

GP Lead
LD Lead



Omane
Lorimy

DA Lead



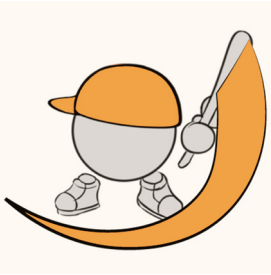
Ethan Azoulay

SD Lead



Lisa Bauder

Research
3D & Anim



Nathan
Jeger-Pariente

GD

[illegible]

Intention Pillars

Blitz Ball was conceived at the start of production with a simple mechanic of throwing and catching a ball. With this tool in mind, we iterated through multiple ideas and concepts changing both how the camera works as well as how the objects in the scene interact.

As we tested, we arrived at multiple conclusions that would influence the final product:

We wanted a multiplayer experience, where we could play a game with the ball with friends and acquaintances.

We wanted a simple gameplay that was easy to understand but hard to master. This came with the chaotic nature of the ball and the idea of creating a control scheme that is both rapid, reactive and effective to use (applying Game Feel principles from Steve Swink).

We wanted a casual experience, allowing players to pick up the controller and play with their friends or family in a fast setting and in a short burst of time (15 to 20 minutes).

CASUAL: The game has to be easy to pick up, to understand and to play. This doesn't mean we cannot have a certain level of difficulty when learning how to play, as a high skill ceiling, but anyone must be able to pick up the gamepad and understand the basic gist of the control scheme.

ENERGETIC: Something has to be happening on the screen constantly, keeping players on their toes. This is mostly achieved with the ball, but level design can impact how players will react to the motions of the game itself.

COMPETITIVE: The game has to foment player attention towards the other player and act accordingly to beat them. This involves also having multiple players at the same time.

Thematical Researches

Sports

In the frame of our project we took a look at multiple sports involving a ball. We took apart the way they defined themselves through rules which expressed themselves in material, spatial limitation, custom, goals, vocabulary... Below is a summary of our researches

DodgeBall



Andrew Ketchum at DodgeBall Worldcup
Dodgeball is one of the rare ball sports where your aim goes directly at another player. The texture of the ball, a softball, and the layout of the arena adapt to it: it is a wide space allowing direct hits but also providing enough space to dodge as its name implies.

Squash



Taking a closer look at squash what interested us is the way it is oriented in the same direction. The way points are counted and rounds are interesting also is something that defines which game one is playing.



Grâce Zaadi at Handball Eurocup 2018

Handball is for us another good example of limiting space with rules. The players have to adapt the way they move, the way they can interact with the ball and with other players depending on when they are on the field.

Pool



Allison Fisher 2019 World 9-Ball China Open

Pool is also a game that is interesting, for us. First because of the way it integrates rebound as the main strategising aspect of the game. Second for the context it is played in, needing a lot of material but also being present in places of social crossing. Finally it has a tendency to branch out into different games with slightly

References

To make it easier for us to conceptualize the project, we drew inspiration from many games based on the rebound of a ball/object. Lethal League is one of them, for several reasons:



Lethal League by Team Reptile

Lethal League has a major focus on speed buildup, and the player has total control over this speed increase. They have a concept of attack direction that influences how much the ball's speed increases. Hitting the ball without adding a direction during the hit will send the ball straight forward with no angle, and the ball will only gain +1 in speed. On the other hand, smashes — when players are in the air and hit the ball using down + direction — will double the speed increase compared to a normal hit. This mechanic can be used to defeat the opponent based on the rebound of the ball, which will no longer go just in front of the player. But what we mainly retain is the influence of ball speed during a hit.

Lethal League's core gameplay mechanic is based on timing, because once the ball reaches a certain speed, it becomes too fast to predict its rebounds and thus unstoppable, unless the player gets lucky with their reaction.

To counter this, they introduced the Hit Stop mechanic.

When a player hits the ball — and this occurs throughout the game — there's a small stop in time before the ball is launched.

The player who needs to intercept the ball can use this pause to react and hit the ball back more easily at the exit of the attacker's hit stop.

The hit stop timing is shown on the UI at the bottom of the screen.

Hit Stop is tied to the ball's speed tiers: the faster the ball goes, the longer the hit stop.

At the beginning of the match, as long as the ball speed is under 180, the hit stop duration equals 2/3 of the ball speed.

Beyond 180, the hit stop duration is always 2 seconds.

This Hit Stop makes it a precise timing game — you have to strike at the exact right moment.

Linking such precise timing is very interesting because it gives players a real

Overall Game Rhythm

The game follows a variable rhythm, generally very fast, but due to its mechanics, the overall speed often slows down or even pauses — only to return to high speed immediately after.

By default, the ball moves very fast throughout the match, but Hit Stop (or release time) allows for a pause, which is necessary for the defending player.

They must either position themselves to intercept the ball when it exits stasis, or calculate the rebounds based on the opponent's attack direction.

This pause also gives the player a mental break before the ball resumes bouncing uncontrollably.

The bunt is a mechanic that drastically slows down the ball and launches it in a pure vertical arc.

It allows players to: have full control over the next shot due to the reduced speed and bait the opponent regarding the actual launch direction.

Rebound Visualization

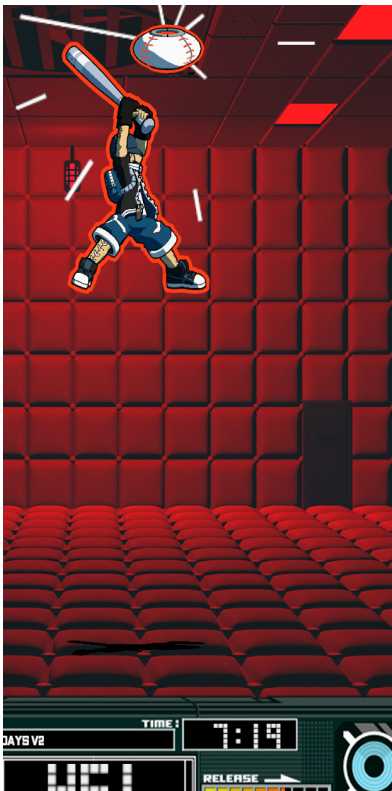
Lethal League heavily relies on making rebound prediction easier for the player.

anticipating the next rebounds.

To achieve this, the game allows the player to see up to 3 previous rebounds when the ball is moving at high speed.

Having a time history of rebounds, along with trails that follow the ball's movement, helps the player mentally draw the ball's path toward its next destination — and thus, prepare to defend.

The addition of pause times, as previously mentioned, also helps with anticipating the next rebounds.

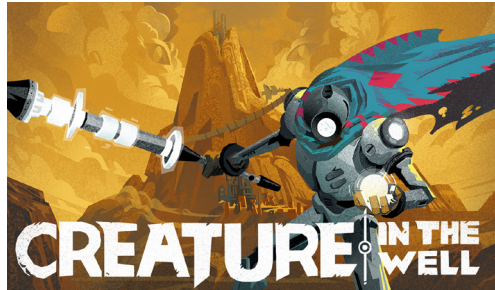


References



Boomerang Fu by Cranky Watermelon

Boomerang Fu was a reference for its atypical level design, but especially for the influence the player has on the throw and the rebounds. The player can influence the direction of their throw with an effect, and thus create a certain variety of possible throw directions that go far beyond strict, straight lines. What's more, they don't play with a ball but with a boomerang, which by default returns to the player on its own, thereby creating a new two-step throwing mechanic — the outward throw and the return. It is therefore possible to play with the return, moving your character so that the opponent ends up between the boomerang and the player who threw it, allowing the returning boomerang to hit the opponent.



Creature in the well by Fight school Studio

Creature in the Well helped us a lot in studying the concept of rebound and aiming with precision. In the game, you can grab the ball and keep it close to you for as long as you hold down the button, in order to position yourself, aim, and shoot precisely where you want to.



Stikbold! by Game Swing

StickBold was our reference for basic dodgeball gameplay. We were able to study the aiming system, which is clearly indicated via a direction indicator and a ball charge indicator. The fact of indicating the direction of the

attack is essential for a multiplayer dodge game. Moreover, their attacks do not kill on the first hit — at first, you are knocked back and stunned, and if you are hit again in that state, you can die. But what interests us is the ability to be stunned after taking a hit.



Creature in the well by Fight school Studio

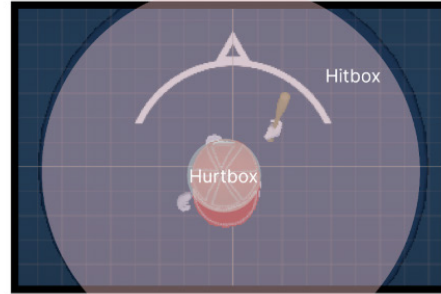
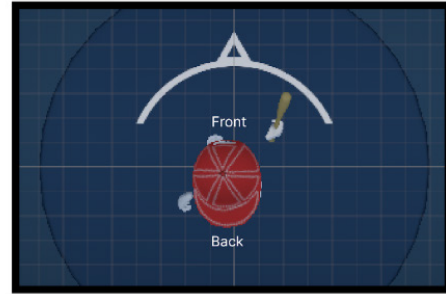
Knockout City was a foundational reference for multiplayer ball-based games. In it, you also find the mechanic of throwing the ball at each other to stun or eliminate opponents. There was no aiming or bouncing, because the player could use auto-aim on the opponent. However, there was also the mechanic of dashing toward an opponent to stun them. The idea of including a dash toward opponents was a real consideration in our project, but we ultimately decided to abandon it.



Mario Striker by Next Level Game

Mario Strikers allowed us to study the concept of an arcade-style game built around the theme of football. Each team must defend its goals while attacking the opponent. It is possible to stun opponents by hitting them with the ball, allowing you to pass more easily and get better access to the opponent's goals. The interest here lies in playing both offense and defense directly with the ball, but also in the concept of how to use the field itself to advance toward the opponents.

3 C's



Character

The main components of the player characters is the ability to show the front of the player, their hurtbox and their hitboxes.

To denote the front of the character, the model counts with an accessory that shows what is the front. In our lead character, we use its cap. Since the cap is on reverse, we can see the player's front clearly, as well as its direction being very evident due to the shape of the character.

The character also has cartoonish proportions to justify the way they interact with the ball.

