



Rapport de projet

L3L1 : Unity 3D - Jeu 3D

Encadrant : Claude LATTAUD

Julien D'ABOVILLE

Timothé MIEL

Simon SEPIOL- DUCHEMIN

2023 - 2024



Référence du projet	L3L1
Intitulé	Unity 3D - Jeu 3D
Date de rédaction du document	3/04
Auteurs	Julien D'ABOVILLE Timothé MIEL Simon SEPIOL-DUCHEMIN
Confidentialité	UFR Math-Info de l'Université Paris-Cité
Date de soumission	3/04



Sommaire

1. Préface.....	5
2. Introduction.....	6
2.1. Contexte.....	6
2.2. Objectifs.....	7
3. Ressources.....	8
4. Outils de développement.....	8
5. Organisation et répartition.....	8
6. Début du projet.....	9
6.1. Reprise du projet.....	9
6.2. Apprentissage Unity et C#.....	9
7. Phase de développement.....	10
7.1. Ajouts.....	10
7.1.1. Nouveaux personnages.....	10
Ajout de la classe de personnage : Bûcheron.....	10
Caractéristiques du Bûcheron.....	11
Ajout de la classe de personnage : Netherfang.....	12
Caractéristiques du Netherfang.....	13
Ajout de la classe de personnage : Lifeweaver.....	14
Caractéristiques du Lifeweaver.....	15
7.1.2. Nouvelles capacités actives.....	16
7.1.3. Nouvelles capacités passives.....	19
7.1.4. Nouvelles cartes.....	20
Arène.....	20
Forêt.....	22
7.1.5. Nouvelle gestion des sauvegardes.....	22
Contexte.....	22
Création.....	23
Bénéfices.....	24
7.1.6. Affichage des compétences dans le menu.....	25
Contexte.....	25
Création.....	25



7.2. Modifications.....	26
7.2.1. Clarification du code.....	26
Détection des clics.....	26
Refonte du script de la caméra.....	26
7.2.2. Résolution de bug.....	26
Hexagones empoisonnés.....	26
Chargement de la carte d'une sauvegarde.....	27
Placement des personnages lors d'une partie chargée.....	27
7.2.3. Amélioration des fonctionnalités existantes.....	27
7.2.4. Gestion des capacités par cooldown.....	28
Contexte.....	28
Création.....	28
Bénéfices.....	28
7.2.5. Équilibrage.....	28
7.2.6. Amélioration des scripts CPU.....	28
Ajout des nouvelles compétences dans les Scripts CPU existants.....	28
Création des Scripts CPU pour les nouveaux personnages.....	29
8. Conclusion.....	30

1. Préface

Dans le cadre de notre 3e et dernière année de licence informatique et applications au sein de l'Université Paris-Cité, il nous est proposé de travailler en groupe sur un projet qui s'étendra sur une période de 4 mois. Ce projet vise à mettre en pratique nos connaissances et capacités face à des problématiques réelles. De plus, ce projet offre l'opportunité de découvrir de nouvelles notions et outils informatiques.

Ce document constitue le rapport final de notre projet intitulé : "L3L1 : Unity 3D - Jeu 3D". Il expose en détail l'ensemble des étapes nécessaires à la réalisation du projet. Il retrace ainsi son déroulement à travers les réflexions émises par le groupe. Notre objectif est de vous présenter un aperçu clair et complet de notre cheminement de pensée, permettant ainsi une pleine compréhension du projet et de son développement.

Dans un premier temps, nous détaillons le contexte dans lequel s'inscrit le projet, nous aborderons ensuite ses problématiques et objectifs avant de parler de sa conception et de son développement. Nous ferons également le point sur les difficultés rencontrées et l'organisation du groupe.

2. Introduction

2.1. Contexte

Pause Café est un jeu vidéo dont le développement a commencé en 2014, toujours au sein de l'Université Paris-Cité sous l'encadrement de M. Claude LATTAUD.

Ce jeu est un jeu vidéo 3D de tactique au tour par tour où deux joueurs s'affrontent. Les joueurs, pouvant être humains et/ou ordinateurs, vont contrôler de 1 à 5 personnages répartis sur le terrain de jeu : une grille hexagonale de 34 par 30 cases. Le but étant de réduire les points de vies de tous les personnages de son adversaire à zéro.

Il y a huit personnages disponibles : le Druide, le Mage, l'Archer, le Voleur, l'Envoûteur, le Soigneur, le Guerrier et la Valkyrie.

Chaque personnage possède ses propres valeurs de statistiques : Points de Vie (PV), Points d'Attaque (PA), Points de Mouvement (PM), Priorité, une attaque basique et deux compétences spéciales.

Chaque attaque et compétence possède également ses propres caractéristiques : Portée, Rayon de l'Effet, si elle vise les ennemis et/ou les alliés et/ou l'utilisateur de la compétence (pour un soin par exemple), l'Effet de la compétence tels que des dégâts, du soin, de l'étourdissement et bien d'autres, la Valeur de cet Effet, le Temps entre chaque utilisation de la compétence (référencé par la suite sous le nom de *cooldown*) et son nom.

Les terrains sont les espaces de jeu où les joueurs vont contrôler leurs personnages et les faire s'affronter. Une zone bonus s'y trouve où si un personnage s'y trouve, un PV supplémentaire lui est conféré. Ce bonus se cumule pour plusieurs personnages mais s'annule si un personnage de chaque équipe s'y trouve.

Deux terrains (ou maps) sont disponibles : Ruines, un terrain prédéfini possédant des murs et des boîtes servant d'obstacles. Aléatoire, un terrain généré aléatoirement à chaque partie où certains hexagones deviennent des murs.

Différentes interfaces sont disponibles au sein du jeu. Dans la première, celle du menu principal, nous pouvons afficher un guide du jeu, les crédits ou passer soit au menu de gestion des sauvegardes soit au menu de sélection des personnages, du mode de jeu et du terrain. Une fois le jeu lancé des interfaces affichant les actions possibles, les personnages sur le terrain et les statistiques du personnage contrôlé et pointé sont constamment affichées. Un menu d'aide est affichable, tout comme un menu pause depuis lequel on peut sauvegarder la partie, charger une sauvegarde ou quitter la partie. Le menu de gestion des sauvegardes permet de sélectionner à quelle sauvegarde jouer.

