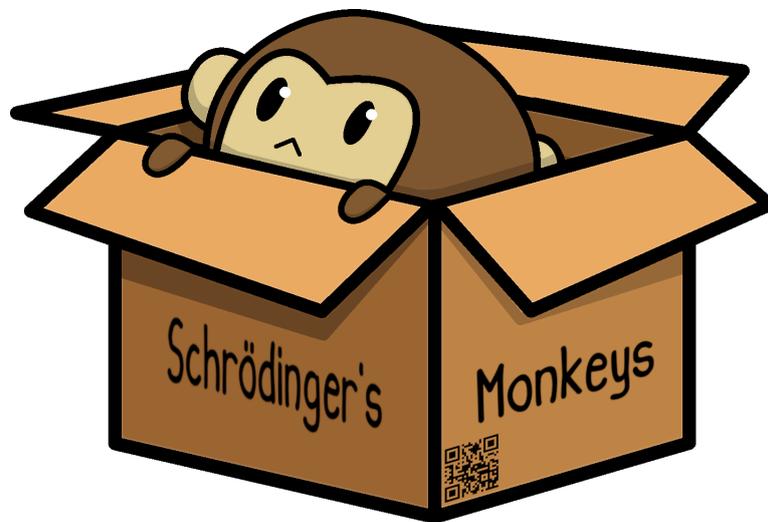


Rapport de soutenance 2

Projet Fracture



Groupe Schrödinger's Monkeys

- Toan Gaucher-Kriete
 - Rapahël Leroy
 - Adam Nessaibia
 - Maëlys Rimbart

Avril 2021

Table des matières

1	Introduction	3
1.1	Première soutenance	3
2	Méthodes de travail	4
2.1	Travail en équipe	4
3	Avancement du projet	4
3.1	Interface graphique	4
3.1.1	Menu de sélection de niveau	4
3.1.2	Barre de vie	5
3.2	Graphismes 2D	6
3.3	Animations	7
3.4	Audio	9
3.5	Construction des mécaniques	10
3.6	Conception de niveau	11
3.7	Multijoueur	13
3.7.1	Photon Unity Networking 2	13
3.7.2	Appels de procédure à distance (RPC)	13
3.7.3	Synchronisation des animations	17
3.7.4	Messagerie instantanée	18
3.8	Site Web (Toan)	19
3.9	Intelligence artificielle	19
4	Difficultés rencontrées et leur résolution	20
4.1	Difficultés individuelles	20
4.1.1	Maëlys	20
4.1.2	Raphaël	20
4.1.3	Adam	20
4.1.4	Toan	21
4.2	Difficultés communes	21
4.2.1	Mise en commun du travail et Git	21
4.2.2	Travail du projet en plus du travail pour les examens	21
5	Avances et retards	22
5.1	Produit d'amusement minimal	22
5.2	Avancement des objectifs fixés par le premier rapport de soutenance	22
6	Prévision de l'avancement à la dernière soutenance	24
7	Conclusion	25

1 Introduction

Fracture est un jeu vidéo de plateformes en deux dimensions, axé sur la coopération de deux joueurs évoluant dans des univers parallèles, en écran scindé. Le jeu prend place dans la Grèce antique et permet uniquement à 2 joueurs de jouer en multijoueur.

Notre équipe est soudée et arbore un nom singulier : les Singes de Schrödinger. Le logo de notre équipe se trouve sur la page de couverture de ce rapport. Vous pourriez demander ce qu'est un singe de Schrödinger ; eh bien en réalité, c'est ce que nous sommes tous. Des singes qui, dès lors qu'ils réalisent leur condition simiesque, changent d'état pour devenir humains. Nous remercions M. Ternier, grâce à qui nous avons enfin pu devenir non-singes.

Nous nous sommes tous mis d'accord sur le principe du jeu, dès le début : la mécanique principale du jeu consiste, pour les joueurs, à échanger de monde afin de surmonter les épreuves mises en place dans les niveaux. Le but est que les joueurs aient besoin l'un de l'autre pour avancer dans deux mondes distincts, d'où le nom *Fracture*.

1.1 Première soutenance

À la première soutenance, nous avons présenté nos méthodes d'organisation, de collaboration ainsi que nos avancées sur la partie multijoueur qui était déjà fonctionnelle, les mécaniques basiques de déplacement qui étaient déjà en place, sur l'audio et les graphismes. Nous avons également présenté nos plans pour la deuxième soutenance, dont le point principal serait l'implémentation de la mécanique principale du jeu : le changement de monde.

2 Méthodes de travail

2.1 Travail en équipe

Nous avons décrit nos méthodes de travail dans le premier rapport de soutenance. Depuis, nous avons tenté de les améliorer en nous mettant d'accord sur des méthodes à adopter.

Par exemple, nous avons mis en avant le partage de tâches. En effet, jusqu'à la première soutenance, nous nous occupions chacun des tâches qui nous avaient été attribuées dans le cahier des charges. Cette approche avait pour avantage de fluidifier le travail, car chacun faisait son travail et n'avait pas besoin d'attendre les autres, mais elle avait pour inconvénient de ne pas permettre une bonne compréhension de l'ensemble du projet. Dans ce but, Toan a par exemple créé des animations pour le joueur ; des tâches de "mécaniques de jeu" ont aussi été assignées à Maëlys (dégâts des ennemis) et à Raphaël (barres de vie).

Nous avons de plus continué d'utiliser Trello, Discord, ainsi que Git, pour suivre l'avancement du projet. Avec le temps, nous nous sommes habitués à ces outils et nous sommes bien mieux organisés que précédemment.

Enfin, nous avons tenté de travailler davantage "en groupe" malgré le distanciel. Le principe était de travailler plus souvent à plusieurs en étant en appel sur Discord afin de créer une meilleure ambiance de travail et d'être plus efficace. Bien que nous ayons appliqué cela quelques fois, cela n'a pas été un franc succès, car nous ne travaillons pas toujours aux mêmes moments au sein du groupe. Cependant, le peu de travail de groupe que nous avons fait a permis de renforcer les liens du groupe.

3 Avancement du projet

3.1 Interface graphique

3.1.1 Menu de sélection de niveau

Le menu principal n'a pas beaucoup changé depuis la première soutenance, cependant les changements principaux se sont fait après celui-ci. Tout d'abord, maintenant, après que 2 personnes ont réussi à rejoindre une partie, nous avons un écran de sélection de niveaux, qui est généré à partir d'un dictionnaire de niveaux et de leur *buildIndex*, ce qui nous évite de devoir refaire ce menu manuellement si de nouveaux niveaux sont amenés à être ajoutés, ou certains à être supprimés.

