

RAPPORT DE SOUTENANCE 1

Fragments of Ember

par Emberfall



Thomas BOBEE
Lucas ETIENNE
Hugo GUYENNET
Alexandre d'AVEZAC de CASTERA

Table des matières

1	Introduction	2
1.1	Objectif et intérêts pour ce rapport	2
1.2	Présentation du rapport	2
1.3	Rappel des différentes parties et avancement	2
2	Interfaces et mécaniques de jeu	4
2.1	Mécaniques de jeu	4
2.1.1	Gestion du Chargement de Scène	4
2.1.2	Mise en pause du jeu	4
2.1.3	Nouveau script pour les interactions	5
2.1.4	Mise en place du mode multijoueur	6
2.2	Interfaces	7
2.2.1	Menus en jeu	7
2.2.2	Menu écran titre	8
3	Univers du jeu	9
3.1	Cartes	9
3.2	Objectifs du jeu	10
3.3	Assets	11
3.3.1	Son	13
3.3.2	Difficultés rencontrées	13
4	Multijoueur	14
4.1	Réseau	14
4.2	Mode de jeu	14
5	Le site web	15
5.1	État actuel	15
6	Conclusion	16
6.1	Avancement du projet	16
6.2	Plan pour le futur	17

1 Introduction

Le projet Fragments of Ember est un jeu vidéo d'aventure et de puzzle développé par notre équipe, Emberfall , dans le but de créer une expérience immersive et réfléchie pour les joueurs. Ce jeu s'inscrit dans un univers riche où chaque salle représente un défi unique, alliant exploration et résolution de mystères. L'objectif principal de ce projet est de concevoir un gameplay original qui stimule à la fois la réflexion et l'interaction avec un environnement captivant.

1.1 Objectif et intérêts pour ce rapport

Dans les sections suivantes, nous détaillerons la conception du gameplay, l'architecture technique, l'interface utilisateur et l'aspect narratif de Fragments of Ember, ainsi que l'approche artistique qui soutient l'ambiance mystérieuse et envoûtante du jeu. Enfin, nous discuterons des perspectives d'évolution du projet et des projets futurs qui pourraient enrichir l'expérience de jeu.

1.2 Présentation du rapport

Ce rapport de soutenance présente les étapes clés de la conception et du développement de Fragments of Ember, en détaillant la vision créative de l'équipe, le processus de développement technique, ainsi que les choix artistiques qui ont façonné l'univers du jeu. À travers ce document, nous mettrons en lumière les défis rencontrés et les solutions mises en place pour offrir un jeu fluide, innovant et captivant. Nous aborderons également la manière dont nous avons intégré les retours des tests et les améliorations continues, afin de garantir une expérience de jeu optimale.

1.3 Rappel des différentes parties et avancement

TABLE 1 – Avancement des tâches par soutenance

Tâches	Soutenance 1	Soutenance 2	Soutenance finale
Scénario	100%	100%	100%
Multijoueur	0%	50%	100%
Réseau	0%	50%	100%
Son	0%	25%	100%
Modèles 3D	10%	70%	100%
Animations	0%	50%	100%
Site Web	65%	85%	100%
Gameplay	5%	85%	100%
Ajout de l'IA	0%	50%	100%
Menu	10%	100%	100%
Documentation	40%	70%	100%

Tâches	Alexandre	Thomas	Hugo	Lucas
Gameplay				
- Mouvements		R		S
- Interaction		R		S
- Pouvoirs		R		S
Graphisme				
- Design	R		S	
- Environnements et niveaux	R		S	
- Animations	S		R	
Intelligence artificielle				
- Comportement des ennemis			S	R
- Interactions avec le joueur			S	R
Multijoueurs				
- Mécanique coopérative	R			S
- Synchronisation des joueurs	R			S
- Réseau	R			S
Site Web				
- Conception		S		R
- Documentation		R	S	
Sons				
- Musiques		S	R	
- Effets sonores		S	R	
Scénario				
- Lore	S			R
- Développement des chapitres		S	R	
- Dialogue	R		S	

