SPEAR - GDD - GOLD



Bynyhonia.

BENRHANNOU - DERIO - DUFOUR - ESMAN - GILLE - GRAND - LARGUIER - LEPRÊTRE - MANSOIS - PAUVRASSEAU - ROUX - VERNET - VREL

Changelog

- 03/10/2019 Document creation (Martin, Simon, Corentin) Added Overview, Game Structure, 3C drafts
- 06/10/2019 (Corentin) Character Abilities, Character Metrics draft
- 08/10/2019 (Martin, Simon) Work on Game Structure, Win and Lose conditions, LD Bricks
- 14/10/2019 (Corentin) Character Metrics, Approx. Vertical Range
- 15/10/2019 (Martin, Simon, Corentin) Game Loops, LD Bricks, LD paper
- 17/10/2019 (Martin, Simon) More LD Bricks, Core feature : Playing music, examples, other small modifications
- 29/10/2019 (Martin, Simon, Corentin) Minor changes, corrections
- 30/10/2019 (Simon) Minor changes, precisions on Heart and Edge and game structure
- 11/11/2019 (Martin, Simon) LD bricks slides (still WIP)
- 12/11/2019 (Martin, Simon) Updating all slides
- 13/11/2019 (Martin, Simon) LD bricks slides, storyboard
- 14/11/2019 (Corentin) Camera dampening
- 25/11/2019 (Simon) Gameflow + Starting Wireframes
- 26/11/2019 (Simon) Wireframes
- 03/12/2019 (Martin) Added player and LD bricks gabarit
- 09/12/2019 (Simon) Reformatting, Changing theme etc...
- 12/12/2019 (Martin) Added elements to Level Design & summary update
- 15/12/2019 (Martin, Corentin) Add Level Design Storyboard + Legend + Philosophy
- 06/01/2020 (Corentin) Updated DC launcher
- 10/01/2020 (Martin, Simon) Update LD and add playtests
- 16/01/2020 (Corentin) Updated Sand/Timed Block
- 20/01/2020 (Simon) Added Tutorial, changed barrel organ to tram, reworked controls slide

Changelog

- 21/01/2020 (Martin, Simon) Updated texts and pics of the first LD zone, Reworked UI
- 22/01/2020 (Simon) Changed LD slides
- 23/01/2020 (Simon, Corentin) Added precisions in camera and tutorial
- 06/02/2020 (Martin, Simon) Clean GDD, delete outdated informations
- 07/02/2020 (Martin) Add new visual
- 08/02/2020 (Kilian, Alexandre, Quentin) Rework Game Feel
- 09/02/2020 (Martin, Kilian, Alexandre, Quentin) Clean GDD for ALPHA + Game feel slides
- 19/02/2020 (Simon) Added precisions and storyboard
- 20/02/2020 (Simon) Storyboard
- 07/03/2020 (Martin, Simon) Piano puzzle
- 10/03/2020 (Martin) Level 1-5 special brick
- 17/03/2020 (Simon) Universe, Backstory
- 26/03/2020 (Simon) Tutorial panels
- 02/04/2020 (Simon, Corentin) General Update, Level Design (Hamlet)
- 03/04/2020 (Simon, Corentin) General Update, Level Design (Palace)
- 04/04/2020 (Simon, Corentin) UI update, Level Design (Orchestra), descriptions of every LD rooms
- 05/04/2020 (Martin, Simon, Kilian, Alexandre, Quentin) Gamefeel update, general clean GDD for Milestone BETA
- 15/05/2020 (Martin, Simon, Corentin) Update GDD for Milestone GOLD
- 16/05/2020 (Martin, Simon) Update GDD for Milestone GOLD
- 17/05/2020 (Martin, Simon, Corentin) GDD GOLD Export



Summary

- <u>Overview</u>
- <u>Game Pillars</u>
- <u>Story</u>
- Fake screenshots
- <u>Game Structure</u>
- Win & Lose conditions
- <u>Game flow</u>
- <u>Game loops</u>
- <u>3C</u>
- Character : Philemon
 - <u>Character Core abilities</u>
 - <u>Character Gabarit</u>
 - <u>Character Abilities</u>
 - <u>Character Metrics</u>
- o <u>Camera</u>
- <u>Controls</u>
- <u>Game feel</u>

BA

- <u>Core feature : Concerto</u>
- <u>Concerto Ideas</u>

- <u>Tutorial</u>
- <u>Storyboard</u>
- <u>LD Bricks</u>
 - Hard Surface
 - <u>Soft Surface</u>
 - Checkpoint
 - Deadly Cogs / Killzone
 - Deadly Cogs Launcher
 - Moving platform
 - **Rotating platform**
 - <u>Timed Block</u>
 - One way platform
 - <u>Pendulum</u>
 - Microphone
 - Bumper drum
 - o <u>Zipline</u>
 - **Collectibles**
 - <u>Piano</u>
 - <u>Tramway</u>

- Rooms
- Legend
- **Rooms Schematics**
- <u>Gameflow</u>
- <u>UI</u>
- <u>Risks</u>





Overview - Specs



Nintendo Switch



Platformer



Single Player



2D Gameplay

Midcore





10 m

Unity 2019.1.8







6

Y A B Press the A button



Overview - Synopsis

"Explore and awaken a gigantic musical realm in this tight and poetic platformer."

"Use your violin bow to make your way through the world, and play the best of concertos!"



- The violin bow is both your **music instrument** and your **means of transport**
- Move swiftly through the air, **bouncing and boosting** through a dream-like environment
- Push through a variety of challenges across various interconnected areas
- Play music to activate elements in the level, and bring life to the scenery

Game Pillars

• The fluidity of the player character's movement The player can move swiftly across the rooms! Jumping and bouncing off the level gives off a pleasant feeling.

• Moves are easy to learn, but hard to master The player can get through the critical path without fully mastering his movements, but mastering them anyway would allow him to reach secrets off the main path, or progress quicker.

• The Level Design evolves as the player progresses through the realm After each concerto, new LD elements appear or begin to move as the realm slowly awakens. The last few rooms of the game challenge the player with many different elements at once.

Universe - Backstory

Symphonia is a mystical realm, composed of mechanical pieces, musical instruments and organic elements, centered around music. The whole ecosystem is powered by music, and the inhabitants are either musicians or manufacturer of musical instruments, or engineers of the city...

For unknown reasons, Symphonia has lost all power and has **fallen into a deep sleep**, empty of all music, for an extensive period of time.

The main character, **Philemon**, a bird-like violinist, enters this asleep realm at the beginning of the game. Wielding his violin and bow, he's about to awaken Symphonia again, by playing music in specific parts of the city, and restore this world to its former glory.



Universe - Game Story

During his journey in Symphonia, Philemon passes through several environments : first a hamlet, seemingly abandoned and empty of life and music; and then a structure with a busier architecture looking like a palace mixed with an opera. This latter place is the power core of Symphonia.

In Symphonia, Philemon also meets a shadow figure that resembles him. The shadow represents the inner will and nature of Philemon, that is to wake Symphonia up, as the conductor of this giant orchestra.