2.2. Objectifs

Notre objectif global était de continuer le travail effectué par les groupes précédents tout en améliorant le jeu. Pour cela, nous avions comme objectifs principaux :

- Introduire une nouvelle classe de personnage dotée de nouvelles compétences et création de son script CPU.
- Ajouter une nouvelle compétence pour chacun des personnages existants.
- Réviser et améliorer les scripts d'IA pour intégrer ces nouvelles compétences.
- Améliorer l'équilibrage des classes.
- Identifier et corriger les mauvaises pratiques, les bugs et les incohérences dans le jeu.

En plus de ces objectifs principaux, nous avions une certaine liberté quant à l'ajout d'autres fonctionnalités. Comme vous le verrez par la suite, nous en avons profité pour ajouter de nombreuses nouvelles fonctionnalités au jeu, dans le but de le rendre le plus complet possible.

3. Ressources

Dans le cadre du développement de notre projet, nous avons obtenu les modèles 3D et les animations des personnages *Netherfang* et *Lifeweaver* grâce à la plateforme Mixamo.

- [Mixamo - Netherfang et Lifeweaver](#)

Pour le personnage du *Bûcheron*, nous avons opté pour un modèle 3D et des animations disponibles sur l'Asset Store d'Unity

- [Unity Asset Store - Bucheron](#)

Concernant les éléments décoratifs destinés à enrichir les nouvelles cartes ainsi que les effets spéciaux, notre choix s'est porté sur l'Unity Asset Store, mettant en avant notre préférence pour cette riche bibliothèque. Les effets sonores, apportant une dimension supplémentaire à l'expérience de jeu, proviennent de divers sites internet offrant des ressources audio libres de droit.

- [Unity Asset Store - Effets](#)

4. Outils de développement

Notre projet a été développé en utilisant Unity version 2023.2.6f1, permettant la création et l'intégration sophistiquée des éléments de jeu. Les scripts utilisés sous Unity sont codés en C#. Le code a été élaboré dans l'environnement Visual Studio. Pour ce qui est de la conception visuelle, les cartes des personnages ont été créées à l'aide de l'outil Gimp . Pour la gestion de version, nous avons utilisé le serveur SVN mis à disposition par l'UFR.

5. Organisation et répartition

En prenant en compte le faible nombre de membres du groupe et pour que chacun puisse appréhender l'ensemble des aspects du projet, il n'y a pas de décomposition en sous équipes. Un diagramme de Gantt a été réalisé au début du projet, pour donner les grandes lignes du développement. Ce dernier prend en compte Abderraouf BEN-HADJ, qui fût déclaré défaillant au cours de la semaine n°7. Nous avons dû réajuster les tâches assignées afin de compenser son désistement du groupe.



Les réunions hebdomadaires ont permis de parler avec l'encadrant des fonctionnalités pouvant être intéressantes à développer qui n'avaient pas été prévus dans le diagramme ou des éventuelles fonctionnalités déjà présentes qui se sont avérées non fonctionnelles ou non utiles et donc de les refaire.

6. Début du projet

6.1. Reprise du projet

Pause Café est donc un jeu en développement depuis 2014, cela implique donc deux contraintes importantes.

Premièrement, il y a eu 7 différentes équipes sur ce projet au fil des années et 25 développeurs ayant travaillé sur le projet, soit sur un aspect précis du projet (comme les groupes L3AL1 et L3AL2 de 2019-2020 et 2020-2021 respectivement) ou sur l'ensemble du projet (comme le groupe L3C1 de l'année dernière).

Cela cause une hétérogénéité notable du code où l'on va trouver des méthodes expliquées, d'autres non, de la redondance, des commentaires parfois présents ou non, clairs ou non. Reprendre un projet d'une telle ampleur implique donc de passer une partie considérable du projet à comprendre les différents aspects du code avant de pouvoir se mettre à coder.

Deuxièmement, certaines technologies et méthodes utilisées dans le code sont obsolètes, principalement car Unity est une plateforme de développement en constante évolution avec 1 mise à jour par trimestre et ce, chaque année depuis ses débuts. Nous avons donc eu grand intérêt à remplacer ces sections obsolètes avec les nouvelles fonctionnalités disponibles afin de rendre le jeu plus performant.

6.2. Apprentissage Unity et C#

Unity est une plateforme de développement nouvelle pour chaque membre du groupe, nous devions avant tout nous y familiariser afin de pouvoir travailler sur ce projet : Scènes, Scripts, Composants, *GameObjects*, Animations et bien d'autres, il n'y a pas que du code pour réaliser ce projet mais également beaucoup de manipulation d'objets dans l'éditeur d'Unity.

Les scripts d'Unity sont donc codés en C#, un langage de programmation orienté objet dérivant de C et C++ ressemblant à Java. Cette ressemblance nous a permis de rapidement comprendre les méthodes inhérentes à C#, certaines méthodes sont également spécifiques à Unity (comme `Update()`, méthode appelée une fois par *frame*).

Pour nous initier à Unity et à C#, nous avons pu réaliser les tutoriels d'Unity 'Junior Programmer' disponibles sur la plateforme en ligne *Unity Learn* donnant des leçons utiles pour démarrer un projet sous Unity.

7. Phase de développement

7.1. Problèmes rencontrés

7.1.1. Gestion de version

Le jeu étant trop lourd pour être envoyé dans sa totalité sur la forge en une fois, nous avons dû choisir quels fichiers envoyer. Cela a créé des problèmes de synchronisation entre les membres car nous n'avions pas suivi tous les fichiers nécessaires sur svn au début du projet. Ce problème a pu être corrigé par la suite en ajoutant les fichiers .meta au svn.