Caméra

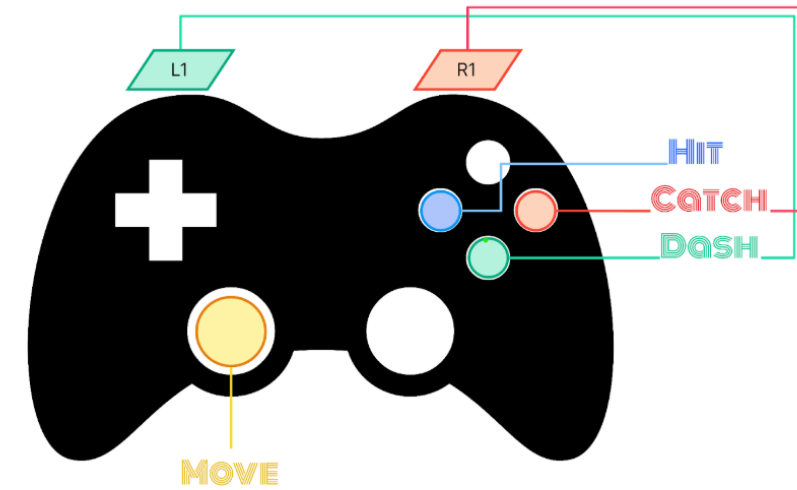
Using a top-down camera, we can see the complete terrain of any

in-game arena. Additionally, the camera moves slightly towards the point between the players, the ball and the center of the arena. This allows two things: It allows players to see the complete arena and be able to better predict the ball's trajectory. Let the players see their own character and be able to dodge and react to the moving parts of the level.

The camera also has a shake that is based on the ball's current speed, and happens specifically when the ball hits a wall. The screen-shake is meant to intensify the danger of the ball, as once it reaches a lethal state the ball has to impose itself by affecting the complete picture.

Controls

The game is best experienced with an XBOX 360 controller. The basic scheme allows the player to move around the arena as smooth and precise as possible, as well as allowing them to aim and plan tactically whenever the ball is around them.



LEFT JOYSTICK: Move in a two-dimensional axis (X and Y).

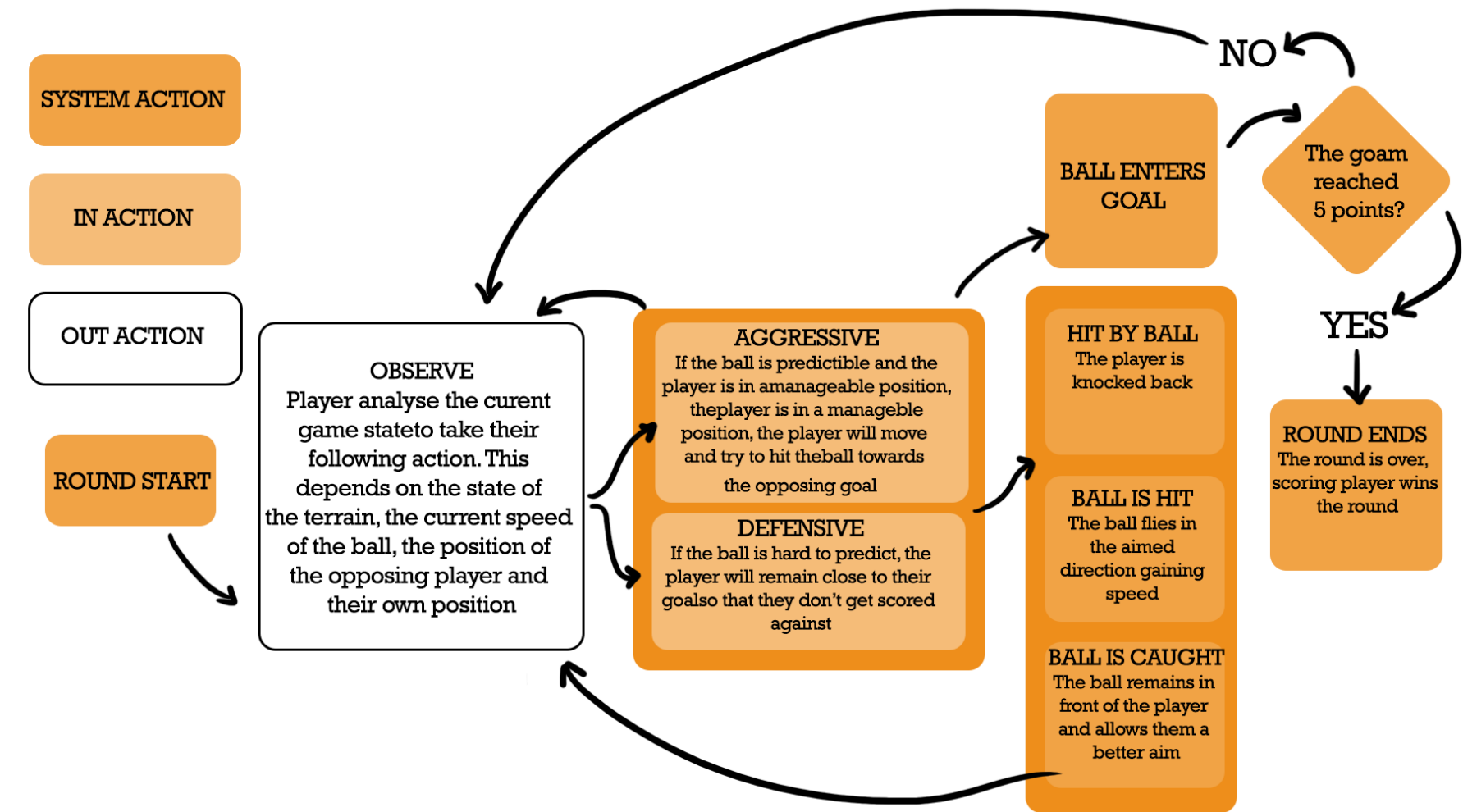
R1 / B (EAST BUTTON): Catch the ball. By holding the button the player cannot move but can rotate and release the ball in the target direction. While holding the ball, the other player can come in and hit it away.

L1 / A (SOUTH BUTTON): Begin sprinting. Sprinting grants a boost of speed that fizzles out into an homogenous faster speed, mixing components of a dash and a running option.

X (WEST BUTTON): Hit the ball and increase its speed. When the ball is touched, the player enters a hitstop of a duration depending on the ball's speed. During this hitstop, they can aim the ball slightly.

Blitz Ball follows a sound art direction based on two core pillars: the

Gameplay Loop



MDA

Sensation

This progression also reinforces our second aesthetic: Sensation – Game as sense-pleasure. It plays on the growing feeling of speed and the rising “pressure” on the players within the arena, as the ball becomes faster and faster, bouncing off all the walls.

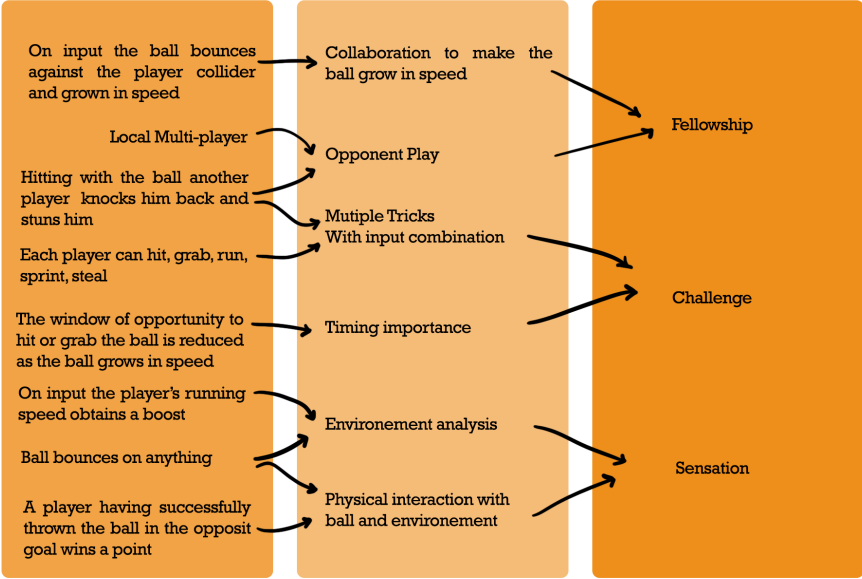
This sense of pleasure is further enhanced through our controller design, which rewards players for executing satisfying action combinations. The game feel improves with the player’s mastery of the controls and the evolving tempo of the round, gradually drawing the player into a deep flow state. In this context, the feeling of “playing well” becomes a challenge and a goal in itself. Much like in competitive sports, improving one’s skills becomes almost as important as winning the match.

Fellowship

The fellowship aesthetic manifests in a unique way. As a local multiplayer game, the competition and player-to-player comparison bring a strong social dimension to the experience. On a more practical level, fellowship is built into our game system at the start of each round, where players must pass the ball back and forth to accelerate it gradually. Additionally, inspired by various ball-based games, we’ve integrated elements like rules and a defined playing field to provide a solid foundation for this aesthetic.

Challenge

The challenge in our game is embodied in the discovery and mastery of the controller. On top of that, we’ve introduced various obstacles within the arena. The difficulty of aiming and the increasing intensity of the gameplay as the round progresses significantly amplifies this aesthetic. The rising curve of tension contributes heavily to the sensation of challenge.



OCR

OBJECTIVE	CHALLENGE	REWARD
Score one point	Angle my hit correctly so that the ball bounces intothe opposing goal	Get a point
Defend my goal	Predict the balls mouvement and position to propely hit it within the allotted opportunity window	Succesfull hit, ownership of the ball
Catch the ball	Predict the balls mouvement and position to propely catchit within the allotted opportunity window	Have a better opportunity to aim the ball
Steal the ball	Move towards a player holding the ball and attempt to hit it	Disrupt the opponent player

Gamefeel

High Level Context

As a large macro aspect of the game, the ability to feel that the game is ramping up in intensity and speed is important to impose on the players. In Lethal League Blaze, we find that as the ball gains speed and it becomes that much more dangerous, the terrain and background change to portray the increasing lethality of the combat. In our game, we can see the same effect as the ball becomes lethal when it reaches a sufficient speed, and the camera shake becomes much more violent. Other details denote this increase in intensity such as the ball's smoke effect against the walls being bigger.

Mid Level Context

Each level has an arena that sports different types of walls with diverse angles. This is done to promote variance in the bounce of the ball, which will inevitably be harder to predict when found in stranger and less open spaces filled with internal walls.

Low Level Context

Both the ball and the player character are the most important conveyors of feedback for the low context of the game. On the side of the player character, the most evident things to denote are the ability to hit, grab and sprint while on the ball we want its trajectory and its bounces being the most important thing to pay attention to.

Hitting the ball produces a big effect around the player, denoting the player's range and giving a clear indicator of where the ball will and won't be hit, granting a small window of opportunity to avoid precise frame inputs.

The grab feedbacks both its catch angle and the effectiveness of the catch. This is portrayed with the particles that are constantly moving towards the player, denoting their pull, and the effectiveness of the pull being feedback by the amount of particles in the animation.

sprinting was explored previously but the feedback to show a proper sprint is subtle yet effective. A trail is generated behind the player character, who changes their pose to signify the sprinting state.

As for the ball, simple is always best: Using the example from Lethal League Blaze, our ball has a trail that increases in length the faster the ball becomes, which grants the player a clear indication of the ball's direction as it's moving. This is a great aid for players as knowing the ball's trajectory gives them an easier time predicting its movements. Then, to signify impacts, we create a particle effect at the impact point that increases in size as the ball goes faster and faster. Additionally, we also have the screen shake that becomes higher and higher.

