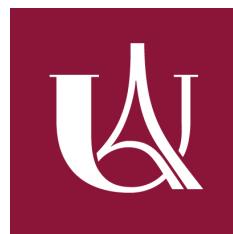


Rapport de projet

Groupe L3C2

Jeu de stratégie et IA

TALEB Rayane, MENGQIAN Xu, NDEYE PATHE Mbaye, TAO Zeyu



Université de Paris
UFR de Mathématiques-Informatique

2022-2023

Remerciements

Nous adressons nos sincères remerciements à Monsieur Claude LATTAUD, notre encadrant, qui durant tout le déroulement du projet nous a orienté et guidé pour atteindre de meilleures perspectives. Nous sommes reconnaissants de l'opportunité qui nous a été donnée de travailler avec lui et nous avons beaucoup appris grâce à ses enseignements.

Nous souhaitons également remercier Monsieur David JANISZEK, responsable de l'unité d'enseignement Projet de Programmation, pour nous avoir offert cette occasion de travailler sur un projet concret en utilisant les connaissances acquises tout au long de notre formation.

Enfin, nous tenons à remercier nos enseignants et l'université pour leur soutien continu et leur engagement envers notre réussite. Nous sommes honorés d'avoir eu l'occasion de travailler sur ce projet et nous sommes reconnaissants de toutes les leçons précieuses que nous avons apprises grâce à cette expérience.

Table des matières

1	Introduction	4
1.1	Présentation	4
1.2	Objectifs	4
2	Guide de lecture	4
2.1	Maîtrise d'œuvre	4
2.2	Maîtrise d'ouvrage	5
3	Démarrage du projet	5
3.1	Début du projet	5
3.2	Outils et langages de programmation	5
3.3	Organisation	6
4	Déroulement du projet	6
4.1	Documentation	6
4.2	Répartition des rôles	8
4.3	Ordre de développement	8
4.4	Points techniques généraux	9
5	Résultats	9
5.1	Fusion du code de l'an dernier	9
5.2	Nouvelles compétences des personnages	9
5.3	Amélioration des scripts CPU	14
5.4	Amélioration de l'interface	15
5.5	Autres améliorations	19
6	Défis à relever	20
6.1	Les différentes réorganisations	20
6.2	Difficultés techniques	20
7	Conclusion	22
8	Références	23
9	Glossaire	23

1 Introduction

1.1 Présentation

Ce document a pour but de documenter l'avancement du groupe L3C2 sur son projet de programmation de fin de licence. Le projet "Pause café" est une Unité d'Enseignement de troisième année de licence Informatique et Applications à l'université de Paris. Le but de ce projet est de montrer que les étudiants, grâce à leurs connaissances acquises lors de la licence, peuvent développer un projet.

Ce projet est la suite d'un jeu de stratégie développé par des étudiants des années précédentes. Le projet est déjà fonctionnel, mais notre objectif est d'améliorer le jeu en fusionnant les codes des deux groupes de l'année dernière et d'ajouter une nouvelle compétence pour chaque personnage. Nous avons également pour objectif d'améliorer les scripts du joueur CPU afin d'améliorer son niveau.

Nous vous détaillerons dans ce document notre organisation pour réaliser ce projet, le travail effectué mais aussi les problèmes auxquels nous avons été confrontés.

1.2 Objectifs

Notre rôle était de continuer le travail effectué par les groupes précédents tout en améliorant le jeu. Pour cela, nous devions :

- Commencer par étudier le code existant des deux groupes précédents et fusionner les parties essentielles.
- Travailler sur l'ajout de nouvelles compétences pour chaque personnage.
- Trouver un meilleur équilibrage entre les différentes classes de personnages.
- Passer du temps à améliorer les scripts du joueur CPU pour qu'il soit plus compétitif.

2 Guide de lecture

2.1 Maîtrise d'œuvre

La maîtrise d'œuvre est assurée par le groupe L3C2, composé par Mengqian Xu, Zeyu Tao, Rayane Taleb, et Ndeye Pathe Mbaye.

2.2 Maîtrise d'ouvrage

Le maître d'ouvrage de ce projet est monsieur LATTAUD Claude, qui communiquera avec le groupe L3C2 par la Forge, également par le biais de réunions hebdomadaires chaque jeudi de 8h à 8h45.

Monsieur LATTAUD Claude s'assurera que le produit final correspond bien aux besoins exprimés par le client.

3 Démarrage du projet

3.1 Début du projet

Cette UE se différencie des autres unités d'enseignement par le fait que nous soyons un groupe d'étudiants avec un même projet à réaliser sur plusieurs mois. Nous avons pu soumettre une liste de vingt voeux parmi une banque de sujets proposés. Après une attribution des différents voeux, le projet L3C – Unity 3D : Stratégie et IA nous ont été attribués.

Après avoir pris contact avec notre encadrant, monsieur Claude LATTAUD, il nous a été assigné la tâche de réaliser plusieurs tutoriels avant la première réunion de groupe afin de prendre en main l'outil qui a permis la bonne exécution du projet : le moteur de jeu Unity 3D. Grâce à cela, nous avons vite pu comprendre les limites du projet ainsi que les possibilités d'améliorations qui étaient à notre portée.