This shadow figure is always ahead of him and is used in the Level Design to show the player how to use the violin and bow. When Philemon needs to play a concerto to awaken the realm, the first time, the shadow is also here to guide him but disappears when the time to play comes.

For the third and last concerto, the shadow stays and plays the final concerto with Philemon. When the city finally wakes up, the shadow fades into the background element as the world wakes up for good.

Early Fake Screenshots





No.





+

VB

Those mockups represent our early concept artworks

+ ×

B

Mockup 3C

Ôn.

This screenshot represents our game artists' 3C scene running in editor

Y

В

Beta mockups

The piano room is an iconic feature of our game.

You will encounter this place two times during the journey, upon different aspects.



в





The Edge, with its huge window, is one of the places where the music of Philemon will awaken the realm.

It is the only place where you can see a glimpse of the outside of Symphonia.

Game Structure

As a full commercial game, Symphonia could be structured in a **tentacular shape**, like a metroidvania, having the player give **a multitude of concertos** before being able to wake the **heart of the machinery** up (cross hatched on the picture).

For the fifteen minutes PFE experience, the game is condensed to only one way to the Heart, and then **one way** back and forth between the Heart and the Edge.

It's possible to use an in game menu to teleport to the last room of each zone.

Leaving the game and coming back to play, starts again from the beginning of the room the player was in when they left.



Game Structure

- A semi-linear **2D platforming game**
- Several rooms of different sizes
- Some rooms are **secret** and reward the player with narration (*this occurrence is not in the game*)
- The Edge and the Heart change drastically the Level Design and the environment

					B				
Sleeping Hamlet		н	Slumbering Palace			E	Illumin Orche	н	
First time entering the Heart, waking Symphonia does not work				Enters the realm, aw Symphoni	Edge aken: a a bi	e of the s it	Second ti Heart, th brings fur life	me entering the is time the musi Ily Symphonia to	

Win and lose conditions

Win conditions

The goal of the player is to cross all the mandatory rooms until the final room, and play the final concerto.

Dying means the player starts again from the beginning of the current room.

Narratively speaking, the goal for the character is to **play music** in different chambers of the realm of Symphonia, so it wakes up from its slumber and comes to life again.

Lose conditions

In *Symphonia*, the character **dies in one single hit** by touching a spike, or falling in a pit, or even being shot by a marble. However, dying is **not very punishing.** There is no game over, and dying takes the character back to the beginning of the room (or the last checkpoint in the room). Dying and trying again is fine and is **part of the experience.** (see "Character: Death & Respawn")

Game Flow

Challenge Rooms

The player is given a huge and diverse set of movement skills, to create a technical, yet dynamic and aerial game flow.

The game presents different types of rooms : the challenge rooms, and the narrative rooms.

When the player progresses in the game, more death hazards appear, but somes places are still left dangerless to let the player rest. Contemplative rooms are also scattered through the whole game.

This means that the game feels harder, but still provides **time to breath** regularly.



Challenge rooms are present all along the game

Narrative Rooms



Narrative rooms are always linked to a musical feature



Game Loops

Місто Loop

At a micro level, the game presents **in a single screen** a mix of **perilous situations**, which tense the player, and challenge them; and **breathing moments**, where the player has no way of dying and can relax for a bit and prepare for the next hazard.

The more the player moves forward into the game, the more dangerous situations there are in the level design.



Game Loops

Mid Loop

At a mid level, the game is separated into mid zones by narrative / sensorial times, when the violinist plays music. Between those times, the game presents a series of **dangerous game situations**, the core rooms, and **out of core gameplay**, like the piano and the train.



Character - Presentation

Philemon

Philemon is the main character of the game.

He is a gracious and volatile figure, a strange creature who looks both of a human and a bird. Philemon inspires tranquility, agility, and charisma.

Coming from the city of Symphonia, his clothing and field of work is tightly related to music.

He is on a journey to awaken the musical realm of Symphonia, fallen into a deep sleep.

He wields a mythical violin and bow, which has unique properties, like playing awakening music, but also physical properties like to be used to bounce around, and to be bended to launch the holder in the air.



Character - Overview

Core Abilities



Walk By holding the Left Stick to the left or to the right, the character moves in said direction.





Jump

When tapping the jump button, the character performs **a small jump**, allowing him to easily cross the smallest gaps and obstacles. Holding down the button lets the character performs **a high jump** instead.



Poke

By pressing either the (R) bumper button or the (ZR) trigger button, the character **performs a poke forward** with their bow.

Character - Overview

Core Abilities



Bounce

When the character pokes a **hard** surface, the character bounces off that surface.



Stick When the character pokes a soft surface, the character sticks to that surface.



When sticked, the player can **launch himself in a direction** by holding the stick in the opposite direction and release it.

Character - Gabarit

The character size and all other gabarits are based on the **Level Editor grid** (intern Tool).

The character height is 2 squares high.

The character width is **1 square wide**.





Character : Abilities (1 / 7)



Run

By holding the Left Stick to the left or to the right (past the stick's dead zone), the character moves in said direction.

The character reaches their full speed 0.1s after the direction starts being held, and the **full speed is the same** no matter how tilted the stick is.

When releasing the stick, the character's horizontal velocity returns to zero in 0.1s

Character : Abilities (2 / 7)





Jump

The character will **immediately** jump when pressing the jump button (default = B) while grounded. No bunny hopping occurs if the jump button is held down. The player's vertical speed is then set to their <[1]jump speed> and remains as such until the button is either released, or it gets held until <[2]max jump duration>, or the player bumps into a ceiling, performs a poke, or any other action that forces a sudden change in velocity.

When the player is airborne and no longer jumping, their vertical velocity decreases every second by <[3]gravity>, causing their vertical speed to first stall, then go into the negatives as the player falls downward. This negative vertical speed is capped by <[4]terminal velocity>.

Character : Abilities (3 / 7)





Poke

By pressing either the (R) (L) bumper button or the (ZR) (ZL) trigger button, the character will **instantly** perform a poke forward with their bow. By tilting the Left Stick, the player can aim their poke sideways, upward, or down if they're airborne.

The poke hitbox is active as long as the button is pressed.

If the tip of the bow touches a surface perpendicular to the direction of the poke, either a <u>Bounce</u> or a <u>Stick</u> occurs depending on whether the surface is <u>hard</u> or <u>soft</u> respectively.

Character : Abilities (4 / 7)



Bounce

When the character pokes on <u>hard surface</u>, the character **instantly** bounces off that surface. The direction the character is sent toward depends of two factors:

- 1. The direction that the surface is facing
- 2. The character's velocity/direction when the surface was hit

A bounce off a hard floor will send the player **further up** than a regular jump would.



Character : Abilities (5 / 7)



Stick

When the character pokes a <u>soft surface</u>, the character sticks to that surface. Both floors, ceilings, and walls can be stuck to as long as they're made of a soft material.

The character remains stuck for as long as they keep the poke button (R or ZR) held down.

If the character releases the button without tilting the Left Stick in any direction, he unsticks and starts to fall normally. If the Left Stick was tilted however, a **Boost** may occur (see next slide).