Finally, to signify the passage of the ball from non-lethal to lethal, the ball will slowly change colors from its normal flying color to its lethal color, as well as a UI element that denotes the current state of the ball.

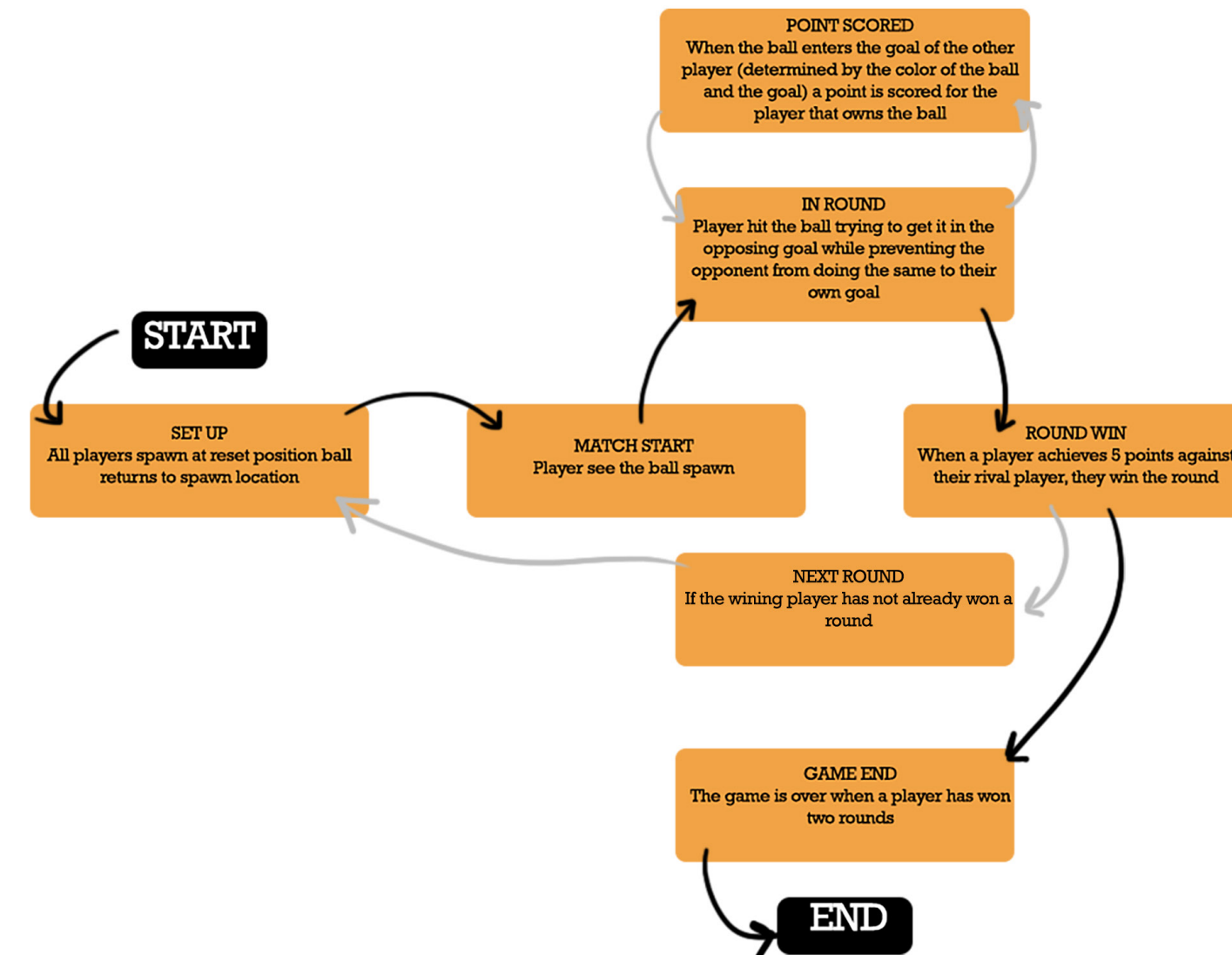
Rules

In Blitz Ball, we take control of a Blitz Ball player : An inhabitant of a ball city where a dangerous game takes place, the titular Blitz Ball.

To win, a player must win two rounds from the three that are to be played. To win a round, players must get the ball into the opposing goal by either hitting it and raising its speed or catching it to stop the ball momentarily and be able to properly aim it.

A player can not score against himself. The last player having touched the ball is considered the owner of the ball.

Matches are 1v1, where only one player can win the match. If no points are scored after a while, the ball becomes faster and faster, forcing the end of the bout as fast as possible.



Features

Control Feel

The main objective of the controller in the game is to allow the player an ample degree of freedom, while respecting the constraints of the level design and the threat of the ball. There are two main ingredients to take into account for this effect: The player's ability to dodge and move, and the player's ability to catch and hit the ball. Additionally, the player needs a certain degree of control over the ball's direction, but within the margins of the increasing intensity that occurs in a round.

The mechanic tools that the player has are the ability to move, the ability to sprint, the ability to catch the ball and the ability to hit it. Each of these mechanics grant benefits and risks.

Sprinting



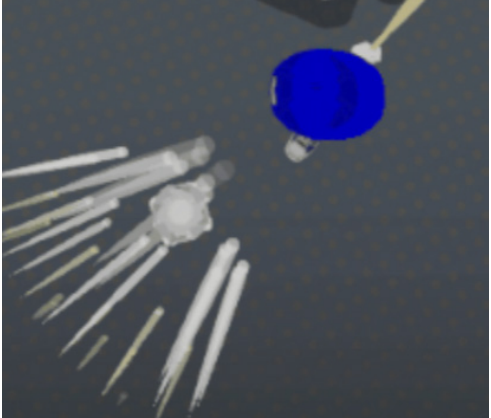
Sprinting is a tool that allows the player to quickly move through the terrain. When sprinting, the player gains a boost of speed during a certain time, which is reduced over the duration and becomes simply a bit faster than the normal movement speed. This boost has a cooldown, which determines how much boost the player will have for their next sprint. The fundamental aspect of the sprint is that it allows players to move quickly but they must stop sprinting to recharge their sprint boost and to hit or catch the ball. The sprint boost uses a curve to determine its lerp towards the basic sprinting speed, which is refreshed depending on the current charge of that sprint boost.

Hitting



Hitting the ball will redirect the ball and increase its overall speed. The hit is not easy to control, although it does establish the new direction for the ball. The value of the hit becomes highest if the player catches the ball before hitting it, effectively allowing them to properly send it in another direction.

Grabbing



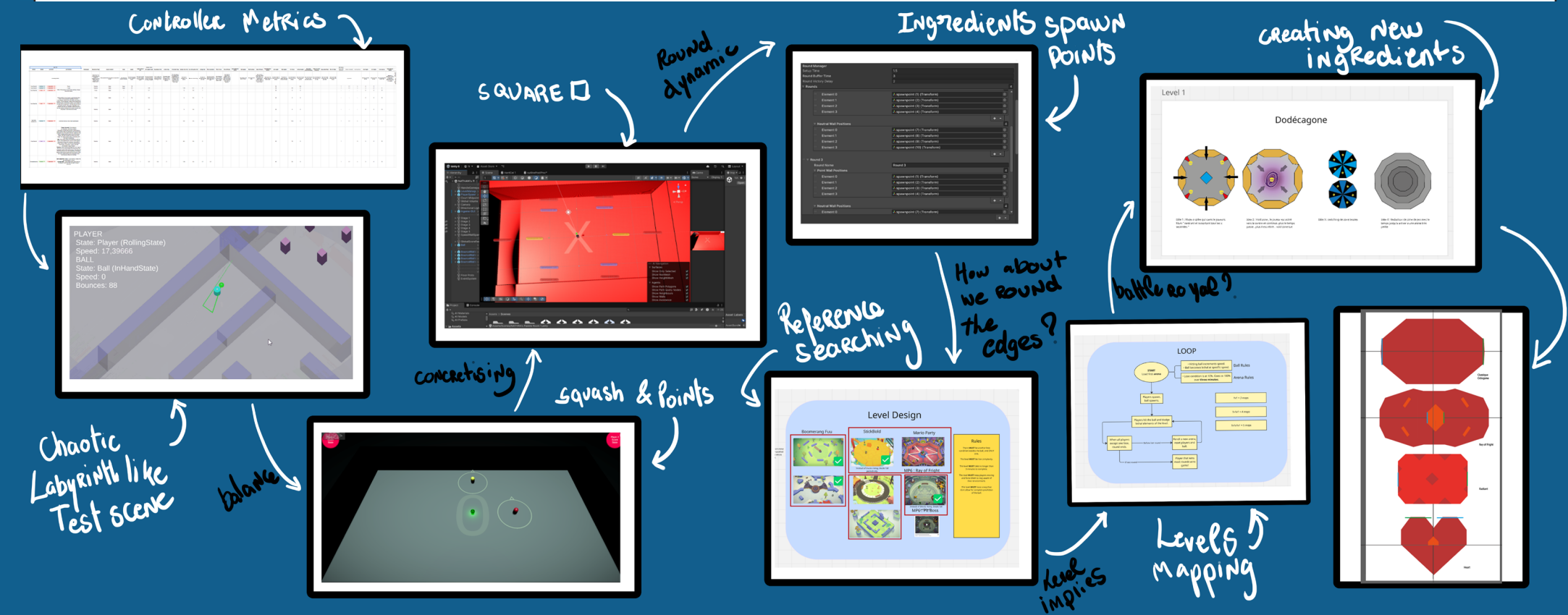
Grabbing or catching the ball will stop it and move it towards the player's front. The ball will remain grabbed until they either release the grab or the other player steals the ball by hitting it. When released, the ball will maintain the speed it was previously at, but can be redirected with a higher degree of control. Grabbing the ball seems like a good option until one realises that the other player can easily come in and hit the ball while it is within the grab. Thus, this mechanic imposes a big risk but also an equivalent reward in having better aim control.

Grabbing has a unique way of working, as the cone of detection for the grab diminishes in size as the button is held. Additionally, grabbing has a certain charge that is depleted after some time, which is recharged after letting go of the button and giving it a slight delay. The grab cone's size transition is also affected by a curve to smooth out the animation. These limiting factors allow the grab to be both strong but limited, since it also stops the player dead in their tracks, making the risk of not catching the ball have a big impact on their ability to retap it.

Sign & Feedbacks

SIGN	ACTION	FEEDBACK
Ball will impact the player (The bell is owned by the opposing player)	Player hits the ball	Player's attack effect starts, as the ball is hit a particle spawns on the ball where it was hit, plus a sound effect.
	Player grabs the ball	The ball is constantly move towards the front of the player, particles fizzle out to denote the duration of the grab.
	Player dodges the ball	The ball does not hit the player
Ball will impact the player (The bell is owned by said player)	Player is hit by the ball	A "stars" effect occurs on the player's head, while a sound effect denotes also the duration of the stun.
	Player hits the ball	Player's attack effect starts, as the ball is hit a particle spawns on the ball where it was hit, plus a sound effect.
	Player grabs the ball	The ball is constantly move towards the front of the player, particles fizzle out to denote the duration of the grab.
	Player dodges the ball	The ball does not hit the player
	Player is hit bythe ball	The ball goes throught the player not touching them

Level Design



Tutorial

Level Design and Tutorial Structure

We divided the project into three distinct levels. The first one is a tutorial map, where players learn the core mechanics of the game. On the ground, tutorial tags are displayed to explain how to use each mechanic. Since our gameplay features only a few core actions, we can afford to integrate the tutorial directly into the arena floor without overwhelming the player.