S'en est suivi la première réunion entre les deux groupes de ce projet, L3C1 et L3C2, et leur encadrant. Celle-ci a pu éclaircir plusieurs points attendus à la fin du semestre tels que les objectifs principaux à atteindre ainsi que les différentes idées des groupes concernant les implémentations à faire et le fonctionnement du jeu.

L'encadrant a exprimé dès le début du projet l'importance d'un travail continu ainsi qu'une bonne entente de groupe afin de pouvoir fournir un travail complet, consistant et robuste qui permettrait d'obtenir un jeu fonctionnel et stable. Il nous a également rappelé les principes de la forge ainsi que chaque document à soumettre avant la date finale de rendu.

3.2 Outils et langages de programmation

Le jeu Pause Café a été créé avec le moteur de jeu Unity 3D, et les scripts sont développés en C#.

Concernant notre avancée dans le code, nous travaillons avec le logiciel de versionning SVN afin de nous permettre de travailler chacun de notre côté sur le même projet et en même temps voir les modifications des autres membres du groupe.

3.3 Organisation

Au début du projet, nous élisions un chef de projet différent par semaine, ce dernier s'occupait d'envoyer le compte rendu de la réunion hebdomadaire sur la forge que nous faisions tous ensemble à l'aide de Google Doc. Le chef de groupe était également là pour essayer de planifier et organiser le travail de chacun.

Pour la rédaction de la documentation, Rayane était le seul pour qui le français est la langue maternelle, il est donc chargé de vérifier les documents pour éviter les problèmes de grammaire et d'orthographe.

Notre répartition du travail et notre communication passent principalement par le logiciel “what's app”, nous utilisons également SVN pour savoir l'avancement des tâches des membres de l'équipe.

4 Déroulement du projet

4.1 Documentation

Pour sa conception, notre projet a suivi le modèle du cycle en V. Le nom de ce modèle de conception fait référence à la vision schématique de ce cycle, qui prend la forme d'un V : une phase descendante suivie d'une phase ascendante.

Le cycle en V associe à chaque phase de réalisation une phase de validation, comme l'illustre le schéma ci-dessous :

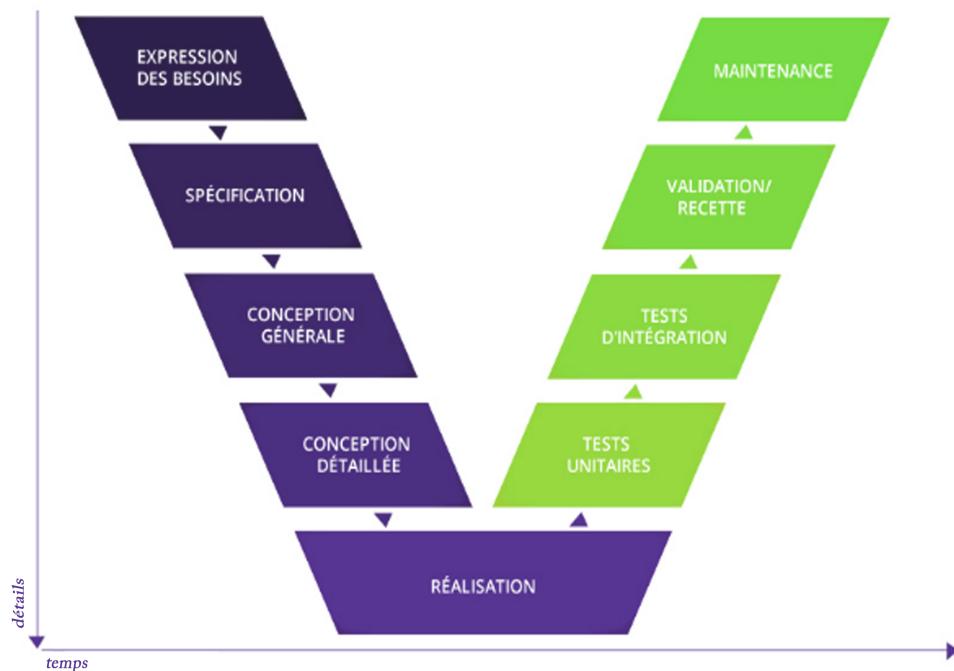


Figure 1 : modèle du cycle en V

Dans ce cadre, les premières phases exigent une documentation posant des objectifs qui seront testés par les dernières phases de ce cycle. Nous avons donc rédigé les documents suivants durant les phases de conception de notre projet :

- **Le cahier des charges** (à rendre pour le 2 février) : c'est un document rassemblant les obligations et les éléments nécessaires pour définir un besoin et les principales contraintes à respecter pour le satisfaire. Le cahier des charges est un élément objectif qui permet à un client de choisir son fournisseur.

- **Le cahier de recettes** (à rendre pour le 9 février) : Le cahier de recette est un document qui sert d'accusé de réception, dont le but est de confirmer la bonne réalisation des fonctionnalités demandées par le client. Le cahier de recette va décrire le contexte, et le détail du déroulement de ces étapes. L'ensemble des tests va permettre de valider le bon comportement fonctionnel de l'application du point de vue du client.

Ces tests permettent à l'équipe de développement de définir ses objectifs fonctionnels et ainsi de savoir comment sera évaluée son application. À l'issue de la recette, selon que l'application passe correctement l'ensemble des tests, le client peut accepter ou refuser le rendu du projet. Étant donné leur importance, ces tests doivent être définis dans le détail.