Character : Abilities (6 / 7)



Boost

When stuck to a <u>soft surface</u>, the player can hold the Left Stick in a direction along the surface, bending the bow, and releasing the poke button propels the character in the opposite direction to the one that the stick was held in. **The player cannot bend the bow up to boost downwards.**

The velocity at which the player is sent is the same no matter how much or how long the stick was tilted.

For the most of the Boost's frames, the character's speed cannot be reduced by tilting the left stick to prevent instant-slowdown.

Character : Abilities (7 / 7)

Play the Violin

When the player press the Y button, the character starts using his violin and **play music**. The music is playing as long as the button is pressed. While the character is on the state "Play music" he can not move, jump or perform any action, the player will needs to release the X button first.

This action is useful to activate some mechanisms in the game (see LD Brick "<u>Microphone</u>"), but is also used as a flavor move. When Philemon plays music at certain places all across the game, his environment may react to its music in a strict graphical or musical way : flowers open, instruments play...

The animation is not the same as the one used when playing music during Concerto.



Here, Philemon plays the violin to open the gate



Character : Former Abilities

NB - We didn't keep this feature in the game





Dash

The player **instantly** performs a dash by pressing the dash button (default = X) while holding a direction on the Left Stick. This moves them quickly in that direction for a set duration, and following a specific velocity. The dash can be performed in 5 directions while grounded, or 8 directions while airborne. Pressing the button without tilting the Left Stick causes the character to dash horizontally in the direction that they are facing.

From the start to the end of the dash, the character's horizontal and vertical velocity are overridden by the dash (if the character is falling, but performs an horizontal dash, their vertical velocity instantly reaches zero and remains zero until the end of the dash). When dashing against a floor, the dash immediately ends. When dashing against a wall or ceiling, the player remains stuck to it until the end of the dash's duration.

The player cannot dash again until they either: land on the ground, perform a Bounce, perform a Boost.

Character : Former Abilities

NB - We didn't keep this feature in the game



Harpoon

After [TBD] the character obtains the ability of throwing their spear with a chain attached to it, allowing them to:

- pull the spear back if it doesn't touch anything or an hard surface
- zip to the spear if it stuck itself on a soft surface
- pull light objects to you(?)
- more (swinging around ?)

Character : Metrics (1/3)

Jump

[1] Jump Speed : Every update, the character rises by this many units while they hold the jump button [2] Max Jump Duration : How long can the player hold the jump button before the Jump automatically stalls (jumping into a ceiling automatically triggers a stalling)

<u>[3] Gravity</u>: Every update, the character falls by this many units while airborne (and no longer jumping/boosting) <u>[4] Terminal Velocity</u>: The maximal vertical speed the character can fall at, under the effects of Gravity

Air control

[5] Horizontal Air Control : How much the left stick affects the character's horizontal speed while airborne.



Character : Metrics (2/3)

Poke

Poking while airborne causes a very short dash that cancels the player's vertical momentum if done horizontally, of horizontal momentum if done vertically)

[8] Poke Speed : Every update, the character moves in the direction of the poke by this many units (Air Control & gravity DISABLED for the duration) [9] Poke Duration : How long will the player remain under the effects of the poke



Character : Metrics (3/3)

Boost (Horizontal)

[10] Horizontal Boost Speed : Every update, the character moves sideways by this many units [11] Horizontal Boost Duration : How long will the player remain under the effects of the boost, unless they hit a wall first (Air Control & gravity DISABLED for the duration) [12] Horizontal Boost Angle : The direction the character is sent in (by default, 0°, horizontally)

Boost (Vertical)

[13] Vertical Boost Speed : Every update, the character moves sideways by this many units
[14] Vertical Boost Duration : How long will the player remain under the effects of the boost, unless they hit a ceiling first. (gravity disabled for the duration, air control still enabled)
[15] Vertical Boost Angle : The direction the character is sent in (by default, 90°, upward)

Character : Death & Respawn

In Symphonia, the character dies in one single hit from either:

- touching a **Spike**
- getting shot by a marble
- falling in a pit

When the player dies:

- their controls are disabled, and their speed is set to (0,0), and they are no longer affected by gravity.
- The character immediately start playing their **death animation**, which causes them to disappear.

A second after the instant they got hit:

- The camera focus on the respawn point
- The player respawns at the last checkpoint activated, which triggers their respawn animation
- The player's controls are **re-enabled**, gravity affects the player again.
- All timed blocks (lyras) in the room respawn immediately
- All microphones in the room deactivate immediately




Camera

General informations

- 2D side-scrolling camera
- Centered on the player by default
- The game is divided in individual scripted screens : it is not a linear open adventure world camera
- Smooth and **light lerp** when the camera follows the player
- A global principle : The camera needs to be adapted to every single game situation, so it never gets in the way of the player



Launch diaporama to activate gifs



Camera - Tools

Tools

- Some tools are used to control how the camera reacts to the different rooms presented in game, and allow a good level design:
- **1. Screen boundaries**: to constitute said rooms and have total control of the camera
- 2. Trigger zones and points: to allow the camera to move towards the challenge and to allow transitions between rooms
- **3. Juice effects** to make the camera more alive and organic



Launch diaporama to activate gifs



Camera - Boundaries

Screen Boundaries

Screen boundaries are a tool used in level design for every screen in the game.

When the camera touches the screen boundary, it **stops** and can't go further. When this happens, the camera can not follow the character properly, which is **no longer at the center** of the screen. (pic 1)

The camera never goes past the screen boundaries of a level, except when the player **exits the room.** In this particular case, the camera moves to the next room **smoothly**, crossing the screen boundaries briefly. (pic 2)





Pic 2

Camera - Boundary Dampening

Boundary Dampening

To prevent the camera from hitting the boundary hard and stopping too suddenly, the camera performs a **dampening** when it nears the boundaries.

When it reaches a distance [x] from the border while moving **toward that border**, the speed at which the camera moves is **halved**. Moving away from the border does **not** cause any slowdown.

When the camera reaches a distance [x/2] from the border, the speed is **halved once again**, resulting in a $\frac{1}{4}$ movement speed of the camera. Allowing the camera to touch the screen boundary at a very small speed.

Normal speed



nom horde

[x/2] Slowdown ++

Position of the camera's right border frame per frame as the player moves at a constant speed from the room border.

Camera - Projected Focus

Projected Focus

The camera follows the **projected** (extrapolated) position of the player, depending on the player's current speed (and its direction).

We are able to **tweak** the coefficient of this projected position to prevent the camera going forward too much, or moving around too quickly when the player switches direction.





Camera - Trigger zones

Trigger zones and points

Trigger zones and points are a tool used in level design for every screen in the game.

When the character touches a trigger zone, the camera is smoothly **sent to the trigger point attached.** It is then deactivated until the character leaves it and crosses it again.

The screen boundaries can be broken by this tool only (the camera can go **past the boundaries** briefly).

This tool is used mainly to allow **screen transitions** (pic 1), but also as a scripted part of level design to move the camera **towards the action.**



Camera - Juice Effects

Juice Effects

- <u>Screen shake</u> is used for juiciness when the player is hit, or when a massive concerto is played.
- <u>Slow motion</u> is a scripted juice effect used very punctually when the player achieves an impressive move in a difficult position.
- <u>Zoom</u> is a tool used for potential narration or when the **room needs a larger vision** of the environment.





