

Rapport de soutenance
Projet Kólasi

La Parité
Marine CHARRA
Nora DELFAU
Kenny FUNG
Corentin OURVOY

16 mars 2018



Résumé

Sur un terrain représentant un labyrinthe, les joueurs doivent atteindre des artefacts dans des pièces. La disposition des murs évolue à chaque tour de jeu.

Table des matières

1	Première Soutenance	4
2	Travail accompli	5
2.1	Menu principal du jeu	5
2.2	Hébergement	5
2.3	Site Web	5
2.4	Map	5
2.5	Avatar	7
2.6	Multijoueur	7
3	Technologies utilisées	8
3.1	Menu principal	8
3.2	Site Web	8
3.3	Map	8
3.4	Avatar	8
3.5	Multijoueur	9
4	Problèmes rencontrés	10
4.1	Menu principal du jeu	10
4.2	Hébergement	10
4.3	Site Web	10
4.4	Map	11
4.5	Avatar	12
4.6	Multijoueur	12
5	Pour les prochaines soutenances	13
5.1	Menu principal du jeu	13
5.2	Site Web	13
5.3	Map	13
5.4	Avatar	13
5.5	Multijoueur	13
6	Autres tâches	14
7	Bibliographie supplémentaire	15

Table des figures

1	Menu Principal	5
2	Blocs en I	6
3	Blocs en T	6
4	Blocs en L	6
5	Les avatars	7
6	Test du multijoueur	7
7	Test du multijoueur	8
8	Apparence du NetworkManagerHud	9
9	Site Web	10
10	Links Hub	10
11	Schéma de déplacement de bloc	11
12	Jouer en réseau	12
13	Twitter	15
14	Facebook	15

Introduction

Ce rapport a pour but de pouvoir suivre notre avancement jusqu'à la première soutenance. Il contiendra nos fausses pistes, nos corrections, ainsi que nos astuces afin d'outrepasser nos problèmes.

Rappel

Notre jeu se déroule dans les Enfers grecques, Kólasi en grec. Les joueurs incarneront des âmes cherchant à s'enfuir des Enfers pour ressusciter. Pour se faire, les âmes auront à regrouper le plus grand nombre possible d'artefacts nécessaire à leur fuite. À titre d'exemple, un trio d'os pour chien pour passer les portes des Enfers gardées par Cerbère.

Ces artefacts sont tous situés dans le Labyrinthe. Les âmes, du nombre de 2 à 4 le parcourront à la recherche des trésors. Mais leur quête n'est pas aisée : le labyrinthe lui même souhaite leur en empêcher. À chaque tour, les murs bougent. De plus, certaines pièces sont piégées. Seule l'âme ayant amassé le plus d'artefacts pourra sortir des Enfers. Le labyrinthe pourra être muni de torches. Nous envisageons de faire un sol de lave avec le Lava Flowing shader sur Unity. Les couleurs se situeront dans les tons rouge, gris, noir, le tout sombre. Pour les textures, nous privilégierons, de la pierre, des effets de fumée et de lave. Le labyrinthe sera à ciel ouvert mais les Enfers auront un plafond de terre.

Le joueur ayant gagné aura son personnage qui pourra traverser le Styx et gagner le monde des vivants.

Le jeu va se scinder en 2 phases de jeux :

1. Une phase de "réflexion"
2. Une phase "d'action"

Dans la première phase, le joueur verra le plateau de jeu en vue du dessus et pourra ainsi voir la forme de chemin que forment chaque pièce (cf. figure 2). De plus, dans cette phase il ne peut voir l'intérieur des pièces. Cette phase est dite stratégique, c'est à ce moment là que le joueur va pouvoir voir où se trouve à la fois son personnage, son objectif ainsi que le chemin qu'il doit suivre pour relier les deux. Mais surtout cette phase de réflexion permettra d'exercer la principale idée du jeu : bouger les murs. En effet le joueur va être équipé d'une pièce de labyrinthe qu'il va pouvoir insérer dans une ligne ou colonne spécifique, entraînant ainsi le mouvement des murs et la destruction de la pièce du bout. Bouger les murs lui permettra donc de se frayer un chemin jusqu'à son artefact ou bien alors bloquer les autres joueurs.

