



by

JOYNEST
STUDIO

Plan:

I) Intro : (page 2)

II) Member roles and task distribution: (page 2)

- Progress schedule:
- Presentation of our progress
- Assignment of tasks:
- Presentation of our tasks

III) State of the Project: (page 3)

- Maps : (page 3)
 - Initial design
 - Unity design
 - Challenges Encountered
- Puzzle Prototypes : (page 8)
 - Medieval
 - Egypt
- Game Mechanics : (page 14)
 - Movement
 - Object Interactions
- Improvement of Multiplayer and Dialogue Box: (page 18)
 - Multiplayer System Optimization
 - Dialogue Box Manager Implementation
 - Teleportation Menu
 - Character Sprite Updates and Movement Handling
 - Explaining the Dialogue Box to Teammates
 - Synthesis
- Characters (page 20)
- Website : (page 24)
 - Concept and Goals for the Website
 - Building the Structure
 - Adding Custom Features
 - Challenges and Solutions
 - Highlighting Key Aspects
 - Reflection and Lessons Learned
- Audio (page 29)

IV) Conclusion : (page 30)

I) Introduction :

The project *Back to EPITA* is more than just a demonstration of our growing skills in programming, game design, and project management—it is also a chance to explore the creative side of software development. This game blends quiz mechanics, split-screen cooperative play, and time-travel elements to provide players with both an immersive and educational experience.

Developing *Back to EPITA* is an ambitious endeavor that requires us to utilize a range of programming tools, software libraries, and game development frameworks. Along the way, we have faced and tackled numerous technical and organizational challenges. Despite these obstacles, our team has remained dedicated to crafting a refined and engaging gameplay experience.

This report details the current progress of our project, outlining the work accomplished, the difficulties encountered, and the solutions we have devised. It will also highlight each team member's role and contributions, emphasizing the collaborative effort that has driven the game's development. Additionally, we will discuss the methodologies, technologies, and tools that have shaped our work, reflecting on our learning journey and the growth we have achieved as motivated first-year students.

Through a combination of creativity, teamwork, and technical problem-solving, *Back to EPITA* aims to showcase how interactive digital experiences can be both educational and entertaining. This report encapsulates our journey so far, as well as the challenges and innovations that lie ahead.

II) Member roles and task distribution:

Progress schedule :

	progress monitoring presentation 1 (13 to 17 january)	progress monitoring presentation 2 (10 to 14 march)	progress monitoring presentation 3 (26 to 30 may)
--	--	--	--

Game programming	33%	66%	100%
Game design	25%	75%	100%
Game mechanics	70%	90%	100%
Graphism	20%	65%	100%
Multiplayer	60%	90%	100%
Audio	0%	40%	100%
Web site	30%	80%	100%

Task Assignment:

	Floralie	Aliya	Oceane	Ponnareay	Arthur
Game programming	R		S		S
Game design		R	S		
Game mechanics	S				R
Graphism		S		R	
Multiplayer	S				R
Audio		S		R	
Web site		S	R	S	

R : In charge / S: Support

III) State of the Project:

1) Maps :

In *Back to EPITA*, maps play a crucial role, serving not only as immersive settings but also as foundations for gameplay mechanics like puzzles and interactions. Each

environment is carefully designed to reflect the historical era it represents, such as the Middle Ages and Ancient Egypt, while also functioning as interactive spaces that encourage player cooperation.

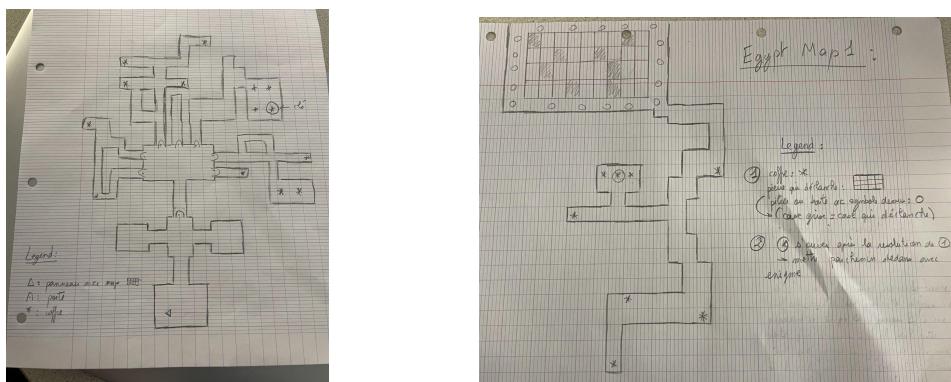
Our goal was to create rich and engaging environments that seamlessly integrate diverse challenges, leveraging asymmetric collaboration. At the same time, we aimed to ensure smooth progression while allowing players a degree of freedom in exploration.

1.1. Initial Design

The initial design phase of the maps was a pivotal stage in the development of *Back to EPITA*, establishing their structure, visual identity, and overall playability. This process began with a thorough analysis of the specific requirements of cooperative gameplay, focusing on how each map could enhance both mechanics and immersion.

Before transitioning to Unity, we created rough sketches to outline the general layout of each map. These drafts allowed us to visualize the arrangement of key areas, such as puzzle sections, exploration pathways, and interaction points. Every map was meticulously designed to strike a balance between aesthetics, fluid navigation, and a logical progression that enhances the player experience.

Example of drafts for the egypt maps:



To address the demands of asymmetric gameplay, we designed distinct but interconnected routes for each player. For instance, one area of a map might require a player to activate a mechanism to open a door or alter the environment for their partner. These plans also anticipated collaborative points where the two players must work closely to solve puzzles.

1.2. Unity Design

To design the game's maps, we chose to utilize pre-existing assets from the Unity Asset Store. This approach allowed us to save considerable development time while maintaining high visual quality, enabling us to concentrate on level design and customization. Using a single set of graphical elements, we created the game's first two levels: the Middle Ages and Ancient Egypt. This method provided a solid and cohesive foundation for crafting visually distinct and immersive environments.