7.1.2. Compréhension du code existant

Au début du projet, il a été compliqué de comprendre le code fourni. En effet, la majorité du déroulement d'une partie est gérée dans le fichier MainGame.cs qui fait plus de 3000 lignes. Le nombre conséquent de groupes ayant participé à la création du jeu fait qu'il reste un grand nombre fichiers non utilisés, ce qui freine la compréhension du code. Le code est assez mal documenté, beaucoup de fonctions ne sont pas commentées.

7.2. Ajouts

7.2.1. Nouveaux personnages

Ajout de la classe de personnage : Bûcheron

La création d'une nouvelle classe était une fonctionnalité obligatoire de notre projet. Nous avons donc choisi d'ajouter un personnage : le Bûcheron. Cette classe a été pensée pour offrir un équilibre entre attaque et défense, permettant aux joueurs d'explorer de nouvelles stratégies de combat.

Pour la modélisation de ce personnage sur la carte, nous avons utilisé l'outil DALL·E.



Figure 1 : Carte du bûcheron

Caractéristiques du Bûcheron

Le Bûcheron possède un total de 35 PV, une attaque de base infligeant 1 points de dégât, il se distingue par sa polyvalence sur le champ de bataille, grâce à ses compétences uniques :

- **Howl** : Cette capacité augmente les dégâts infligés par le Bûcheron ainsi que ses points de vie de 1 point avec un cooldown de 8, offrant un avantage significatif lors des affrontements critiques.
- **Hammer Explosion** : Attaque de zone dévastatrice, elle permet au Bûcheron d'infliger 3 points de dégâts aux ennemis à une portée de 2 hexagones avec une zone de dégâts de 3 hexagones de rayon, idéale pour attaquer un groupe d'ennemis, cooldown de 12.
- **Hammer Launch** : Cette attaque à distance donne au Bûcheron la capacité d'atteindre un ennemi situé jusqu'à 6 hexagones en infligeant 2 points de dégâts avec la possibilité d'étourdir l'adversaire 1 fois sur 4, le tout avec un cooldown de 4, ajoutant une couche stratégique à l'approche du combat.

Le modèle 3D et ses animations sont libres de droits et trouvés sur l'Asset Store.



Figure 2 : Modèle 3D du bûcheron ainsi que sa zone de déplacement

Ajout de la classe de personnage : Netherfang

La création d'une deuxième nouvelle classe était une fonctionnalité optionnelle de notre projet. Nous avons pris la décision d'intégrer un nouveau personnage : le Netherfang. Cet ajout est une initiative visant à diversifier l'expérience et enrichir l'univers de jeu.

La conception du Netherfang a été méticuleusement réfléchie pour introduire une classe axée sur la résilience, avec une quantité accrue en PV au prix de dégâts réduits par rapport aux autres classes. Cette approche distinctive permet d'insuffler une nouvelle dynamique au jeu, encourageant les joueurs à adopter et expérimenter de nouvelles stratégies de combat.

Pour la modélisation de ce personnage sur la carte, nous avons utilisé l'outil DALL·E.



Figure 3 : Carte du Netherfang

Caractéristiques du Netherfang

Le Netherfang débute avec un total de 45 PV, une attaque de base infligeant 1 point de dégât. Il se distingue par sa capacité à être résistant sur le champ de bataille, grâce à ses compétences :

- **Howl** : Cette capacité booste les dégâts infligés par le Netherfang ainsi que ses points de vie de 1 point avec un cooldown de 8, offrant un avantage significatif lors des affrontements critiques.
- **Freeze** : Attaque permettant de geler son adversaire, ce dernier ne pouvant plus bouger pendant 1 tour, avec un cooldown de 4.
- **Jump Explosion** : Cette attaque le propulse dans les airs et va engendrer une explosion large de 3 hexagones à l'atterrissement qui inflige 5 points de dégâts à tout ennemi dans cette zone, cooldown de 12.

Le modèle 3D et ses animations sont libres de droits et trouvés sur Mixamo.



Figure 4 : Modèle 3D du Netherfang ainsi que sa zone de déplacement

Ajout de la classe de personnage : Lifeweaver

Bien que l'ajout d'une troisième nouvelle classe ne fût initialement pas prévu dans notre projet, la présence limitée d'un seul soigneur parmi les personnages disponibles a mis en lumière la nécessité d'élargir cette fonctionnalité essentielle. C'est dans cet esprit que nous avons décidé d'intégrer un nouveau personnage : le Lifeweaver.

La conception du Lifeweaver a été effectuée avec une attention particulière, donnant naissance à une classe capable non seulement de soigner ses alliés mais aussi de se défendre efficacement contre les adversaires. Cette dualité de rôle ouvre de nouvelles avenues stratégiques pour les équipes, permettant une flexibilité et une adaptabilité accrues pendant les combats.

Un des atouts majeurs du Lifeweaver est sa capacité unique de téléportation, lui permettant de se déplacer instantanément vers un allié pour le soigner ou se retirer stratégiquement du combat. Cette compétence renforce son rôle de support vital tout en offrant des options tactiques pour naviguer dans le champ de bataille avec une mobilité inégalée.

Pour la modélisation de ce personnage sur la carte, nous avons directement utilisé le modèle 3D disponible sur Mixamo.



Figure 5 : Carte du Lifeweaver

Caractéristiques du Lifeweaver

Le Lifeweaver commence la partie avec un total de 25 PV et une capacité de soin de base qui restaure 3 PV à l'un de ses alliés. Le Lifeweaver se distingue par sa capacité à être plus mobile ainsi que la possibilité d'infliger des dégâts en plus d'en restaurer. Ses compétences sont les suivantes :

- **Damage** : Cette capacité inflige 3 points de dégâts à un adversaire situé jusqu'à 6 hexagones de distance avec un cooldown de 4.
- **Absorb**: Cette compétence procure de l'absorption à un allié : lui procurant instantanément 3 PV supplémentaires, suivis de 3 autres PV après 2 tours, elle possède un cooldown de 8.
- **Téléportation** : Compétence qui permet de se téléporter instantanément à une portée de 7 hexagones avec un cooldown de 12.

Le modèle 3D et ses animations sont libres de droits et trouvés sur Mixamo.