TABLE 2 – Répartition des tâches par membre de l'équipe

2 Interfaces et mécaniques de jeu

2.1 Mécaniques de jeu

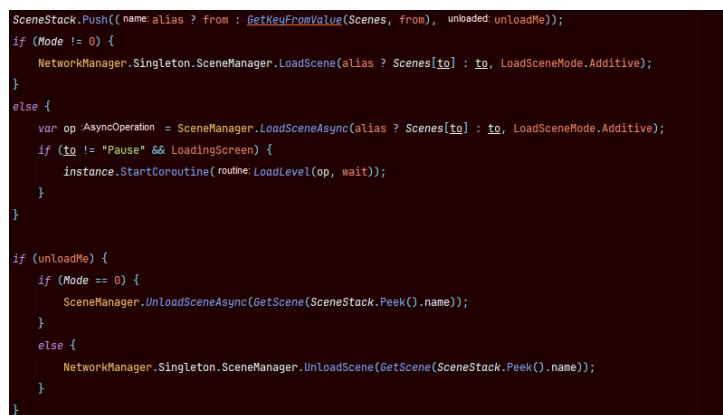
2.1.1 Gestion du Chargement de Scène

Le système fonctionne en empilant les scènes chargées dans une structure SceneStack, permettant ainsi de suivre les transitions effectuées. Lorsqu'une nouvelle scène est chargée, elle est ajoutée à la pile avec son identifiant (name) et une indication sur la nécessité de décharger l'ancienne scène (unloadMe).

Avant de charger la scène, on l'ajoute à la pile SceneStack, qui contient toutes les scènes du jeu qui ont été chargées.

Chargement de la scène : Si le mode de jeu (Mode) est 0 (Menus) on utilise le chargement de scènes unity via SceneManager.LoadSceneAsync en additif, permettant d'avoir plusieurs scènes chargées en même temps. Sinon, le jeu est soit en solo soit en multi, donc il faut utiliser le NetworkManager, permettant de gérer les scènes sur un ou plusieurs appareils. Si le paramètre LoadingScreen est true, une coroutine est lancée afin d'afficher un écran de chargement avec une barre de chargement. à l'exception de la scène pause, toutes les scènes peuvent lancer un écran de chargement.

Déchargement de la scène précédente : Ce cas est géré lorsque le paramètre unloadMe est true, cela signifie que l'on souhaite décharger la scène actuelle pour charger la suivante. Ainsi, de façon analogue au chargement les cas menus et jeu sont différencier



```
SceneStack.Push((name:alias ? from : GetKeyFromValue(Scenes, from), unloaded:unloadMe));
if (Mode != 0) {
    NetworkManager.Singleton.SceneManager.LoadScene(alias ? Scenes[to] : to, LoadSceneMode.Additive);
}
else {
    var op : AsyncOperation = SceneManager.LoadSceneAsync(alias ? Scenes[to] : to, LoadSceneMode.Additive);
    if (to != "Pause" && LoadingScreen) {
        instance.StartCoroutine(routine:LoadLevel(op, wait));
    }
}

if (unloadMe) {
    if (Mode == 0) {
        SceneManager.UnloadSceneAsync(GetScene(SceneStack.Peek().name));
    }
    else {
        NetworkManager.Singleton.SceneManager.UnloadScene(GetScene(SceneStack.Peek().name));
    }
}
```

FIGURE 1 – Code extrait du loading screen

2.1.2 Mise en pause du jeu

Dans ce projet, la gestion de la pause du jeu est mise en place à travers plusieurs étapes clés. Tout d'abord, une variable paused est définie et assignée à true, indiquant que le jeu entre en mode pause.