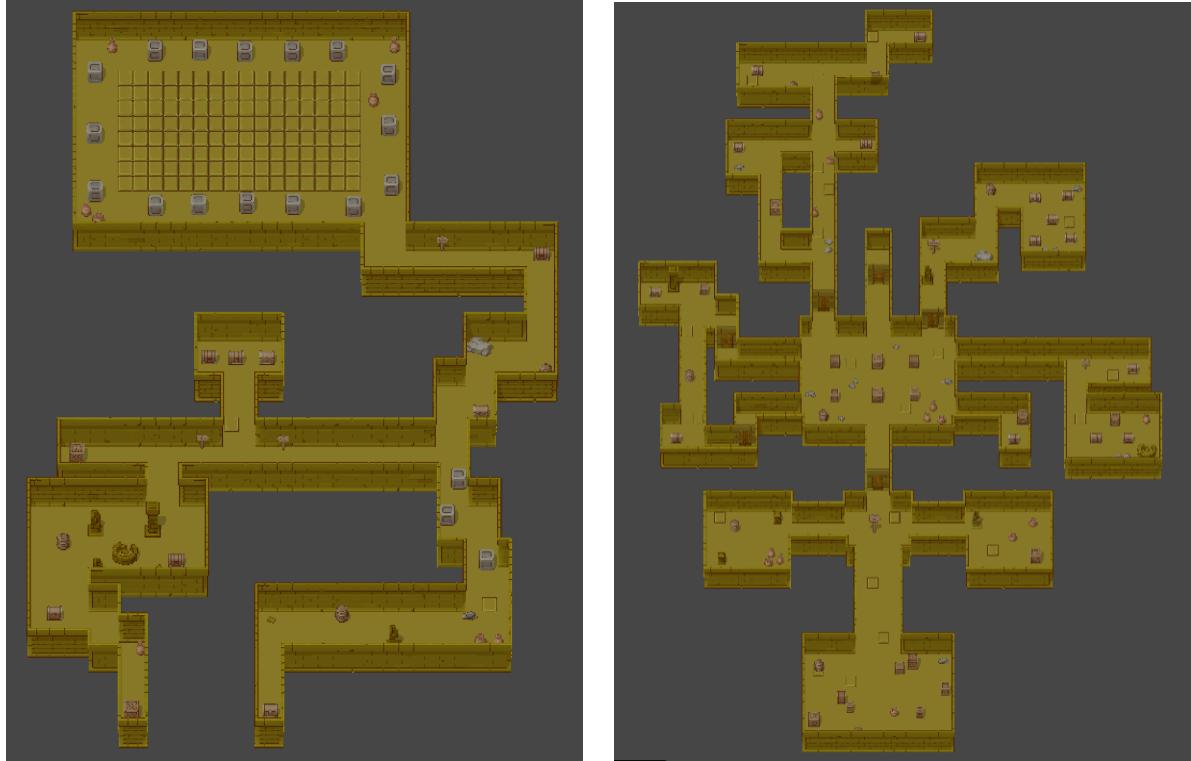
For each historical period, we developed two separate maps tailored for individual player exploration. This decision aimed to enhance replayability, offering players a fresh and engaging experience when switching roles. While each map is unique, they are carefully designed to align seamlessly with the visual and narrative themes of their respective time periods.

In the case of the medieval maps, we maintained a consistent overall structure to ensure coherence, while introducing significant variations in specific areas to accommodate unique puzzles for each player. For instance, one player might interact with a section featuring glowing pillars, while the other navigates a mysterious graveyard to solve puzzles essential for progression. These differences create complementary yet interconnected experiences, reinforcing the cooperative nature of the gameplay.



For the Egyptian level, the maps for both players have been fully completed. To maximize efficiency, we once again utilized the same assets from the medieval level while carefully selecting only the props that fit within the Egyptian environment. Additionally, several adjustments were made to enhance the setting's authenticity. For instance, the tilemap colors were modified to incorporate warmer tones, with

shades of yellow, sand, and ochre, evoking the desert landscapes characteristic of this era. These refinements allowed us to repurpose existing resources while ensuring the Egyptian level maintains a distinct and immersive identity.



Designing puzzle-specific areas required meticulous planning, particularly in terms of space distribution and player guidance throughout their progression. Each level had to be structured in a way that naturally led players toward their objectives while maintaining an engaging and immersive experience. This process involved carefully balancing open exploration with more confined puzzle areas to encourage collaboration without making navigation feel restrictive. Ensuring that players intuitively understood where to go and what actions to take, without explicit instructions, was a key challenge that demanded thoughtful design choices.

While the maps are now complete, we anticipate further refinements as we finalize the implementation of puzzles. These adjustments will allow us to tailor the level structure to the gameplay mechanics, ensuring a seamless and coherent experience. Refining movement paths, repositioning interactive elements, and adjusting environmental cues will help create a smoother flow of progression while reinforcing the cooperative nature of the game. This iterative process will also allow us to fine-tune level pacing, making sure that challenges remain engaging without causing unnecessary frustration.

Ultimately, map design was not just about placing assets—it required the deliberate integration of both visual and interactive elements to heighten immersion and reinforce the game's collaborative mechanics. Every historical setting was crafted to be more than just a backdrop; it plays an active role in the player experience, shaping how puzzles are approached and how teamwork unfolds. The result is a dynamic and varied gameplay experience, where each level transports players to a distinct historical period, offering unique challenges that encourage exploration, communication, and problem-solving.

1.3. Challenges Encountered

Designing and creating 2D maps in Unity presented several challenges, particularly when it came to maintaining both performance and visual coherence across the game. One of the major difficulties was managing Unity's **Tilemap** system. While the Tilemap tool is powerful, it can be tricky when working with large or complex maps. Organizing and structuring tilemaps effectively to avoid performance issues required a clear strategy, as managing multiple layers of tiles can quickly become overwhelming. Additionally, ensuring tiles align properly, especially when working with custom assets, proved to be time-consuming.

Another challenge was the management of layering and sorting. Sorting the correct layers in a 2D map can be difficult, particularly when elements need to interact across layers, such as characters passing in front or behind specific objects. Ensuring that objects didn't overlap incorrectly or fail to display in the correct order required constant fine-tuning of the sorting layers and order in layer settings. Mismanagement of these settings often led to visual glitches, where elements appeared out of place or behind other objects unintentionally.

Performance optimization was also a significant challenge, especially with large numbers of interactive elements. Each object added to the scene contributed to the computational load, and the more assets added, the higher the risk of lag, particularly on lower-end machines. We had to carefully consider each asset's impact on performance, ensuring that the game could run smoothly without compromising visual quality or gameplay experience.