La deuxième phase de jeu est en vue à la première personne et consiste à se déplacer dans le labyrinthe à travers les différentes pièces dans un temps imparti. Le joueur n'aura plus accès à la carte et devra utiliser sa mémoire afin d'aller à son objectif. Le déplacement se fera de façon semi-automatique (c'est-à-dire que le joueur ne se déplacera pas tout seul mais pourra choisir la pièce où il pourra aller). Lorsque le joueur entrera dans une pièce il pourra faire face à plusieurs situations :

1. La pièce est piégée il devra effectuer un QTE pour éviter un malus
2. La pièce contient un bonus
3. La pièce contient un joueur, les 2 devront effectuer un QTE afin de pouvoir expulser l'autre de la pièce
4. la pièce contient l'artefact cherché
5. La pièce est vide

Jusqu'ici nous avons fait ou commencé l'hébergement, le site web, le menu principal, la map, les avatars ainsi que leurs déplacements et le mode multijoueur.

1 Première Soutenance

Pour la première soutenance nous devons avoir fini l'hébergement et le site web. Nous avons pour objectif d'avoir avancé la carte et l'implémentation des personnages.

Tâche	Avancement
Hébergement	100%
Design site	80%
Carte	50%
Avatars	30%
Multijoueur	30%
Caméra	10%
Menu Principal	0%
QTE	0%
Malus/Bonus	0%
Musiques/Sons	0%

TABLE 1 – Avancement première soutenance



2 Travail accompli

2.1 Menu principal du jeu

Malgré le fait que le menu principal n'était pas à faire, le temps de recherche de création de la map nous a permis de nous intéresser au menu principal en attendant. Le menu est assez sobre et simple et contient donc :

1. Un menu pour les options (musique, effets, langues)
2. Un menu pour héberger une partie
3. Un menu pour rejoindre une partie
4. Un menu pour afficher les règles

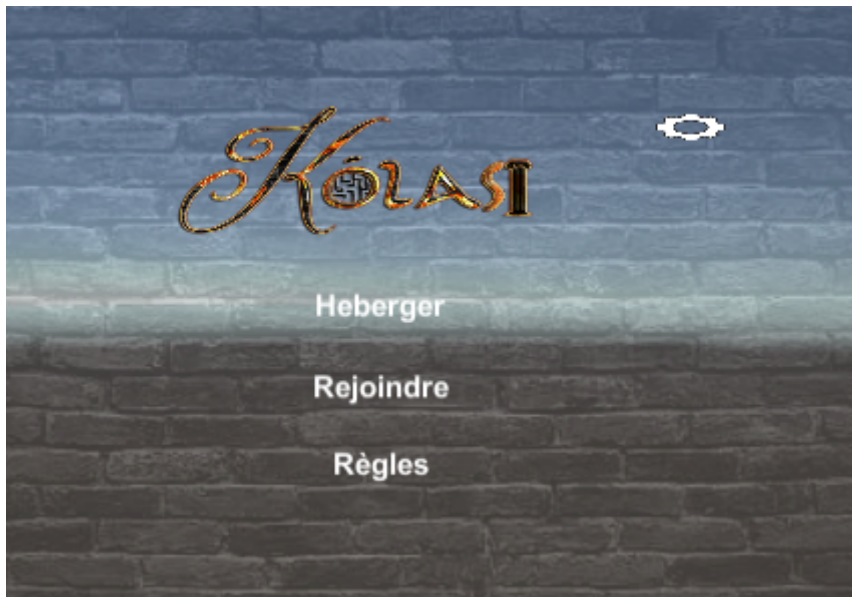


FIGURE 1 – Menu Principal

2.2 Hébergement

L'hébergement et la mise en commun de nos fichiers se font par GitHub grâce à l'outil GitHub-Desktop qui nous permet d'avoir une interface graphique avec notre stockage. De même, notre site est hébergé sur GitHub mais cette fois-ci dans un autre dossier car demandant des spécificités de nom.

2.3 Site Web

Le site web donne accès à un lien de téléchargement pour Windows et MacOSX. Y sont glissés les contacts vers nos réseaux sociaux et notre email pro. Il est constitué d'une page avec des ancres.

2.4 Map

Sur la map nous avons réussi la plupart des objectifs fixés. En effet il nous est possible de créer une map contenant 3 différents types de bloc (Bloc en T, Bloc en I, Bloc en L) placés et tournés de façon aléatoire pour qu'à chaque partie la carte soit différente et que les possibilités de jeu soient diverses.