Ensuite, une boucle foreach est utilisée pour parcourir tous les objets actifs du jeu qui doivent être affectés par cette pause. Deux types d'objets sont concernés :

Les joueurs (PlayerScript) : Chaque joueur dans ActiveObjectScript.players est parcouru, et la méthode pause() est appelée sur lui. Cela permet d'arrêter ou de suspendre les actions des personnages jouables. Les objets basiques (BasicObject) : De la même manière, tous les objets présents dans ActiveObjectScript.basicObjects sont mis en pause grâce à l'appel de pause(). Enfin, la transition vers l'état de pause est complétée par un appel à la fonction LoadScene(), qui charge la scène intitulée "Pause". Le paramètre unloadMe : false indique que la scène actuelle n'est pas déchargée, permettant ainsi de reprendre facilement la partie une fois la pause désactivée.

Ce mécanisme garantit que tous les éléments interactifs du jeu sont bien suspendus lorsque le mode pause est activé, améliorant ainsi l'expérience utilisateur en assurant une interruption propre et contrôlée du gameplay.

```

paused = true;
foreach (var scr :PlayerScript in ActiveObjectScript.players) {
    scr.pause();
}

foreach (var scr :BasicObject in ActiveObjectScript.basicObjects) {
    scr.pause();
}

LoadScene(from, "Pause", unloadMe: false);

```

FIGURE 2 – Code extrait de la mise en pause du jeu

2.1.3 Nouveau script pour les interactions

Avant toute action, il est nécessaire de vérifier si le jeu n'est pas en pause à l'aide de la condition suivante :

```
if (!GlobalSceneScript.paused) { ... }
```

Si le jeu est en pause, aucune interaction ne peut avoir lieu, évitant ainsi tout comportement indésirable.

Gestion de la rotation L'objet effectue une rotation autour de l'axe **X** selon deux scénarios :

- Si l'objet n'est pas activé (`!activated`) et que $i > 0$:

- Une rotation est appliquée via :

```
transform.Rotate(new Vector3(dir, 0, 0));
```

- La variable i est ensuite décrémentée (`i--`), limitant ainsi la rotation.

- Si l'objet est activé (`activated`) et que $i \leq I$:

- La rotation est appliquée de la même manière.

- Cette fois, i est incrémentée (`i++`), ce qui permet à la rotation de continuer ou d'inverser le mouvement.

Ce mécanisme est particulièrement utile pour les éléments interactifs tels que les portes, les leviers ou tout autre objet nécessitant un mouvement en fonction de l'activation du joueur.

```

if (!GlobalSceneScript.paused) {
    if (!activated && i > 0) {
        transform.Rotate(new Vector3(dir, 0, 0));
        i--;
    } else if (activated && i <= I) {
        transform.Rotate(new Vector3(dir, 0, 0));
        i++;
    }
}

```

FIGURE 3 – Code extrait pour les interactions

2.1.4 Mise en place du mode multijoueur

Ce script initialise le mode multijoueur en lançant un hôte, tout en gérant l'approbation des connexions et la synchronisation des scènes.

Configuration du mode multijoueur Les paramètres suivants sont définis pour activer le mode multijoueur :

- **Mode = 2** : Définit le jeu en mode multijoueur.
- **ConnectionApproval = true** : Active l'approbation des connexions via `ApprovalCheck`.
- **addListeners()** : Ajoute les gestionnaires d'événements réseau.
- **StartHost()** : Démarrer l'hôte, combinant les rôles de serveur et de client.
- **ActiveSceneSynchronizationEnabled = true** : Synchronise les scènes entre les joueurs.

Grâce à cette configuration, le jeu peut héberger une partie, accepter les connexions des joueurs et assurer une synchronisation fluide de l'environnement de jeu.

```
Mode = 2;  
NetworkManager.Singleton.NetworkConfig.ConnectionApproval = true;  
NetworkManager.Singleton.ConnectionApprovalCallback = ApprovalCheck;  
addListeners();  
NetworkManager.Singleton.StartHost();  
NetworkManager.Singleton.SceneManager.ActiveSceneSynchronizationEnabled = true;
```

FIGURE 4 – Code extrait du loading screen

2.2 Interfaces

2.2.1 Menus en jeu

Comme illustré ci-dessous, le menu pause est fonctionnel et intègre les boutons "Resume", "Setting" et "Exit", répondant ainsi à l'ensemble des objectifs fixés pour ce menu. La section "Setting" offrira la possibilité d'ajuster le volume général du jeu ainsi que celui des musiques. Cependant, cette fonctionnalité n'est pas encore opérationnelle à ce stade.