Integrating assets into Unity also presented difficulties. While there were many resources available on the Unity Asset Store, ensuring they matched the game's aesthetic and time period required modifications. Some assets didn't fit the intended visual style, necessitating adjustments to textures, colors, or the overall

look of the environment. Moreover, certain assets weren't originally designed for 2D gameplay, leading to issues with resolution, scaling, and alignment, which needed to be addressed to maintain consistency across the levels.

These challenges provided valuable learning experiences and pushed us to refine our skills in Unity's 2D map creation. As we encountered and overcame these obstacles, we developed strategies to optimize the workflow, ensuring we could create polished and immersive maps for the game.

2) Puzzle Prototypes :

While researching and conceptualizing ideas for our puzzles, we realized that not all puzzles are suited for every type of gameplay experience. Some are better designed for solo play, while others are specifically meant for multiplayer interactions. Poorly crafted multiplayer puzzles can quickly diminish the quality of the experience. For example, a puzzle that requires minimal collaboration loses its significance in a cooperative setting. If one player can complete a task on their own while the other simply watches, the essence of cooperation is lost, which undermines the whole concept of multiplayer gameplay. Additionally, tasks that feel repetitive or disconnected from the main objective—such as carrying an object across a vast area with no real obstacles—can quickly become tedious and slow the overall pace of the game. Another pitfall is ignoring the skill differences between players. Puzzles that rely on specific or difficult knowledge or skills can alienate certain players, creating an imbalance and frustrating the less experienced participants.

To illustrate, let's take an example from the game *We Were Here Too*. In this game, one player must wait while the other solves a maze. If the player navigating the maze struggles, the waiting player may quickly grow bored, as their involvement is limited. This type of puzzle fails to foster cooperation, weakening the overall experience. Similarly, in *Keep Talking and Nobody Explodes*, players must work together to defuse a bomb, with one player handling the bomb and the others providing instructions from a manual. Some of the puzzles require complicated tasks, like performing mathematical calculations or solving algebraic equations. Players who are less comfortable with math may struggle, which can disrupt the cooperative experience for everyone.

With these design principles in mind, we made sure to craft our puzzles to promote meaningful collaboration. We only deviated from these guidelines when doing so

served a deliberate and beneficial purpose. Every exception was carefully considered to improve, rather than detract from, the overall experience.

When designing multiplayer puzzles, we prioritized ensuring that cooperation was essential, but we also aimed to keep the gameplay challenging and rewarding for both players. Each puzzle needed to create a sense of progression, with tasks that not only encouraged communication but also built upon previously discovered clues. This is where the balance between complexity and collaboration became key. In our levels, we created puzzles that required players to engage deeply with each other. They were designed to promote shared problem-solving, with challenges that increased in complexity over time. By building on each player's strengths and encouraging active participation from both sides, we made sure the gameplay remained engaging and cooperative.

Let's now explore how these design principles were applied to our specific levels, demonstrating how we crafted a cooperative experience that flows smoothly, maintains momentum, and keeps players fully engaged.

Medieval (likely to change):

4.2.1 "An Angel's Encoded Guidance"

Red Player's Experience

Players spawn in separate maps with complete freedom of movement. Red discovers a stone with cryptic inscriptions near their spawn point containing:

- A message about "a soft heavenly voice whispers of guidance for the lost"
- A hint suggesting their companion (Blue) might have heard similar tales
- A warning that "guidance will only befall those who are worthy"
- A cipher key showing symbols (Ѥ, റ, ല, ള) with corresponding shift values

A second pillar nearby provides crucial context: "An ancient king... mastered a secret language... twisted words, rearranging their letters... 'Veni, Vidi, Vici!'" This subtly indicates the use of a Caesar Shift Cipher, where letters are shifted along the alphabet.

Blue Player's Experience

Blue must search for divine symbols. Throughout their map, they discover:

- Candles with messages like "She watches over us" and "Her gaze appeases those who fell for their King"
- An Angel Statue that, when interacted with, reveals two encrypted messages:
 - "陛 SDANA | 皇 WKH | 朝 GHDG | 帝 XQJJUX"
 - "帝 AMMS | 朝 NBY | 朝 DPMVNOT | 朝 YP | 朝 ROMNZ"
- An alphabet guide (A-Z) to assist with decoding
- A candle with the message about loyal souls lingering, bound by "the rebellion's cruel hand"

Collaboration Mechanics

The puzzle requires players to share information:

- Red provides the cipher key: 陛 = +3, 朝 = +5, 朝 = -4, 皇 = -2, 帝 朝 = -1
- Blue shares the encrypted messages
- Players must realize that each word has a different shift value indicated by the symbol
- The symbols show what shift was applied to the original word, so players must apply the opposite shift to decode

For example, to decode "皇 YLECJ":

- 皇 indicates a -2 shift was applied to the original word
- Players must apply a +2 shift to "YLECJ" to get "ANGEL"

When correctly decoded, the messages reveal "Where the Dead Sleeps" and "Seek the columns of light" – clues directing players to their next locations.

4.1.2 : "The Royal Graveyard / The Trapped Souls"

Blue Player's Experience

Following the clue "Where the dead sleeps," Blue locates a graveyard on the southern part of their map containing:

- The King's cross-grave (CRG) with the message: "The fallen King's final resting place. You can feel holiness emanating from underground. Perhaps doing something honorable will reveal its secret"
- Eight additional graves (4 tall-graves and 4 short-graves)
- Each grave displays detailed information when interacted with:
 - Name and title (e.g., "Vernatius Willigher")
 - Years of life (e.g., "1355-1415")

- Description of their role and legacy (e.g., "Keeper of the Royal Library, Scholar at the Royal Academy and Seeker of Knowledge")
- Their connection to the King
- Sometimes their family coat of arms

Blue can either read the grave inscriptions or "Press F to pay Respect" at each grave.

Red Player's Experience

Following "Seek the columns of light," Red discovers an area with glowing pillars on the eastern side of their map:

- A stone with the message explaining that loyal servants' souls are trapped in pillars due to treachery
- Five glowing pillars arranged in a circular formation, initially dark
- Each pillar, when interacted with, reveals:
 - A Roman numeral (I through V) indicating sequence order
 - A riddle-like description of one of the deceased royal servants

The five pillars correspond to the Princess, Scholar, Chronicler, War General, and Steward.