Il est possible de quitter ce menu à tout moment, ce qui aura pour conséquence de faire quitter la salle multijoueur aux deux joueurs, pour éviter des conflits si quelqu'un d'autre essayait de rejoindre la salle alors qu'une des personnes aurait quitté.

```
public static readonly Dictionary<string,int> scenes = new Dictionary<string,int>()
{
    {"Level 1",3},
    {"Level 2",4},
    {"Level 3",5}
};
```

Dictionnaire de niveaux



Sélection de niveau appelée avec le dictionnaire du dessus

3.1.2 Barre de vie

Nous avons également implémenté un système de vie, qui sera détaillé plus tard dans la partie mécanique. Celle-ci est représentée par une barre de vie comme dans la plupart des jeux, et nous avons fait en sorte que lorsque les deux joueurs échangent de place, celles-ci échangent aussi pour éviter la confusion entre les deux joueurs. Nous avons utilisé le système de RPC afin de synchroniser les barres chez les joueurs, sinon la barre de vie d'un joueur ne se mettrait à jour que de son côté, ce qui n'est pas très pratique dans le cas d'un jeu multijoueur.



Barres de vie avant/après échange de position des 2 joueurs

3.2 Graphismes 2D

Le logiciel principal utilisé pour les graphismes est resté Gimp. L'avancée la plus importante dans les graphismes a été la création du premier personnage ; celui-ci a été fait à partir du dessin-concept qui se trouvait dans le rapport précédent.

L'arrivée de ce premier personnage dans le jeu a permis d'accumuler une expérience sur le mode de fonctionnement d'Unity par rapport aux personnages. L'implémentation du second personnage sera donc plus rapide. En effet, lorsque nous avons commencé la réalisation de ce personnage, nous ne savions pas exactement comment fonctionnait Unity. Le lutin a donc dû être refait plusieurs fois, par exemple à cause du fait que ses membres doivent être séparés pour créer des animations.



Le premier personnage

La seconde avancée dans cette partie a été la création des graphismes d'un second ennemi : la harpie. Cet ennemi sera, dans le futur, un ennemi volant. Elle a été terminée peu avant la soutenance, c'est pourquoi elle n'est pas encore tout à fait implémentée dans le jeu.

Une harpie, selon la mythologie grecque, est un animal au corps d'oiseau et à la tête de femme. Celle-ci a donc été réalisée à partir d'une photo d'un vrai oiseau. Elle a été fortement retouchée par la suite, mais les ailes sont très proches de l'image d'origine. Bien sûr, la tête et les serres ont été rajoutées à la main.



Second ennemi : une harpie

Les tuiles, quant à elles, restent les mêmes que précédemment. Le seul ajout, mineur, a été un trait horizontal noir, permettant de séparer le "monde du haut" du "monde du bas".

3.3 Animations

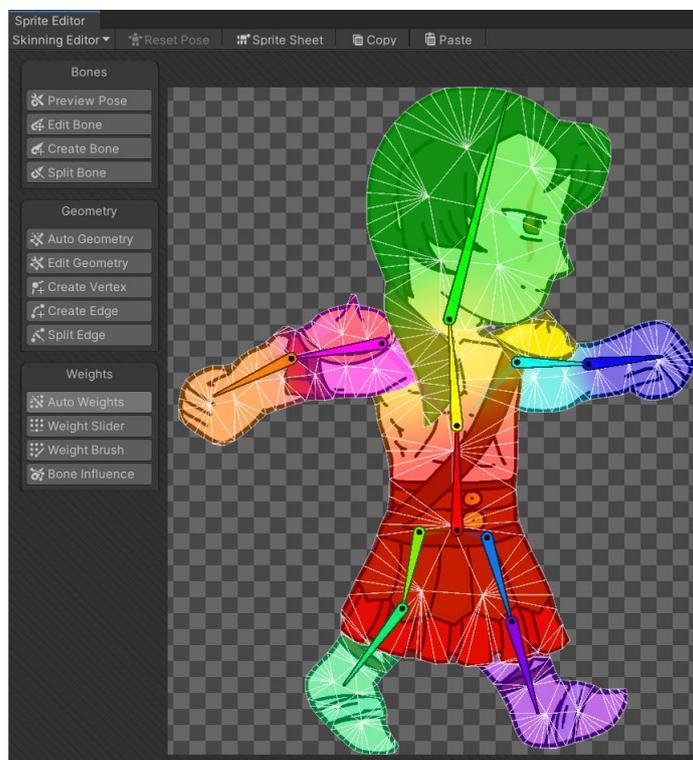


Découpage du personnage pour les animations

Comme dit dans la partie "Graphismes", les animations ont nécessité un découpage particulier du personnage. En effet, chaque membre doit être séparé du reste du corps, afin qu'il puisse être indépendant. Ce découpage est fait automatiquement par Unity grâce au paquet "PSD Importer". Gimp ne supportant pas le format ".psb", il a fallu renommer les fichiers graphiques avec cette extension. Par la suite,

en les exportant dans Unity, chaque calque est automatiquement transformé en une "partie" du personnage.

Par la suite, il a fallu créer les "os" du personnage. Cela se fait grâce au "Skinning Editor" de Unity. Sur l'image ci-dessous, les traits de toutes les couleurs représentent les os. Chaque membre du personnage est pourvu d'un os qui permettra de le faire bouger dans le cadre des animations. On peut également voir que le personnage est multicolore : il s'agit des poids, qui déterminent la dépendance de chaque membre par rapport aux os. Par exemple, la tête est uniquement dépendante de l'os vert, ce qui signifie que modifier la rotation de l'os bleu n'aura pas d'influence sur la tête. La modification de certains os peut entraîner des déformations ; par exemple, si on bouge un avant-bras du personnage, le reste du bras se "plie" pour le suivre. Les déformations ainsi créées dépendent de la géométrie du lutin, et donc des vecteurs, qui sont les traits blancs que l'on peut voir partout dans le personnage.

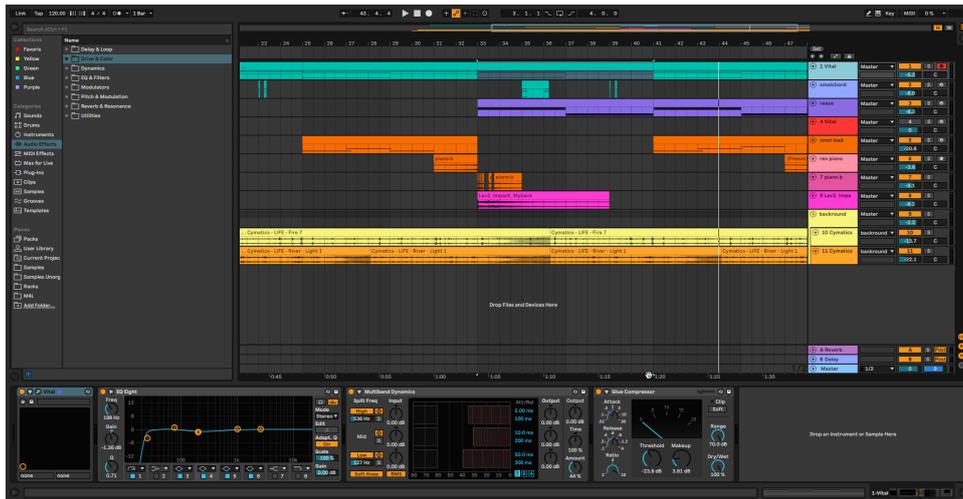


Éditeur de lutin de Unity

Une fois les os et les dépendances gérés, il est possible de commencer à créer des animations grâce à l'éditeur d'animation d'Unity. Celui-ci nécessite le paquet "2D Animation". Les animations sont très intuitives à créer, car il suffit de faire bouger les os du personnage en fonction de ce que l'on souhaite lui faire faire : si on veut qu'il lève le bras, alors on lui fait lever le bras. Il faut donc créer différentes poses que l'on agence comme on le souhaite dans une ligne du temps. Après avoir créé l'animation étape par étape, celle-ci se fluidifie automatiquement entre les différentes poses que l'on a pu donner au personnage.