Controls - Joy Cons

Ân.



5



Controls - Pro Controller

Ô.



-15

Game Feel - Essential Animation principles

Serenity



Squash and stretch animation response to interaction. It increases the sensation that the environment reacts to the player's will. It empowers his control impression. Trust



Arcs : propulsion trajectory will follow arcs, in this way, the player can easier predict his destination. It also gives a fluidness feeling. Amazement



Secondary actions gives the scene more life, it also tells the player that he's evolving in a huge system.



Game Feel - Complementary animation principles

Serenity



Anticipation : the player is aware of the incoming events, preparation frames help him understand the current situation. Trust



Timing : The character seems to obey the laws of gravity, the player feels more confident about his abilities as he knows they follow a logical rule. Amazement



Staging : The player understands he achieved something, he believes something important is going on.

Game Feel - Emotions

Symphonia is a **challenging** platformer, with an **exploration** component that uses *easy to learn* controls.

Some of the emotions conveyed in Symphonia are :

- Serenity : The player evolves in a calm atmosphere, he knows his environment isn't a danger. He commands the rhythm of the game.
- **Trust**: The player is confident in his abilities, he's got an high precision in his controls, and his fluid movements don't hinder his actions.
- **Amazement :** The player goes through rooms where gigantism and poetry inspires the artistic direction.





Game Feel - Advanced design rule



Coyote Jump

The player can still jump a small amount of time after falling off of a platform. The player can redirect his attack when airborne.

Airborne Attack Redirection



Different deadzone ground/airborne

The player has different deadzone when he's on ground and airborne to maximise jump precision.

Game Feel - Advanced design rule



Spear width

The spear has a certain **width** to facilitate hitting object.

We simulate a **rectangular hitbox** to smooth the player movement at the edges of the platforms.

Rectangular hitbox simulation

If the player hits jump **too soon**, the input is still **registered** to avoid frustration.

Jump buffer







Game Feel - Advanced design rule



Stick reaction

When **sticked** to a soft surface, your **orientation speed** is doubled when you go from the outside to the inside to left to simulate the bow elastic force and facilitate fast **position switch**.



Stab buffer

After a poke, the **bow stays out** for a tiny amount of time to avoid the **frustration** of having the bow retract at the exact time that it was supposed to hit the ground.



Terminal Velocity

The player has a **maximum fall velocity,** to make every fall easier to control.

Game Feel - Violin Mechanic

Play Violin



Zoom centered on the Player. Ambient sounds and music fade away to hear the **sound of the violin** in order to highlights the serenity of the instant.

Concerto



Dynamic Animation. **Zoom out** to show the player's action greatness.

Game Feel - Exotic Situation

Zipline



Progressive zoom through the ride, **collectibles** and **no risk of dying** are used here to **reward** the player on a mysterious slide through Symphonia : it's a **narrative beat**.

Tramway



The camera zooms out during zipline, which lands the player right on a tramway wagon. The **tramway** is one of the most **surprising** features of the game, an absolutely **twist** of the player's expectations.

Game Feel - Exotic Situation

Air Thruster



The physic of the Character changes for more **aerial physics**. The Player can experiment his **Air Movement** with the Air Thruster.

Elevator

Marbles Burst Shot



After the "Concerto", we let the Player **rest** in the Elevator. During this Phase the Player doesn't have **any control** on the Character. It's a transition phase.



Used to tense the player on his **timing** skills. The **asynchronous** wave provides a juicy effect comparable to a metronom.

Game Feel - Intentions

Non Playable Character





Those little characters can **interact** with the Player, And bring **more life** inside the world of Symphonia.

They bring **Lore's** element throughout the whole game.



Intention

Cutscene Introduction



This is the first contact between the **player** and his **character**, discovering for the first time the graceful movements of the character.

It is also the first contact the player has with the world of **Symphonia** as he discovers the outside of the village.

The closing of the gate, in addition to the environment, creates a feeling of **solitude** within this universe, reinforcing the bond that the player will have with his character.

Intention

Heart of Symphonia Phase 1



This is when the player discovers for the first time the inside of the palace of Symphonia, hidden by the big curtains.

The dezoom shows the player the greatness of the world of Symphonia.

He can gradually discover how Symphonia is made.

The fact that the curtain doesn't open up completely, teases to the player a possible second passage in this room to play a new concerto.

Story Board

Ba

The Edge Cutscene





Zoom when the player approches the scroll



Center Camera

The scroll starts scrolling The player goes on the plateform



The player starts playing violin, instruments start appearing



Platform starts elevating.



Stop dezoomzing

Platform stops



Stop dezooming

The player stop playing on a big note, a shock change the assets and/or the color grading.



Camera Shaking after the shock wave



are now working.





Platform stops, all intruments and gear



-Notes comes from the gramophones. -Offset on the forest that goes down so the whole level seems going up. -Rocks are falling from the window.



Level continues to going up.

Something appear behind the mountains



Platform goes down

Intentions

The Edge Cutscene



This cutscene shows to the player the **Awakening** of a part of the Symphonia Structure.

During this cutscene the game universe evolves from a still "**sleeping**" world to an "**awaken**" world.

The musical instruments will all start to play in rhythm to offer the player a **concerto** worthy of the name.

Intentions

Heart of Symphonia Phase 2



This passage corresponds to the **end of the game** and the last moment the player spends with his Character.

Here, the player can finally **discover** completely what was behind the great curtain.

This represents the full awakening of Symphonia with the **bell** ringing again.

Several elements in the Background and the Foreground should also come to life to symbolize the **awakening** of the whole structure.

Game Feel - Tool

When pressing the start button in editor, the designer can stop the game and have an **overview of the character trajectory over time**.



showcase of the player's movement abilities

Dynamic timeline of the different state of the character during a level



Stick

Core level design feature : Concerto

What is it?

Punctually, the violinist plays a concerto in **dedicated rooms**, in a cinematic scripted event. This concerto gradually **awakens the machinerie** in which the character is evolving, affecting the level design. The player plays music 3 times in 2 different rooms during the 15 minutes experience. A concerto cutscene lasts about 10 to 15 seconds. The concertos are given in **the Heart** and in **the Edge** rooms.

What does it do?

Playing music radically **changes the level design** of the game. This has various effects that depends of the specific chamber : modifying the state of LD bricks, making them more active or unstable, modifying the structure of walls; or even activating musical machinery! In every case : those are **permanent modifications and the character must adapt** and continue.





Core level design feature : Concerto

How does it happen?

The violinist playing a concerto is a **cutscene** (scripted event), possible only at two places in the game, where they must press the X button to start playing. Therefore, the player is **not able to interfere** with the cutscene and is fully aware of the importance of what is happening. After the cutscene is finished, the player is able to start wandering around again and interact with the new various elements.



Core level design feature : Concerto

What does it looks like?

The concerto rooms contain a **pedestal and a music stand**, as well as a partition. The room is empty of other gameplay elements. That way, the player knows when they are gonna have to give a concerto.