- **Plan de développement** (à rendre pour le 16 février) : Le plan de développement est détaillée présente toutes les fonctions, méthodes, classes, paquetages, librairies... qui seront nécessaires au bon développement de l'application. Ce document vise à faciliter l'implémentation de notre projet par les développeurs et vise à garantir que le fonctionnement du logiciel produit correspondra bien aux besoins de l'utilisateur final.

- **Manuel d'installation et manuel d'utilisation** (à rendre pour le 20 avril) : Le manuel d'installation est un document rassemblant l'ensemble des procédures nécessaires à la mise en place de notre jeu dans son environnement de production. Le manuel d'utilisation est un document décrivant l'ensemble des fonctionnalités de notre projet ainsi que les actions à réaliser afin de le mettre en œuvre. Le manuel d'utilisation doit permettre à l'utilisateur final d'exploiter ses fonctionnalités.

- **Plan de test** (à rendre pour le 20 avril) : Le plan de test peut être considéré comme un manuel. Il décrit les objectifs des tests (ce qu'il faut vérifier et/ou valider), la portée des tests (ce qui sera et ne sera pas testé), ainsi que le calendrier général voire détaillé des activités à effectuer (comment le tester et quand le tester).

4.2 Répartition des rôles

Afin d'ordonnancer et de gérer le temps, nous avons utilisé un diagramme de Gantt pour la conception de notre projet.

Voici notre diagramme de GANTT :

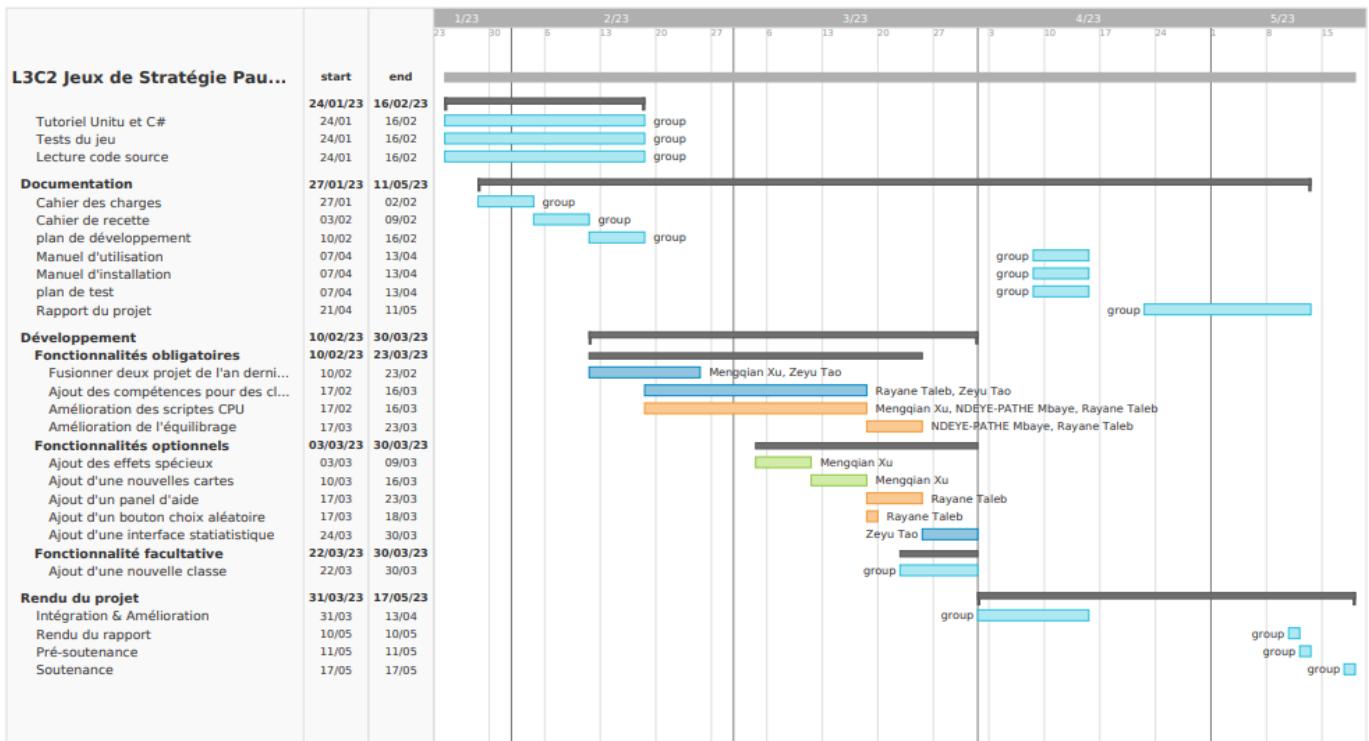


Figure 2 : Diagramme de Gantt

Dans l'ensemble, nous avons tout au long du projet respecté la répartition des tâches et des dates. Pour ce qui est du développement, nous avons en plus amélioré l'interface et ajouté une nouvelle carte que nous n'avions pas prévue lors de la réalisation de ce diagramme de Gant.

4.3 Ordre de développement

Lors de la première semaine du projet et lors de notre prise en main du code et du jeu, nous avons pensé et imaginé les points d'améliorations pour le jeu, ce qui nous a permis d'établir les différentes fonctionnalités citées dans le cahier des charges et le cahier de conception.