3.4 Audio

Nous avons maintenant une nouvelle musique de niveau qui vient s'ajouter à celle du menu, qui était déjà présente lors de la deuxième soutenance. C'est la première fois que nous nous essayons à la composition pour un jeu vidéo et il n'a été possible que d'en ajouter une jusqu'à cette deuxième soutenance ; nous sommes cependant satisfaits du résultat.



Projet du menu principal

Nous avons modifié notre AudioManager présenté à la dernière soutenance afin de pouvoir gérer les effets sonores et de pouvoir superposer certains sons et pas d'autres à l'aide d'attributs booléens. Nous avons maintenant des effets sonores pour la marche, le saut et l'accélération. A l'exception de la marche qui a été récupérée sur internet, les effets sonores ont été synthétisés avec Vital, un synthétiseur logiciel gratuit.



Capture d'écran de Vital (Saut)

3.5 Construction des mécaniques

La construction des mécaniques regroupe plusieurs choses : les mécaniques de jeu, le timing avec lequel nous les donnons au joueur et enfin l'équilibrage de ces mécaniques afin qu'aucun des deux joueurs ne soit laissé pour compte.

Sur ces sujets, après mûre réflexion, nous avons déterminé que les joueurs n'auront pas les mêmes capacités excepté pour la mécanique principale du jeu : échanger de position pour pouvoir passer certains obstacles. Au début du jeu, un des joueurs aura une capacité d'accélération lui permettant de se projeter vers l'avant à une grande vitesse. Cela lui permettra de passer de gros gouffres. L'autre joueur aura une capacité de double saut lui permettant d'atteindre des positions élevées sans avoir besoin de plateformes. Avec ces capacités en main, les joueurs devront atteindre la fin du niveau. Après avoir vaincu des boss, les joueurs se verraient octroyer de nouvelles capacités pouvant être spécifiques à chacun ou communes. Cela permettrait de donner un vent de fraîcheur et de nouveaux outils aux joueurs après qu'ils auront exploré les précédentes capacités au maximum.

Les niveaux seront agencés par groupes de 2 niveaux. Chaque groupe aura des concepts particuliers. Nous pensons faire 8 niveaux et 2 boss qui devront tous tourner autour de la mécanique principale. Il faudra donc proposer des niveaux asymétriques permettant aux deux joueurs de s'amuser et de s'entraider.

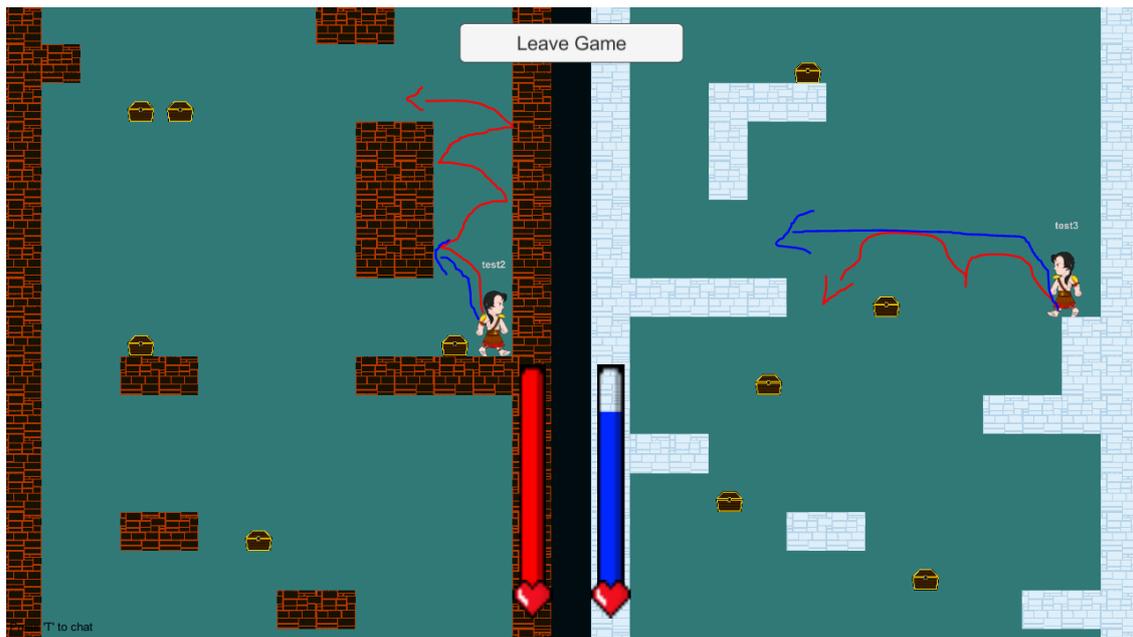
Cela sera à la fois un défi technique et un défi créatif. Nous devons réussir à être ambitieux et originaux sans pour autant nous perdre dans un torrent d'idées.

3.6 Conception de niveau

La conception du niveau a été particulièrement intéressante et amusante tout en étant réellement difficile. Contrairement à n'importe quel jeu, *Fracture* repose sur un écran scindé en permanence. De ce fait, nous ne pouvons pas utiliser pleinement l'axe Y. De plus, nous sommes obligés de placer des éléments bloquants pour les joueurs, les obligeant à échanger leurs positions. Cette restriction se fera sous la forme de deux niveaux asymétriques ne proposant pas les mêmes obstacles à passer. C'est ce qui permet d'avoir une rejouabilité mais surtout, c'est ce qui rend notre jeu unique.



Vous pouvez voir sur l'image ci-dessus l'un des exemples de différence ; en rouge, on peut voir le premier joueur et la trajectoire qu'il peut prendre avec ses capacités actuelles. Ensuite, on peut voir la même chose pour le second joueur, en bleu. Sur la partie du haut, le joueur doit effectuer un double saut pour passer l'obstacle tandis qu'en bas, le joueur doit effectuer une accélération. Cet exemple est rudimentaire car ce niveau est un niveau d'introduction au jeu et à la mécanique du jeu. Dans les prochains niveaux, les possibilités seront variées et demanderont de plus grandes capacités d'analyse. La mécanique d'échange peut sembler être une restriction cependant il est intéressant de les placer et la création de ce premier niveau qui est encore à l'état d'un niveau de démonstration nous a permis d'entrevoir toutes les utilisations de cette mécanique pour les prochains niveaux du jeu.