Figure 6 : Modèle 3D du Lifeweaver ainsi que sa zone de déplacement

Ajout de la classe de personnage : Personnage Custom

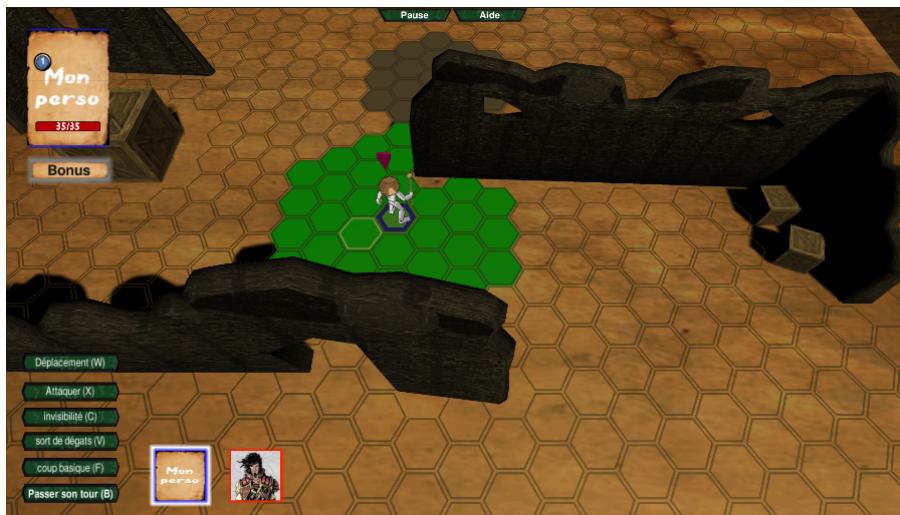
Dans les jeux nintendo, on retrouve parfois parmi les personnages propre au jeu, notre propre personnage appelé un mii (“Mii — Wikipédia”). La création du personnage custom s’inspire de ce principe. Le but n’est pas d’avoir un personnage équilibré en continuité avec ceux déjà ajoutés, mais d’avoir un personnage qui peut être ce qu’on veut.

Une page "Mon Perso", accessible depuis le menu principale, permet de choisir les pv, pa, pm et priorité du personnage custom dans un premier temps. Ensuite il est possible de choisir ses attaques parmi celles des personnages existants. Pour ce faire, pour chaque attaque il y a un menu déroulant permettant de choisir de quel personnage on veut prendre une attaque, puis un deuxième menu déroulant est rempli automatiquement avec les attaques de ce personnage. Il faut ensuite cliquer sur sauvegarder pour entrer les personnages créés dans une base de données.



Le personnage est ensuite sélectionnable dans l'écran de lancement d'une partie. De part la diversité des attaques qu'il peut avoir, il n'est pas possible de faire jouer une IA avec une équipe qui comporte le personnage custom.

En jeu, le personnage custom a une apparence bien différente des autres personnages, comme pour les Mii, ce qui montre qu'il est en rupture avec le jeu normal, et est là pour tester de nouvelles choses.



Ce personnage peut aussi servir pour les futurs développeurs, s'ils souhaitent ajouter de nouvelles attaques à différents personnages, il sera possible de les tester toutes en même temps en créant leur personnage custom avec ces attaques.

7.2.2. Nouvelles capacités actives

L'équilibrage n'étant pas finalisé, les valeurs données ici sont amenées à changer.

L'ajout d'une nouvelle compétence à chaque personnage était l'un des objectifs initiaux du projet. Le but étant de complexifier la stratégie du jeu, nous avons opté pour différents types de compétences donc certaines étant totalement nouvelles au jeu. Parmi celles ajoutées, nous trouvons des compétences offensives, faisant des dégâts sur le coup et/ou sur la durée, des compétences d'améliorations de statistiques mais aussi des compétences défensives, et également des compétences plus complexes.

Les compétences des nouveaux personnages sont également listées.

Les nouvelles capacités ont été réfléchies pour être cohérentes avec les différents personnages et leur style de combat. Le style heroic-fantasy du jeu a été respecté.

Archer : Flèche enflammée

Un type d'attaque BURNING a été créé. Lorsqu'un ennemi est touché par une attaque de type BURNING, il subit sur le coup le nombre de dégâts indiqué dans le constructeur de l'attaque, puis $\frac{1}{3}$ de ces dégâts le tour suivant. Cette capacité déclenche un effet visuel de flamme sur la cible touchée.

Ce type d'attaque est implémenté chez l'Archer avec sa compétence *Flèche enflammée*, qui a une portée de 5 hexagones, une zone d'1 hexagone de rayon, inflige 3 points de dégâts sur le coup et par conséquent 1 point de dégât le tour suivant, son cooldown est de 8.

Envoûteur : Blocage des capacités

Un type d'attaque DISABLESKILL a été créé. Lorsqu'un ennemi est touché par une attaque de type DISABLESKILL, il voit ses 3 capacités être bloquées pour un nombre de tours indiqué dans le constructeur de l'attaque. Il peut toujours se déplacer et utiliser son attaque de base.

Ce type d'attaque est implémenté chez l'Envoûteur avec sa compétence *Blocage des capacités* qui a une portée d'1 hexagone, un zone d'effet d'1 hexagone de rayon, n'inflige aucun dégât et bloque les capacités pour 5 tours, avec un cooldown de 2.

Valkyrie : Tourne hache

La nouvelle capacité *Tourne Hache* de la valkyrie consiste en une attaque circulaire avec sa hache. C'est une attaque de type DAMAGE avec une portée d'1 hexagone, une zone d'effet d'1 hexagone de rayon infligeant 4 points de dégâts et possédant un cooldown d'1.

Mage : Bouclier

Un type d'attaque SHIELD a été créé. Lorsque le personnage active son bouclier, celui-ci peut résister à n'importe quelle attaque. Il y a un effet sonore lors de la désactivation de son shield.

Ce type d'attaque est implémenté chez le mage et sa compétence *Bouclier* s'active sur lui-même avec un cooldown de 8.