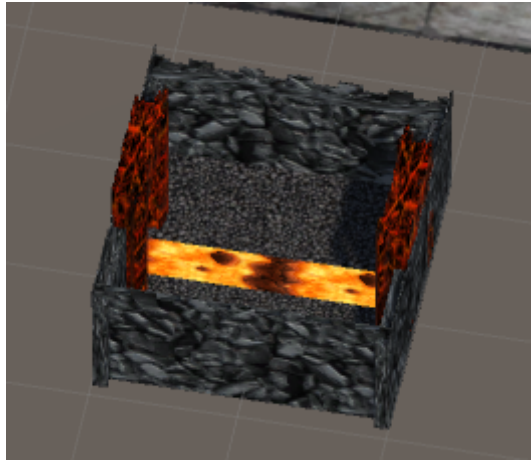


FIGURE 2 – Blocs en I



FIGURE 3 – Blocs en T

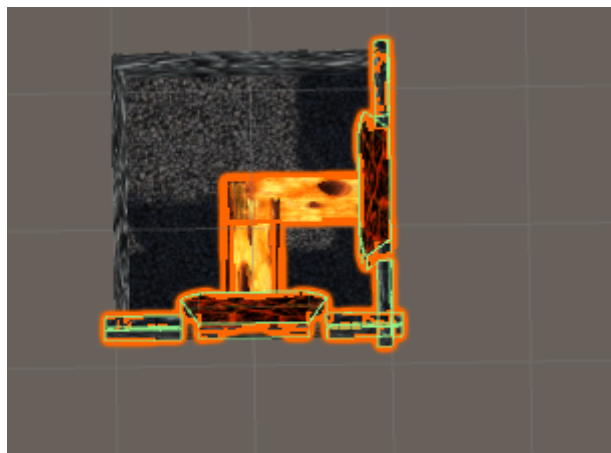


FIGURE 4 – Blocs en L

2.5 Avatar

Le développement des personnages a suivi plusieurs modifications physiques avant d'arriver à sa forme finale. Il s'agit de quatre fantômes animés par des systèmes de particules colorées représentant les âmes. Ils sont capables de se déplacer en avant, en arrière et sur les côtés.



FIGURE 5 – Les avatars

2.6 Multijoueur

A l'aide du Network Manager de Unity, nous avons réussi à créer un serveur dans lequel les joueurs peuvent se rejoindre. Ainsi, chaque joueur peut distinguer son avatar des autres et le faire se déplacer sans que ce dernier ne se confonde avec ceux des autres joueurs dans la partie. Dans le cas présent, le joueur distingue son avatar des autres car le sien est bleu, ceux des autres sont blancs.

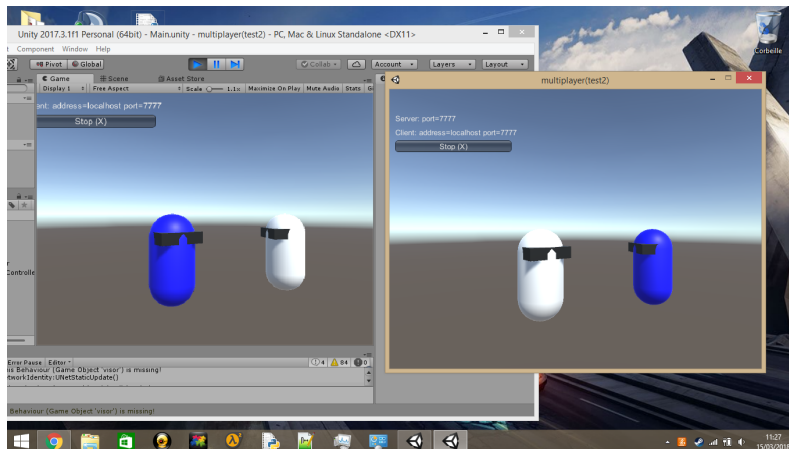


FIGURE 6 – Test du multijoueur

3 Technologies utilisées

Nous avons pour objectif de réaliser un jeu vidéo sans dépenses. Nous nous sommes alors imposés une contrainte lors du choix des différentes technologies. Chacun des logiciels ou assets cités dans cette partie est gratuit et en open source.

3.1 Menu principal

Pour le Menu Principal nous avons utilisé Unity. Plus particulièrement les Canvas et les Boutons, spécialisé pour l'interface.

3.2 Site Web

Pour le site web nous avons utilisé le template "prologue" de HTML5 UP. Nous avons aussi testé wix.com mais n'avons pas retenu ce site.

3.3 Map

Pour la map nous avons utilisé beaucoup d'asset du Unity Asset Store :

1. Castle Supply LITE
2. Yughues Free Cobble Materials
3. Lava Flowing Shader
4. Cracked stone filled with lava

Nous avons créé des prefabs et écrit des scripts en C# pour composer et contrôler le comportement de la map. Nous nous sommes servis du Rigidbody afin que la matière ne traverse pas notre labyrinthe.