The mechanics introduced are: the Hit, the Grab and the Sprint and Initial Boost. To aid comprehension and mastery, the tutorial level is designed with simple rebound scenarios that help players learn how to anticipate trajectories. The layout also includes a large open area, allowing players to easily understand the movement of their character and get comfortable with the fast movement mechanics.

To encourage predictable and readable rebounds, the layout needed to rely on a familiar and basic shape. If players can instantly recognize the shape of the arena, they will more easily visualize how the ball will bounce off each surface. That's why we initially designed the level using a regular polygon as a base.

We first experimented with a rectangular layout, as it's universally understood and provides straightforward, intuitive rebounds. The ball would often bounce repeatedly between parallel walls, eventually reaching the opposite side of the arena. While this made for easy and accessible rebounds, it also removed the incentive for players to move around. Many would simply camp near their goal, waiting for the ball to come to them, with no need to anticipate or reposition.

This limitation led us to research games with similar gameplay, particularly those involving a defined arena and a bouncing projectile. These references guided the evolution of our level design toward more dynamic and engaging layouts.



References: Boomerang Fu

A multiplayer arcade-style game where players have to throw boomerangs, to which they can apply an effect in order to hit other players in the match. The interest in this reference revolves around the act of throwing an object and applying various direction modifiers to it.

Adding an effect is too complex for our current plans, but I'm more focused on their level design, which makes it possible for rebounds to happen when a boomerang, thrown at full speed, hits a wall. This rebound remains very limited, because the sharpness factor of the boomerang depends on its speed, and after hitting another wall, the boomerang's speed drops drastically.

So, the game prioritizes direct throws toward enemies, but still allows for rebounds, which, when combined with the game's various power-ups, give the whole mechanic its full meaning.

The Value of Rebound-Based Level Design

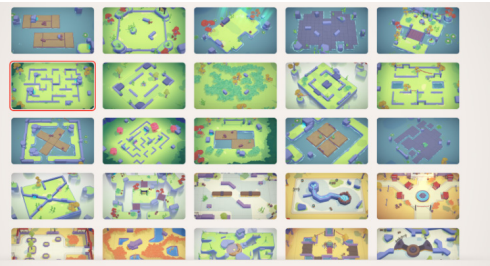
Their rebound system is very pleasant and fairly easy to calculate; it generally represents what we're looking for in terms of rebound application and rebound prediction during a throw.

So we can start to take an interest in their level design, and we find a certain consistency in their designs: most of the time, the arena shape is quite close to a square or rectangle, with level design components often symmetrical, and elements often positioned at 45° angles relative to the walls.

The goal of this system is to enclose the player in a shape they are familiar with, where they feel comfortable—they know the limits of the arena, they know there are no dead ends, and they are free to move around inside. The small walls placed at 45° angles to the main walls are used both to separate players and to vary the throws and rebounds.

References

Having a wall that goes against the direction of the base layout makes the rebounds off of it act as imperfections compared to the classic rebound possibilities in the level, and therefore introduces unexpected outcomes for the opponent. This is how the player is encouraged to play better and win the match



The Role of Rebounds in Our Project

The core principle of our game lies in scoring points in the opponent's goal. To do this, you need to aim well and take advantage of moments when the opposing player is out of position to defend their goal. The best way to put the opponent in a bad position is through rebounds—the more complex a rebound is, the less predictable it becomes.

All it takes is a miscalculation for a player to misread a rebound and end up losing. A player might try to intercept the ball, but if they misjudge the rebound, they could end up conceding a point.



In the Mario Party games, we often find fast-paced mini-games where the main idea is to throw a ball at each other or to dodge one, but the most relevant mini-games were Ray of Fright and Pit Boss from Mario Party 6.

References

Arcade Game Mario Party

In the Mario Party games, we often find fast-paced mini-games where the main idea is to throw a ball at each other or to dodge one, but the most relevant mini-games were Ray of Fright and Pit Boss from Mario Party 6.

Ray of Fright



Ray of Fright is a mini-game from Mario Party 6 that is played 1 versus 3, in which the solo player is placed in the center of the map inside a “laser cannon” and must shoot at the other players. The thing is, they don’t fire a single laser beam, but rather one in front of them and one behind them.

These lasers have the ability to bounce twice, which means the lasers cover a large part of the arena. The surviving players have to anticipate 6 simultaneous rebounds, making it a very difficult mini-game.

Ray of Fright led us to think about more complex level shapes than standard ones. A field with no flat sides allows for a maximization of “unpredictable” rebound play. And even though it’s too difficult in the level of Ray of Fright, it remains interesting to study the rebounding capabilities of an arena that only has hard-to-predict walls, while still maintaining total symmetry (a dodecagon).

Pit Boss



Pit Boss is also a mini-game from Mario Party 6, which also involves dodging an object that bounces off walls. In this case, a ball is launched and bounces off the walls, then other balls are launched, and all the objects bounce off each other. The players must survive until all the balls stop moving.

The difference compared to other games is that this time, the arena is circular, and the rebounds happen both on the outer walls and with the moving objects in the level. The dangers are located around the players, as well as within the arena.

This approach allowed us to think about circular level designs, but the prediction became too unpredictable, so we quickly abandoned the idea of a circular arena. The idea of multiple objects also came to us, but we preferred to focus on a single ball in order to make it easier to see and predict the ball’s behavior.

Reference Research Results

After analyzing several ball-based games in which gameplay revolves around rebounds, we identified a certain interest in having level designs based on a mix between straight faces and faces rotated at 45°, in order to maximize the ability to predict certain rebounds, while others remain more difficult to anticipate.

There is also value in placing walls at the center of the arena to avoid empty space and to add more complex and rewarding rebound possibilities for players.

It is therefore interesting to move toward complex yet symmetrical shapes, with a minimum of 8 sides and not exceeding 12 sides.

To meet all these requirements, we explored diamond-shaped forms, which are always parallel, while still being fairly complex and classic in shape.



With these shapes, many of them could have been usable because even behind their circular forms for some, they are still transformed to be flat. So, we really liked the concept of the Radiant, which keeps the simplicity of the rectangle while adding some faces at 45°, as was sought earlier.



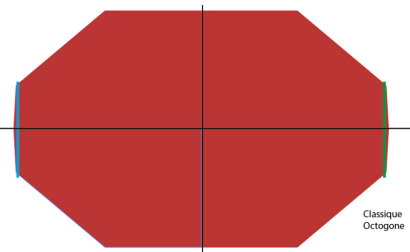
But this arena was too flat, too empty, and there wasn’t enough interest in the rebounds. So, I added spikes to bring in those irregular and hard-to-predict angles while keeping the principle that rebounds use the spikes as a main tool to direct the ball toward the enemy’s goals.

Arenas

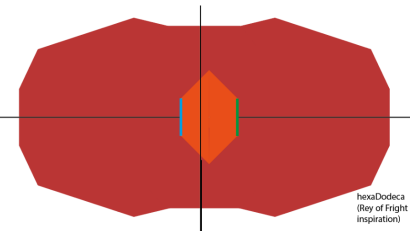


With this arena, the spikes all serve as dangers for the players’ goals from multiple directions, making it a level with a high rebound potential and therefore very dangerous for each player. It will be a perfect final level.

Starting from this principle, I was able to rethink an octagon, which would be a perfect level for a tutorial and level 1. The goals would be face to face, with no objects in the center to block passage. Players could learn to play and aim with the help of tags on the ground that teach them each control.



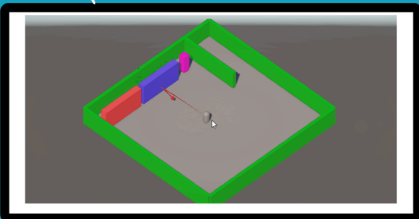
Then, to find our final level, we wanted to start with a unique shape inspired by the Ray of Fright level, turning it into a hexadecagon (16 sides) to allow for very unusual geometric shapes while continuing to use the pattern of straight walls and walls at 45°. But it seemed too similar to the tutorial level, so to change it, we added a prism in the center and placed the goals also in the center, in order to encourage players to rely on rebounds and move around in order to score.



Game Programming



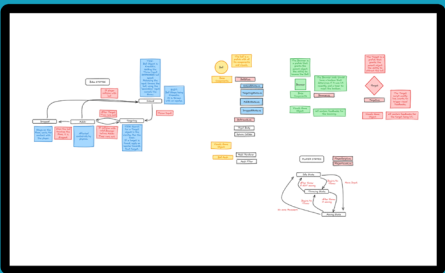
Metrics Variables



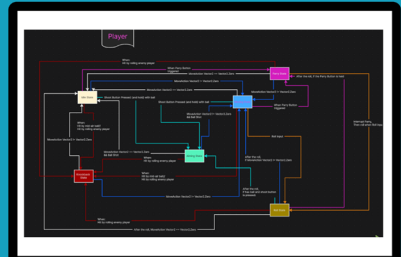
First Bounces

• cc
• Ball Behavior

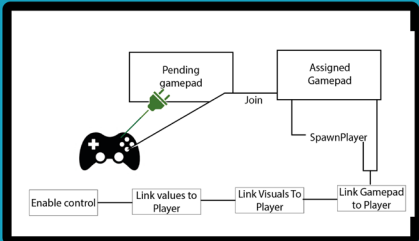
Retouch Camera



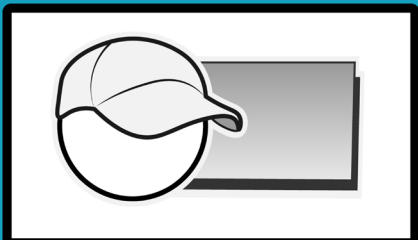
Integrating Features



States Machines



Multiplayer



Menu & UI



Bug Fixing

Basics

Technical Development

For this project, we made use of Unity v6 with the Universal Render Pipeline (URP).

We divided our work into different aspects: The player's control, the ball's different states, the level states that manage the main loop and a game state machine that manages all interfaces.

Within the editor, we worked with the new Input System which provides versatility and more straightforward ways of creating code. We also used JetBrains Rider 2024 with the plugin Github Copilot. We used these tools to streamline repetitive code and support for unknown errors and bugs, as well as to improve and refine code so that it takes less space.

Physics

We use Unity 6's base physics, with the default settings. The only change we worked with was changing the Default Contact Offset from 0.2 to 0.01, so that contact is less recursive. This prevents several bugs with the collision of the ball with the player and is more precise.