Mengqian et Zeyu ont d'abord fusionné les codes des deux groupes. Le projet Q1 étant plus complet, après discussion en groupe, nous avons décidé d'ajouter le personnage de Q2 à Q1.

Par la suite, les membres du groupe décident par la discussion de déterminer les nouvelles compétences de chaque personnage. Après avoir rassemblé nos codes sur la même branche, Rayane s'est occupé d'ajouter la compétence du druide du projet précédent ainsi que les nouvelles compétences pour les huit personnages, y compris un nouveau bouton qui est nommé "Compétence 2(B)" afin d'utiliser cette compétence. Nous nous sommes ensuite concentrés sur l'amélioration du script de la partie CPU, en même temps, Rayane implémente les nouvelles compétences pour que le CPU les prenne en compte.

Puis Mengqian s'est occupée de l'amélioration de l'interface, et elle a défini une fonction pour créer des obstacles générés aléatoirement sur la carte. De plus, elle a ajouté une nouvelle carte.

Rayane a ajouté un bouton “choix aléatoire” dans la sélection des personnages ainsi qu'un panneau qui, lorsque la souris passe sur une compétence, affiche un descriptif de la compétence ainsi que les dégâts de celle-ci.

À noter qu'avant, durant et après chacune de ces fonctionnalités, un travail d'équilibrage des personnages constant est effectué afin d'essayer de rendre le jeu le plus équilibré possible pour que chaque classe soit aussi utile, voire aussi essentielle que les autres.

4.4 Points techniques généraux

Unity 3D est un ensemble de composants logiciels qui centralisent la gestion des images, des sons et de la physique, le tout permettant de réaliser des jeux compatibles avec la plupart des appareils.

Dans le cadre de notre projet, nous avons eu à utiliser les scripts pour gérer et contrôler les interactions des personnages entre les différents GameObjects du jeu. Et pour pouvoir améliorer l'expérience utilisateur, nous avons eu à améliorer le design du jeu en utilisant les assets Store.

5 Résultats

5.1 Fusion du code de l'an dernier

En tant que projet proposé il y a huit ans, des groupes ont travaillé chaque année pour l'améliorer. Cette année, notre première tâche était de fusionner les réalisations des deux groupes de l'année précédente. Jusqu'à l'année dernière, ce jeu avait un total de 6 personnages. Les deux groupes de l'an dernier ont été invités à ajouter un nouveau personnage, Q1 a implémenté le druide et Q2 a implémenté la valkyrie. Ainsi, nous devions les fusionner.

Notre premier défi a été de comprendre le code. Nous avons passé du temps à comprendre le code des deux groupes. Finalement, nous avons décidé de prendre le projet de Q1 comme base et d'y ajouter le contenu de Q2. Actuellement, ce jeu a un total de 8 personnages.

De plus, nous avons fusionné les scripts CPU du Druide et de la Valkyrie.

5.2 Nouvelles compétences des personnages

La création de nouvelles compétences des personnages était une fonctionnalité obligatoire de notre projet. Nous avons voulu respecter le rôle des personnages et leur fournir une deuxième compétence en accord avec la première compétence afin de renforcer ce rôle.

Guerrier :

Le guerrier est un personnage possédant beaucoup de points de vie. De ce fait, nous avons choisi de lui implémenter une deuxième compétence telle que lorsque le guerrier l'active, il gagne un point d'action en plus et inflige 2 points de dégâts d'attaque. Pour des raisons d'équilibrage, cette compétence nécessite d'infliger 6 points de dégâts supplémentaires comparés aux autres secondes compétences des personnages pour être réutilisé.



Figure 3 : Compétence 2 du guerrier

Les cases bleue représentent la portée de l'attaque

Les cases orange représentent la zone ciblée de l'attaque

Voleur :

Le voleur est la classe “perturbatrice”, c'est le personnage qui est censé semer le trouble dans la formation adverse, ce personnage se distingue grâce notamment à sa grande mobilité.

Pour être en accord avec ce rôle, nous avons choisi de lui implémenter une deuxième compétence telle que lorsque le voleur l'active, il gagne 2 points de vie en plus et inflige 2 points de dégâts d'attaque ce qui pourrait lui permettre de survivre d'un assaut et renforce le côté perturbateur. La portée de cette attaque est de 2.



Figure 4 : Compétence 2 du voleur

Archer :

L'archer est la classe avec la plus grosse portée du jeu, grâce à son arc il pourra lancer un déluge de flèches qui inflige un point de dégâts à chaque ennemi se trouvant dans la zone. Il attaquera à une distance de 12 et la largeur de la zone ciblée est de 2.



Figure 5 : Compétence 2 de l'archer

Les cases bleue représentent la portée de l'attaque

Les cases orange représentent la zone ciblée de l'attaque

Mage :

Nous avons donné au mage une toute nouvelle compétence qui agit tour par tour. En effet, la seconde compétence du mage permet d'empoisonner un personnage adverse en infligeant un point de dégâts chaque tour jusqu'à un maximum de 3 tours. La portée de cette attaque est de 6.