Druide : Tempête

Un type d'attaque STORM a été créé. Lorsque le personnage déclenche la tempête, tous les ennemis dans la zone de son attaque perdent un nombre de points de vie considérable. les ennemis touchés ont une tornade qui s'active sur eux.

Ce type d'attaque est implémenté chez le druide. Sa compétence *Tempête* a une portée de 8 hexagones avec une zone d'effet de 8 hexagones de rayon, faisant 6 dégâts sur tous les adversaires dans la zone. Il possède un cooldown de 12.

Guerrier : Rage

Un type d'attaque RAGE a été créé. Lorsque le personnage active sa rage, ses PV et ses dégâts sont boostés. La durée est fixée à 8 tours. Cette rage génère un effet visuel de rage tout au long de son activation.

Ce type d'attaque est implémenté chez le Guerrier. Sa compétence *Rage* procure 4 points supplémentaires à ses PV et à son attaque de base, le tout avec un cooldown de 10.

Soigneur et Lifeweaver : Absorption

Un type d'attaque ABSORB a été créé. Le personnage applique une absorption aux PV d'un allié : cela lui ajoute instantanément les PV indiqués dans le constructeur, puis une autre fois après 2 tours. Cette capacité déclenche un effet visuel de soin lors des 2 activations.

Ce type d'attaque est implémenté chez le Soigneur et le Lifeweaver. Leur compétence *Absorption* procure au total 6 PV d'absorption, a une portée de 6 hexagones avec un cooldown de 6.

Voleur : Invisibilité

Un type d'attaque INVISIBLE a été créé. Lorsque le personnage active son invisibilité, ses dégâts sont augmentés selon la valeur indiquée dans le constructeur et ce pour une durée de 4 tours le rendant complètement hors des radars des autres personnages durant cette période.

Ce type d'attaque est implémenté chez le voleur. Sa compétence *Invisible* s'active sur lui-même : il inflige 5 points de dégâts supplémentaires durant celle-ci et sa compétence possède un cooldown de 11.

Bûcheron et Netherfang : Hurlement

Un type d'attaque HOWL a été créé. Cette capacité unique booste les dégâts infligés par le personnage ainsi que ses points de vie d'une valeur indiquée dans le constructeur. Celle-ci déclenche un effet sonore lors de son activation ainsi qu'un effet de flammes sur le personnage.

Ce type d'attaque est implémenté chez le Netherfang et le Bûcheron. Leur compétence *Hurlement*, s'active sur eux-mêmes, leur procurant 3 points de dégâts et de vie supplémentaires avec un cooldown de 7.

Bûcheron : Hammer Explosion

Un type d'attaque HAMMER a été créé. Cette attaque de zone permet au personnage d'infliger des dégâts aux ennemis selon les valeurs indiquées dans le constructeur. Celle-ci génère un effet visuel d'explosion ainsi qu'un effet sonore.

Ce type d'attaque est implémenté chez le Bûcheron. Sa compétence *Hammer Explosion*, a une portée de 2 hexagones sur une zone de 3 hexagones, et inflige 4 points de dégâts à tous les adversaires dans la zone avec un cooldown de 8.

Bûcheron : Lance Marteau

Un type d'attaque HAMMER_LAUNCH a été créé. Cette attaque donne la possibilité d'étourdir l'adversaire 1 fois sur 4, ses autres caractéristiques sont dépendantes du constructeur. Un effet visuel apparaît sur l'ennemi touché.

Ce type d'attaque est implémenté chez le Bûcheron. Sa compétence *Lance Marteau* peut atteindre une cible jusqu'à 6 hexagones de distance, et inflige 4 points de dégâts avec un cooldown de 6.

Netherfang : Jump Explosion

Un type d'attaque JUMP_EXPLOSION a été créé. Cette attaque propulse le personnage dans les airs et crée à l'atterrissement une explosion dont les dégâts et le rayon d'effet dépendent du constructeur.

Ce type d'attaque est implémenté chez le Netherfang. Sa compétence *Jump Explosion* lui permet d'effectuer un saut d'une case, infligeant 5 points de dégâts sur un rayon de 3 hexagones avec un cooldown de 3.

Lifeweaver : Absorption d'âme

Un type d'attaque SOUL_ABSORPTION a été créé. Cette attaque fait des dégâts sur un adversaire avec un effet visuel qui apparaît sur celui-ci.

Ce type d'attaque est implémenté chez le Lifeweaver. Sa compétence *Absorption d'âme* lui permet d'infliger 3 points de dégâts sur une cible jusqu'à 5 hexagones de distance avec un cooldown de 4.

7.2.3. Nouvelles capacités passives

Nous avons décidé d'introduire des compétences passives dans le jeu pour enrichir la complexité et la diversité stratégique offerte aux joueurs.

Contrairement aux compétences actives qui requièrent une action immédiate de la part du joueur, les compétences passives se déclenchent automatiquement sous certaines conditions, ajoutant ainsi une couche stratégique passive qui fonctionne en arrière-plan.

Cette approche permet aux joueurs d'élaborer des stratégies plus profondes et réfléchies, prenant en compte non seulement les actions immédiates mais également les configurations à long terme.

L'introduction de ces compétences passives augmente la variété des tactiques disponibles, offrant aux joueurs la possibilité de concevoir des plans de jeu qui s'adaptent et répondent à une gamme plus large de situations.

Les compétences passives sont gérées de manière à refléter leur nature automatique et conditionnelle. Voici comment nous avons implémenté ces compétences pour différents personnages dans notre code :

1. Archer, Envoûteur, et Soigneur :

- Méthode nextTurn() : Pour ces personnages, leurs compétences passives sont vérifiées et activées à chaque tour dans cette méthode. Cela permet d'assurer que les effets passifs sont évalués régulièrement à chaque cycle de jeu.
 - Exemple : L'Archer avec la compétence "Œil de faucon" augmente sa portée de tir sous certaines conditions de santé, ce qui est vérifié au début de son tour.