Collaboration Mechanics

- Red must communicate the riddles and their Roman numeral order to Blue
- Blue must identify which grave matches each riddle and "pay respect" to them in the correct sequence
- Three of the eight graves are decoys and should not be honored
- Each correct grave honored lights up the corresponding pillar on Red's side
- If Blue pays respect to an incorrect grave, all lit pillars go dark, forcing them to restart the sequence

When all five pillars are illuminated, the souls are released with angelic music playing, and the King's grave begins to glow, signaling completion.

4.1.3 : "Honor Ernest and Set him Free"

Red Player's Experience

After completing Puzzle 2, Red receives a message about a loyal soldier buried in solitude "where the Sun rises." Red must:

- Find Ernest's lone grave on the far right side of the map
- Read the gravestone: "Sir Ernest, 1338-1384, 'Here rests Sir Ernest the Unyielding, who stood alone against many...'"
- Interact with Ernest's soul who requests: "Take me to the false throne, the seat atop the highest point, where the Sun falls upon hollow glory"
- Locate the specified bench (the "false throne") at the highest point of the map
- When Ernest reaches this location, his soul is freed, and he drops a time travel device ready for use

Blue Player's Experience

Blue must:

- Interact with the now-glowing King's grave to receive the "Key of Kings"
- Follow the message to find a chest "atop the highest ground" of their map
- Use the key to open the chest, obtaining a powerless time device
- Read the message instructing them to go "where day begins" (the east side of the map)
- Find and sit on a bench on the east side to activate the device

Completion

When both players activate their time travel devices simultaneously, they are transported to the Egyptian level, completing the Medieval section of the game.

Egypt:

4.3.1. The Map and the Symbols on the Ground

The first player discovers an ancient map with enigmatic drawings and specific symbols. While exploring the environment, they notice that these symbols match markings on the ground in the second player's area. Each symbol is linked to a particular object that the second player must place in the correct locations on the ground.

Objective: The first player must communicate with the second player to indicate the correct places where specific objects should be placed. These objects bear symbols corresponding to those on the map. The second player, who can see the markings on the ground, must place the objects in the right locations based on the

hints provided by the first player. When they succeed, a code or a hidden mechanism is triggered, unlocking a new area or access point.

The goal of this puzzle is simple: encourage communication. The first player will explore their map until they find a grid on the ground, while the second player will attempt to decipher a parchment containing different symbols. Their objective is to communicate so that the first player places the objects in the correct spots according to the second player's indications.

Map 1:

- A grid on the ground with different stones and objects featuring various symbols.
- Different symbols to match objects.

Map 2:

- A panel displaying a grid and the different symbols corresponding to the ones on the objects.
- A parchment (to be designed) containing hints.

4.3.2. The Chests and the Egyptian Riddle

After solving the map and symbols puzzle, the first player encounters several locked chests in their environment. Each chest requires a specific hint to determine which one should be opened. However, the first player lacks the crucial information needed to identify the correct chest.

Objective: The second player, in a different area, discovers an Egyptian riddle (an ancient hieroglyph or a secret inscription) which, once deciphered, provides a clue about the correct chest to open. By solving the riddle, the second player gives the necessary information to the first player, who can then open the correct chest. Once the right chest is opened, a mechanism or a new area is unlocked, allowing the players to progress further.

Once the first puzzle is solved, a chest and the first door in Map 2 open within Map 1. Inside this chest, the players find a new riddle (Riddle 1) instructing them to locate another chest bearing the same symbol.

In Map 2, there are seven locked doors, each leading to a hidden area containing chests. To access these areas, the first player must successfully keep a door open by

playing a mini-game, allowing the second player to enter. If the first player fails, the second player is immediately teleported back in front of the door.

Map 1:

- **Riddle 1:** The parchment contains a symbol and an enigmatic phrase:
"Like the sand of the desert, each chest holds a fragment of the past, a key to reveal what is hidden."
Along with this phrase, an image is displayed as a clue.

Map 2:

- Empty chests contain messages such as “**Try again**”.
- The correct chest contains a key and **Riddle 2**. After solving the riddle, the players gain access to the key and can be teleported to the next world.

Riddle 2: *I am an ancient legend, built by forgotten hands, my stones will guide you to a secret. Who am I?*

Answer: *Pyramid*

When the player enters *Pyramid*, they successfully complete the level and gain access to the key or the next objective.

3) Game mechanics :

3.1. Movements: Managing Physics and Conditions

For the character movement, we focused on simplicity and precision, allowing players to move their characters in four cardinal directions: up, down, left, and right. We used Unity's Input System to process directional inputs, translating them into velocity updates for the characters' Rigidbody2D components.

Collision Setup

To manage collisions efficiently, we used both BoxCollider and PolygonCollider components. BoxCollider was primarily applied to simple, rectangular objects where a straightforward collision boundary sufficed, optimizing performance by reducing unnecessary complexity. Meanwhile, PolygonCollider was used for irregularly shaped objects, allowing for precise hitboxes that better matched the

contours of the environment and interactive elements. Carefully selecting the appropriate collider type for each object helped ensure accurate detection while minimizing computational overhead, contributing to smoother gameplay and improved performance.

For objects requiring dynamic physical behavior, we integrated Rigidbody components. Unity then handled their movement and physical interactions automatically, accounting for parameters like gravity, friction, and constraints we have set. This system was particularly useful for simulating realistic behaviors in puzzles or mobile scenery elements, minimizing the need for complex scripting.

Collision Management and Detection

Collision events were detected using specific C# methods, such as OnCollisionEnter, OnCollisionStay, and OnCollisionExit. These functions enabled us to respond dynamically to various collision-related events, whether at their initiation, during their occurrence, or after they ended. By leveraging these methods, we could retrieve critical information—such as identifying the objects involved—and adjust effects or behaviors accordingly. For example, this data allowed us to trigger animations, modify object states, or activate interactive mechanisms essential for gameplay.

Triggers for Non-Physical Collisions

In cases where collision detection was needed without triggering physical behaviors, we used triggers. These zones detect interactions without enforcing physical constraints, proving particularly useful for detection zones or complex mechanism activations. For example, triggers were used to handle events like opening doors or starting visual effects without disrupting the scene's physical balance.

This implementation of collisions and interactions allowed us to create a robust and flexible structure. By fully leveraging Unity's capabilities, we designed an immersive and precise interaction system vital to the game's mechanics, ensuring a smooth and coherent player experience.