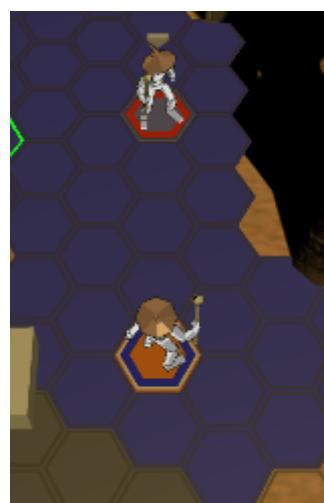


Figure 6 : Compétence 2 du mage

Les cases bleue représentent la portée de l'attaque

Les cases orange représentent la zone ciblée de l'attaque

Soigneur :

Le soigneur possède la capacité de faire regagner à ses alliés des points de vie en zone et un faible nombre de points de vie à une cible unique. Nous avons estimé qu'il n'était pas assez influent sur le jeu c'est pour cela qu'on a donné au soigneur la capacité de redonner au maximum 9 points de vie à un de ses alliés ce qui permet au soigneur de changer le cours d'une partie en choisissant avec minutie l'allié à soigné. La portée de cette attaque est de 3.



Figure 7 : Compétence 2 du soigneur

Les cases bleue représentent la portée de l'attaque

Les cases orange représentent la zone ciblée de l'attaque

Envouteur :

L'envouteur est le personnage qui donne des bonus aux alliés, il possède actuellement la capacité d'augmenter les dégâts ainsi que les points d'actions de ses alliés. Il était donc logique que nous décidions de permettre à l'envouteur de faire gagner un point de mouvement supplémentaire à son équipe. La portée de cette attaque est de 4 et la zone ciblée est de 2.



Figure 8 : Compétence 2 de l'envouteur

Les cases bleue représentent la portée de l'attaque

Les cases orange représentent la zone ciblée de l'attaque

Valkyrie :

La valkyrie se distingue par sa plus grande capacité de déplacement (un point de mouvement de plus que les autres classes). Sa lance lui permet d'effectuer des attaques de zone, tout autour d'elle. Nous lui avons donc permis grâce à sa lance et son épée de se focaliser sur un ennemi et lui infliger quatre points de dégâts d'attaque.



Figure 9 : Compétence 2 de la valkyrie

Les cases bleue représentent la portée de l'attaque

Les cases orange représentent la zone ciblée de l'attaque

Druide :

Le druide est le personnage qui permet de contrôler les ennemis. Il possède actuellement la capacité d'étourdir l'ennemi et donc de lui bloquer son prochain tour. Pour respecter l'axe contrôle de foule du personnage, nous avons choisi de lui donner la capacité d'immobiliser un ennemi en mettant ses points de mouvements à zéro. L'ennemi touché sera donc dans l'incapacité de bouger pour le prochain tour. La portée de cette attaque est de 3.

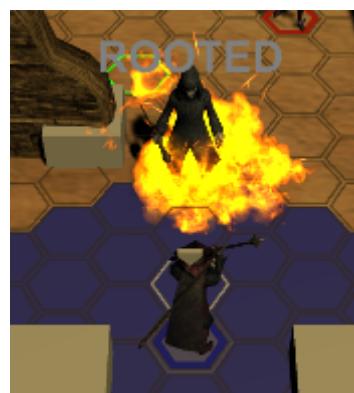


Figure 10 : Compétence 2 du druide

Les cases bleue représentent la portée de l'attaque

Les cases orange représentent la zone ciblée de l'attaque

5.3 Amélioration des scripts CPU

L'amélioration des scripts CPU est une des fonctionnalités obligatoires. Elle consistait à optimiser le comportement du joueur CPU lors des parties. Les scripts contenaient une bonne base de réactions offensives, défensives et en cohésion. Nous avons donc rajouté les scripts pour le druide et la valkyrie des projets précédents.

Avec l'ajout des nouvelles compétences, nous avons apporté aux scripts CPU les éléments suivants :

- Les personnages CPU peuvent désormais prioriser l'utilisation de leur seconde compétence. Celle-ci peut être réutilisée lorsque le personnage inflige 16 points de dégâts et 22 pour le guerrier après activation de la compétence.
- Le guerrier va se focaliser sur le druide car comme ce dernier peut le contrôler il représente une grande menace pour le guerrier.
- Réciproquement, le druide se focalise sur le guerrier si cela est possible.
- Le mage priorisera les attaques sur le guerrier si celui-ci est à portée d'attaque car désormais le guerrier et le voleur.

Le rôle du Druide est donc de contrôler le Guerrier et inversement, s'il en est possible, cela contribue à réduire l'impact dans le jeu du personnage et donc donnera un avantage considérable à son équipe.

Le guerrier étant une des classes les plus dangereuses au corps à corps du jeu, le mage va donc essayer de réduire son impact en se concentrant sur le guerrier si ce dernier vient à être à portée d'attaque.

Les nouvelles compétences implémentées ainsi que l'ajout de priorité de cible pour les attaques du druide, du guerrier et du mage rendent le joueur CPU plus agressif, plus compétitif et plus difficile à battre.

5.4 Amélioration de l'interface

Nous avons trouvé intéressant d'ajouter un descriptif des compétences lorsque la souris passe sur cette dernière. Cela permet de différencier les deux compétences principales et facilite l'expérience du joueur car avant cet ajout, le joueur devait connaître les compétences des personnages au préalable ce qui aurait complexifié la tâche avec l'ajout de la deuxième compétence.