The **lights fade**, and only let a **spotlight** directed onto the violinist. Instruments and music notes are visible. The camera focuses on Philemon and his maestria.

The environment starts **making weird mechanical or musical sounds** during the concerto and after the end, with a screen shake and chromatic aberration, to underline the fact that something changed in the world.



Concerto: Ideas

Effect example : Modifying existing bricks

Playing music in a Concerto Room can permanently modify existing level design elements :



The ground level changes here and there because of cogs and wheels in the ground, can be pulsing in real time or stay inert



The inert block evolves into a rotating block and then into a rotating moving block



This wall was only paved with hard and soft blocks, but now, deadly cogs are being spit out from some hard blocks!

Concerto: Ideas

Effect example : Modifying existing bricks

In this situation, the violinist comes from the left to play music to the far right. When it happens, everything starts moving and **bricks collapse, new paths appear.** The player has to take advantage of bricks in the level design that they didn't need or didn't exist before, so they can exit the room.



going this way

Before Concerto



Concerto: Ideas

NB - We didn't keep this feature in the game

Effect example : Rotating the level

In this situation, the violinist comes from the left to play music to the far right. When it happens, everything starts moving and **rotates 90° to the left.** The player has to take advantage of bricks in the level design that they didn't need before, so they can exit the room.





Tutorial in video games

What is usually done :

No explanations

Text is displayed to explain mechanics to the player

A **video** is shown to the player in a separate window

Pictograms and drawings are shown in the background to explain the mechanics

A character **opens the path**, showing what to do

Tutorial

Our proposition for Symphonia :

Pictograms and drawings are shown in the background to explain the mechanics

A character **opens the path**, showing what to do

Our idea is to introduce the mechanics in several rooms with intradiegetic **pictograms and drawings** in the background, but to couple it with the appearance of a mysterious character who is sometimes seen by the player opening the path : **the shadow**.

The Shadow is a character a bit transparent and ethereal, like a ghost. That's because he's just an instinct of our main character, Philemon.

The shadow can be shown using the violin bow and playing music at strategic points in the game. At the same places it appears, there will often be in game pictograms, like small billboards, to show what button to use to achieve what it is doing.

Tutorial Example

A diegetic pictogram shows him how to progress, and in front of him, the shadow does the movement and then leave.

The pictograms are integrated to the background according to the theme.







List of pictograms



Tutorial : Benefits

Why we chose this option :

- We can't afford to not explain our mechanics
- Using text in the game is out of the question for the team (we wanted to release the in-game experience without dialog)
- We can't afford to open an entire new window to show a video
- Showing just a pictogram of the buttons has been tried in a 3C build playtest, but is not enough

Why we think it's a great idea :

- The shadow is the only other character resembling the avatar in the game, so the player will be focused on him
- The pictogram, which was not enough, is now clear because there is "video proof" of what is supposed to happen
- The shadow looks like the main character and uses the same bow but is very slightly different, which teases the player and adds narration to the game
- This solution allows the game artists to re-use the main character's animations, slightly adapted, to introduce a powerful narrative and learning tool, for a limited amount of work
Storyboard : Tutorial

Places where we show the shadow :

- **Room 2** : Sticking and boosting upwards
- Room 3 : Bouncing upwards and boosting sideways
- Heart of Symphonia 1st time: Leaving the room
- Room 2-5 : Watching the player
- Room 3-2 : Running towards the next room
- Heart of Symphonia 2nd time : Playing violin again (lore)



The Shadow

Tutorial : Room 1

NB - We didn't keep this occurrence in the game

• **Room 1** : Playing music to open door

As Philemon walks to the bottom of the staircase (1), the player is forced to stop and the camera focuses on the shadow, standing still near the door (2). He is shown playing music (3). He then leaves through the door and disappears in a fading effect (4). The camera travels back to Philemon who has the ability to move again (5).



Tutorial : Room 2

• Room 2 : Sticking and boosting upwards

As Philemon walks to the edge of the false platform (1), the player is forced to stop and to watch the shadow, standing still near the wall (2). He is then shown using his bow to stick to the wall (3) and slowly boost himself upwards at the top of the platform (4). He then leaves through the right of the platform (5). The camera travels back to Philemon who has the ability to move again (6).



Tutorial : Room 3

• Room 3 : Bouncing upwards and boosting sideways

As Philemon enters the room, the player is forced to stop and to watch the shadow, standing still near the wall (1). He is then shown using his bow to bounce up and reach the platform above him (2). After this, the shadow walks to the right and sticks to the soft ground (3). He then slowly boosts to the left and lands (4). Then he slowly fades as he walks to the left (5).



Tutorial : Heart

• Heart: Playing violin

As Philemon enters the room, the player can see the shadow standing on the pedestal and leaving the room (1). When Philemon gets closer to him, he fades (2). Philemon is all alone when he reaches the point where the shadow was standing (3).







Storyboard : Ending scene

At the end of the game, Philemon reaches the Heart room again to play a third and final concerto.

The room is empty (1). When Philemon approaches and climbs on the pedestal, the shadow appears (2).

When Philemon starts playing, the shadow plays along, as they give the most memorable concerto Symphonia has heard in a long time (3).



LD Bricks - Hard Surface

- The player can walk normally on it.
- When the player pokes a hard surface, he **bounces off** this surface.
- Hard blocks can be placed on floors, walls or roofs.
- A tile can either be Hard OR Soft, not both at the same time.







LD Bricks - Soft Surface

- The player can walk normally on it.
- When the player pokes a soft surface and holds the poke button, the character and their bow get **stuck on that surface**. (see Character : Abilities 6)
- If the player releases the poke button without tilting the stick, **they fall down**
- If the player releases the poke button after tilting the stick, **they perform a boost** (see <u>Character : Abilities 6</u>)
- Soft blocks can be places on floors, walls or roofs.
- A tile can either be Hard OR Soft, not both at the same time.



Legend Soft Surface



LD Bricks - Checkpoint

A checkpoint is an **invisible trigger**. If the player dies, they respawn at the last checkpoint they've touched.

The checkpoint consists of 2 parts:

-The trigger zone : the area that detects the collision with the player on his first pass. It has 2 states: On and Off

-The respawn point : the exact location where the player respawns after his death





LD Bricks - Regular killzone "deadly cogs"

A killzone is an area that kills the player instantaneously if they touch it.

A killzone has either a visual representation or is represented by an **offscreen pit**.

Reminder: if the player dies, they respawn at the previous checkpoint.

<u>Gabarit:</u> We should be able to put a killzone on one square and put them side to side together.







LD Bricks - Marble, marble launcher & socket

The **marble** is a **1x1 block** that sends deadly marbles flying through the air. The marbles can be sent either downwards or sideways at a regular interval, and their paths follows a straight line.

Before sending the marble, the launcher makes one appearance for a short moment. This state is call **Launch preview**.

Marbles disappear when they enter in a **marble socket**, or when they fall in the foreground after hitting a **piano key**. Like other killzones, it **kills the player** if there is any contact.

Metrics that we should be able to balance :

- The **speed** of the marble
- The rate between each shots
- The launch preview duration
- Whether the cog will be fired to the **left**, to the **right**, to the **bottom o**r to the **top**



Gabarit: Launchers, Sockets and Marbles all have a has a 1x1 size, similar to a regular hard block.