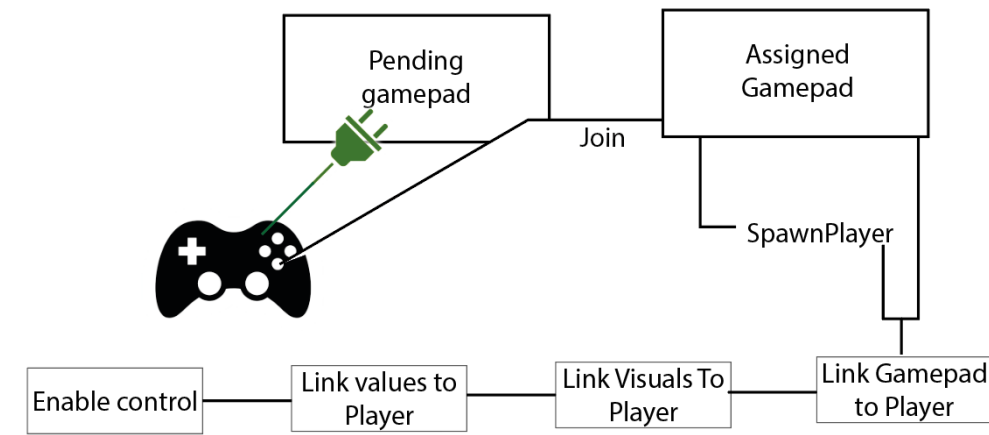
All game objects that exist in-game have a Rigid Body component that is used for all calculations.

Movement for the player consists of applying changes to its velocity by multiplying their speed variable against the Vector 2 generated from using the left joystick on the gamepad. The ball works in a similar way, where instead of using the Vector 2 we simply apply the direction of the player's forward vector and a force that uses both a modifier from the player (derived from charging the attack and the base power it has) and the ball's current velocity. This way, the ball never slows down.

After these two elements we also have the obstacles that act as walls. All of these objects have a physics material, which allows them to have no drag or roughness and allow the ball to bounce off them without losing any speed.

Technical Development

Our multiplayer connection script happens in 3 steps. When I connect a controller to my Unity, the system assigns the controller as a 'pendingController'. From there, I can identify who the gamepad belongs to by using Unity's GamePad HashSet. After that, I can start linking my controller to all its dependent objects.



The purpose of this feature is also to be able to access, thanks to the recorded HashSet of each controller, which controller has been connected. So, if a controller happens to connect, we can recognize the new pending Gamepad as the Assigned Gamepad and assign the controller to the player.

Ball States

DROPPED STATE: The ball begins with this state at the start of the game.

CAUGHT STATE: Whenever the player catches the ball. During this period, the ball will move towards a spot in front of the player that we call the player's hand. It will continuously check if it is already on that spot, moving towards it using an animation curve. If already on the spot, it will not move anymore. The caught state will end if the ball is hit or the player's catch charge is depleted. In both cases, the ball will move away from the player (if hit it will use the hit direction, if the catch charge is depleted it will move in the direction that it was moving when caught).

FLYING STATE: In the flying state, the ball simply travels. It can be considered as the "neutral" state of the ball. It is not affected by gravity, and will continuously move around. During the start of the flying state, the ball checks if the current speed is less or more of the last recorded speed, so

HIT STATE: This state has a short duration, starting when the ball is hit and swiftly ending. This state makes the ball have no collisions, which prevents the player from hitting themselves with the ball if they were to hit it from behind.

SOME IMPORTANT EVENTS, OnNeutralBounce: This event triggers when the ball bounces off a wall. OnBallFlight: This event triggers when the ball flies.

OnBallCaught: This event triggers when the ball is caught. OnHit: This event triggers when the ball is hit. OnHitStateStart: This event triggers when the hit state begins. CaughtStateEnd: This event triggers when the catch state ends.

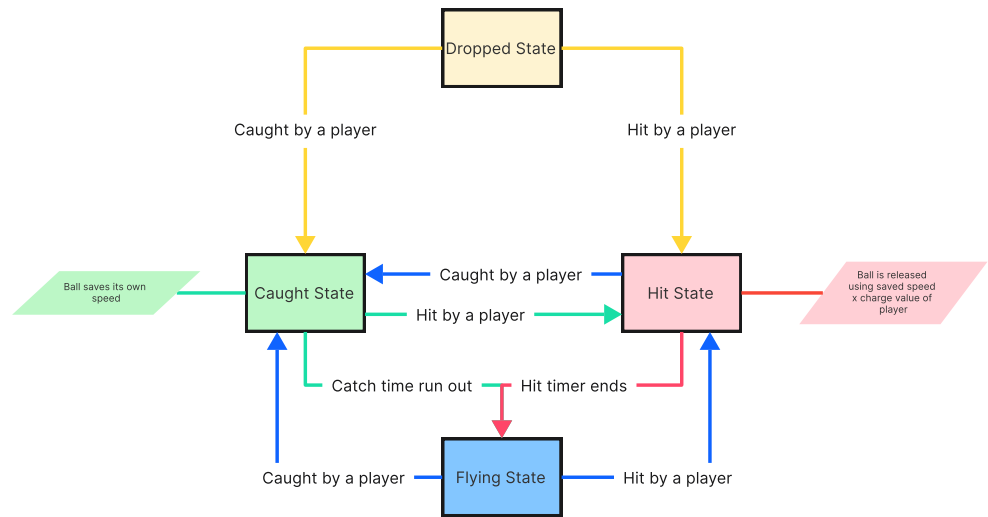
Ball Propulsion Variables: The maximal and minimal **speed** and **height** that can be attained by the ball

Ball Physics Variables: **Linear damping** and **Mass** when flying and grounded

Immunity Time : PlayerImmunityTime: How long does the ball ignore the player's collisions.

BallOwnerPlayer: Determines what player is the owner of the ball.

BallSpeedFloor: The ball's speed's minimum value, which is increased every time the ball's linear velocity is increased.



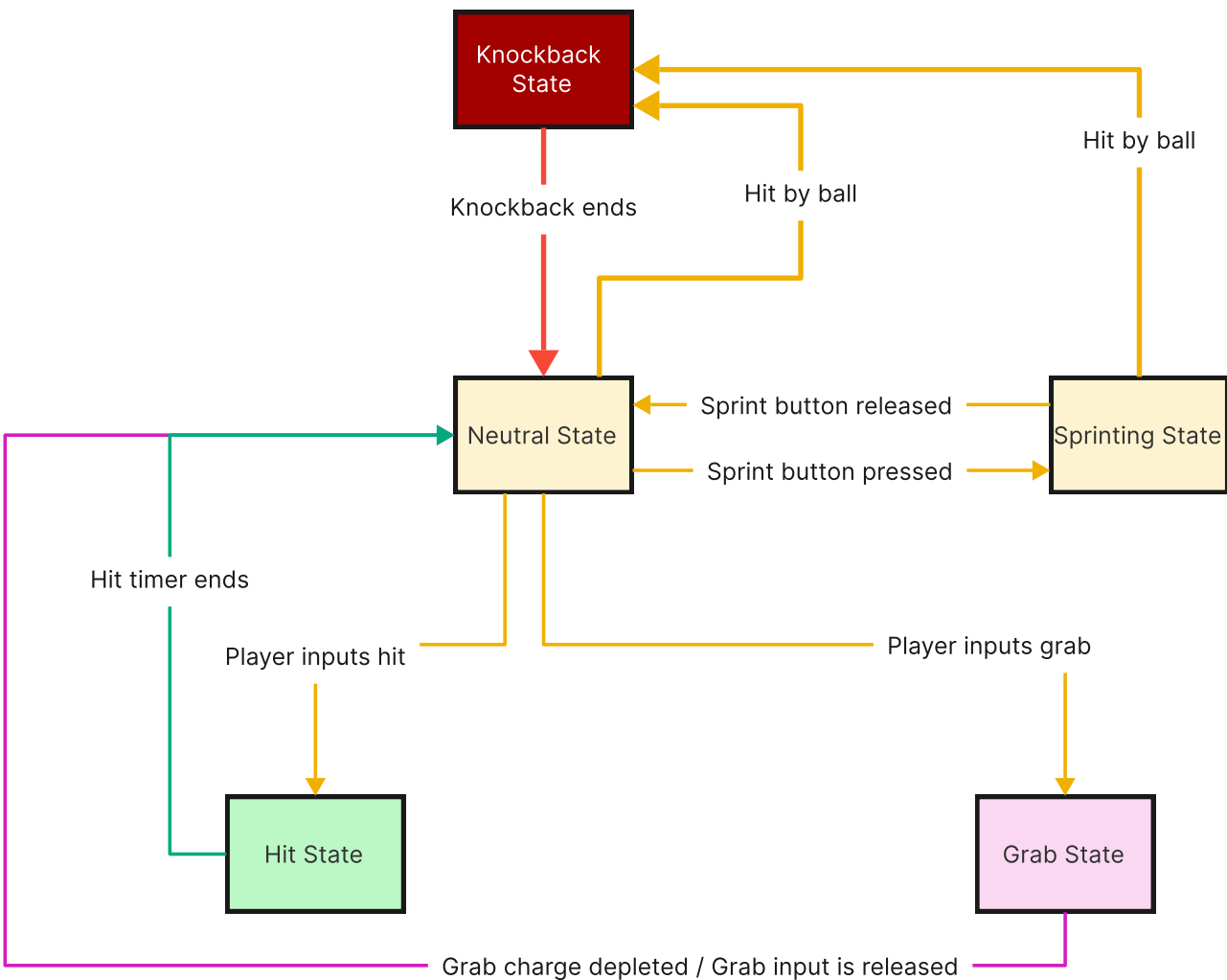
Player State

NEUTRAL STATE: The player's basic state. On neutral, the player can only move around and swap into the other states depending on their inputs.

SPRINTING STATE: The player's sprinting state, which similarly to the neutral state uses the move function with the sprint boost and the sprint speed, effectively making the player faster. The player cannot hit or grab the ball while sprinting, but it does stop the movement applying the inputs this way.

HIT STATE: The player swings their bat. They can move freely in this state.

GRAB STATE: The player starts pulling the ball towards themselves if it is in range. This state has an established duration depending on the player's grabbing charge, and a range that is reduced depending on this charge value.



Multi-Scenes

Our game works on a level progression system, so in order to avoid as many problems as possible when changing scenes, we decided to go with a multi-scene management system. The purpose being to be able to use a Level Manager script in all the scenes, without having to reconfigure it for each scene we want to use. So we have a container scene, which contains all the necessary objects for each level — the Level Manager, the UI, a floor, and lighting. When switching to the container scene via the Start menu, the Level Manager will automatically load the other necessary scenes. To be able to load the correct scenes, I use a list of levels that I set from my Game Manager. I load them all and deactivate all the GameObjects in each scene directly in order to activate only the GameObjects of the scene I want to play. I use my Build Index Scene to access each of my scenes and to be able to activate the GameObjects of the scene I need to play. This way, I can call this function each time I want to change levels to activate the GameObjects of the next level.