3.4 Avatar

Pour les avatars nous avons créé des script C# pour contrôler les mouvements de notre personnage. Pour les assets nous avons utilisé :

1. Pour la matière, nous avons utilisé l'asset "Level 1 Monster Pack".
2. Le Modèle utilisé pour obtenir la forme du fantôme provient du même asset.
3. Le système de particule animant l'avatar vient quant à lui de l'asset "PinwheelFantasyEffectFreePack".

Nous nous sommes aussi servis des Rigidbody afin que les personnages soient soumis à la force de gravité et qu'il ne puisse pas traverser les murs et le sol. De plus, chacun des avatars est muni d'une caméra interne qui permettra aux joueurs d'avoir le point de vue de leur fantôme.



FIGURE 7 – Test du multijoueur

3.5 Multijoueur

Avec Unity, la création et la gestion d'un réseau multijoueur s'effectue à l'aide du réseau multijoueur intégré à Unity ainsi que du High Level API (HLAPI). Pour cela, nous avons besoin d'un Gameobject vide (que l'on appelle NetworkManager) et d'y ajouter deux composants scripts "Network" qui sont le Network Manager et le Network Manager HUD.

Le composant NetworkManager gère l'état du réseau du jeu.

Le composant NetworkManagerHUD fonctionne avec NetworkManager et fournit une interface simple pour contrôler l'état du réseau lorsque le jeu est en cours d'exécution.

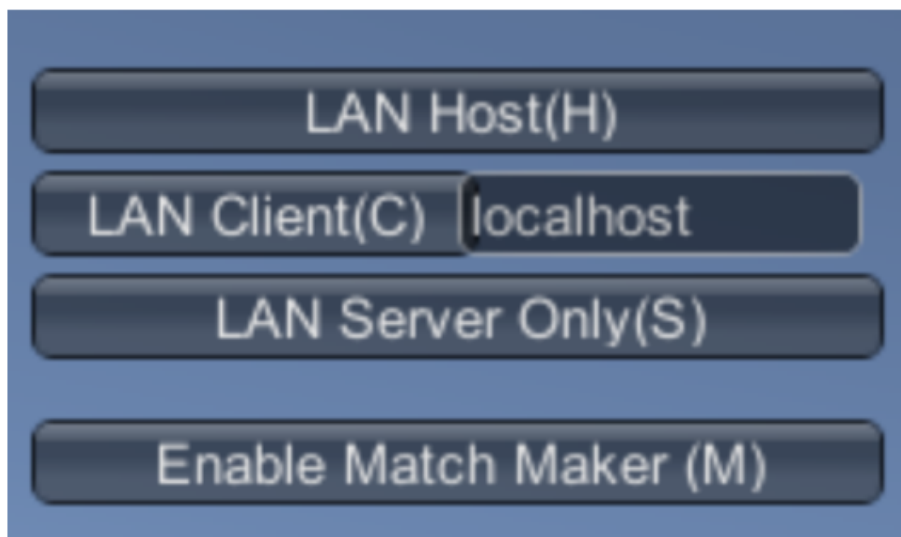


FIGURE 8 – Apparence du NetworkManagerHud

Mais ce n'est pas tout, il faut aussi ajouter à l'avatar du joueur un certain nombre de composants pour que celui-ci puisse échanger correctement les informations envoyés par le joueur avec le serveur. Voici une liste de tous ces composants scripts :

- Network Identity
- Player Controller
- Network Transform

Le composant NetworkIdentity est utilisé pour identifier l'objet sur le réseau et pour en informer le système de gestion de réseau. Le script PlayerController permet au joueur de contrôler le Player GameObject.

Le script NetworkTransform synchronise la transformation de l'avatar du joueur sur le réseau. C'est à dire que le joueur déplace son avatar détenu localement et seulement l'avatar détenu localement. Le script synchronise ensuite la position, la rotation et l'échelle de la transformation de l'avatar du joueur à travers le serveur et tous les clients.

4 Problèmes rencontrés

4.1 Menu principal du jeu

Sur la construction du menu nous n'avons pas rencontré beaucoup de problèmes, car Unity est assez explicite et facile à utiliser pour créer des menus.

4.2 Hébergement

Le principal problème rencontré avec GitHub est le fait qu'il bride la connexion lors du téléchargement depuis l'hébergement. De plus le fait que nous ne soyons pas tous sur le même système d'exploitation (mac-OS et Windows) pose des problème lors du passage de mac à Windows car celui créer des fichiers caché qui font totalement bugger le système lorsque qu'un pc veut renvoyer son contenu sur le GitHub.