Sur l'image ci-dessus, on a un autre exemple de différence. Cette différence met en valeur une de nos nouvelles mécaniques, le *wall-jump* ou saut mural. Le personnage pouvant faire un saut mural doit l'utiliser pour passer sur la gauche ; celui de droite doit utiliser sa propulsion habituelle.

Depuis la première soutenance nous sommes passés d'un seul et unique niveau horizontal à deux niveaux horizontaux et un niveau vertical. Ce niveau vertical a été atypique à conceptualiser, contrairement aux niveaux horizontaux. Cela a été dur d'intégrer la patte de fracture à ce niveau : mettre en valeur les moments d'échanges tout en offrant une construction de niveau qui permet aux joueurs d'y progresser de manière intuitive.

3.7 Multijoueur

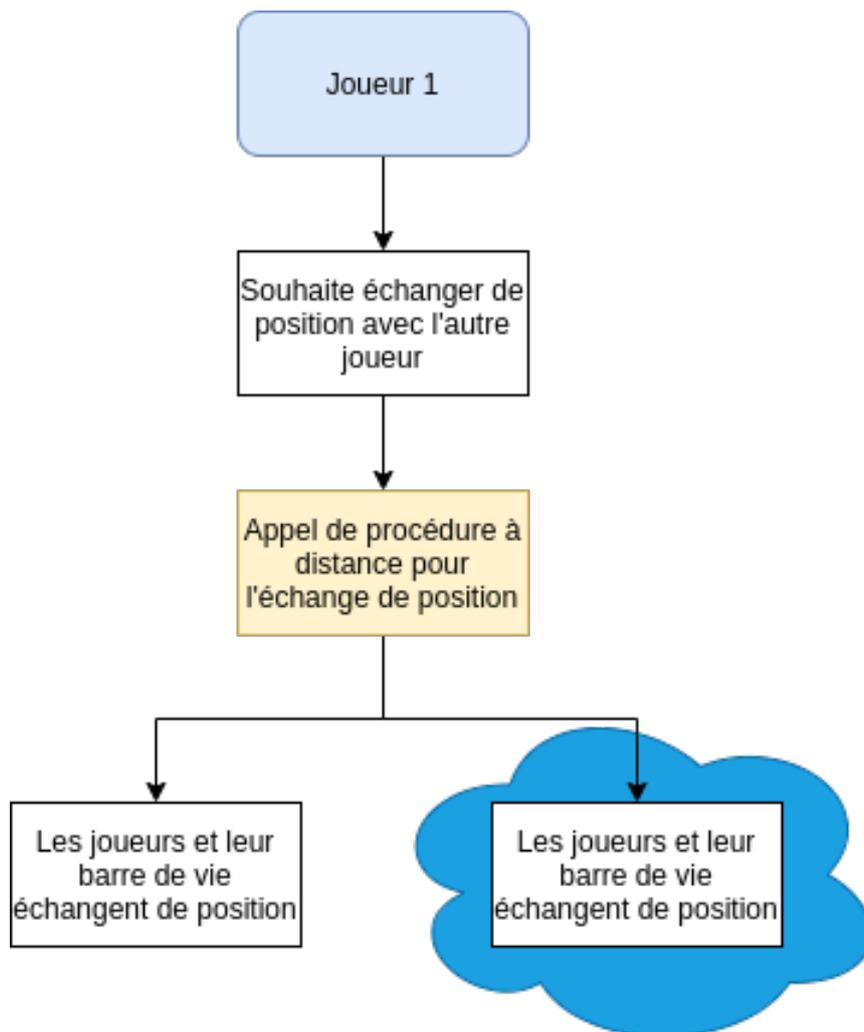
3.7.1 Photon Unity Networking 2

C'est Toan qui s'est chargé de l'implémentation du multijoueur du jeu. Afin d'y parvenir, nous avons utilisé la version gratuite de l'asset [Photon Unity Networking 2](#) (PUN2) de Exit Games, présent sur l'Asset Store de Unity. Ce dernier permet de connecter jusqu'à 20 joueurs dans une même partie.

De plus, Photon est particulièrement intéressant quand il s'agit de créer un jeu avec des parties comprenant un nombre limité de joueurs, ce qui est notre cas puisque nous souhaitons réaliser un jeu jouable à 2 joueurs. Cet asset nous laisse la liberté de choisir un système de partie publique ou de parties privées joignables avec un code. Nous avons décidé d'implémenter les deux !

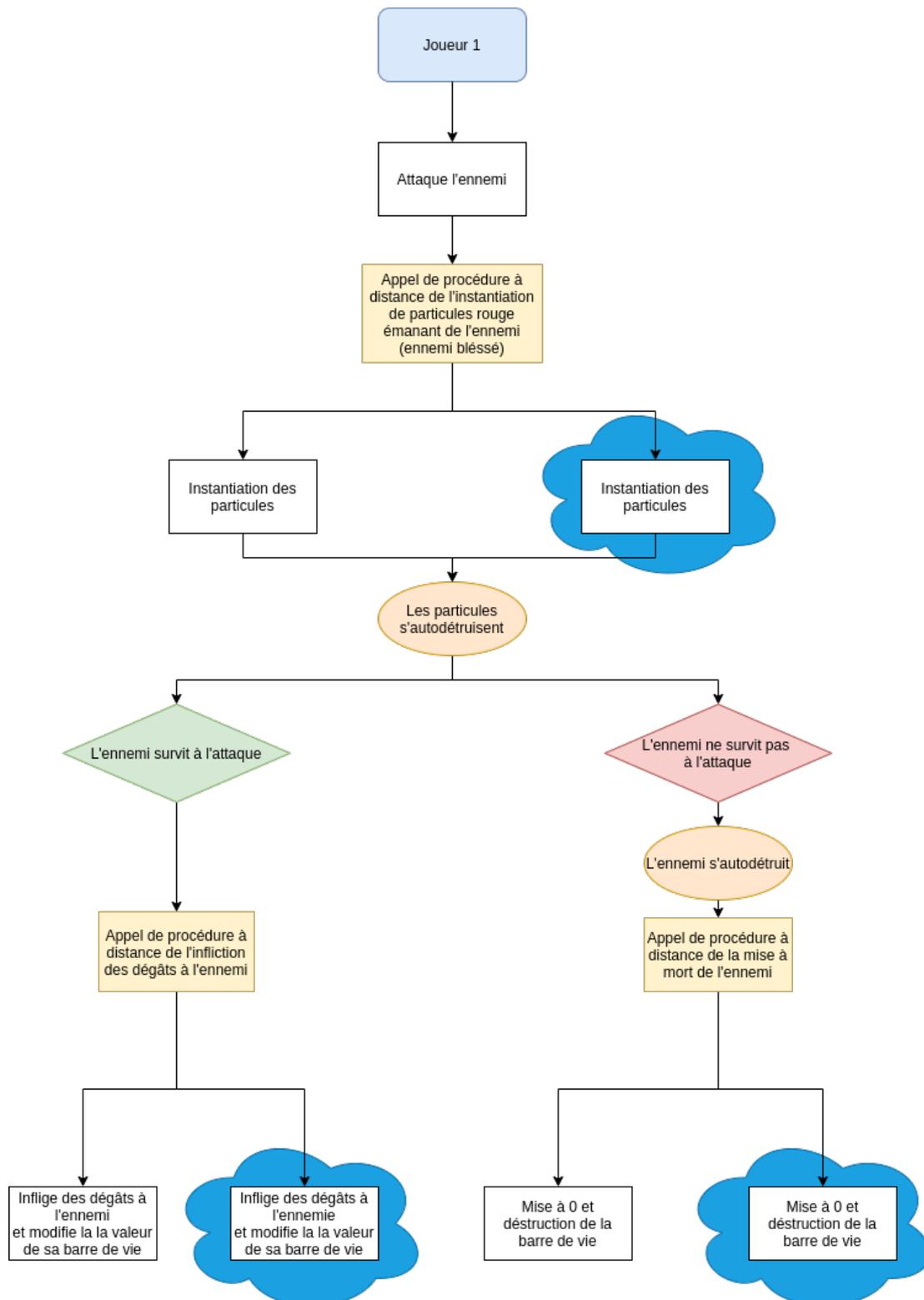
3.7.2 Appels de procédure à distance (RPC)