Container	Build Index
Level 3	1: Octogone
Level 2	2: Ray of Fright
Level 1	3: Radiant
All activated scenes	
Order of scenes to play	

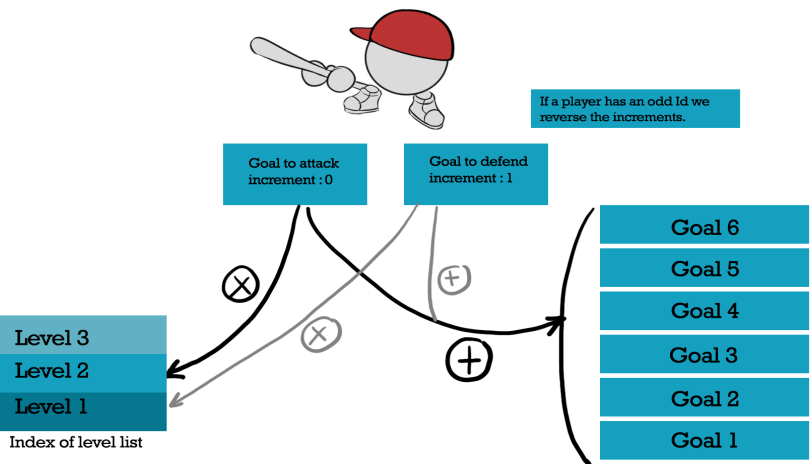
Once my scene is properly loaded, I can start configuring my level so that it adapts to the loaded scene.

The biggest part of the level configuration is the goals. Each player must be assigned a goal to defend and a goal to attack. Since each level has different shapes and therefore different goal positions, the variable goal-ToDefend and the variable goal-ToAttack must be adapted each time a level is loaded. To do this, I use a mathematical function. I make it so that, when my goals are instantiated, I add them all to a list of goals, and I work with this list and my current level to find the correct ones.

Following this calculation, both players will always have an attacking and defending goal at each level change. However, this script has flaws — if a goal is spawned in excess, it becomes misaligned and incorrect, and we cannot allow unbalanced matches. In the worst case, it would be possible to have an extra player on one team, but this isn't really a problem since we can only have a maximum of two players.

Once done, each player knows their two goals — the one to attack and the one to defend. Knowing who each goal belongs to allows us to score only in the 'goal to attack'.

Each goal keeps track of the number of points it has received and allows us to know the progress of the match.



Direction Artistique Visuelle



Intention

Intentions

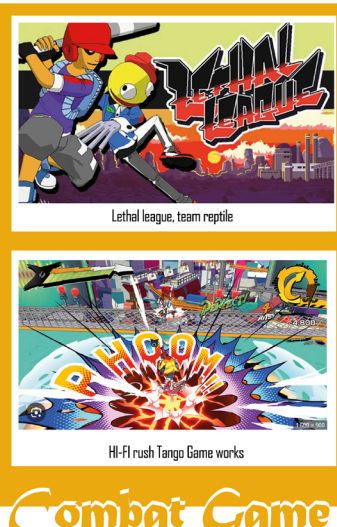
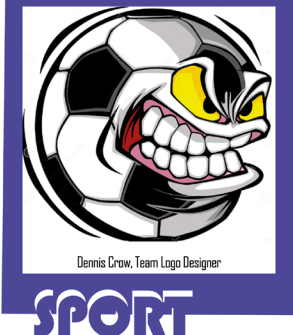
For Blitz Ball, we wanted to have an aesthetic that combines simplicity and readability with something a bit more fun and goofy, allowing us to convey our party game vibe.

To do that we took a large range of references that could be divided into 3 main aspects:

- A cartoonish esthetic expressing itself through round shapes, plain colors and goofiness in movement.

- An inspiration in childhood with reference evoking ours with the intentions of smiling at it. In this aspect we did some research on pop art which adopted a similar dynamic by infusing pop culture into traditional arts. We got inspired in shapes, pattern and color from this movement

Moodboard



Character

Character Design Constraints

For the character, we are working with several constraints:

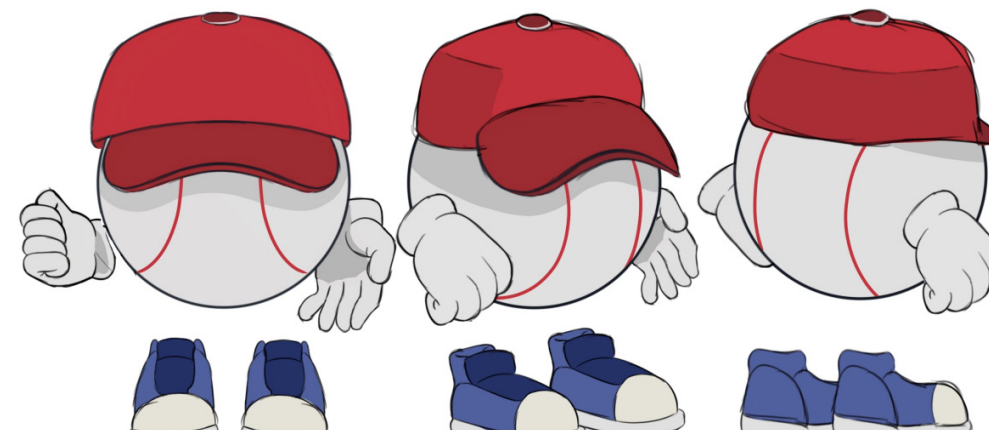
Visual Feedback: The character must clearly convey a number of key visual cues—its forward-facing direction, hitbox and hurtbox, movement animations, and attack actions. It also requires a tool or element used to strike or attack.

Multiplayer Adaptability: Since the game is multiplayer, the character must be available in multiple variations that are easily distinguishable from one another during gameplay.

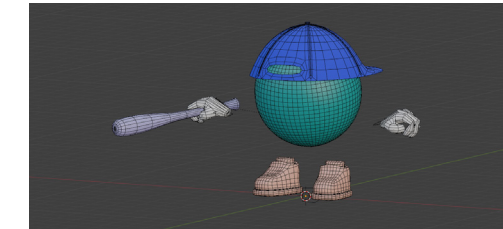
Shape and Readability: We chose to work with a round shape for the character. However, we had to carefully consider how to create visual contrast between the character and the ground or the ball to ensure good readability.

Production Constraints: Due to a strict time constraint, we opted for simple-to-produce assets to streamline the development process.

Concepts Arts



3D Model



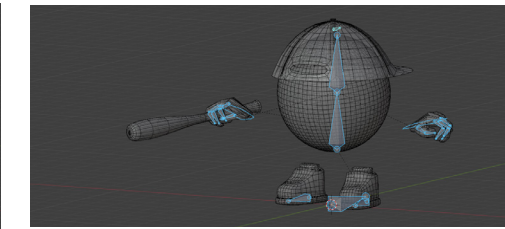
The model is fairly simple. To make it easier to create multiple character variations, we organized the UVs and materials so that some can be reused across all characters, some only require a color change, and others can be replaced with different shapes.

Textures



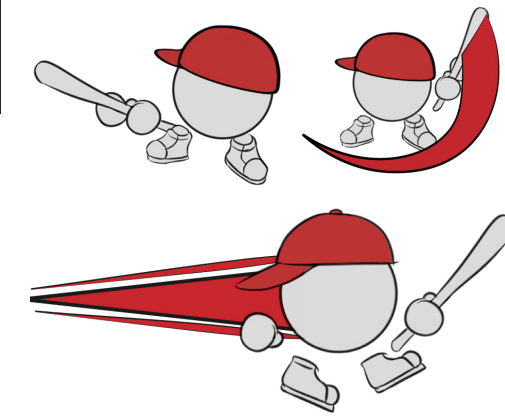
The textures are primarily made up of flat, matte colors. We added stitching details using hand-drawn paths to give them a slightly “trashy paint” or rough, handmade aesthetic.

Animation



We created a simple humanoid rig. Since we don't use arms or legs, the hands and feet are parented directly to the spine. At the top of the head, we added a bone that controls the cap.

The animations are very simple. We play with exaggerated disarticulation to create a comical effect. We also add a bouncy feel by stretching and compressing the two central spine bones.



Environment

Arena Design Constraints

For the Arenas, we are working with several constraints:

Visual Neutrality: The arena must not get in the way of what's happening in it. We made sure to use light colors with no pattern, especially on the ground.

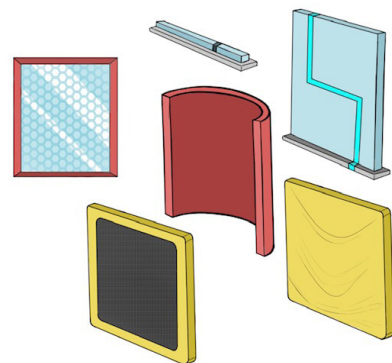
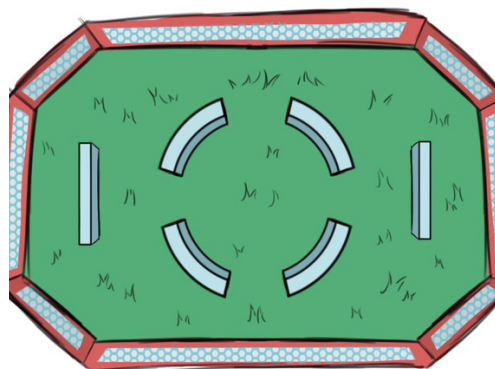
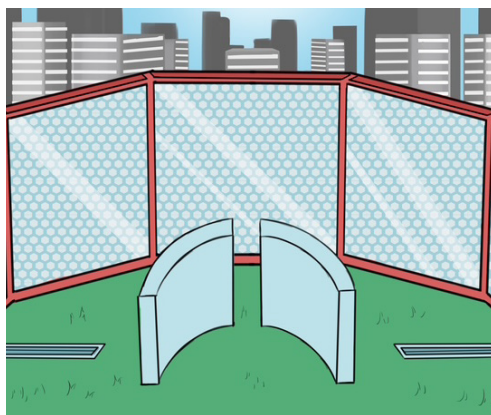
Highlight LD: Each arena has its own element that should be highlighted because they cause the determining parts of rebounds in the level.

Shape and Readability: We chose to work with a square shape to contrast with the character.

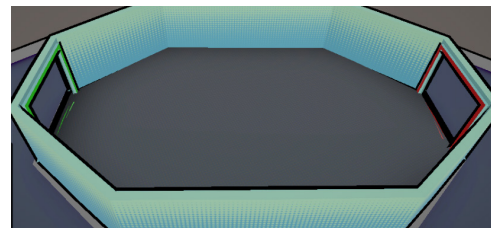
Responsivness: We want the arena to give some feedback of the current game state so it participates to the narration intradiegetically

Production Constraints: Due to a strict time constraint, we opted for simple-to-produce assets to streamline the development process.