Legend



Marble Launcher / Marble Socket

LD Bricks - Moving platforms

A moving platform is a LD block (with hard or soft surface) that **moves in any pattern.** The platform makes a movement in as many directions as needed, then starts the animation again from the start at the end of the loop. The **movement speed is variable.**

- Once the pattern is finished it can repeat itself
- The platform can stop for a variable time at any point (that time can be equal to 0)
- The player must be able to stick (and stay sticked) or bounce to a moving platform (following the surface) without an issue
- There can have **more than 2 points** that define the pattern

Metrics that we should be able to balance :

- The movement speed at any point of the pattern
- The direction of motion at any point of the pattern
- The stop time at any point of the pattern
- Has an active or inactive trigger

Gabarit: Usually 1x2 or 2x1









Moving Platform (Soft)



LD Bricks - Rotating blocks

The block turns **90° clockwise or counter clockwise**. **Each surface** of the block can be **hard, soft** (that means the player can **stick to it**) or **a killzone**. The rotating platform has **2 differents states**:

- **Inert state:** When the block is not rotating. In this state, if the surface is soft, the player **can stick** to the block. He can also bend or boost.
- <u>Rotating state:</u> When the block is rotating. In this state, the player can not stick to the block (unless they were already sticked). If the player was bending his bow, the system forces the character to return into the idle stick state (when the player is sticked to the block but not bending). This reset is effective even if the player was holding the joystick in a direction before the rotation, if they release the joystick the character stays in the idle stick state.

<u>Gabarit:</u> The rotating platforms has an unique size of 2x2.





LD Bricks - Rotating blocks

During the **inert state**, the light in the middle **changes color** before the rotation as an indicator for an upcoming **rotation**. First it turns **yellow**, then **orange** and finally turns **red** when the block is rotating. For the timer we have 2 differents ones:

- <u>Global Timer:</u> includes the inert + rotating state
- **<u>Rotation Timer:</u>** includes only the rotating state

To define the inert timer, we just need to **subtract the rotation timer from the global timer.**

Metrics that we should be able to balance :

- The direction of rotation
- The global timer
- The rotation timer
- The exact moment from when you switch to yellow light
- The exact moment from when you switch to orange light
- Has an active or inactive trigger





LD Bricks - Timed Block

All values marked as "X seconds" bellow should be tweakables

The Timed Block (previously "Sand Block") is a surface that disappears **X seconds** after the player either lands on it, or bounces off/sticks on it. The block **cannot** be crossed through from the bottom like a one-way platform, however bumping your head on it will not cause it to disappear. There are **two versions** of this block: Hard or Soft. Allowing the player to bounce or stick on it respectively.

- The block visibly shakes/animates X seconds before disappearing
- The block reappears X seconds after its disappearance
- When the player respawns, the block also immediately respawn

Gabarit: The timed blocks have an unique size of 3x1

Legend



LD Bricks - Fake Block

A fake element is a **part of the foreground**. When the player collides with an **invisible trigger link to the fake element** it disappears revealing a hidden entrance. Each fake block will emit a sound (the same for all of them) to give an hint to the player about their location. Once this element disappears it **does not reappear**. Use to hide the entrance of the <u>secret rooms</u>.

Metrics that we should be able to balance :

- Trigger size
- Time it takes for the element to disappears
- Adjust the sound coming from the fake block
- Adjust the zone where the player can hear the sound

Gabarit: Variable. Talk to a Game Designer to see the necessary size of each fake block.

NB - We didn't keep this brick in the game

\$*†*



Legend

Zone hidden by Fake Block

LD Bricks - One way Platform

This is a platform that allows the player to **pass in one direction but not the other**. By default, the player can **pass through from below** but **not from above**.

This platform can be soft or hard.

Gabarit: One single one way platform has a width of 1 tile but the length represent only 0.5 tile







LD Bricks - Pendulum

A pendulum is a platform suspended by a string.

The platform can behave like **either a hard block**, **a soft block**, **or a kill zone**, allowing the player to stand on top of it, and poke it with their bow. The string and anchor that suspend the block are intangible.

The pendulum moves back and forth along a curve, with its movement following a regular pendulum motion, depending on the original position of the platform in relation to the anchor. If the player jumps or boosts off a moving pendulum, its current momentum gets added to the player's inertia.

If the platform is placed immediately below the anchor, the pendulum will be inert, behaving like a regular block.



The pendulum's platform can behave like either a hard block, a soft block or a killzone.



LD Bricks - Microphone

The microphone is an interactive LD brick that activates a change in the LD. Bricks can start moving, rotating, or appearing from the background for a limited time.

In this situation, the door opens when Philemon plays his violin in the microphone.





Microphone/ **Concerto Trigger**

LD Bricks - Microphone

The microphone works because it puts in resonance the machinery in the background because of the vibration produced by the violin bow on the string. Therefore, it is a very graphical brick.

Metrics that we should be able to balance :

- The blocks linked to the microphone
- The action the blocks perform when activated:
 - Platform appears or disappears
 - Activate or deactivate a rotating platform
 - Activate or deactivate a moving platform
 - Change a deadly pendulum into a soft or hard Pendulum
- The duration of the timer
- The number of bricks that the microphone activates or deactivates

Gabarit: Each microphone has a length of 3 tiles and a width of 2 tiles.



Activated for a limited time Deactivated

LD Bricks - Contextual : Bumper Drums

The bumper drums are a special LD brick, appearing just after The Heart. When the player pokes it, they are **violently repelled to another bumper drum** a bit further up, and they must **poke each bumper drum in rhythm**, until they get to the top of the room.

The bumper drum is a brick that only repels the player if they **poke the membrane**. If the player tries to land on the membrane, they slide on it and eventually fall down. If the player tries to poke the drum in another part that is not the membrane, they get a simple bounce, which is not enough to get up there.

Metrics that we should be able to balance :

- The strength of the impulsion when the bumper is poked
- The angle of projection

Gabarit: Each bumper drum has a size of 3x3 tiles.







LD Bricks - Zipline

The zipline is a LD brick used a few times in the game. When the player jumps on the zipline, they start **sliding down** until they reach the end of the zipline.

The player can jump from the zipline, if they fall down on it again, they start sliding down again.

Metrics that we should be able to balance :

- The start and end of the zipline
- The acceleration on the zipline
- The **max speed** on the zipline







LD Bricks - Collectible - Musical Fragment

The musical fragment is a **collectible item**. When the character hitbox collides with the musical pick up, they collect it and add 1 unit to the musical fragment meter. This meter is visible after the credits, at the end of the game.

The Musical meter rewards the player with one of 5 available titles, changing following the number of fragments collected.

This collectible is common: there are 200 in the game.

Gabarit: One tile for each.



Collectible





LD Bricks - Collectible - Automaton NPC

NB - We didn't keep this feature in the game

The Automaton is a collectible item. Three of them are hidden behind secret passages and in zones that are hard to reach.

When the player finds one of them, they wake up and run out of screen.