4.3 Site Web

Nous avons mis du temps mis du temps à choisir notre template. Ensuite il a fallut comprendre l'html, le css et le php. Un fois cela fait, nous avons du supprimer une partie du site (formulaire de contact) car il n'était pas aisé à implémenter. Nous avons mis du temps à comprendre comment gérer un site web, mais une fois fait, c'était aisé. Nous avons aussi créé des réseaux sociaux pour promouvoir notre jeu :

1. Facebook : <https://www.facebook.com/K%C3%B3lasi-851271011712431/>
2. Twitter : <https://twitter.com/KolasiMaze>



FIGURE 9 – Site Web

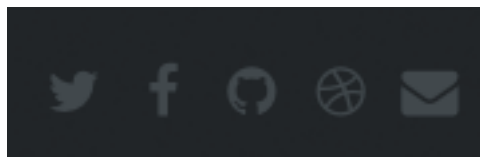


FIGURE 10 – Links Hub

4.4 Map

On a eu plusieurs problèmes sur la map, le premier était de savoir comment créer la Map et y afficher différents blocs. Pour cela nous avons créé deux tableaux, un qui gère les blocs : un objet que nous avons créé comprenant comme paramètres la rotation, un type (I,T,L) et son artefact (qui sera utilisé pour plus tard) ; le deuxième est un tableau de VisualBloc comprenant tous les murs et portes d'un bloc qu'on choisira d'afficher (ou non) en fonction de son type et c'est donc cet objet qui est affiché au joueur.

Mais tout d'abord, il fallait créer les blocs. Nous avons assemblé quatre murs et un sol dans un GameObject vide. Ensuite nous avons divisés en trois sous parties le reste, représentant les trois types de blocs. Ceux-ci contiennent les portes et leurs murs ainsi que les chemins de lave. Il y a alors fallu créer un prefab d'après le GameObject contenant les blocs.

Puis comme compris dans le cahier des charges, il faut pouvoir bouger ces blocs, nous sommes donc partis sur le fait d'ajouter un rigidbody (une fonction de Unity qui ajoute un corps physique à notre objet) à notre VisualBloc, afin de lui ajouter une force de déplacement. Sauf de part la façon dont était défini notre VisualBloc, il nous était impossible d'accéder à son rigidbody. Il a donc fallu changer de stratégie, nous avons donc trouvé qu'en appliquant la fonction Transform.Translate directement sur notre cube au lieu de son rigidbody nous arrivions au même résultat.

Cependant, pour la composante rigidbody, nous avons mis du temps à remarquer que changer les blocs dans le jeu ne modifiait pas le prefab. Après cela, il y avait toujours un bug car toute notre map marchait sauf le mur gauche qui s'envolait. Il est apparu que ce mur avait un rigidbody (sans IsKinematic, permettant à l'objet de ne pas être soumis aux contraintes physiques) qui contraignait celui implémenté sur le bloc.

Ensuite il a fallu déplacer ces blocs, la condition était que lorsque l'on bouge les blocs d'une ligne ou d'une colonne tous doivent bouger dans le même sens. A cause de la rotation de nos blocs, trouver la bonne direction fut compliqué, les blocs partaient dans tout les sens. On a donc fait des conditions pour chaque en fonction de la direction voulue. Par exemple si l'on veut que la colonne 3 bouge vers le sud mais que le bloc est orienté vers l'est alors ce bloc bougera vers sa droite afin d'avoir un déplacement global vers le sud.

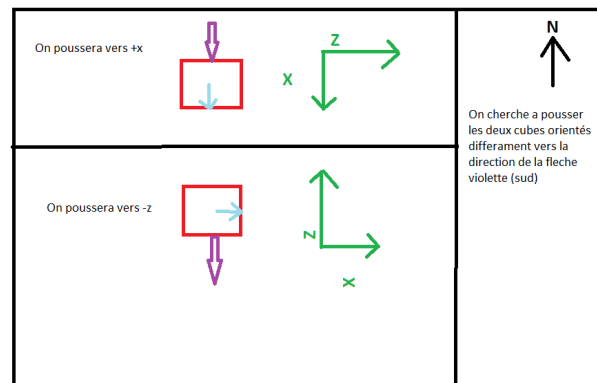


FIGURE 11 – Schéma de déplacement de bloc

Enfin lors du déplacement du personnage il se prenait tout les boîtes de collisions des portes ainsi, pour résoudre ce problème il a fallut les désactiver.