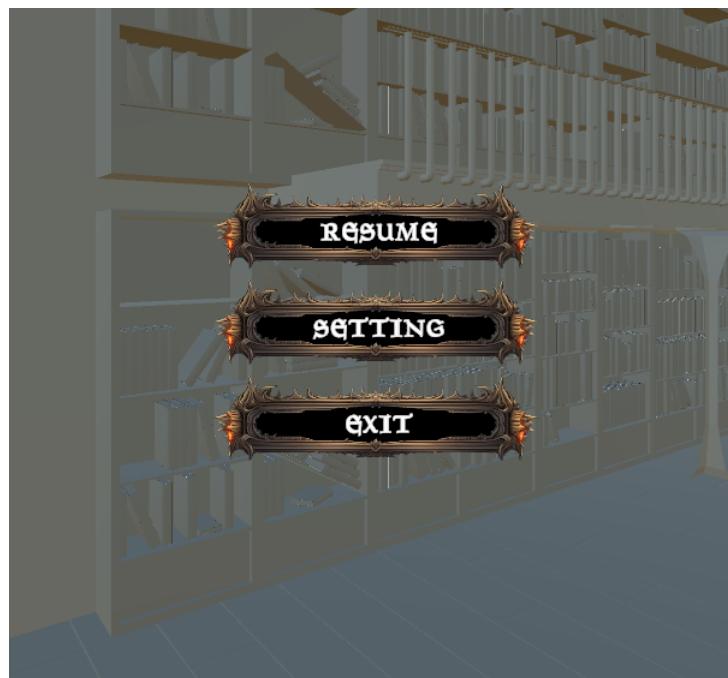


FIGURE 5 – Exemple de menu pause en jeu

2.2.2 Menu écran titre

L'écran titre, interface principale du jeu, est actuellement fonctionnel. Il donne accès aux options pour personnaliser les paramètres, tels que le volume (et, à venir, les commandes), ainsi qu'au lancement du jeu, qui propose deux modes : solo et multijoueur. Une option "Quit" permet également de quitter le jeu.

Son design, en harmonie avec l'univers médiéval du jeu, intègre des textures, des couleurs et des typographies spécifiques à l'ambiance globale.



FIGURE 6 – Exemple de l'écran titre.

Le menu multijoueur, encore en développement, est une fonctionnalité clé. Il permettra de gérer les interactions en ligne via un lobby, espace intermédiaire où les joueurs pourront créer ou rejoindre des parties, configurer des paramètres comme la salle de départ, le nombre de participants et les règles, ou communiquer avant la partie grâce à un chat.

L'interface visera une ergonomie intuitive et une immersion cohérente avec l'univers du jeu. Elle devra également répondre à des contraintes techniques telles que la stabilité réseau, la gestion des latences et la synchronisation des actions. Des éléments visuels et sonores viendront renforcer l'ambiance et l'expérience utilisateur.



FIGURE 7 – Menu play.