Concepts Arts

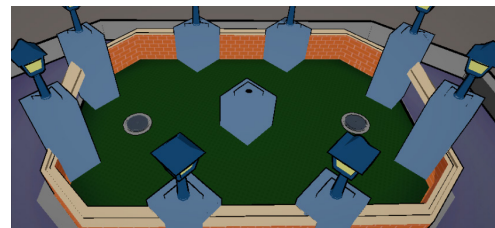


Octogone



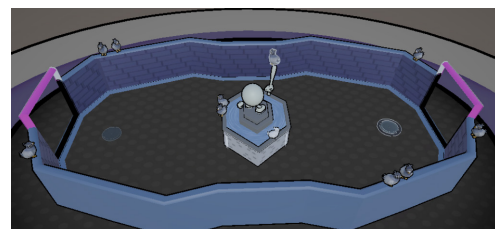
The octogone is our tutorial level. We chose to show the inputs of the game on the ground. We also signaled the player's goal apparence with color, this is done in a uselessly complicated way but which allows us to dynamically allow us to assign a color because said player color is randomly assigned at the beginning of the game

Radiant



Working on the Radiant Arena we encountered the problem of how to balance decoration without creating blind spots.

Ray Of Fright



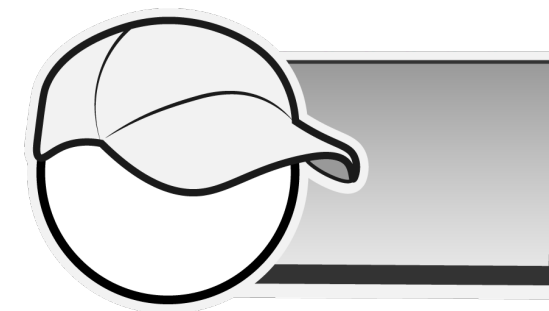
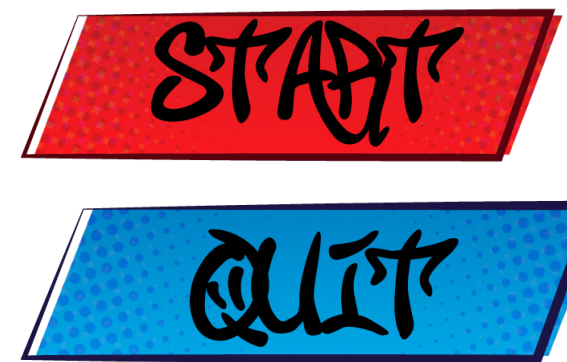
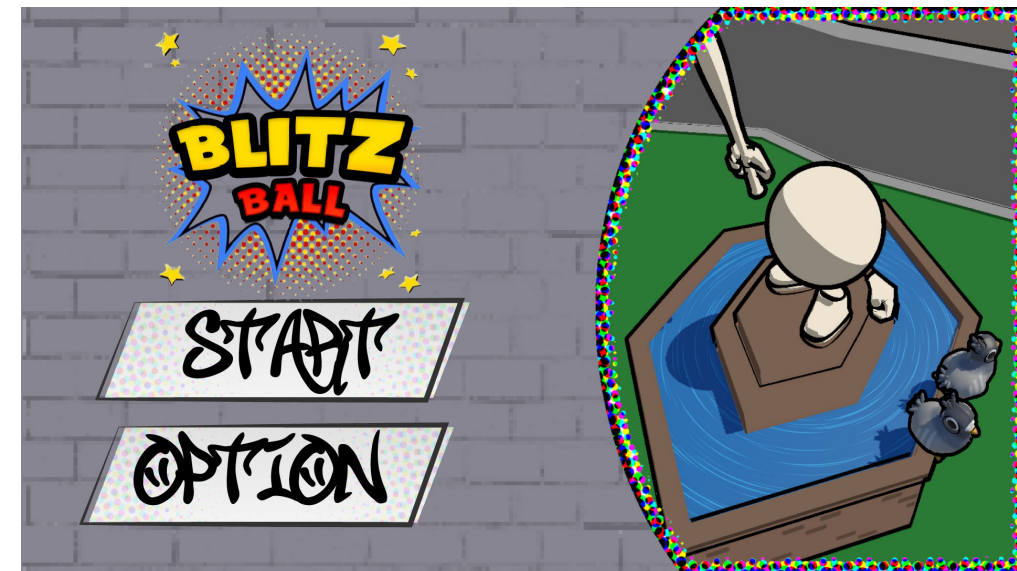
With this arena we base ourselves on a fountain square, generally a place that would host an improvised game of ball as it is frequented by children. We also invited its full-time resident the pigeon

UI

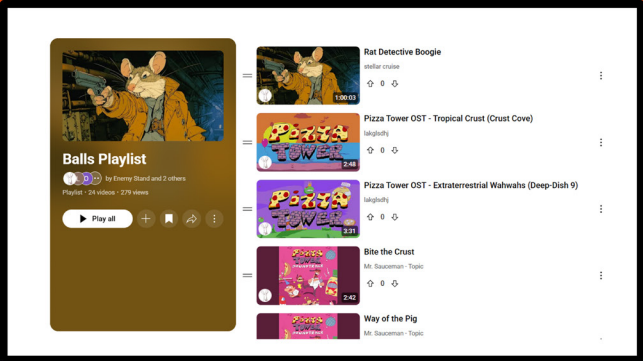
The challenge of the interface was to translate the aesthetic of Blitz Ball (cartoon, pop, and street art) into a UI that is readable and intuitive. The initial iterations above were more sport-oriented than street art, hence the presence of a smoother style representing a basketball.

As production progressed, the needs of the art direction shifted toward a less polished, less vector-based aesthetic. Brighter colors were used, and the presence of an expressive icon for the character was added... The overall shape became a frame with a drop shadow, while still keeping a character to represent the player (and the logo was changed to black and white to make it easier to integrate the UI into the game).

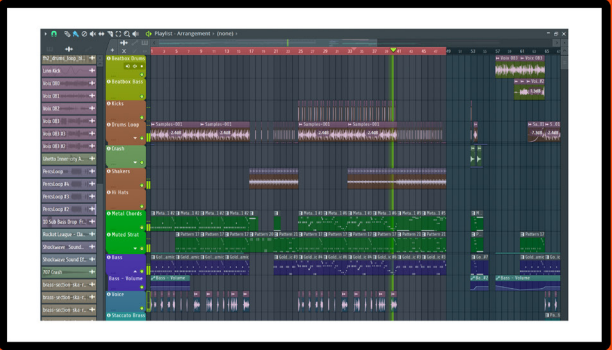
In its most recent versions, the street art style took precedence over the 'cartoon' aesthetic, with the use of a wall as the background to give the impression that the buttons are painted over it (while still maintaining some readability in their presentation). To strike a balance between street art and 'cartoon,' the final version features a simplified logo and background, with brighter colors.



Direction Artistique Sonore



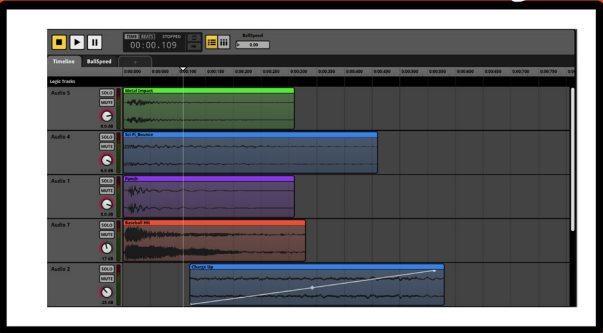
Collecting
References



Sound
Designing
&
Mixing

Sound Name	Description	Category	File Name	Usage
Ball, Sound	Ball rolling on a flat surface	Sound	Ball_01.wav	Used in the game
Ball, Sound	Ball rolling on a rough surface	Sound	Ball_02.wav	Used in the game
Ball, Sound	Ball rolling on a wet surface	Sound	Ball_03.wav	Used in the game
Ball, Sound	Ball rolling on a wooden surface	Sound	Ball_04.wav	Used in the game
Ball, Sound	Ball rolling on a metal surface	Sound	Ball_05.wav	Used in the game
Ball, Sound	Ball rolling on a glass surface	Sound	Ball_06.wav	Used in the game
Ball, Sound	Ball rolling on a stone surface	Sound	Ball_07.wav	Used in the game
Ball, Sound	Ball rolling on a brick surface	Sound	Ball_08.wav	Used in the game
Ball, Sound	Ball rolling on a concrete surface	Sound	Ball_09.wav	Used in the game
Ball, Sound	Ball rolling on a asphalt surface	Sound	Ball_10.wav	Used in the game

Sound
List



Integration

Intentions

Blitz Ball follows a sound art direction based on two core pillars: the sensation of power and speed, and the creation of a stylized urban atmosphere. These two directions aim to enhance player satisfaction while maintaining strong consistency with the game’s visual universe.

Sensation of Power and Speed :

One of the key intentions in the sound design of Blitz Ball was to convey a growing sense of intensity and dynamism with each ball exchange. Every strike with the bat had to be not only visible but also felt through the sound.

Impact sounds were designed to reflect the increasing power as the ball gains speed. The more it accelerates, the more intense and layered the sound becomes, using evolving elements like sharp transients, percussive lows, and controlled distortion to emphasize the rising force.

The ball’s sound changes depending on the type of surface it hits, whether it’s the bat or the wall, and based on its current velocity.

Special attention was given to spatialization and reverb to strengthen the sensation of speed and the perception of distance in a 3D environment.

As the ball moves across the field, a series of dynamic and synthetic whooshes underline the growing tension and the anticipation of impact. These elements help to translate speed and danger directly through sound.

Creation of a Stylized Urban Atmosphere :

Beyond raw impact, Blitz Ball is rooted in a strong urban identity inspired by 1980s American street culture.

The game incorporates textures and layers reminiscent of street environments like urban playgrounds and improvised tournaments. Background ambiances feature distant crowd chatter, voices, horns to bring life and spontaneity to the experience.

This atmosphere supports the visual and gameplay intentions by immersing players in a vibrant setting that feels playful and grounded at the same time. The soundscape becomes an active part of the game’s personality, reinforcing its fun, energetic, and competitive nature.

The sound design of Blitz Ball was conceived to serve the action, amplify sensations, and immerse players in a coherent and exciting world. Every audio element contributes to the physical impact and the urban flair of the game, merging both into a powerful and unified experience.

References

One of the primary sound design references for this project was Lethal League Blaze. This game stood out for its dynamic and impactful audio work, particularly in how it handles ball interactions and velocity-driven sound variations.

A key element of inspiration was the way Lethal League Blaze communicates speed and intensity through sound. As the ball accelerates, each successive impact is accompanied by increasingly aggressive and exaggerated audio feedback. This creates a strong sense of momentum and power, which enhances both the gameplay and the player’s emotional response.

The sound of the bat hitting the ball was also a major influence. In Lethal League, the impact sounds are carefully crafted to emphasize the force of the hit, using a combination of punchy transients and crisp metallic textures. These sounds are not only satisfying but also help convey the kinetic energy behind each strike.