Ajout de la mécanique d'échange de position à l'aide de RPC



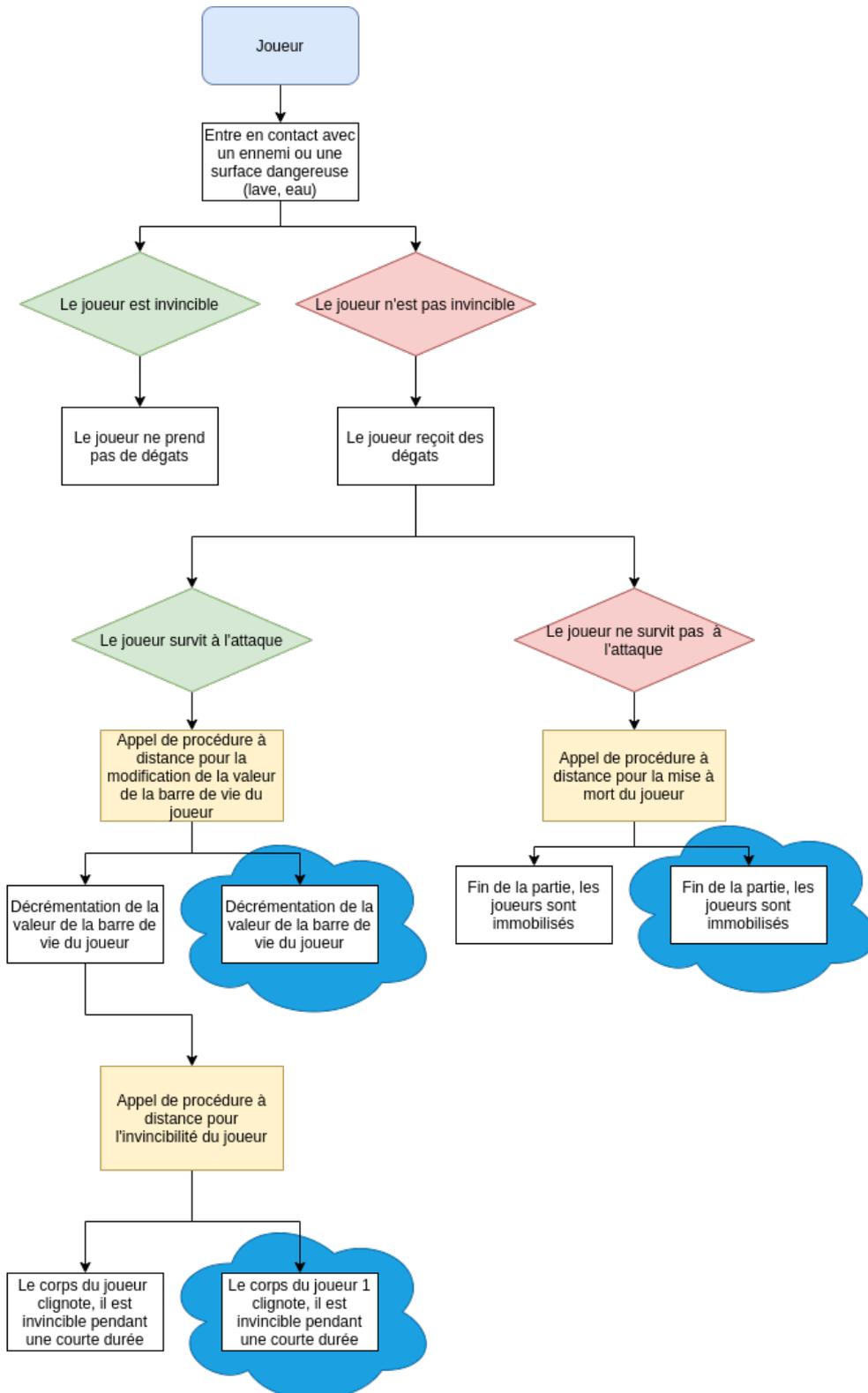
Flow chart du fonctionnement de la mécanique d'échange de position

Synchronisation des dégâts infligés aux ennemis via des appels de procédure à distance.