3 Univers du jeu

3.1 Cartes

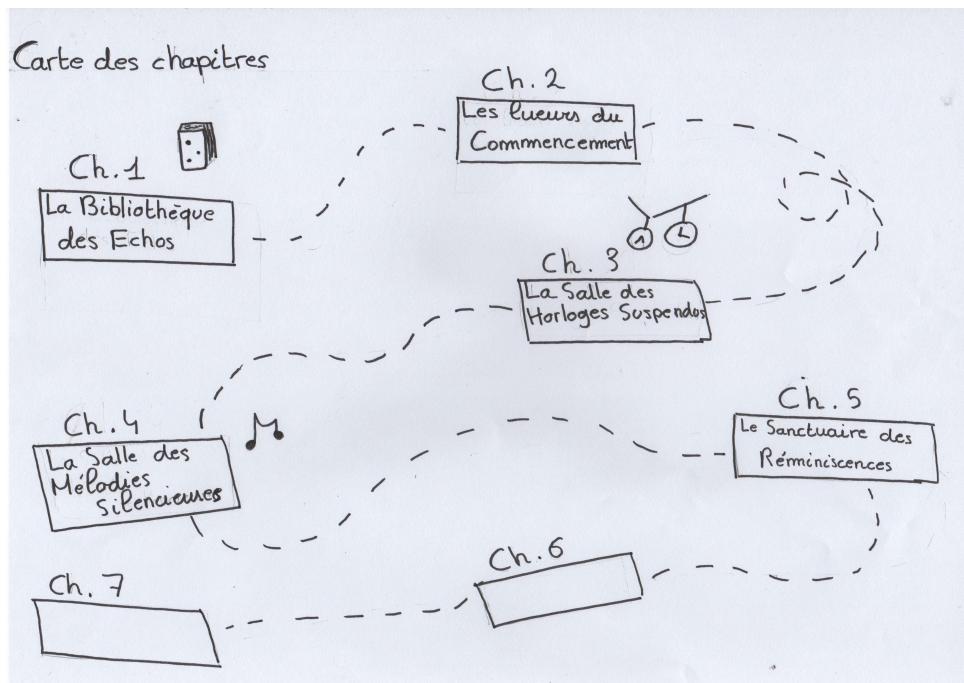


FIGURE 8 – Carte du jeu

Le jeu sera composé de 5 chapitres, chacun relié entre eux.

3.2 Objectifs du jeu

Dans *Fragments of Ember*, le joueur incarne un héros plongé dans un monde mystérieux composé de salles énigmatiques. L'objectif principal est de restaurer l'équilibre d'Ember en collectant et en assemblant les Fragments, des artefacts essentiels dispersés dans chaque salle. Chaque environnement propose un défi unique, mêlant puzzles logiques, mécaniques, environnementaux et narratifs. Le joueur doit explorer ces espaces clos, observer et écouter attentivement les indices laissés par un monde vivant qui réagit à ses actions.

Le jeu est structuré en 7 chapitres, chacun représenté par une salle distincte. Chaque chapitre se termine par la résolution de l'énigme principale de la salle, permettant d'accéder au chapitre suivant. Certaines salles renferment également des pouvoirs cachés. Les 7 salles forment une suite logique et chronologique, mais le joueur a la possibilité de revenir dans une salle précédente pour tenter d'obtenir ces pouvoirs cachés, en s'appuyant sur des indices découverts ultérieurement.

L'histoire se dévoile progressivement à travers des interactions avec des gardiens éthérés et des inscriptions cryptiques, qui révèlent la mission du héros et son lien profond avec Ember. La réflexion et l'observation sont essentielles, car les énigmes reposent sur des éléments variés tels que des motifs, des sons et des lumières. En relevant les défis propres à chaque salle, le joueur répare les liens brisés entre les différentes parties du royaume.

Au fil de sa progression, le joueur découvre également des fragments du passé d'Ember, enrichissant sa compréhension de cet univers complexe. Restaurer l'équilibre final nécessite de surmonter les obstacles, de rassembler tous les Fragments et de lever le voile sur les mystères entourant le héros et son rôle dans ce monde fascinant.

3.3 Assets

L'avancement des modèles 3D a été significatif, avec la finalisation de plusieurs environnements et éléments essentiels du jeu. Les décors de la bibliothèque sont désormais complets, comprenant des étagères détaillées, des meubles et des objets immersifs qui renforcent l'atmosphère du lieu. Par ailleurs, les grottes liées à la bibliothèque ainsi qu'à la forêt ont été intégralement modélisées, offrant des environnements variés et cohérents avec la direction artistique du jeu.