2. Autres Personnages :

- Méthode useAttack() : Pour les autres personnages, les compétences passives sont intégrées dans la logique d'attaque, où les conditions sont souvent liées aux actions offensives.
 - Exemple : Le Voleur avec la compétence "Attaque Surprise" a une chance d'étourdir son ennemi lorsqu'il attaque, ce qui est vérifié à chaque utilisation de l'attaque.



Des effets visuels ont été ajoutés sur les compétences passives suivantes : Lien Vital, Guérison Forestière, Siphon d'Âme, Soif de Sang, Aura.

Voici la liste des nouvelles compétences passives :

Lifeweaver : Lien vital

Quand le Lifeweaver soigne quelqu'un, il reçoit aussi 40% de cette guérison. Lorsqu'il est tout seul dans son équipe, sa compétence 'absorption d'âme' n'a plus de cooldown.

Voleur : Attaque Surprise

Le Voleur a 20% de chances d'étourdir son ennemi quand il l'attaque.

Envoûteur : Ascension Arcanique

A partir de 15 tours, l'Envoûteur gagne 2 points de base de PA supplémentaires.

Bûcheron : Guérison Forestière

Si le Bûcheron possède moins de 25% de ses PV initiaux, il récupère 1 point de vie à chaque attaque et gagne 1 point de base de PA supplémentaire.

Druide : Siphon d'Âme

Le Druide absorbe les points d'action des ennemis qu'il bat.

Valkyrie : Protection Magique

La Valkyrie reçoit 2 points de dégâts en moins des attaques magiques. Elle a également une régénération de sa santé à hauteur de 33% des dégâts de son attaque.

Netherfang : Soif de Sang

Le Netherfang récupère 50% des dégâts infligés sous forme de PV lorsqu'il est en dessous de 50% de ses PV initiaux.

Soigneur : Aura

Le Soigneur active une guérison automatique de 1 point de vie à chaque point d'action, pour tous les alliés situés à 2 hexagones ou moins de lui.

Guerrier : Mur de fer

Renforcement de la défense du Guerrier, qui diminue les dégâts subis de 1 point, lorsque ses PV sont réduits de 50%.

Mage : Conduit arcanique

Diminution de 2 pour les cooldowns du mage lorsque des alliés à proximité utilisent leurs compétences.

Archer : Oeil de faucon

Augmentation de la portée de tir de l'archer de 2 hexagones lorsque sa santé tombe sous 1/3 de ses PV initiaux.

7.2.4. Nouvelles cartes

Le jeu en l'état initial ne possédant que 2 cartes : *Ruines* et *Aléatoire*, nous avons jugé nécessaire d'enrichir le choix de cartes qui est un aspect fondamental d'un jeu de stratégie.

En plus d'avoir ajouté des cartes fonctionnelles, nous avons également pris soin d'ajouter des événements spéciaux à ces cartes : des portails de téléportation et des buissons d'invisibilité. Le tout dans le but d'offrir une expérience la plus diverse et complète que possible au jeu.

De plus, le bonus anciennement généré aléatoirement sur le terrain à chaque partie peut désormais être assigné à un groupe d'hexagones précis. Cela permet de créer des cartes plus diverses mais également de permettre une meilleure modularité à l'avenir.

Arène

L'Arène est la première carte supplémentaire, parsemée de sombres murs, l'espace large aux extrémités hautes et basses devient de plus en plus exigu à l'approche du milieu où se trouve la zone bonus. Il y a de l'autre côté des 2 murs centraux, en diagonale, les portails reliés entre eux : ces derniers vont permettre aux personnages de passer d'un côté à l'autre de la carte, et ce, une fois tous les 3 tours. Ce nouveau mode de déplacement puissant (environ 25 cases en un mouvement!) ajoute un nouvel aspect à prendre en compte dans la stratégie des joueurs.



Figure 7 : Carte Arène

Forêt

La Forêt est la deuxième carte supplémentaire au jeu, celle-ci mime une forêt possédant de nombreux arbres et buissons, puis au milieu une clairière où des buissons et la zone bonus se retrouveront. Ces buissons vont permettre de rendre invisible tout personnage y entrant. Cette fonctionnalité ajoute de nouvelles possibilités stratégiques non négligeables pour obtenir la victoire.



Figure 8 : Carte Forêt

7.2.5. Nouvelle gestion des sauvegardes

Contexte

Dans l'état initial du projet, les sauvegardes étaient implémentées, mais non fonctionnelles. Après débogage du système, il est possible de faire une sauvegarde d'une en cours, mais pas sur la map aléatoire. En quittant la partie, s'affiche alors dans le menu principal un bouton permettant de reprendre cette partie. Un clique sur ce bouton reprend la partie, mais il y a un bug avec les statistiques de fin de partie. Si l'on quitte la partie après l'avoir chargée, le bouton pour la reprendre n'est plus là. La sauvegarde se trouve donc supprimée au moment de son chargement.

Création

Une première modification a été faite pour débugger l'écriture et la lecture de sauvegardes. La possibilité d'avoir 3 sauvegardes qui ne se suppriment pas après chargement a été ajoutée pour pouvoir garder plus d'une partie en mémoire, ce qui nous sera d'ailleurs utile pour préparer des scénarios de jeu pour la soutenance.

Suite à une réunion, M. LATTAUD a proposé de plutôt avoir un système pouvant gérer un nombre infini de sauvegardes.

Une deuxième modification a été effectuée afin de pouvoir gérer un nombre infini de sauvegardes et les supprimées. Les fonctions d'écriture et de lecture ont été modifiées en conséquence, et il a été ajouté :

- Un bouton "Sauvegarder" dans le menu pause qui fait apparaître un nouveau panneau permettant de nommer la sauvegarde (si la partie n'est pas déjà une sauvegarde).
- Un bouton "Redémarrer à la dernière sauvegarde" dans le menu pause
- Un bouton "Voir les sauvegardes" dans le menu principal menant à une nouvelle scène pour visualiser les différentes sauvegardes.