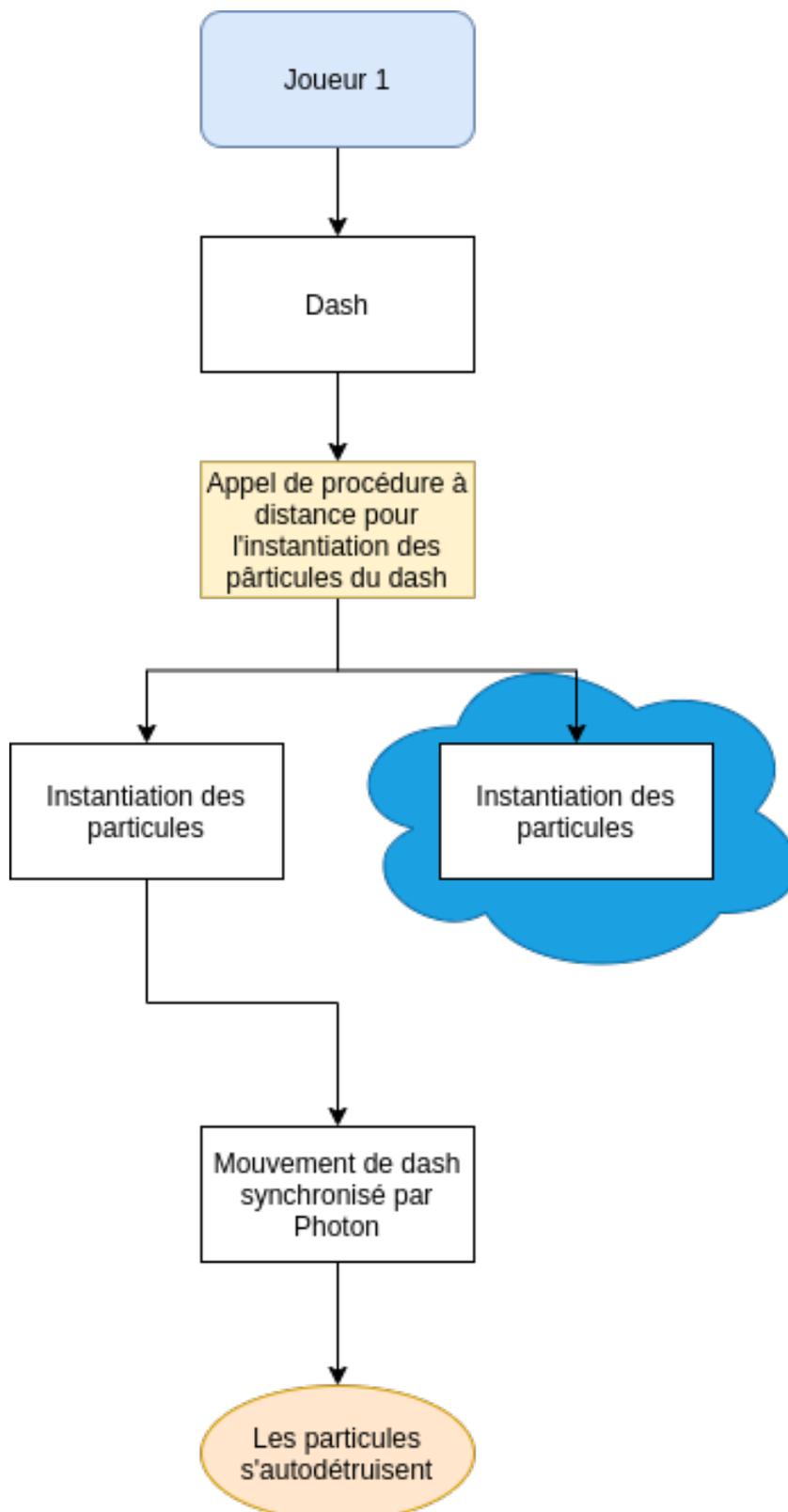
Flow chart de la synchronisation des dégâts infligés aux ennemis

Synchronisation des dégâts infligés aux joueurs via des appels de procédure à distance.



Flow chart de la synchronisation des dégâts infligés aux joueurs

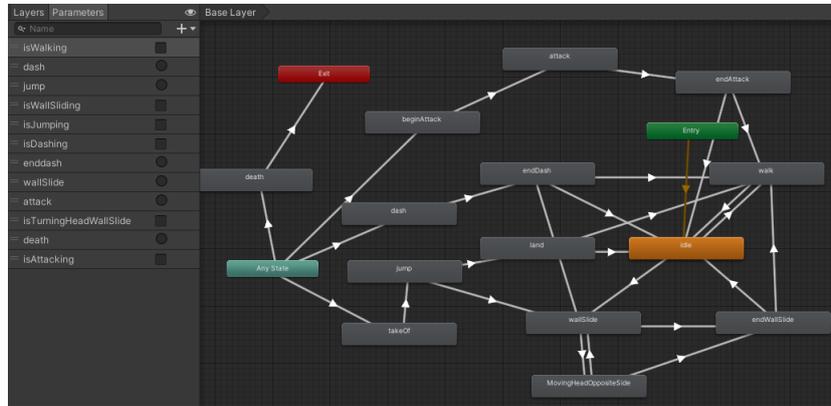
Ajout de la mécanique de propulsion en avant à l'aide de RPC



Flow chart du fonctionnement de la mécanique de propulsion en avant

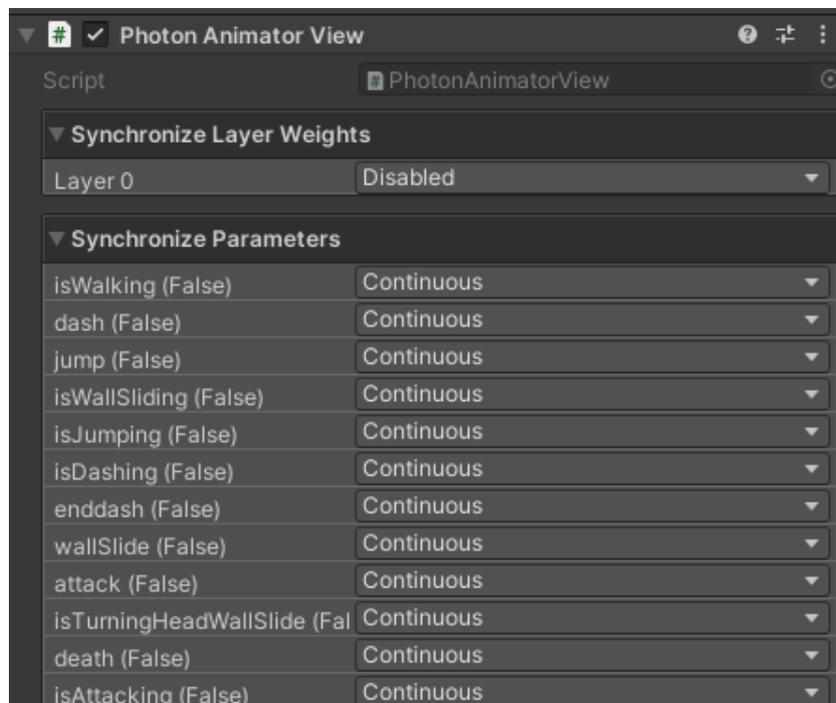
3.7.3 Synchronisation des animations

Une fois les animations créées il a fallu les déclencher au bon moment dans le code, mais aussi les synchroniser entre les deux joueurs. Il faut donc synchroniser les paramètres utilisés dans le régulateur d'animations.



Régulateur d'animations avec ses paramètres sur la gauche

Pour accomplir cela, rien de plus simple, il suffit d'ajouter un composant PhotonView à l'objet qui contient le régulateur d'animations et de demander à synchroniser tous les paramètres.

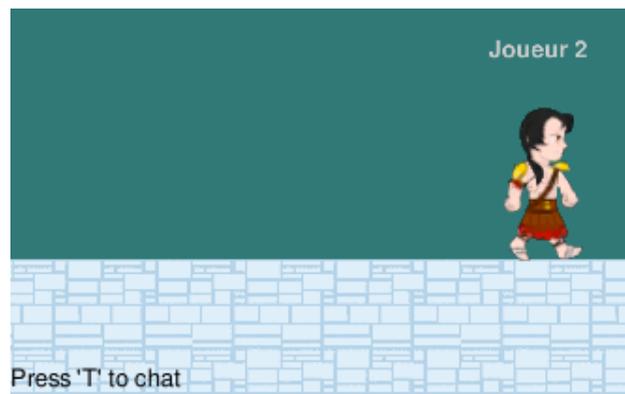


Flow chart du fonctionnement de la mécanique de propulsion en avant

La valeur *Continuous* implique que ces paramètres seront synchronisés à une très haute cadence, ce qui est exactement ce que nous souhaitons.