Additionally, Lethal League Blaze excels in using audio cues to signal when the ball reaches extreme speeds. High-pitched whines, layered distortions, and time-stretched elements are integrated into the mix to create a sense of chaos and danger. This inspired a similar approach in our project, where the sound design evolves alongside the ball’s velocity to communicate escalation clearly to the player. By studying and drawing inspiration from Lethal League Blaze, the sound design in our game aims to achieve a comparable level of energy, responsiveness, and clarity, ensuring that every hit feels powerful and every high-speed moment sounds intense.



Lethal League by Team Reptile



Jet Set Radio By SEGA

Another major influence on the sound design of our project was Jet Set Radio, particularly in terms of establishing a vibrant and immersive urban atmosphere. Jet Set Radio is known for its unique audio identity, blending energetic music, street-level ambience, and stylized sound effects to create a dynamic cityscape. The game’s soundscape plays a crucial role in conveying its rebellious, fast-paced tone and underground culture, which served as a strong inspiration for the mood and setting of our own game. In our project, the goal was to capture a similar sense of urban energy—gritty, lively, and rhythm-driven. The influence of Jet Set Radio is especially present in the background ambience:

distant traffic noise, subtle crowd murmurs, graffiti spray sounds, and mechanical elements like trains or neon buzzes all contribute to creating a world that feels alive and grounded in a modern, street-style environment. Moreover, Jet Set Radio inspired the layering of environmental sounds with music in a way that feels cohesive rather than chaotic. The game integrates its soundtrack with environmental cues to keep the momentum going without overwhelming the player. This approach influenced how we crafted transitions between gameplay and ambient moments, ensuring the atmosphere remains rich and engaging throughout the experience. By taking cues from Jet Set Radio, the sound design in our game strives to reflect a similarly bold, street-inspired identity that enhances the visual style and supports the energetic gameplay.

References



Knockout City by Velan Studios

Knockout City served as a key reference in shaping the core mechanics of ball-related sound design in our game. Its approach to impact sounds, ball textures, and speed dynamics provided a solid foundation for our own audio strategy. One of the most striking elements in Knockout City is the way it handles ball interactions. Every throw, catch, and collision is supported by distinct, satisfying sound cues that clearly communicate both force and gameplay feedback. This inspired a detailed study of how to represent the physicality of a fast-moving object purely through sound. In particular, the design of ball textures in Knockout City—from the lightest glide to a full-speed

smash—helped guide our own search for materials and tonal qualities. The ball in our game required its own unique sound identity, and the richness of Knockout City’s layers (soft friction, air movement, and sharp impacts) played a major role in defining how we approached this. Additionally, the way Knockout City builds a sense of acceleration and anticipation through audio was highly influential. Subtle increases in pitch, additional layers of wind or whoosh, and evolving impact transients all contribute to the perception of speed, tension, and power. These techniques were directly adapted to emphasize the increasing velocity of the ball in our game, making each hit feel more dangerous and intense. Lastly, the impact of the ball against the player in Knockout City—often exaggerated, expressive, and stylized—offered a perfect example of how to make contact feel visceral without relying on visual cues alone.

In our game, similar attention was given to hits and collisions, ensuring they are both readable and impactful through sound alone. Drawing inspiration from Knockout City allowed us to refine the expressiveness and clarity of our sound design, especially in gameplay-critical moments, enhancing both feedback and immersion.

Sound List

The Sound List is a comprehensive table used to manage and track every audio element in the game, from sound effects to ambient layers and music. It was designed as both a creative and technical tool, serving as a central reference point for the entire team throughout development. Each row in the table corresponds to a specific sound asset, and the columns are organized to provide essential information for both sound designer and programmers. The first column identifies the name of the event or sound. This is followed by a category column that classifies the audio as Music, SFX, or UI, helping to clarify its role in the game. The action column describes what in-game event triggers the sound, allowing the design and development teams to synchronize implementation more effectively. The description column offers a brief but precise explanation of what the sound is and what it’s meant to convey in context. Another column indicates how the sound is played, whether as a one-shot or loop, which helps determine how it should be integrated into the audio engine. A key part of the Sound List is the progress status column, which shows whether each sound is in concept, production, or final stage. This has been particularly helpful for tracking deliverables and managing priorities during development cycles. The variable column outlines any gameplay-related parameter that might affect the sound. For example, the ball’s speed, the number of rebounds, or player states might alter the pitch, layering, or intensity of a given sound. This allows for dynamic audio behavior that reflects gameplay feedback in real time.

2	Hit_Sound	SOUND EFFECTS	Le joueur frappe dans le vide.	Le son est différent en fonction du maintient du bouton.	ONE SHOT	75%	Temps de maintient du bouton hit. Float	PlayerScript: OnPlayerHitReleased	Le son se lance lorsque le joueur frappe le ballon.
3	BallTouched_Sound	SOUND EFFECTS	Le ballon est touché lors du hit.	C'est un son de percussion plus ou moins puissant en fonction du maintient du bouton hit.	ONE SHOT	75%	Temps de maintient du bouton hit. Float	PlayerScript: OnBallHitByPlayer	Le son se lance avant que le ballon soit hit.
4	KnockBack_Sound	SOUND EFFECTS	Un joueur est touché par la balle.	Son de percussion plus ou moins puissant en fonction de la vitesse de la balle.	ONE SHOT	50%	Vitesse de la balle. Float	PlayerScript: OnPlayerHitByBall	
5	Dash_Sound	SOUND EFFECTS	Lorsque le joueur appuie sur un bouton pour dasher.	/	LOOP	50%	/	PlayerScript: OnPlayerDash	Le son Dash ne se lance pas.
6	ClassicBounce_Sound	SOUND EFFECTS	La balle rebondit sur une surface classique.	Le pitch augmente à chaque rebond.	ONE SHOT	75%	Quantité de rebond sur une surface classique. Integer / Vitesse de la balle. Float	Ball SM: OnNeutralBounce	Le son se lance lorsque le joueur frappe le ballon.
7	ScoringBounce_Sound	SOUND EFFECTS	La balle rebondit sur une surface de score.	/	ONE SHOT	0%	?	?	
8	PlayerSpawn_Sound	SOUND EFFECTS	Un nouveau joueur rejoint l'arène.	/	ONE SHOT	75%	/	LevelManager: OnPlayerSpawn	Le son PlayerSpawn ne se lance pas.
9	Player2Spawn_Sound	SOUND EFFECTS	Un deuxième joueur rejoint l'arène.	/	ONE SHOT	0%	/	?	
10	Grab_Sound	SOUND EFFECTS	Le joueur attrape la balle.	/	ONE SHOT	75%	/	PlayerScript: OnPlayerCatch	
11	Navigate_Sound	UI	Le joueur se déplace dans le menu.	/	ONE SHOT	0%	/	?	
12	Selection_Sound	UI	Le joueur sélectionne dans le menu.	/	ONE SHOT	0%	/	?	
13	Return_Sound	UI	Le joueur retourne dans le menu précédent.	/	ONE SHOT	0%	/	?	

Another important column is the event name, which refers to the precise technical event or parameter in the engine where the sound is implemented. This ensures that both audio and programming teams are aligned when integrating the assets. Finally, the review column is used to document whether a sound is functioning correctly in-game or needs adjustments. This has made it much easier to communicate issues between sound designers and programmers and to keep track of ongoing changes or bugs.

What does the sound produce in the game?

On the sound level the player must feel something almost uncontrollable, an almost instantaneous speed. The player must imagine that the ball has reached a speed that would allow him to destroy everything. The ball changes its speed to the point of exceeding the wall of sound, the sound effects during these moments must be well defined the state of the ball and what it can do.

The parameters

The different parameters will vary mainly with the speed of the ball. The sound of the impacts, the sound of the ball passing from one gear to another. Subtle spatial effects like reverb or distance filters can be added to enhance immersion and depth. The resulting sound is tested across various gameplay situations and camera angles, ensuring it remains consistent and satisfying regardless of the action’s context.

Process

Step by step

Creating the sound of the player hitting the ball :

- 1) Sound analysis and preparation. Keep in mind the different movements, objects used and materials of interfering elements with the sound of hitting. Here, a baseball bat and a ball. The movement is a kind of swing. The baseball bat is made of metal and the ball will be made of rubber.
- 2) Creating the core sounds. Instead of aiming for perfection from the start, the focus is placed on constructing the essential sound elements. A swoosh is created to represent the swing of the bat and convey motion speed. The impact between the bat and the ball is designed with a sharp, metallic quality mixed with a rubbery texture. A quick whoosh is used to signal the ball’s sudden acceleration after the hit. These foundational elements are tested directly in the game to ensure they fit naturally within the gameplay rhythm and energy.

- 3) Adding details. With the base in place, additional layers are added to enhance realism and texture. Subtle metallic resonance helps emphasize the material of the bat. A faint acoustic bounce suggests the elastic response of the rubber ball after the impact. Since the intensity of hits increases as the game progresses, dynamic sound layers are created to reflect this evolution—such as mild distortion or tonal changes for high-powered strikes.

- 4) Mixing and integration In this final phase, the sounds are refined and balanced to sit well within the overall game environment. The volume of each element—swing, impact, and launch—is adjusted to maintain clarity and cohesion. Equalization is applied to give each sound its place in the frequency range. Subtle spatial effects like reverb or distance filters can be added to enhance immersion and depth. The resulting sound is tested across various gameplay situations and camera angles, ensuring it remains consistent and satisfying regardless of the action’s context.

Music

The music in the game was composed by Théo Bucamp, the first goal to reflect its fast-paced, competitive, and energetic nature. Inspired by urban culture, the soundtrack adopts a distinctly urban style, drawing heavily from hip hop aesthetics. Elements such as vinyl scratches, punchy drum loops, and rhythmic sampling techniques were used to give the music a raw, street-level feel. This musical direction helps reinforce the visual identity of the game and complements the setting, which is grounded in dynamic city-inspired environments. Beyond its stylistic choices, the music plays an active role in enhancing the player’s experience through real-time interaction with gameplay events. One of the key interactive features is the pitch modulation of the music, which changes when the score shifts during a match. As one team gains the upper hand or stages a comeback, the music subtly increases in pitch. This technique adds urgency and emotional intensity, reinforcing the competitive stakes and signaling to the player that the momentum is changing. It creates a high-energy moment that feels musically and mechanically unified.

In addition to pitch variation, the soundtrack evolves progressively across rounds. At the start of the game, the music begins with its core rhythm. As the match advances and rounds change, new instrumental layers are introduced. First, a bass layer is added, bringing depth and drive to the track. Then, an electric guitar enters, injecting attitude and flair. Finally, a layer of trumpets is introduced, giving the music a bold, triumphant quality that marks the climax of the game. These added layers serve both as musical escalation and as gameplay feedback, signaling progression and raising the emotional stakes. This dynamic music system was designed not only to match the game’s visual and mechanical intensity but also to make each match feel like a unique musical performance. It ensures that the soundtrack remains fresh and responsive, enhancing the immersion and reinforcing the rhythm and flow of the gameplay.