Cette nouvelle scène possède une page défilante (scrollView) dans laquelle sont créés dynamiquement les boutons permettant d'accéder aux différentes sauvegardes et de les supprimées. Les boutons sont générés à l'aide d'un patron de bouton (prefab) et sont mis dans un vertical layout group. L'ensemble de ces éléments permet de s'adapter, peu importe le nombre de sauvegardes. La scène reprend l'esthétique du reste du jeu.

Cette nouvelle scène comprend également un bouton "Retour" pour retourner au menu principal.



Figure 9 : Écran de gestion des sauvegardes



Bénéfices

- Rechargement d'une partie à la dernière sauvegarde effectuée : Il est possible par exemple faire une sauvegarde avant de tenter un coup risqué puis revenir en arrière.
- Le partage des parties contenant un bug : Si un bug est trouvé dans une partie avec un certain personnage, une certaine map et des IA ou non, il est possible de partager la configuration de cette partie en envoyant le fichier de sauvegarde aux autres membres du groupe.
- Pouvoir préparer différents scénarios de jeu : Dans le but de les présenter lors de la soutenance.

7.2.6. Affichage des compétences dans le menu

Contexte

Nous nous sommes rendus compte qu'il n'existe que 2 guides dans le jeu : un dans le menu principal et un autre en jeu. Hors la connaissance de ces compétences est essentielle au moment de la sélection des personnages. En effet, les compétences de chaque personnage sont essentielles à connaître car elles décrivent presque entièrement le rôle d'un personnage. Il a donc été décidé d'ajouter en complément de ces 2 guides, un affichage des compétences des personnages dans le menu de sélection.

Création

Pour chaque carte de personnage a été ajouté un texte représentant au mieux ses compétences, un script a été ajouté aux cartes des personnages pour détecter quand la souris passe au-dessus d'elle. Un texte avec un script qui lui permet de suivre la souris a été créé.

Quand une carte détecte la souris au-dessus d'elle, sa description des compétences est transmise au texte qui suit la souris pour l'afficher au joueur.



Figure 10 : Affichage des compétences dans le menu de sélection des personnages

7.2.7. Nouveau guide

Le guide précédent étant trop limité en taille, une nouvelle page "Guide", accessible depuis le menu principal a été ajoutée. On retrouve dans ce nouveau guide des informations plus claires et plus nombreuses. Pour chaque personnage il y a son image, ses pv et ses compétences spéciales. Ces informations permettent de comprendre l'utilité d'un personnage.



7.3. Modifications

7.3.1. Clarification du code

Détection des clics

Dans le code du jeu fourni, la détection des clics de la souris sur un bouton s'effectue en détectant un clic de la souris puis en vérifiant si la position de la souris était entre certaines coordonnées inscrites en dur dans le code. Cette façon de faire n'est pas adaptée à l'amélioration du jeu car pour chaque création ou chaque ajustement de leur position, il faut réécrire les coordonnées à vérifier dans le code.

Une fonction IsPointerOver a donc été créée, elle prend en entrée un RectTransform et renvoie un booléen indiquant si la souris est sur ce RectTransform. Chaque GameObject de l'interface possède un RectTransform donc cette méthode peut être réutilisée dans tout le projet.

Refonte du script de la caméra

L'un des nombreux avantages d'Unity est de pouvoir attribuer un script à chaque GameObject existant, en l'occurrence la caméra du jeu. Les déplacements de cette caméra étaient gérés dans le script MainGame, script s'occupant du déroulement de la partie. Ne pas avoir de script séparé gérant la caméra nous a paru absurde. C'est pourquoi nous avons créé un script CameraController attribué à la caméra du jeu, afin de rendre le code plus lisible et de mieux séparer les différentes parties de ce dernier. 2 méthodes permettant la communication entre MainGame et CameraController ont notamment été créées : IsOnlyCPUGame indiquant à la caméra si elle doit être immobile tout le long de la partie et GoToCharacter indiquant quel personnage la caméra doit observer.

7.3.2. Résolution de bug

Hexagones empoisonnés

Correction d'un bug qui attribue les dégâts des hexagones empoisonnés au mauvais personnage. Les dégâts étaient attribués au personnage qui joue actuellement et pas au personnage qui a empoisonné les hexagones. Un attribut whoPoisonedIt référencent un personnage a été ajouté au hexagones du jeu. Cet attribut est utilisé lors de l'attribution des dégâts.

Chargement de la carte d'une sauvegarde

Correction d'un bug dans loadGame, qui ne choisissait pas la bonne carte au moment du chargement d'une sauvegarde. Une requête SQL supplémentaire a été ajoutée pour lire dans le fichier de sauvegarde de quelle carte il s'agit.

Placement des personnages lors d'une partie chargée

Correction d'un bug lors du chargement d'une partie via une sauvegarde, le tableau endGameDataCharacter était mal initialisé. En effet, la fonction de création de ce tableau récupérait les personnages en les plaçant sur les hexagones où ils ont commencé la partie. Or, lors du chargement d'une sauvegarde, les personnages ne reprennent pas forcément la partie à leur emplacement de début de partie. Le tableau est désormais initialisé en utilisant hexaGrid.charList qui contient tous les personnages en jeu indépendamment de leur position.

7.3.3. Amélioration des fonctionnalités existantes

L'ancien affichage des actions ne donnait pas le nom des différentes compétences du personnage contrôlé, il affichait juste "Compétence 1", "Compétence 2". Nous avons décidé de changer cela afin que les joueurs aient une meilleure compréhension des compétences à leur disposition. Nous avons aussi modifié cet affichage afin de montrer le nombre de tours restants avant de pouvoir réutiliser une compétence lorsque celle-ci a été utilisée.

Voici un tableau résumant ces changements :

Anciens boutons en jeu	Nouveaux boutons en jeu
	

7.3.4. Gestion des capacités par cooldown

Contexte

Initialement le cooldown d'une capacité se faisait en infligeant 10 dégâts à des ennemis et ce, quelle que soit la capacité. De plus ce compteur de dégâts infligés était commun à toutes les capacités ce qui cause un bug "de logique" car cela permettait d'utiliser une première capacité puis infliger 9 points de dégâts, utiliser une seconde capacité puis infliger seulement 1 point de dégât pour récupérer les 2 capacités.