Physics in Puzzles

Beyond basic movement, physics elements play a crucial role in some of the game's puzzles. Characters can interact with blocks or objects they must push or move to

create paths or activate mechanisms. These physics-based interactions require precise cooperation between the two players, reinforcing the importance of teamwork.

3.2. Interactions: Collaboration at the Core of Gameplay

Interactions between players and the environment are the heart of Back To Epita. Each puzzle is designed to demand close cooperation between the two participants, highlighting the importance of communication and mutual support.

Collaborative Puzzle Design

For instance, some puzzles require simultaneous activation of mechanisms, like levers or switches, located in different areas of the map. Players must coordinate their actions precisely to progress.

Another approach involves separating exploration zones for each player. These areas are distinct but interdependent, forcing players to collaborate to solve puzzles. In the medieval level, for example, one player may activate glowing pillars visible only to their teammate, who then uses these clues to decipher a puzzle. This type of interaction emphasizes the collaborative nature of the game, making each player indispensable to progress.

Interactive Objects

Numerous interactive objects enhance the gameplay experience, allowing players to engage with their environment in meaningful ways. Players can interact with objects to discover hidden keys, reveal text written on surfaces, or uncover other clues vital to solving riddles. These interactions encourage exploration and create moments of discovery, adding depth to the overall experience.

Triggers also play a pivotal role in managing these interactions. These invisible zones detect player actions and initiate specific events, such as revealing hidden objects, activating timed puzzles, or displaying narrative elements like dialogue or inscriptions. By incorporating these mechanics, we ensured an engaging and immersive gameplay dynamic that rewards curiosity and teamwork.

Interactive Dialogue System for Puzzle Clues and Enigma Solving

The system was implemented to allow players to interact with objects in the game and display text in a dialogue box based on the object they interact with. This

functionality is managed by the DialogueManager script, which handles both the interaction and the display of text in the dialogue box.

The script begins by finding the dialogue box and its associated text component (TMP_Text). Initially, the dialogue box is set to inactive to ensure it doesn't appear before an interaction occurs. When the player interacts with an object, the method RpcShowDialogue is triggered. This method uses a Remote Procedure Call (RPC) to ensure the dialogue is displayed to the local player interacting with the object. It receives a tag as a parameter, which corresponds to a specific object in the game, such as graves or stones. Based on the tag, the method calls SelectMessage, which returns a pre-defined message associated with that tag. For instance, if the player interacts with an object tagged "Stone 1 red," the message, "Amongst the stones, a soft heavenly voice whispers..." is shown in the dialogue box.

Once the player finishes reading the message, the HideDialogue method can be called to hide the dialogue box. This ensures that the dialogue does not remain on the screen when it is no longer needed. The system checks the local player's interaction to ensure proper functionality, particularly in the multiplayer setting.

This system will be particularly useful for implementing all the enigmas based on written puzzles that players will need to decode. By displaying context-specific text in the dialogue box, it allows for the presentation of cryptic clues or puzzle hints that players must decipher. This will help create a more engaging and interactive puzzle-solving experience, where the clues are tied directly to the objects the players interact with, fostering deeper collaboration and critical thinking.

In summary, this system enables dynamic interactions with in-game objects, ensures that context-specific text is displayed in the dialogue box, and is perfectly suited for puzzles requiring written clues to be decoded, all while maintaining synchronization across players in a multiplayer environment.

Challenges Encountered

While implementing these features, several challenges arose that required careful consideration and problem-solving. One of the primary difficulties was identifying the most appropriate tools and pre-implemented functions in Unity for managing interactions effectively. For instance, choosing between OnTriggerEnter and OnCollisionEnter necessitated experimentation and analysis to determine which was best suited for the mechanics we aimed to implement—such as triggering text

displays without disrupting the physical flow of the scene. This decision was critical because it impacted the smoothness and accuracy of the interactions.

Another challenge involved debugging the code. Unity's debugging tools can be both powerful and tricky, especially when dealing with interactions triggered by multiple objects at once. Identifying why certain interactions failed to trigger or how they were being processed in the wrong order was time-consuming. Debugging these scenarios often revealed hidden dependencies between objects or components that weren't initially obvious. Unity's interface can sometimes make it difficult to track down the exact cause of an issue, especially with complex interactions that require precise timing and coordination between game elements. This process involved a lot of trial and error, as well as the need to test under a variety of conditions to identify all possible edge cases. We overcame many of these obstacles through iterative refinement and by adding robust error handling to prevent interactions from breaking under unexpected circumstances.

In addition, finding an efficient way to detect interactions in a way that didn't interfere with the game's performance proved to be difficult. Handling collisions and triggers without creating redundant checks or adding unnecessary complexity was a delicate balance. This required optimization to ensure that the game would run smoothly, especially in multiplayer settings where performance is critical. Issues like these forced us to reconsider how we approached interaction detection, experimenting with different methods to optimize both functionality and efficiency.

Lastly, designing interactions that felt intuitive and rewarding for players required fine-tuning the placement, timing, and visual feedback of interactive elements. Ensuring players could easily discern what was interactable and how to interact with objects without confusion was a key challenge. Developing clear indicators and responsive mechanics became essential to align with players' expectations, making the experience seamless and immersive.

These hurdles highlighted the importance of a deep understanding of Unity's tools, iterative testing, and careful consideration of user experience. By addressing these challenges, the team was able to create a robust and immersive dialogue system that supports meaningful puzzle interactions, contributing to the cooperative gameplay experience that Back to Epita is built around.

4) Improvement of Multiplayer and Dialogue Box:

Multiplayer System Optimization

Since the last defense, my primary focus has been on improving the multiplayer system using Mirror in Unity. Given that our game is strictly two-player, I optimized network usage by limiting the maximum connections to two. This prevents unnecessary overhead and ensures that the server does not handle excess connections, improving efficiency. Additionally, reducing the number of connections allows for smoother synchronization and better handling of player interactions without unnecessary network congestion.

One key improvement was in how player actions are managed over the network. Previously, each player had separate scripts handling interactions, leading to redundant code and potential synchronization issues. I refactored the system to use a single script to handle both players, ensuring more efficient event management. This consolidation reduced the risk of desynchronization and provided a more streamlined way to manage game events. For example, the dialogue box now only appears for the player who interacts with an object, instead of being triggered for both players. This prevents unnecessary UI updates and keeps interactions more immersive, ensuring that each player experiences the game world in a consistent manner.