Figure 11 : Panel d'aide pour les compétences

Toujours dans la quête d'améliorer l'expérience du joueur, il est maintenant possible grâce à l'ajout de ce bouton de choisir son équipe et celle de l'adversaire aléatoirement, ce qui peut rajouter un côté amusant pour les joueurs voulant se fixer des défis.

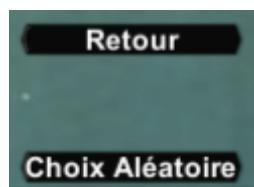


Figure 12 : Bouton choix aléatoire

De plus, nous avons embellie l'interface en utilisant les ressources que nous avons téléchargées sur internet.

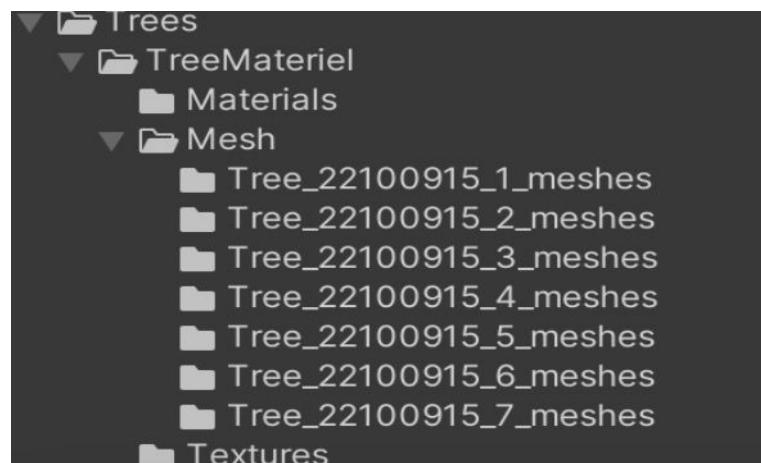


Figure 13 : Ressources de l'arbre

Nous avons ajouté une carte, ainsi il y a au total 3 cartes qui peuvent être choisies :

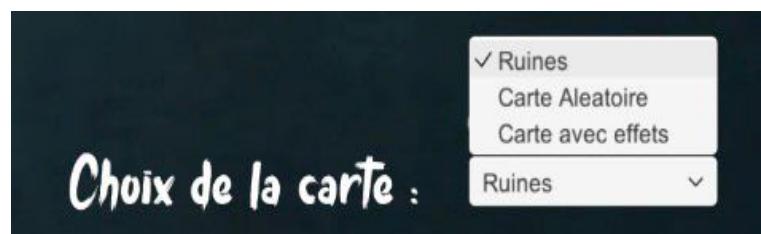


Figure 14 : Choix de la carte

Pour la première carte (Ruines), il y a des murs comme obstacles, le drapeau est présent aléatoirement dans la carte.



Figure 15 : La carte "Ruines"

Pour la deuxième carte (Carte aléatoire), il n'y a pas de mur, mais des hexagones peuvent être générés aléatoirement en tant qu'obstacles, et la carte générée à chaque mise à jour du jeu est différente.