La forêt a également bénéficié d'un travail approfondi, avec la finalisation des modèles 3D de la végétation, du relief et des éléments architecturaux spécifiques à cet environnement. Ces ajouts permettent de renforcer le contraste entre les différents lieux explorables du jeu, tout en maintenant une harmonie visuelle.

En parallèle, la production des personnages est en cours. Leur conception suit une direction artistique cohérente avec l'univers du jeu, avec un travail minutieux sur les proportions, les détails vestimentaires et les animations prévues pour renforcer l'immersion. Ces modèles seront intégrés progressivement, avec des ajustements pour garantir une parfaite interaction avec les décors existants. Sachant que la majorité des personnages seront créés à partir de la même base : une brume vivante. Cette base nous permettra de réaliser des personnages différents avec chacun leurs propres petits détails sans perdre de temps.

Ces avancées permettent d'assurer une progression fluide dans la construction du monde de *Fragments of Ember*, en maintenant une attention particulière aux détails visuels et à l'impact narratif des environnements.



FIGURE 9 – Chapitre 1 - La bibliothèque des échos



FIGURE 10 – Chapitre 2 - Les Lueurs du Commencement

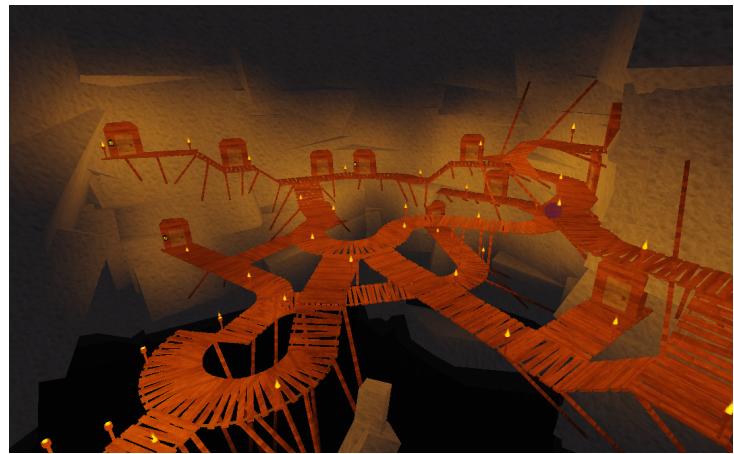


FIGURE 11 – Interlude

3.3.1 Son

Dans le cadre de notre développement, l'ambiance sonore de *Fragments of Ember* a été significativement enrichie grâce à l'intégration de musiques et d'effets sonores immersifs.

Plusieurs compositions originales ont été réalisées pour accompagner l'expérience de jeu. Une musique dédiée au menu du jeu a été créée, offrant une introduction sonore cohérente avec l'atmosphère du projet. De plus, une ambiance musicale spécifique a été conçue pour la bibliothèque, renforçant l'immersion dans cet espace.

Côté effets sonores, nous avons privilégié une approche artisanale en enregistrant des sons à partir d'objets du quotidien. Par exemple, des bruitages de portes ont été obtenus en manipulant des éléments réels afin d'apporter une dimension plus authentique à l'environnement du jeu.

Enfin, la forêt bénéficie désormais d'une ambiance sonore immersive, mêlant bruits d'oiseaux et échos mystérieux, renforçant ainsi le caractère énigmatique de cet environnement. Ces éléments sonores viennent enrichir l'expérience globale du joueur et accentuer l'atmosphère unique de chaque lieu.

3.3.2 Difficultés rencontrées

Lors du développement du son dans notre jeu vidéo, nous avons rencontré plusieurs problèmes techniques et artistiques. Par exemple, lors de la composition des musiques jouées en fond sonore, un choix précis et cohérent d'instruments est absolument nécessaire. Ce choix n'est pas anodin, car il influence directement l'expérience immersive des joueurs.