The automatons can later be found in the Heart of the machinery, and when the player plays the final concerto at the end of the game, the automatons that have been found **play along with one instrument**, **adding layers to the final music** of the game as a reward.

When the player found a NPC, he needs to stand next to it and use the play violin action. After 1 second, the NPC awaken and start moving. When the NPC can finally move, it greats the player and he can be found at the end of the game on the final concerto.



LD Bricks - Piano

A piano tile is a block with a piano key shape. When the player walks on it, or lands on it, or hit it with their bow, the tile **goes down from X units** and the sound of a piano key is played. The block remains down as long as the player remains on top of it. It also behaves like a **hard block**, which means that the key will immediately start moving back up if the player hit it with their bow because they are **bouncing**.

The sound continues to play with its volume slowly decreasing as long as the player remains on top of it and the key is down. If the player jumps/the key moves back up, the volume then decreases almost instantly, stopping the sound.

In addition, a corresponding piano hammer falls into the background at the same time as the sound is emitted.

Metrics:

- note pitch
- distance that the key moves down

Gabarit: Each piano tile has a width of 1 tile.





LD Bricks - Piano Puzzle (100m 2-4)

On the first passageway on the piano room, all the tiles are deactivated (except one) and the exit is locked. That means only one tile makes a piano sound and activates a hammer in the background. The only tile activated has shiny particles on it and a music note.

When the player steps on this note, the tile is deactivated and another one activates instead. This pattern repeats a few times (3 times?) until the final tile is pressed.

When it is pressed, a little melody plays. Now, all the tiles are activated, they all make a piano sound, and activate hammers in the background. And the exit of the piano room is open.

Metrics:

- Which piano key is what element of the puzzle (first, second, third piece etc)
- Sound that is played
- Hammer that is activated





LD Bricks - Tram

The tram is a special LD brick. It's composed of differents little train cabins, following each other like a tramway or telecabin. When the player steps on it at first, the cabins are static, but then they start to move towards the right edge of the screen, where a killzone is placed, putting the player in danger.

The trams acts like a **treadmill**, taking the player with it, and is also a **hard surface** (allows bouncing).

The pattern in which the trams spawn can and will be customized so some trams are missing on the rail, causing the player to have to jump so they does not fall. The speed of the trams is also customizable.

In addition, the train plays **spatialized sound when it is activated**, partially covering the initial background music. It also stays activated, once it is activated.





LD Bricks - Tram

On the technical side, the tram brick works with "slots" system. The slots move from the left to the right. These slots are created on the left, the creation point (Green box) and destroyed on the right, the destroying point (Red Box). Each slots can have 2 states:

- Full: That means the slot is represented as a Tram. A Tram is • full of Hard Surfaces and its movement is similar to a moving platform.
- *Empty*: That means the slot is represented with nothing, no colliders, if the player walks on it, he falls.

We can define if each slot is full or empty with a list. Once this list come to an end, it will loop the actual list. Metrics:

- General scroll speed of the slots
- Being able to define if a slot is full or empty (via a scroll down list on the hierarchy for example)
- The distance between the creation point and the destruction point
- Being able to switch the trams from hard surface to neutral surface



LD Bricks - Wind column

The wind instrument sticks out from the ground and **blows air** through the pipe upwards. If the player goes to the wind column, he **floats** in the air as long as he's on the column. While on this floating state, the player **speed is reduced**, but he can still go left and right but by floating.

The wind instrument can be flute, trumpet, pan flute etc. While the air blows, we can have a musical sound according to the visual.

Metrics:

- Size of the area of effect of the column
- **Reppel power** of the air blow
- Speed reduction





Level Design: Philosophy

The game is set inside a **gigantic town** built-out of **music instruments**, inhabited by **sleeping machinery** and **automatons**.

As the player **progressively awakes** the machines by playing music at specific places, the scenery puts itself into motion, and new gameplay elements appear. This leads the levels to have a greater variety of elements and tougher challenges the further the player gets into the game.





Level Design - Levels

The game is divided into several areas. Three "Levels", separated from each other by two Rooms called the Heart and The Edge. The game starts in the Sleeping Hamlet, then the players goes through the Heart to reach the Slumbering Palace, at the end of which the player finds The Edge, opening the path to the Illuminated Orchestra, that brings the player back to the Heart.



Level Design - Rooms

Hamlet

1-3

- One-Way Platform

- Bounce

1-2

- Boost

1-4

- Kill Zones

Each room up until the last few ones introduce new elements, while also re-using elements from the previous rooms.

Also each level has its own artistic style, increasingly grand and animated. Both graphically, and musically.

1-5



3-1

- Pendulum (Kill)

3-2

- Marble Launcher

- Jump - Play Violin

Start

1-1

- Run

- Collectibles

Level Design - Secret Rooms

The level design includes **secret rooms**, about 3 to 5 in the whole game.

These secret rooms are **hidden in the level design** and their entrances are revealed by the fade out of a <u>fake block</u>.

The secret rooms are small areas with **no gameplay elements** inside, their goal is to display informations about the **universe of Symphonia** via **environmental narration** (Background elements, posters...), as well as possible collectibles.

The informations displayed and their size are to be consulted on a case-by-case basis. NB - We didn't keep this feature in the game





Level Design - Legend

10 th



Zipline

Tramway

Pendulum (Hard)

Pendulum (Soft)

Pendulum (Kill)

Wind

1

Level Design - Room 1-1



The first room in the game serves to introduce the player to the basic movements: **walking** and **jumping**. The player will find **collectibles** on their path that they will pick up as they touch them.

Finally they have to play music in front of the gate at the right end of the room to raise it, and progress to the next room.

Level Design - Room 1-2



In room 1-2, the player is introduced for the first time to **soft surfaces**. They will now have to **poke** them with their violin bow to **boost** their way past the pillow gap.

A more experienced player may also accumulate momentum and **bounce** past the gap entirely.




To get past the entrance of room 1-3, the player will have no choice but to perform a **bounce** by **poking** the **hard floor**, ensuring that the player cannot go past this room without having used bouncing and boosting at least 2-3 times each.

One-Way platforms are placed to help the player progress, and collectibles indicate the path to take.



As the last real challenge of the Hamlet, room 1-4 serves as a test to make sure that the player is comfortable interacting with both **hard and soft surfaces**, as well as introducing them to **Kill Zones**. Checkpoints are placed to the start and end of each room as a safety measure, but this is where they might first see some actual use.



After several rooms of increasing challenge, room 1-5 offers the player a breather before the Heart takes their breath away.

Here the player can optionally use the **Wind** coming out of instruments to catch the collectibles placed in the air, as well as bounce to gain momentum and catch the coins.

Level Design - Heart



The **Heart** serves as a transition between the Hamlet and the Palace, here the player has to **play music** in front of the stand to awake the automatons who will activate the **hard moving platform** for them, lifting them to the next room, towards the next step in the violinist's quest : **The Edge**.





The first room of the Palace poses little threat, only requiring the player to bounce between the **Bouncing drums** with timing.

The collectibles add to the musicality of the room as a double ring is heard after every drum bounce.



In Room 2-3, the player has to deal with more interactive elements, as **timed blocks** fall a second after the player steps on them. The first one is placed over a solid floor to introduce the element to the player in a safe setting. The other two will require the player to bounce of them to clear the room. A stack of collectible awaits the player at the end to reward them.