4.5 Avatar

En ce qui concerne les avatars, les problèmes sont survenus au niveau des déplacements. Il est question d'obtenir un mouvement fluide actionné par un appui simple d'une touche "flèche". De plus, comme la caméra suit le personnage dans ses déplacements, il faut que ce dernier pivote sur lui-même lorsqu'il se dirige vers la gauche ou la droite ou lorsqu'il fait demi-tour. Nous avons commencé par utiliser des fonctions permettant de décaler l'objet d'un point à un autre. Or, celles-ci font un effet de téléportation, ou du moins un déplacement trop saccadé, brutal. Nous avons alors cherché à décaler le personnage sur des petits intervalles jusqu'à un certain cas d'arrêt de façon à obtenir la fluidité recherchée. La rotation du personnage provoque cependant des complications à propos du cas d'arrêt. Après avoir pivoté une fois, le joueur ne semble plus s'arrêter dans son mouvement. Nous sommes actuellement capable de créer un mouvement fluide mais sans rotation du personnage, ou l'inverse. Afin de pouvoir jouer dès maintenant, nous avons mis en place un déplacement qui engendre la rotation et la téléportation de l'avatar.

4.6 Multijoueur

Le problème majeur du multijoueur était qu'il ne fonctionnait que sur un seul ordinateur à la fois. En effet, le but de notre jeu est qu'il soit essentiellement multijoueur... Le problème était donc de savoir comment faire fonctionner le serveur sur plusieurs clients.

Finalement, le problème venait du fait que le BUILD de l'application du jeu n'était pas exécuté au bon emplacement dans le dossier du projet. La solution était donc de créer un dossier (à l'intérieur du dossier projet) dans lequel on plaçait le BUILD de l'application.

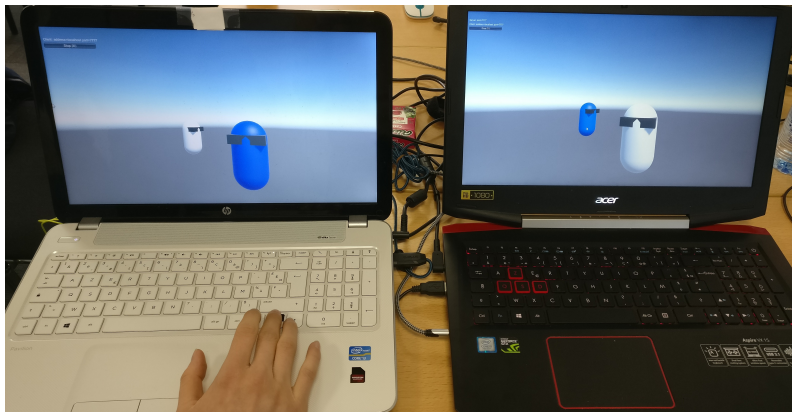


FIGURE 12 – Jouer en réseau

5 Pour les prochaines soutenances

5.1 Menu principal du jeu

A l'heure d'aujourd'hui, le menu n'est pas fini, car il n'est pas relié au multijoueur, pareil pour la barre de la musique et des effets qui n'est pas relié au son et à la musique actuelle du jeu car étant inexistant et donc de même pour le changement de langue. De même nous aimerions encore améliorer son design.

5.2 Site Web

Le site Web est quasiment fini et ne nécessitera pas de grande modification pour les prochaines soutenances. Nous envisageons peut-être d'ajouter plus de texte ainsi que les règles du jeu ou une vidéo de promotion.

5.3 Map

Pour la Map, il reste à ajouter un système de flèches pour montrer au joueur où il va pousser son bloc, d'ajouter un système de bloc au joueur, d'ajouter un nombre de bloc fixe sur la Map, de trouver un équilibre du nombre de type de bloc et d'ajouter des effets de fumée/brume afin d'éviter que le joueur puisse voir à travers les portes le contenu des salles adjacentes. De même il faudrait faire en sorte que les portes donnant sur l'extérieur du labyrinthe soient inaccessibles.

5.4 Avatar

Le déplacement des avatars reste à être amélioré. Il est question d'apporter une fluidité au mouvement et de peut-être implémenter un effet flottant du fantôme. Son domaine d'accessibilité sur le plateau doit aussi être défini. L'ajout de contrainte dans son déplacement éviterait que le personnage sorte du labyrinthe.

5.5 Multijoueur

Le multijoueur sera l'une des parties les plus complexes à gérer du jeu. En effet, il s'agit de la partie ayant pour rôle d'assembler et de faire fonctionner EN LIGNE les autres parties du projet (cartes, avatars, caméra, QTE,...). Pour l'instant nous n'avons pas encore implémenté cette partie dans notre jeu mais avons réussi à faire communiquer plusieurs ordinateurs en réseau.