Nous devons donc analyser attentivement chaque scène de notre jeu, en prenant en compte l'ambiance visuelle, narrative et émotionnelle, afin de trouver des instruments qui, auditivement, s'harmonisent parfaitement avec la mise en scène. Cette étape est cruciale pour garantir une immersion totale.

En effet, si elle est négligée, une musique qui ne correspond pas à l'atmosphère d'une scène peut non seulement perturber l'expérience des joueurs, mais aussi rendre certaines séquences inconfortables, voire déplaisantes. Cela pourrait avoir un impact négatif sur la qualité perçue du jeu, voire nuire à son succès global.

Pour éviter de tomber dans cet écueil, nous avons choisi d'investir un temps conséquent dans le processus de sélection des instruments et l'analyse minutieuse de chaque scène.

Ce travail préliminaire nous a permis de poser des bases solides pour la composition musicale. Grâce à cette méthodologie rigoureuse, nous sommes désormais en mesure de nous concentrer pleinement sur la création musicale elle-même, en utilisant les instruments que nous avons soigneusement sélectionnés.

Ce processus, bien qu'exigeant, nous assure que chaque note et chaque mélodie contribueront à renforcer l'immersion des joueurs dans l'univers de notre jeu, tout en rendant leur expérience mémorable et agréable.

4 Multijoueur

4.1 Réseau

Le multijoueur de Fragments of Ember sera disponible en mode local (LAN) et en ligne (Online).

En mode local, les joueurs pourront se connecter via un réseau local, ce qui est idéal pour jouer dans un même espace physique. Une fois les appareils connectés au même réseau, un joueur pourra héberger la partie et les autres pourront rejoindre facilement en utilisant une fonction de découverte automatique ou en entrant une adresse IP locale.

L'intégration de ces fonctionnalités dans le jeu est réalisé à l'aide de Netcode for gameobjects.

4.2 Mode de jeu

Le multijoueur de Fragments of Ember repose sur la coopération. Les joueurs évoluent dans les mêmes salles, résolvant ensemble les énigmes.

Chaque joueur contribue activement, avec des actions simultanées ou complémentaires.

L'objectif est de favoriser le travail d'équipe pour une expérience immersive et collaborative.

5 Le site web

5.1 État actuel

Le développement du site web dédié à Fragments of Ember a connu une progression significative. L'architecture globale a été consolidée, garantissant une meilleure stabilité et une navigation plus fluide. Les pages principales ont été enrichies avec des contenus optimisés, mettant davantage en valeur les objectifs du projet ainsi que la présentation de l'équipe.

La mise en page a été affinée pour offrir une expérience utilisateur plus intuitive et immersive. La timeline du projet, élément clé de la communication, a été retravaillée pour être plus dynamique et facilement actualisable.

Par ailleurs, l'identité visuelle a été renforcée avec des ajustements graphiques visant à harmoniser les différentes sections du site. Le design des cartes de présentation des membres de l'équipe a été revu afin d'améliorer leur lisibilité et leur intégration au sein de l'ensemble du site.

Les prochaines étapes se concentreront sur l'optimisation et l'intégration de nouveaux contenus interactifs pour rendre le site encore plus engageant.