Additionally, I refined how the network synchronizes player states and actions, making sure that only necessary data is sent across the network, reducing bandwidth usage. This was achieved by implementing targeted Remote Procedure Calls (RPCs) and utilizing the NetworkTransform component more effectively. By carefully selecting which data needs to be synchronized and optimizing its transmission, I was able to minimize unnecessary data transfers, thus improving overall performance and reducing lag.

Dialogue Box Manager Implementation

Another major addition was the dialogue box system, which allows players to interact with objects in the game world. This system was built using a simple UI overlay that activates when a player faces an interactive object, such as a grave, and presses a designated interaction key. The dialogue box system was designed to be lightweight yet effective, ensuring that interactions remain smooth and do not introduce delays or performance issues.

To ensure accurate interactions, I implemented a Raycast system that detects objects in front of the player. Each interactable object is assigned a specific Tag, allowing the system to recognize whether an object should trigger the dialogue box. This tagging system ensures that only specific objects activate the dialogue UI, preventing unintended interactions and making the system more robust.

The UI is purely local, meaning that it only appears for the player performing the interaction and does not sync across the network. This prevents unnecessary synchronization calls and maintains a smooth multiplayer experience. By keeping the dialogue system independent for each player, I was able to reduce network overhead while maintaining a seamless and responsive UI.

Teleportation Menu

To facilitate debugging and navigation, I developed a teleportation menu that allows players to instantly move between different game worlds. This menu is accessed through a button in the corner of the screen, which, when clicked, opens a UI panel containing multiple teleportation buttons. The teleportation system was designed with flexibility in mind, allowing for quick transitions between game areas without disrupting gameplay.

Each button corresponds to a predefined teleport location. When a player selects a destination, their Transform position is updated locally, allowing for instant movement. Since the player object is a NetworkObject, the server automatically propagates the position change to both players, ensuring that each player's location is correctly updated across the network. This ensures consistency in player positioning while minimizing network latency.

Because this feature is primarily intended for debugging, no animations or cooldown mechanics were added to the teleportation process. However, the system remains flexible and could be expanded in the future if needed. For instance, visual effects or transition animations could be added later to enhance the immersion of the teleportation feature.

Character Sprite Updates and Movement Handling

Lastly, I made changes to the player's appearance by updating the character sprite. This was done by modifying the player prefab, ensuring that the new sprite is applied consistently across all instances of the player object in the game. The new

sprite was chosen to improve character visibility and enhance the overall aesthetic of the game.

To ensure the correct facing direction, I implemented a sprite rotation system that adjusts the player's visual representation based on movement input. The system ensures that the player sprite correctly rotates when moving left, right, up, or down. While I don't recall the exact method used for animation handling, the approach involved flipping or rotating the sprite appropriately based on movement direction. If necessary, an Animator could be integrated later to further refine sprite transitions and ensure smoother animations.

Since these changes are applied to the player prefab, they automatically sync across the network, ensuring that both players see the correct sprite orientation at all times. By making sure the sprite updates dynamically based on player input and network conditions, I was able to create a consistent visual experience for both players.

Explaining the Dialogue Box to Teammates

In addition to implementing the dialogue box system, I took the time to explain its functionality and integration process to my teammates. This ensured that everyone on the team understood how it works and could make use of it effectively in their own implementations. By providing clear explanations and documentation, I helped facilitate smoother collaboration and prevented confusion regarding its usage.

Synthesis

In summary, the improvements made to the multiplayer system have significantly enhanced both performance and gameplay experience. By consolidating scripts and reducing unnecessary network traffic, the game now operates more efficiently with fewer risks of desynchronization. The addition of the dialogue box system and its local-only functionality ensures that interactions remain smooth and immersive. The teleportation menu, while primarily designed for debugging, provides a quick and effective way to navigate between game worlds. Lastly, updating the character sprite and refining its directional handling has improved the visual quality and responsiveness of the player character.

These modifications not only optimize the game's functionality but also lay the groundwork for future improvements. The systems implemented are designed to be

scalable and adaptable, ensuring that as new features are added, they can be integrated seamlessly without disrupting the core gameplay mechanics.

5) Characters

In the development of our game, acquiring high-quality assets was a crucial step in creating a visually appealing and engaging experience for players. To expedite the development process while maintaining artistic quality, we decided to purchase assets from the Unity Asset Store as a foundation for the game's 2D characters. This approach allowed us to establish a solid starting point and then customize the assets to align with the specific artistic direction and thematic elements of our game.

Acquisition of Assets from the Unity Asset Store

The Unity Asset Store is a widely recognized marketplace offering a variety of assets, including 2D and 3D character models, animations, environments, and user interface elements. After careful consideration, we selected a set of 2D character assets that closely matched our game's visual style and aesthetic requirements. These assets provided well-designed characters with high-resolution sprites, layered components for customization, and animations that could be adapted to fit our needs.

By purchasing pre-made assets, we were able to save valuable development time that would have otherwise been spent on designing characters from scratch. This decision enabled us to focus on refining gameplay mechanics, level design, and story elements while ensuring that our characters remained visually appealing and unique to our project.

Modifications and Customization

While the purchased assets served as a strong starting point, they required significant modifications to better fit our game's theme and character identity. Several key aspects were altered to ensure uniqueness and alignment with our vision:

- 1. Character Features Adjustments** – The original designs provided a base structure, but modifications were necessary to ensure each character had a

distinct personality. Facial expressions, hairstyles, and body proportions were adjusted to create more diversity and originality among characters.

2. **Clothing and Equipment Redesign** – The default outfits and accessories included in the purchased assets did not entirely fit the theme of our game. Using digital painting and sprite editing techniques, we modified the clothing styles, colors, and materials to match the lore and setting of our game world. Additional accessories and props were also created to enhance the depth of customization.
3. **Animation Tweaks and Enhancements** – Some animations required refinements or additional frames to improve smoothness and responsiveness. Using animation software, I adjusted movement sequences, idle animations, and attack sequences to better suit the gameplay dynamics.
4. **Character Editing in Unity** – After modifying the assets externally, I imported them into Unity for further customization. Within Unity, I adjusted the sprite sorting layers, tweaked animation states using the Animator tool, and implemented sprite swapping techniques to enable dynamic character variations. I also utilized Unity's 2D rigging system to enhance character movement and responsiveness, ensuring they fit seamlessly into the game environment.