6 Autres tâches

Il nous reste à travailler les QTE(Quick Time Event), la caméra les malus/bonus ainsi que l'IA et les musiques et sons du jeu. Enfin, nous aimerions créer un décor extérieur à notre labyrinthe (un plafond et un sol).

Les QTE seront les seuls actions possibles entre les joueurs : lorsqu'un joueur se retrouvera dans la même salle qu'un autre, un QTE se déclenchera durant lequel les deux joueurs devront effectuer des actions indiqués à l'écran le plus vite possible. Le joueur qui réussira le mieux recevra un bonus, tandis que le joueur qui réussira le moins bien recevra un malus.

La caméra permettra de changer du mode observation au mode jeu immersif. De même il faut faire en sorte que la carte vu du haut reste lisible.

Les malus/bonus permettent d'avoir un avantage/désavantage par rapport aux autres joueurs. Un bonus pourrait être par exemple un ajout de temps supplémentaire ou bien posséder une carte qu'il est possible d'afficher lors de la phase d'action. Nous envisageons aussi des plateformes permettant au joueur de se téléporter. Les malus seront affectés au joueur lorsque celui-ci ratera une phase de QTE. Il pourront être de plusieurs formes comme du temps en moins, une vision obscurcie ou encore l'obligation de passer son tour.

Pour l'IA, nous ne la finirons pas pour la seconde soutenance mais nous tenterons de réfléchir à un algorithme permettant la résolution du labyrinthe en plusieurs mouvements. Il est peu probable qu'il y ait un rendu visible de l'IA pour la seconde soutenance.

Pour la seconde soutenance, nous gardons notre plan :

Tâche	Avancement
Hébergement	100%
Design site	100%
Carte	90%
Avatars	80%
Multijoueur	70%
Caméra	100%
Menu Principal	30%
QTE	20%
Malus/Bonus	10%
IA	20%
Musiques/Sons	0%