Here the player arrives for the first time in the Piano Room. A gate at the end prevents the player from progressing until they solve the puzzle of the room by pressing on specific **piano keys** as they are highlighted.

Once the puzzle is solved, the player can press each piano key to make a collectible drop from the ceiling.

(see slide 96 "LD Bricks - Contextual : Piano Puzzle (room 2-4)")



Room 2-5 introduces Rotating Blocks, with both hard and soft faces.

The player will find out that they can remain stuck to a soft face as the block rotates, which will potentially help them clear the room more easily.

An extra collectible at the end rewards the risk taking players.



start

This room tests the player's understanding of previously introduced elements, along with kill-faced rotating blocks, then rewards them with a beautiful slide down a **zipline**.

next





This highly vertical room gives the player an opportunity to have fun performing several consecutive **boosts** in a row.

A **soft moving platform** in the middle adds a bit of flavor to this otherwise easy room (compared to the ones before and after)





The last room of the Palace challenges the player's timing and measuring skills as they have to jump upward from **pendulum** to pendulum to reach The Edge.

The room can either be easily cleared by taking breaks on the one-way platforms and **bouncing** from them onto the next pendulum, or faster but a bit more challenging by bouncing directly from a pendulum to the next.



Level Design - The Edge



The Edge marks the midpoint of the player's round-trip from the Heart. By playing music in it, they will awaken the rest of the realm's machinery, and the rooms the player just went through will have been replaced with new and more dynamic challenges.





From this point onward, the player will mostly move from the right to the left, marking his return toward the Heart.

Marbles are introduced in this room, the player has to time his jump to fall through them. The **timed blocks** in this room last a bit longer than in other rooms.



This is the only room in the game where **"Kill" pendulum** appear.

The player has to boost through the right one, then let themselves fall through the left one.

The player's feet don't touch the ground until they have cleared both **pendulums**.







The zipline at the start of this room breaks quickly, causing the player to fall on top of a **tramway** moving right. The player has to keep jumping and **bouncing** to not get dragged into the kill zone, and progress to the next room.

(see slide 99 "LD Bricks : Tram")



Room 3-4 serves as a "breather" as the player easily jumps from pendulum to pendulum horizontally. There is a challenge at the end where the player has to boost from the **soft pendulum** through a wave of marbles.

Performing the boost while the **pendulum** moves left will send the player through easily.





The player returns to the piano room, but this time the challenge is drastically different. Instead of having to press piano keys in an order to unlock the path to the next room, marbles rain from the top of the room and onto the piano keys, playing in sync with the background music.

The few tiles that never have any marbles fall on them are of a slightly different tint.



The last two rooms of the game don't introduce any new elements, instead challenging the player's mastery of all the elements introduced through the game, in quick succession.

After his fall at the end of the room, the player is rewarded with the sight of a beautiful clockwork, which react to the violinist music.





For this last challenge room, the player discovers that moving platforms can block marbles, and they have to move over and below the platform, as it scrolls left to arrive safely back at **the Heart**.



Level Design - Heart (end)



Hard Surface



Moving Platform (Hard)

Microphone / Concerto Trigger

Back at **the Heart**, the player moves down the elevator and can admire the scenery one last time before playing the musical finale and finishing the game in the most impressive way.



Ô.

Q

UI : Splash Screens



Those are the opening screens. We need one to display ISART Digital and one for the software used (Unity, FMod...). We could also have a team logo and display it here.

Those screens don't require an input from the player to advance.

Connected to : Title Screen

(B)

Splash Screens (Software, ISART)

Title Screen

UI : Title Screen



This is the screen featuring only the title / logo of our game and eventually a nice background.

The player need to press a button to advance.

Connected to : Main Menu

Title Screen

Main Menu

UI : Main Menu



This is the main menu of the game, featuring our title logo, and four buttons to advance in the menus or quit the game.

The cursor at the left of the buttons could be the violin bow.

Connected to : Title Screen, In game, Settings, Credits

Main Menu

Title Screen

In Game

Settings

Credits

UI : In Game



This is the game featuring the core gameplay.

There is no extra-diegetic HUD planned, as the collectibles are featured in the environment of the central room.

Connected to : Pause Screen

In Game

Pause Screen

UI : Pause Screen



This is the pause menu, featuring a button to resume the game, a button to access the settings, and a button to go back to the main menu.

In the background, the game is frozen and darker.

Connected to : In game, Pause Settings, Back to main menu screen, checkpoint select screen



UI : Pause Settings



This screen features the settings of the game : a button to select a language, two sliders for the sound and music, and a possibility to enable or disable the screen shakes and console vibrations.

The player needs to press up and down to select the category and then use left and right to change each setting.

Connected to : Pause Screen

Pause Screen Pause Settings

UI : Pause Screen

The se



This is the screen where the controls of the game are reminded to the player in case they need it.

Connected to : Pause Screen

Pause Screen

Controls

UI : Checkpoint Select Screen



This screen allows to select the point in the game where the player begins to play.

Pressing the A button selects a zone and goes to the confirmation overlay, and then pressing A again launches the game.

Connected to : Main Menu, In Game

Screen

Checkpoint Select Screen

In Game

UI : Collectible Count



This screen shows how many collectibles the player accumulated, and rewards them with a specific title.

Pressing the A button sends the player back to the main menu.

Connected to : In Game

Collectible Count

In Game

UI : Return to Main Menu?



This screen is the confirmation screen to leave the game and go back to the main menu.

The player need to press down and A to confirm. Press B to go back to the Pause Screen.

Connected to : Pause Screen

Pause Screen Return to Main Menu?

Main Menu

UI : Settings



This screen features the settings of the game : a button to select a language, two sliders for the sound and music, a possibility to enable or disable the screen shakes and console vibrations.

The player needs to press up and down to select the category and then use left and right to change each setting.

Connected to : Main Menu

Main Menu

Settings

UI : Credits

and the



This screen displays the credits rolling upwards.

Press B to go back to main menu.

Connected to : Main Menu

Main Menu

Credits

General Production Risks

GD

Player metrics not balanced disrupting the game flow \rightarrow More time testing and tweaking

The LD doesn't properly showcase our core gameplay → To thoroughly iterate on precise LD game situations

The controls or the game are too tough to apprehend \rightarrow **To heavily playtest the game** and iterate

GP - GDP

Too many elements to code and implement → To keep an eye on the scope and iterate

The code is not optimized enough and makes the game slow down \rightarrow To simplify the code and avoid coroutines

The Switch development is trickier than planned \rightarrow To manage our time and the scope carefully

GA - MS

Too much 2D and 3D content to create \rightarrow To limit quantity of assets and LD bricks

The graphic integration is too complicated or unreadable → To ensure a simple and solid graphic chart

Lack of musical coherence and harmony → To avoid too many musical sound effects and randomization

ALL

Too many bugs and crashes in the game → More time testing and tweaking

Losing our progress and getting slowed down → To save and push regularly

The team starts to split up in vision or in work quantity → To actively listen to each other and inquire about others' work