Benefits of This Approach

The decision to acquire and modify pre-existing assets had multiple benefits, including:

- Time Efficiency: By leveraging pre-made assets, we accelerated the development timeline while ensuring quality.
- Cost-Effectiveness: Instead of commissioning entirely custom artwork from scratch, modifying existing assets provided a balance between affordability and originality.
- Flexibility and Scalability: The asset modifications allowed for easy expansion, ensuring new character variations could be quickly created as needed.
- Improved Game Cohesion: Customizing assets ensured that all characters aligned with the overall artistic vision and narrative of the game.

Purchasing assets from the Unity Asset Store proved to be a strategic decision in the development of our 2D game. While the acquired assets provided an essential foundation, the extensive modifications performed ensured that they were fully

adapted to our unique game world. This approach not only saved time and resources but also allowed for a level of customization that met our creative and gameplay expectations. Moving forward, these customized assets will serve as the basis for expanding our character roster and refining the game's visual experience.

The 1st character:



The 2nd character:



6) Website:

As part of our project to design and develop a video game, we are also tasked with not only creating the game but also building a professional presence to showcase our work. At first, we had decided to use AI tools to streamline the process while ensuring the final website was engaging, functional, and visually appealing. However, since it doesn't represent our ability to code and create a website, we created it from scratch using JavaScript. The website serves as a digital hub to represent our company and promote our game. It includes three main sections: a home page, a game page, and a page about Joynest Studio, the team members and the journey behind the concept of our game.

6.1. Concept and Goals for the Website

The primary goal of the website was to effectively communicate the essence of Joynest Studio, showcase our game, and highlight the contributions of our team members. To achieve this, we outlined several key objectives:

- 1. User-Friendly Design:** The website needed to be intuitive and easy to navigate. Each section should provide clear information while maintaining a visually cohesive layout.

2. **Highlighting the Game:** As the centerpiece of our project, the game deserves special attention. It was essential to create a dedicated page that would draw visitors in and provide detailed information about the game's concept, features, and visual assets. We also did not forget to include a download link of our project report and in the future when the game is ready, we will be sure to include a link to download the game.
3. **Showcasing the Team:** Our team worked collaboratively on this project, so it was important to recognize and show each member's role and contributions. We also showed how we have progressed, the issues we have encountered and the solutions found.

Having little but some experience with JavaScript, we managed to create something simple yet perfect for this project that corresponds to our needs.

6.2. Building the Structure

We began by using AI to design and give us a general appearance for the website. Using Visual Studio Code we then created the website following the design, and adapted it to what we knew how to do. We aligned with Joynest Studio's branding by incorporating our logo, color scheme, and typography. The structure of the website includes three main pages:

1. **Home Page:** This page serves as the gateway to our website. It features a visually striking hero section with an image related to our game and a tagline that encapsulates Joynest Studio's creative vision. Below the first section, we included a brief introduction to our studio and links to additional information.
2. **Game Page:** The game page is dedicated to our project—a video game that represents the culmination of our efforts. We used this page to explain the game's premise, highlight its unique features, and showcase images. We also embedded a few photos to give visitors a glimpse of the gameplay. Additionally, the page featured detailed sections on the game mechanics, its target audience, and the inspiration behind its development. Links to tools, software, and libraries we used in game development were also included to credit resources and offer transparency.
3. **About Us Page:** This page focuses on Joynest Studio as a company and introduces the team members. We included a brief history of the studio, our

mission statement, and profiles of each team member. Each profile features a photo, name, and a description of their role within the project. We also added a section on the values that drive Joynest Studio, emphasizing creativity, teamwork, and innovation. Additionally, we incorporated a timeline showcasing the key milestones in the project, from the initial brainstorming to the final presentation.

6.3. Adding Custom Features

To make the website more interactive and engaging, we incorporated several features:

- **Dynamic Animations:** Subtle animations were added to enhance the user experience. For example, buttons change slightly when hovered over. This created a sense of movement and modernity.
- **Call-to-Action Buttons:** We included prominent buttons on each page to guide users, such as “Start your Adventure,” and “Explore more”. These buttons were styled to stand out and encourage engagement.
- **Download Links:** We added download links to our website to enhance accessibility and user experience by allowing visitors to easily obtain important resources. These links enable users to download our game, including the full version and beta releases, ensuring they can test new features and provide valuable feedback. Additionally, we included download options for reports and other key documents, allowing users to visualize data and insights offline. By providing these direct download links, we improve convenience, encourage engagement with our content, and support our community in accessing the materials they need efficiently.

6.4. Challenges and Solutions

Creating the website came with its fair share of challenges. One of the biggest hurdles was achieving a cohesive and professional design within a tight timeframe. Additionally, learning to use JavaScript and integrating unfamiliar elements proved to be a significant challenge. Understanding the language while also ensuring the site was visually appealing required both time and effort. Despite these obstacles, overcoming them was a valuable learning experience that ultimately improved our technical skills and design capabilities.

6.5. Highlighting Key Aspects

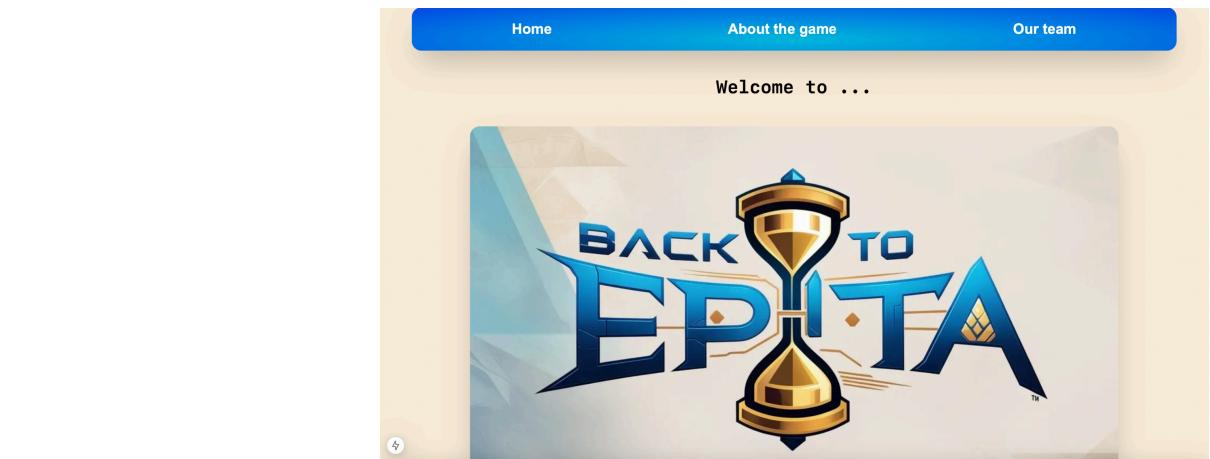
Throughout the design process, we paid special attention to certain elements to ensure they stood out:

- **Visual Appeal of the Game Page:** This page was the centerpiece of the website, so we used bold visuals, large images, and an eye-catching layout to draw visitors in. Additional features such as the members' opinions, and an encouraging words section helped break down the game's unique selling points.
- **Team Member Profiles:** We wanted each team member to feel recognized, so we designed the profiles with a consistent format, including professional-style headshots and personalized descriptions.
- **Resource Links:** To provide transparency and value to visitors, we included a section that listed the tools, software, libraries, and assets we used during development. This served both as a credit to these resources and as inspiration for others looking to create similar projects.

6.6. Reflection and Lessons Learned

Creating the website for Joynest Studio is an experience that teaches us the importance of planning, adaptability, and leveraging technology effectively. This experience also underscored the value of collaboration, as input from everyone helped shape the content and design choices.

In the end, the website successfully showcases our game and studio in a professional and engaging way. It serves as a platform to celebrate our work and demonstrates the potential of combining creativity with technology. The website is also a reminder of the importance of balancing efficiency with creativity. By prioritizing user experience and branding, we were able to deliver a polished product.



Home About the game Our team

Dive into a world of mystery and adventure with Back To EPITA, where every riddle reveals the story and every step can't be made alone.

[Start your adventure →](#)

Ponnareay
Audio and Graphics Leader

"Elements of Back Two bring the immersive atmosphere that makes the game world to life."

Floralie
Game developer

"Crafted with passion, this game is a masterpiece of storytelling and gameplay."

Océane
Project Manager

"Our team has poured their hearts into creating an unforgettable experience for players."

Arthur
Game Developer

"Back Two EPITA offers an immersive experience that keeps you on the edge of your seat..."

Discover the talented individuals who bring creativity and passion to our captivating world.

About the team

We're a dynamic group of individuals who are passionate about what we do and dedicated to delivering the best results for our clients.

Océane Delogé
Project Manager

Born in Kenya, Océane grew up developing a love for video games. Her passion for IT started in her early school years and she pursued her career as a computer engineer at EPITA.

[in](#)

Aliya Aparicio
Gameplay Leader

Originating from Clermont-Ferrand, France, Aliya has always been fascinated by numerical interfaces. At EPITA, she quickly blossomed in this sector by perfecting her competencies in programming and game design.

[in](#)

Floralie Niort
Game Developer

Born in the south of France, Floralie has been brought up in the universe of video games from a very young age. At EPITA, she acquired solid foundations, as she got familiar with multiple programming languages.

[in](#)

Arthur Brune
Game Developer

Originating from Ain, France, Arthur is an expert in programming and has a passion for creating solid infrastructures. He also has previous experience for developing mobile games.

[in](#)

Ponnareay Rith
Audio and Graphics Leader

8) Audio (Pending Implementation)

Ambient music and sound effects are essential components for creating a deep sense of immersion and enhancing the overall gaming experience. This is especially important in a game that spans multiple time periods, as sound plays a crucial role in establishing the atmosphere, intensifying emotions, and supporting the narrative. By distinguishing each era with its own soundscape, we can make the players feel fully immersed in the different worlds they travel through. For instance, classical instruments can evoke the ancient past, while electronic sounds can transport players into a futuristic setting.

Sound effects, like the opening of a time portal or unique noises for each time period, offer immediate feedback to players, helping them navigate the world and stay oriented. Additionally, music has the power to shape the pace of the game, adding excitement to action sequences or bringing a sense of tranquility to slower moments. In a game that spans different eras, sound is key to enriching the player's experience, making each period distinct and memorable.

In our game, music holds a special significance, particularly in relation to one of the puzzles we've designed. In this puzzle, the music itself is an integral part of progression. We've crafted a subtle interaction between the player and the game's soundscape, using music to guide players through clues and mechanics that can only be uncovered through careful listening.

To achieve this, the music will vary slightly over time, with repetitions or interruptions that allow players to piece together the correct sequence of actions. This puzzle would not be possible without a thoughtfully composed soundtrack that complements the game's atmosphere and context, while also providing key auditory clues for solving the challenge. To bring this vision to life, we plan to compose our own original music, aided by access to a MIDI keyboard. Through this approach, music becomes a storytelling tool and a means of interaction within the game world, deepening immersion and offering players a unique auditory experience driven by collaboration and perception.

IV) Conclusion

In summary, the development of our game has seen significant progress in both gameplay mechanics and narrative integration. Over the course of this phase, we have successfully implemented puzzle mechanics across various interactive objects in the first map, allowing players to engage with the environment in a meaningful way. Each object serves as a puzzle element, contributing to the progression and adding complexity to the gameplay experience.

We have also designed and integrated a user-friendly menu that serves as the central hub for players to interact with the game's settings, navigate through different sections, and manage their gameplay experience. The menu design aims to provide a seamless interface, ensuring that players can easily access and adjust features without disrupting the flow of the game.

In addition to these gameplay elements, we have made significant strides in character creation by implementing two-player functionality with customized characters. These characters bring unique personalities and abilities to the game, enhancing the cooperative experience and offering varied gameplay styles for the players to explore.

One of the major highlights of our progress lies in the puzzle mechanics. We've carefully crafted a series of enigmas that span across different environments and time periods. Each puzzle is designed with a mix of interaction, observation, and critical thinking, requiring players to collaborate and communicate to solve them. From manipulating objects in the game world to interpreting clues hidden in the environment, each puzzle serves to challenge players in new ways, enhancing the depth and replayability of the game.

Overall, these advancements in puzzle design, character implementation, menu creation, and collaborative mechanics are setting the foundation for a compelling and immersive gaming experience. Moving forward, we aim to refine these systems further, enhancing both the complexity and the overall cohesiveness of the game to create a truly engaging experience for players.