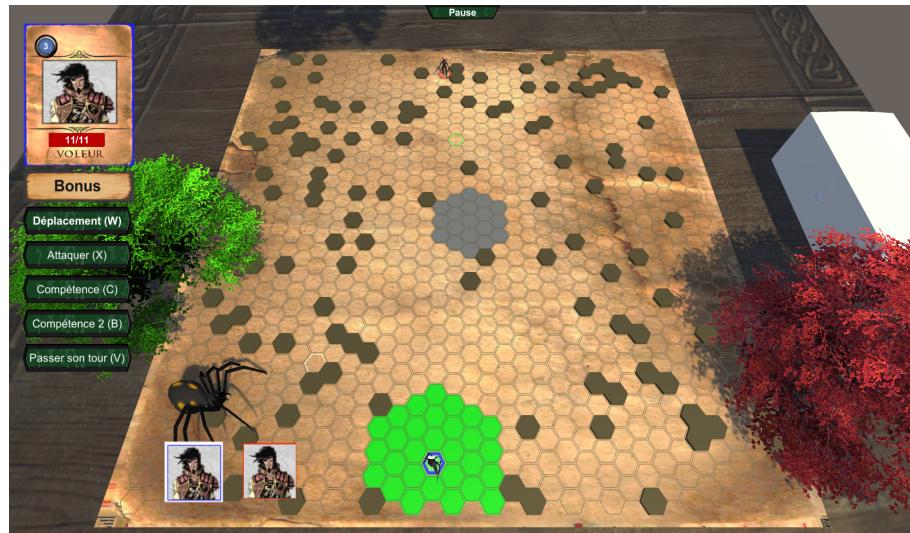


Figure 16 : La carte aléatoire

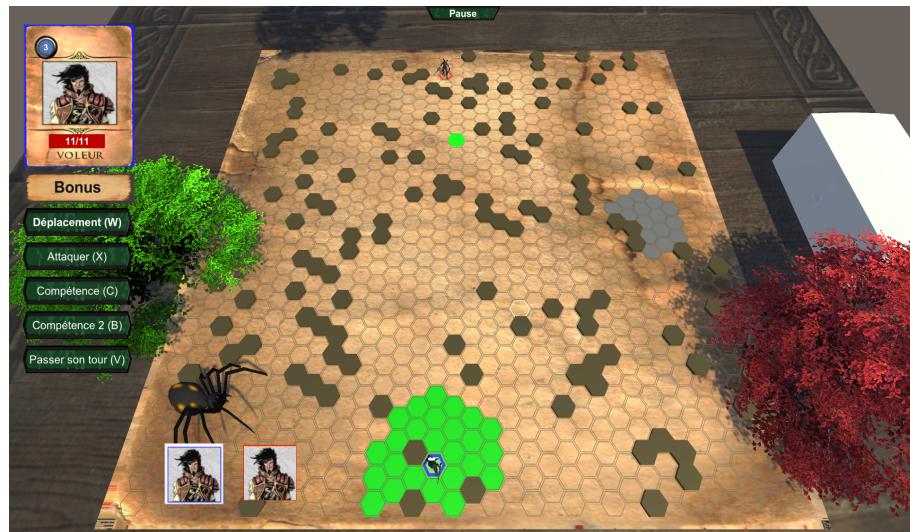


Figure 17 : Redémarrage pour la deuxième carte

Pour la troisième carte, elle est relativement simple car il n'y a pas d'obstacles, mais afin de rendre la carte plus riche, nous avons ajouté quelques effets spéciaux, tels que le feu explosif, le méthane, le brouillard et d'autres phénomènes.



Figure 18 : La carte avec effets

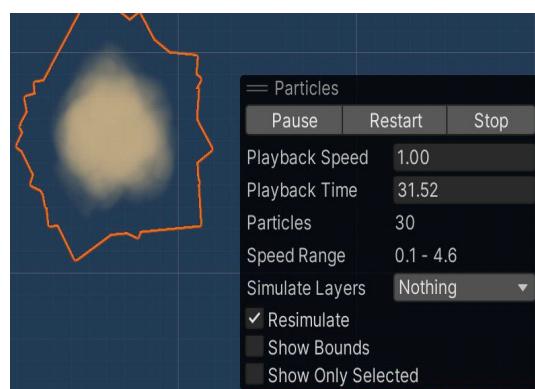
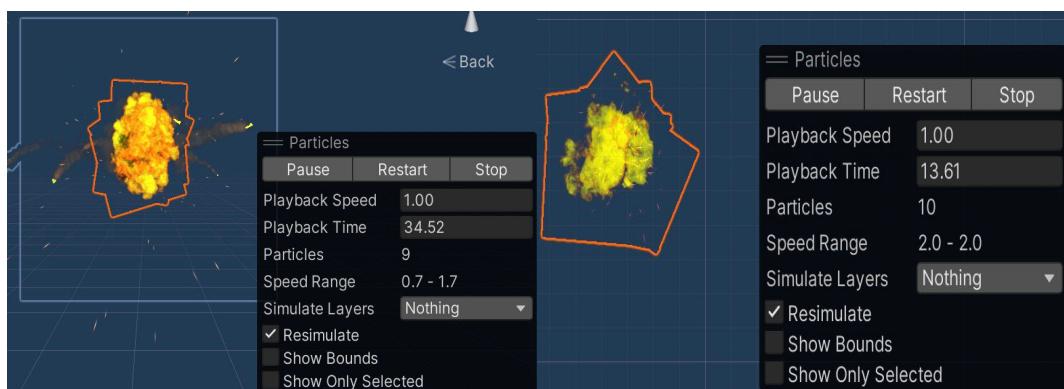


Figure 20 : Exemple d'effet spécial

5.5 Autres améliorations

Une autre fonctionnalité que nous avons ajoutée à ce projet est une interface qui permet de faire tourner le jeu en mode ordinateur contre ordinateur plusieurs fois et fournit le taux de victoire de chaque personnage, ce qui nous permet d'ajuster l'équilibre du jeu en conséquence. Ce test est uniquement basé sur le niveau du script.

Il implique que le jeu peut ne pas être bien équilibré si les personnages sont contrôlés par une personne. Cependant, il peut encore révéler certains problèmes d'équilibre du jeu, ce qui nous permet d'ajuster l'équilibre du jeu en conséquence. Étant donné que cette fonctionnalité n'est utile que pour le développement et l'équilibrage des personnages, nous ne pouvons voir cette interface que dans l'environnement de développement ; en d'autres termes, elle est invisible pour l'utilisateur.



The screenshot shows a table of statistics from a game. At the top, there are two buttons: "Pause" on the left and "continuer" on the right. The table has three columns: "Charactor" (Character), "Nb de gagner/jouer" (Number of wins/played), and "Probabilité de gagner" (Win probability). The data is as follows:

Charactor	Nb de gagner/jouer	Probabilité de gagner
Guerrier	461/925	49.8%
Voleur	502/948	53.0%
Archer	447/874	51.1%
Mage	424/884	48.0%
Soigneur	454/885	51.3%
Envouteur	464/939	49.4%
Valkyrie	440/881	49.9%
Druide	423/894	47.3%

Figure 21 : Exemple de test de l'interface

Nous avons trouvé intéressant d'avoir un résumé des statistiques globales à la fin d'une partie, c'est pourquoi nous avons ajouté la possibilité d'avoir un résumé des dégâts par équipes à la fin de la partie, en effet cela permet de voir le nombre de dégâts que chaque équipe a infligé ainsi que le pourcentage de dégâts infligés par chaque personnage.