TABLE 2 – Avancement deuxième soutenance

7 Bibliographie supplémentaire

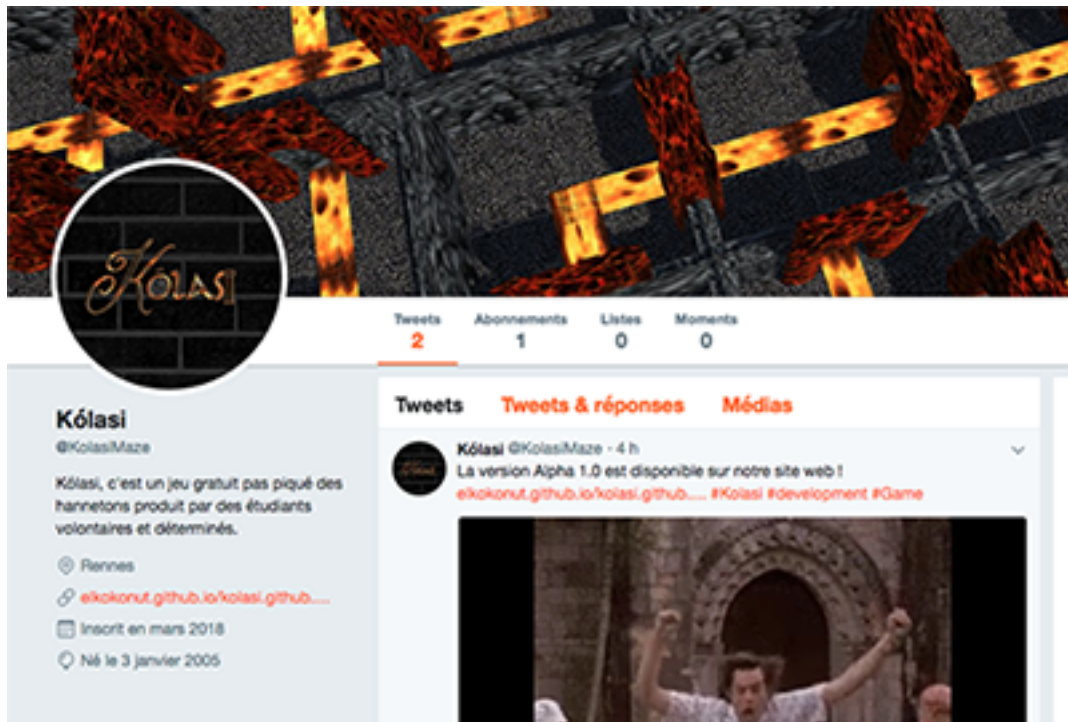


FIGURE 13 – Twitter

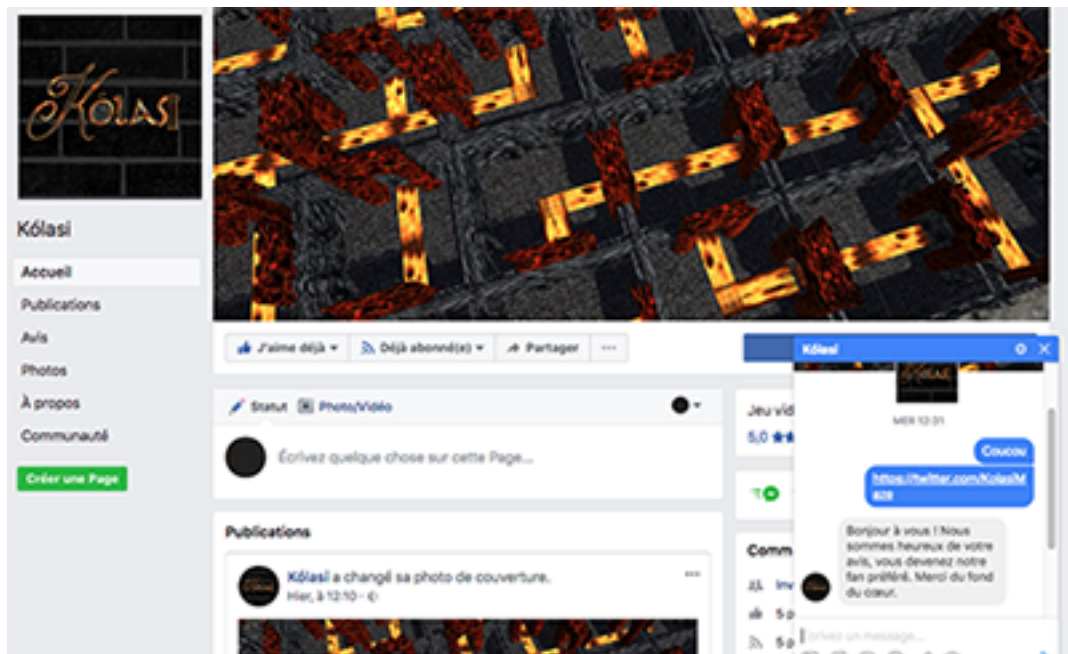
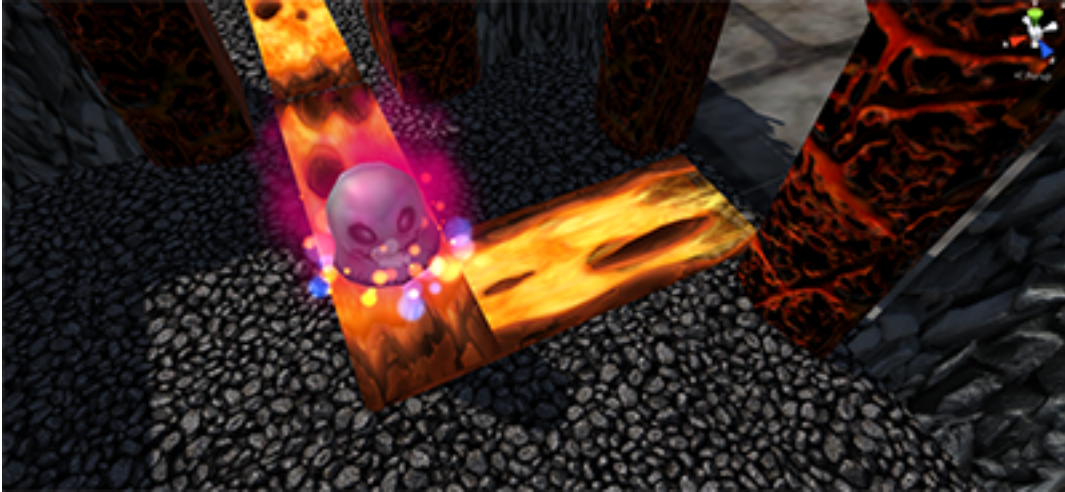


FIGURE 14 – Facebook

Conclusion

Pour cette soutenance, nous avons terminé à 100% l'hébergement du projet sur GitHub. Le site internet du jeu est presque totalement terminé. La carte est bientôt terminée. La création de l'avatar progresse comme prévu. Nous avons fini le réseau multijoueur. Il ne reste qu'à l'associer aux autres parties du projet (Map, avatars, menu principal,...) Le menu principal est bientôt terminé. Pour les prochaines soutenances nous commencerons à travailler sur les QTE, la caméra, les bonus/malus, l'IA et les sons/musiques.



LA PARITÉ ↗