3.7.4 Messagerie instantanée

Nous avons décidé d'implémenter un système de messagerie instantanée entre les deux joueurs une fois qu'ils sont sur le même niveau pour aider leur coopération. Pour pouvoir taper un message il faut d'abord appuyer sur la touche 'T' pour ouvrir le menu de discussion. Le message envoyé apparaîtra sur l'écran des deux joueurs grâce à l'utilisation d'appel de procédure à distance et sera suivi du nom de l'auteur.



Messagerie instantanée fermée



Messagerie instantanée ouverte



Discussion avec la messagerie instantanée

3.8 Site Web (Toan)

Toan s'est vu attribuer la réalisation du site web. Notre site est consultable à l'adresse : <https://fracturegame.github.io>. Il a été réalisé à l'aide du générateur de sites statiques HUGO et du thème [min_night](#).

Pour le moment, le site possède 5 pages :

1. L'accueil, qui présente le projet
2. La présentation des membres du groupe
3. L'historique de notre projet avec les liens permettant de télécharger le cahier des charges et les rapports à venir
4. La chronologie de réalisation du projet avec les liens des outils utilisés
5. Une version web de notre jeu

Le site possède également un en-tête qui sert de barre de navigation et un bas de page. Il s'adapte à la taille de l'écran sur lequel il s'affiche afin que tout le monde puisse le consulter sans gêne.

Le gain de temps apporté par l'utilisation d'un thème a pu profiter au reste du site puisque davantage de soin a pu être apporté à la création des pages en HTML et CSS. Il a tout de même fallu adapter le thème qui est pensé pour être utilisé en tant que blog.

3.9 Intelligence artificielle

Depuis la première soutenance, la partie de l'intelligence artificielle n'a que très peu avancé, au profit des animations ainsi que des mécaniques de jeu. Le seul ennemi possédant une intelligence artificielle reste ainsi le blob.

4 Difficultés rencontrées et leur résolution

4.1 Difficultés individuelles

4.1.1 Maëlys

Pour cette soutenance, la compréhension des logiciels a été un problème beaucoup moins important que précédemment. En effet, j'ai dû apprendre à créer et utiliser les systèmes d'os et de dépendances pour les animations sur Unity, mais après quelques heures de pratique, cela était bien plus simple. Le plus compliqué a été de m'adapter aux outils de Unity : j'ai dû modifier les graphismes du joueur de nombreuses fois, car je ne comprenais pas bien comment implémenter les animations. Grâce à des vidéos YouTube, j'ai finalement réussi à comprendre le fonctionnement du système d'animations. Un autre problème était la gestion de mon espace de travail : ayant un ordinateur portable et un fixe, je dois m'organiser afin de toujours travailler sur un seul des deux à la fois. Pour résoudre ce problème, j'ai simplement décidé de travailler exclusivement sur mon ordinateur portable. Ensuite, j'ai rencontré des problèmes d'ordre matériel, avec un disque dur dysfonctionnel à cause duquel je n'ai pas eu accès à mon travail pendant plusieurs jours. Récupérer mes données a été long, mais heureusement, rien n'a été perdu.

4.1.2 Raphaël

Comme expliqué dans les difficultés communes, j'ai eu du mal à gérer mon temps entre les examens, TPs et le projet, mais mes plus gros problèmes étaient liés à git étant donné que j'étais chargé de résoudre les conflits de merge. Je n'ai au final pas réussi à le faire avec git, et nous sommes repartis de la branche de Toan et avons refait nos modifications dessus. J'ai aussi eu quelques soucis pour la gestion de la barre de vie afin de la synchroniser entre 2 clients même si j'ai au final réussi à les régler avec l'aide de Toan et des forums Unity.

Il était cependant plus facile de manier Unity qu'à la dernière soutenance, même si j'ai encore du mal avec la gestion de l'affichage notamment pour la résolution de l'écran, car certains éléments de l'interface utilisateur ne se positionnaient pas bien entre les 2 jeux lorsqu'ils avaient des résolutions différentes, mais ce problème est finalement résolu.

4.1.3 Adam

Pour cette seconde période de travail j'ai rencontré beaucoup de problèmes. La première difficulté fut de sauvegarder mon travail, car j'ai perdu en tout 3 fois ce que j'ai fait. Heureusement, j'avais noté le niveau 3 sur des feuilles. L'utilisation d'Unity pour créer les animations fut difficile et malheureusement, j'ai perdu lesdites animations. Comme d'habitude, Git a causé des problèmes. Sinon, j'ai beaucoup apprécié le fait que nous ayons échangé certaines de nos tâches ; cela nous permet d'acquérir des compétences et cela améliore l'esprit d'équipe. La création du niveau 3 a été très intéressante ; ce niveau est assez atypique mais tout de même amusant à parcourir.

4.1.4 Toan

Durant cette seconde période de développement, j'ai rencontré plusieurs difficultés. C'est tout d'abord la mécanique d'échange de position qui m'a posé problème. En effet, lorsque j'ai commencé à la développer, je ne connaissais pas les *appels de procédures à distance* ou RPC et ai d'abord tenté plusieurs approches plus compliquées les unes que les autres et peu efficaces. J'ai finalement découvert cette fonctionnalité et l'ai utilisé à de nombreuses reprises pour les mécaniques, pour les variables relatives aux objets telles que le nombre de points de vie ou l'invincibilité et l'apparition de particules chez les deux joueurs.

De plus, j'ai également eu des difficultés à réaliser le régulateur d'animations. De fait, il y a de nombreux paramètres à prendre en compte pour chaque transition et le timing entre les animations est souvent très serré.

4.2 Difficultés communes

4.2.1 Mise en commun du travail et Git

Notre organisation se porte mieux qu'avant la dernière soutenance et nous avons plutôt bien réussi à travailler sur des branches différentes et à mettre en commun nos résultats toutes les 2 semaines. Cependant, lors du merge final avant la soutenance, nous avons souffert de nombreux "merge conflicts", que nous n'avons pas réussi à résoudre, principalement sur des fichiers scènes d'Unity. Nous avons essayé le YAML Merge tool proposé par Unity, qui avait marché avant la première soutenance mais qui n'a pas fonctionné ici, ainsi que de les résoudre à la main sur l'interface web de Github. Nous avons fini par opter pour refaire les travaux les moins longs sur la branche qui contenait les modifications les plus importantes, ce qui a plutôt bien marché, même si beaucoup plus fastidieux que si le merge s'était bien passé.

4.2.2 Travail du projet en plus du travail pour les examens

La soutenance faisant suite à une période d'examens, il a été difficile pour nous de consacrer du temps au projet car nous devions réviser et faire nos TP, qui augmentent en difficulté. Nous avons tout de même réussi à travailler sur le projet avant cette période, mais nous n'avons pas pu bénéficier autant des semaines précédant la soutenance que lors de la première. Il a également été compliqué de commencer le rapport de soutenance en avance, la semaine de vacances ayant été dédiée principalement aux derniers ajouts sur le projet et à la mise en commun du travail, qui a comme expliqué ci-dessus été plus problématique que prévu.