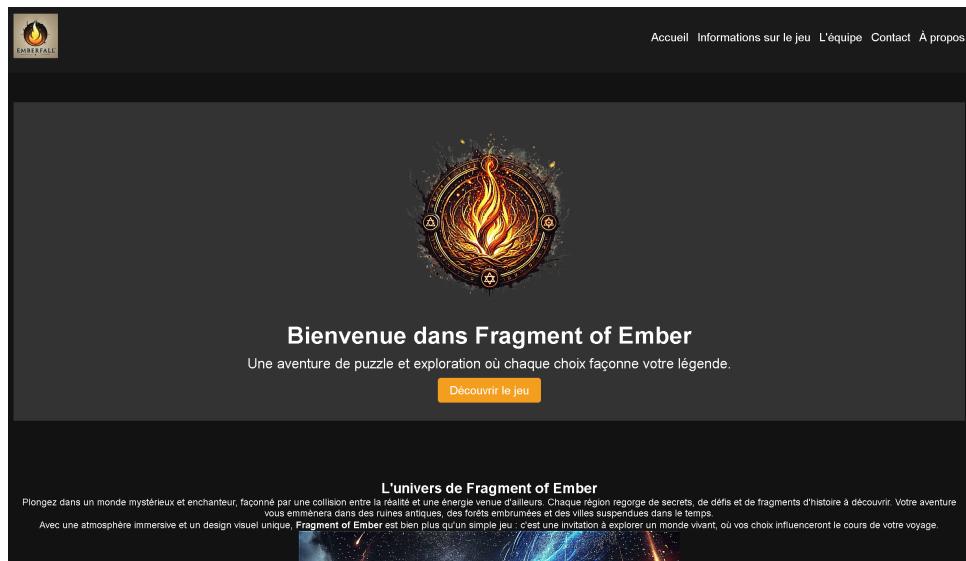


FIGURE 12 – Site web - Accueil

6 Conclusion

Le projet Fragments of Ember continue de se développer en fusionnant innovation, réflexion et créativité. À travers ce second rapport de soutenance, nous avons approfondi les fondements du jeu, tout en explorant des aspects plus avancés de son développement, notamment la consolidation des mécaniques de jeu, l'enrichissement de l'univers immersif, et l'optimisation des éléments techniques et artistiques.

Les progrès réalisés depuis la première soutenance témoignent de l'engagement et de la synergie croissante au sein de notre équipe. Chaque étape, qu'il s'agisse du perfectionnement du gameplay, de l'amélioration des graphismes ou de l'intégration de la bande-son, a été franchie avec soin, en prenant en compte les retours reçus et en affinant notre approche du projet.

Cette deuxième phase du développement marque des avancées notables, notamment avec la mise en place de nouveaux systèmes de jeu et la finalisation de certains éléments essentiels. Cependant, le projet est toujours en évolution, et nous continuons à explorer des opportunités pour perfectionner l'expérience proposée.

En conclusion, Fragments of Ember est bien plus qu'un simple jeu ; il représente une expérience immersive et dynamique qui prend forme et évolue grâce à la passion et au dévouement de toute l'équipe. Nous sommes impatients de partager avec vous les prochaines étapes de cette aventure et d'en dévoiler davantage lors de la dernière soutenance.

6.1 Avancement du projet

TABLE 3 – Avancement des tâches par soutenance

Tâches	Soutenance 1	Soutenance 2	Soutenance finale
Scénario	40%	100%	100%
Multijoueur	0%	100%	100%
Réseau	0%	100%	100%
Son	5%	50%	100%
Modèles 3D	10%	60%	100%
Animations	0%	25%	100%
Site Web	65%	85%	100%
Gameplay	10%	85%	100%
Ajout de l'IA	0%	0%	100%
Menu	70%	70%	100%
Documentation	25%	70%	100%

	En retard
	En avance

6.2 Plan pour le futur

Nos prochaines étapes incluent l'achèvement de tous les chapitres du jeu, avec une attention particulière portée aux détails des salles, aux énigmes et à la narration. Nous finaliserons également l'implémentation des mécaniques de jeu pour assurer une expérience fluide et intuitive. Cela inclut :

- L'amélioration et l'intégration des derniers modèles 3D dans le moteur de jeu.
- L'affinement du sound design pour renforcer l'immersion.
- La finalisation du système d'énigmes et des interactions.

Nous continuerons également le développement web pour garder un site web à jour face aux avancements de notre jeu. Enfin, nous réaliserons une série de tests approfondis afin d'identifier d'éventuels problèmes imprévus et d'assurer une expérience de jeu fluide et optimisée.