Figure 22 : Interface afficher statistique

6 Défis à relever

6.1 Les différentes réorganisations

La majorité du temps, notre groupe travaillait en coordination à distance. Étant donné que nos emplois du temps ne se coordonnaient pas et que certains d'entre nous sont salariés, il a été très difficile de s'harmoniser dans un premier temps, mais nous avons su pallier ce problème trouver une manière plus régulière de travailler.

La réalisation de la documentation en groupe a été assez compliquée à gérer. Nous avons dû rechercher les meilleurs outils pour travailler ensemble et de manière synchrone. Comme il y a trois personnes dans notre groupe qui n'ont pas la langue française comme langue maternelle, il y a souvent des problèmes d'orthographe.

Enfin, le véritable défi de ce projet fut la compréhension du code déjà existant. Il a donc été primordial de passer du temps sur la compréhension du code pour ne pas réaliser des absurdités, et nous avons en plus de cela pris soin d'ajouter des commentaires aux parties que nous avons ajoutées, afin de simplifier la compréhension pour les prochaines années.

6.2 Difficultés techniques

Afin de mener à bien ce projet, nous avons dû rapidement prendre en main Unity et le langage C# pour comprendre le jeu et ses scripts. Pour ce faire, nous avons effectué des tutoriaux sur le site dédié Unity Learn et nous nous sommes documentés sur internet. Nous avons continué ce travail durant les premières semaines du projet, en parallèle de la rédaction des documents demandés afin d'être dans les meilleures conditions pour coder.

L'une des principales difficultés était de comprendre un projet déjà commencé auparavant par plusieurs groupes différents. Dans un premier temps, la compréhension du code et des scripts liés au jeu nous ont pris la majeure partie de notre temps ce qui est tout à fait normal lors d'une reprise d'un projet déjà commencé mais nous pensons que nous avons passé un peu trop

de temps sur la compréhension ce que nous expliquons peut-être par un manque d'expérience avec Unity et le langage de programmation C#.

Nous avons également mis l'accent sur le fait de nommer les variables et fonctions en respectant les règles de codage, nous avons également commenté le plus clairement possible les parties que nous avons codées car nous pensons que cela est un gain de temps et facilitera la reprise du projet pour les étudiants de l'année prochaine.

7 Conclusion

En conclusion, nous avons dans l'ensemble réussi à atteindre nos objectifs malgré les problèmes rencontrés et nous sommes satisfaits dans l'ensemble du travail fourni tout au long de ce semestre.

Effectivement, nous avons fusionné les deux projets précédents, ajouté les deux personnages implémentés par ces derniers et créé des compétences pour chacun des personnages existants. L'ajout de nouvelles compétences diversifie grandement la manière de jouer ce qui accroît l'aspect tactique du jeu. En plus de cela, l'ajout des compétences renforce le rôle donné aux personnages.

En plus d'avoir complété nos objectifs initiaux, nous sommes contents d'avoir pu ajouter une nouvelle carte, un descriptif en jeu pour les compétences ainsi qu'un bouton qui permet de choisir ses personnages aléatoirement.

En outre, ce projet nous a également permis de nous familiariser avec Unity et le C#, que nous n'avions jamais étudié auparavant, cela a renforcé nos compétences individuelles et notre expérience. Nous sommes donc satisfaits des aspects techniques de notre projet, tant par leur apprentissage que leur réalisation.

Enfin, le jeu dans l'état actuel peut encore être amélioré de plusieurs manières. En effet, il serait intéressant d'ajouter une troisième compétence pour chaque personnage cela pourrait apporter une nouvelle dimension à l'expérience de jeu et permettre aux joueurs de personnaliser encore davantage leur approche de chaque situation. Également, nous avions pensé au fait d'implémenter un nouveau mode de jeu, nous avons imaginé un mode dans lequel chaque personnage verrait ses compétences changer aléatoirement parmi les compétences des autres personnages.

8 Références

- Rapport de projet du groupe L3Q2 de l'année dernière
- Rapport de projet du groupe L3Q1 de l'année dernière
- Lien vers le site de projet

9 Glossaire

Forge : Logiciel en ligne de collaboration entre personnes d'un groupe pour le partage de documents tel que du code ou une documentation de projet. Ce groupe-ci utilise la forge Redmine fournie par l'université Paris Cité.

IA : Intelligence Artificielle, se dit du code d'une machine qui sert à simuler l'intelligence humaine. Cela inclut généralement la capacité de se baser sur des expériences passées.

Moteur de jeu : Environnement logiciel dédié à la création de jeux vidéo. Un moteur de jeu, souvent appelé moteur graphique, simplifie le prototypage et le développement de jeux en implémentant un bon nombre de fonctionnalités nécessaires au fonctionnement d'un jeu, notamment graphiques, mais également logique selon le moteur, en temps réel. Unity est un moteur de jeu.

CPU : Acronyme anglais de Central Processing Unit, signifiant Unité Centrale de Traitement, et faisant référence à un processeur. Dans un jeu vidéo, ce terme est souvent utilisé comme nom de joueur pour désigner un joueur contrôlé par une IA.

Script : Suite d'instructions que l'ordinateur doit exécuter. Peut inclure des boucles d'instructions, ou instructions à exécuter sous certaines conditions.

C# : Langage de programmation orienté objet (paradigme de programmation consistant en la représentation de concepts au travers d'objets). Les scripts sous Unity sont en C#.