5 Avances et retards

5.1 Produit d'amusement minimal

Note importante : Afin d'être plus conscient des avancées et retards de notre projet, nous avons choisi de ne plus les représenter sous forme de pourcentage, une présentation qui n'était pas assez concrète et qui pourrait, selon nous, se révéler sournoise. Nous avons donc décidé de nous baser sur notre produit d'amusement minimal.

Tâches	2ème soutenance	Produit d'amusement minimal
Interface graphique	Oui	Menu basique
Graphismes 2D	ensemble de tuiles, 2 ennemis (blob, harpie), arme du personnage, 1 personnage manquant	ensemble de tuiles, 2 personnages, 1 ennemi
Animations	Animation d'un ennemi et d'un personnage	Animation d'un ennemi et des deux personnages
Audio	1 musique de menu et une musique de jeu	1 musique de jeu
Construction de mécaniques	6 (Saut sur les murs, changement de monde, ruée, saut, double saut, marche, attaque)	4 mécaniques avec le changement de monde
Conception de niveau	3 niveaux	1 niveau
Multijoueur	Oui	Multijoueur fonctionnel
Site Web	Oui	Pas nécessaire
Intelligence artificielle	Oui, l'intelligence artificielle du blob	Intelligence artificielle d'un ennemi

5.2 Avancement des objectifs fixés par le premier rapport de soutenance

Interface graphique Premièrement, en ce qui concerne l'interface graphique, nous pensons être en accord avec nos estimations du dernier rapport. En effet, nous avons ajouté un écran de sélection des différents niveaux. A chaque nouvelle fenêtre, il y a toujours un bouton 'Back' nous permettant de revenir en arrière et il est également possible de quitter une partie via l'interface graphique de notre jeu en informant le joueur qui reste seul de cette action. Les cas où le maître ou le client s'en irait avant le début de la partie ont également été gérés dans notre menu.

Graphismes 2D Pour ce qui est des graphismes, Maëlys a réalisé le premier personnage. De plus, les graphismes d'un second ennemi ont aussi été réalisés. Pas de retard à signaler à ce niveau-là non plus.

Animations Les animations, quant à elles, ont bien avancé durant cette deuxième période de développement. En effet, nous avons réalisé le système d'os du personnage ainsi que ses animations d'inaction, de marche, saut, propulsion en avant, glissade sur le mur et saut depuis le mur et enfin celle de sa mort.

Audio L'audio, réalisé par Raphaël, est en très bonne voie avec l'ajout d'une musique de jeu et d'effets sonores pour la marche, le saut et la propulsion en avant du joueur.

Conception de niveau Concernant la création des niveaux, nous avons ajouté 2 nouveaux niveaux : un horizontal et un vertical.

Construction des mécaniques Cette deuxième soutenance voit l'ajout des mécaniques de changement de monde, de glissade et saut sur les murs. Nous avons donc réalisé l'ensemble des mécaniques essentielles au déplacement de notre personnage.

Multijoueur La partie multijoueur a surtout consisté à la synchronisation des nouvelles fonctionnalités que nous avons ajoutées. Son avance est en accord avec ce que nous avions prévu.

Site internet Le site connaissait une légère avance par rapport à la soutenance précédente. Il est désormais dans les temps avec la mise à jour de la page *Chronologie*.

Intelligence artificielle Enfin, le développement de l'intelligence artificielle de nos ennemis connaît un léger retard même si notre projet comporte tout de même celle du blob. Nous avons revu à la baisse nos objectifs pour cette tâche qui étaient peut-être un peu trop ambitieux. Il faudra en faire une priorité pour la dernière soutenance.

6 Prédiction de l'avancement à la dernière soutenance

Interface graphique

Nous avons réussi à avoir un menu fonctionnel. A présent, nous allons essayer de l'embellir avec un écran d'accueil plus beau et sophistiqué ; vous trouverez en annexe un concept graphique pour ce menu.

Graphismes 2D et Animations

Nous espérons ajouter le second personnage ainsi qu'au moins un ennemi de base, plus robuste que les précédents. De plus, il faudrait créer un boss de fin. Pour ce qui est des animations, il faudra créer les animations du 2ème personnage ainsi que des ennemis supplémentaires. Enfin, il faudrait ajouter un fond aux niveaux, qui les rendrait plus beaux.

Construction des mécaniques

Les mécaniques que nous souhaitons sont toutes implémentées.

Construction de niveau

Nous espérons réussir à produire au moins 5 niveaux d'ici à la prochaine soutenance.

Site internet

Nous devons mettre un lien de téléchargement du jeu.

Audio

Nous espérons faire au moins 3 musiques de niveaux différentes d'ici à la prochaine soutenance.

Intelligence artificielle

Nous espérons créer une intelligence artificielle pour deux nouveaux ennemis ; une intelligence artificielle qui tenterait de suivre le joueur et une autre qui serait plus complexe (et qui pourrait être celle du boss de fin, par exemple).

7 Conclusion

Arrivés au terme de cette deuxième période de développement, nous sommes fiers du travail accompli jusqu'à aujourd'hui et confiants quant à l'avenir du projet pour la dernière soutenance. Nous, les Singes de Schrödinger, avons su utiliser les connaissances acquises afin de réaliser nos tâches respectives avec plus d'efficacité. C'est donc avec fierté que nous présentons une version bêta de notre jeu *Fracture*.

Cette version comporte déjà de nombreuses composantes du jeu final attendu pour juin. En effet, elle est composée d'une interface graphique sobre mais élégante représentant différents menus. Elle est également munie d'une musique principale et permet aux joueurs d'effectuer les actions de base que l'on attend d'un jeu vidéo. Le joueur a le choix entre plusieurs systèmes de jumelage pour rejoindre un coéquipier dans une partie et plusieurs niveaux. Une fois entré dans le niveau, le joueur peut constater le travail fourni au niveau des graphismes 2D faits maison mais aussi de la réflexion nécessaire à la conception du niveau et des mécaniques de jeu. Le jeu possède les animations du personnage et sa première intelligence artificielle. Il nous semble important de préciser que la partie créative de notre jeu est entièrement conçue par nous-même.

Afin d'aboutir à ce résultat, il a fallu conserver, voire améliorer nos méthodes de travail et notre organisation en tant que groupe. Cela s'est notamment fait à l'aide de l'application Trello et des réunions de groupe qui ont lieu tous les lundis soir.

Le chemin pour arriver jusqu'ici a été semé d'embûches mais les difficultés rencontrées ont toujours pu être surmontées, telles que la gestion de la mise en commun du travail à l'aide de Git ou encore le fait de toujours devoir travailler en distanciel.

Les tâches sont dans leur globalité en bonne voie même si certaines auraient pu bénéficier d'un peu plus d'attention. Il faudra redoubler d'efforts pour atteindre les objectifs que nous nous sommes fixés pour la soutenance finale.