Création

Un attribut "coolDown" a été ajouté à la classe Attack.

La fonction checkAndSetBackSkill a été écrite. Elle vérifie à chaque tour pour le personnage actuellement contrôlé si les coolDown de chacune de ses attaques sont finies et réactive les capacités si besoin. Elle est appelée dans la fonction nextTurn.

Lorsqu'une compétence a été utilisée, le nombre de tours restant avant de pouvoir la réutiliser est affiché dans le bouton lui correspondant en jeu.

Bénéfices

L'ajout de cooldown permet un meilleur équilibrage.

Correction du bug de récupération des capacités.

7.3.5. Équilibrage

L'équilibrage est effectué en modifiant les valeurs de dégâts, de portée, de dégâts de zone, de cooldown des attaques des différents personnages ainsi que leurs points de vie et points d'action.

Il est fait de manière itérative en testant différentes valeurs, en faisant combattre les IA entre elles, mais aussi des joueurs humains.

7.3.6. Amélioration des scripts CPU

Ajout des nouvelles compétences dans les Scripts CPU existants

Soigneur

Sa compétence *Absorption*, offrant un soin réparti sur 2 tours, a été intégrée dans les scripts CPU. Elle est désormais la deuxième compétence prioritaire à être utilisée, après son soin de zone, dès que les conditions de distance le permettent.

Druide

La compétence *Tempête* du druide, reconnue pour ses dégâts de zone considérables et sa grande portée, a été programmée comme la première action dans son script CPU.

Envoûteur

Sa compétence *Blocage des capacités*, étant la plus puissante car permettant de désactiver les compétences ennemis, est maintenant la première à être déclenchée dans son script CPU.

Mage

Sa compétence *Bouclier* a été ajustée pour être utilisée préférentiellement si les compétences de dégâts de zone ou d'empoisonnement ne sont pas applicables, en se basant sur une distance optimale inférieure à 10 hexagones de l'ennemi.

Valkyrie

L'introduction de la compétence *Tourne Hache* dans le script CPU cible des situations spécifiques où les deux autres compétences, *Coup de Hache* et *Gel*, ne sont pas disponibles, favorisant le combat rapproché.

Archer

Sa compétence *Pluie de Flèches* est désormais priorisée dans les scripts CPU suivant l'utilisation de *Flèche Améliorée*. Cela permet de maintenir une pression continue avec des dégâts de zone sur 2 tours.

Voleur

L'intégration d'*Invisibilité* dans le script CPU se fait à condition que la distance avec l'adversaire soit inférieure ou égale à 10, augmentant les dégâts de 5 pendant 4 tours et offrant une furtivité stratégique.

Guerrier

Dans les scripts CPU, la priorité est donnée à l'utilisation de la compétence *Coup d'Épée* à courte distance. La compétence *Rage* devient primordiale lorsque l'adversaire est à moins de 10 hexagones, augmentant les dégâts de base et les PV.

Création des Scripts CPU pour les nouveaux personnages

Bûcheron

Commence le combat avec *Hurlement* pour une augmentation immédiate de ses dégâts et PV. *Hammer Explosion* est réservée pour les affrontements rapprochés, tandis que *Hammer Launch* est utilisée pour étourdir et se rapprocher des adversaires distants, créant des opportunités d'attaque stratégiques.

Netherfang

Favorise *Hurlement* pour un avantage initial en dégâts et PV. Il tente de se rapprocher pour utiliser *Jump Explosion* pour de larges dégâts de zone, suivi par *Gel* pour contrôler l'ennemi avant d'appliquer son attaque de base *Coup de Griffes*.

Lifeweaver

Le Lifeweaver est stratégiquement optimisé pour rester au plus près de ses alliés. Grâce à la compétence *Téléportation* il peut se positionner à côté de l'allié le plus proche, maximisant l'efficacité de son soutien en soins. Celui-ci peut via la compétence offensive prioritaire *Absorption d'Âme* aider à l'attaque ses alliés en absence de besoin immédiat de soins.

8. Conclusion

La prise en main d'une nouvelle plateforme de développement pour nous qu'est Unity fût un challenge ayant testé nos capacités d'adaptation et la reprise d'un code nous a permis une sorte d'immersion du développement dans des entreprises, où différentes personnes travaillent avant nous sur un projet et d'autres après nous : le but est de comprendre leur code, puis de travailler dessus en utilisant les outils à disposition, le tout en restant le plus clair possible pour les équipes futures. Ce projet est une expérience très enrichissante dont nous sommes fiers d'avoir fait part.

L'objectif initial de l'ajout d'une nouvelle classe, d'implémentation d'une troisième compétence à chaque personnage, de l'équilibration des personnages et de l'amélioration des CPU est clairement atteint et c'est donc les autres nouvelles fonctionnalités que nous avons implémentées qui marque un avancement conséquent dans le développement du jeu *Pause Café*.

Nous pouvons lister les principales améliorations comme suit :

- Ajout de 4 nouveaux personnages.
- Implémentation d'une 3e compétence à chaque personnage.
- Ajout de 2 nouvelles cartes et de nouvelles fonctionnalités à ces dernières.
- Ajout d'un nouveau système de sauvegarde.
- Ajout des compétences passives.
- L'amélioration des scripts CPU.
- Résolution de bug des années précédentes et clarification du code.



Les principales difficultés ont été de comprendre le fonctionnement d'Unity et de prendre avantage d'un maximum des fonctionnalités mises à disposition, de comprendre et travailler sur le code des 7 précédents groupes ayant travaillé sur le projet et de travailler sur le système de joueur CPU.

Nous espérons que les prochaines équipes de développeurs sauront rendre ce jeu encore meilleur qu'il ne l'est à l'avenir.

9. Perspectives

Les principales possibilités d'amélioration sont de refactorer le code en séparant mieux les différentes classes, l'amélioration du personnage custom, par exemple en permettant de créer son apparence, ajout d'un mode multijoueur, ajout de nouveau mode de jeu, par exemple un mode combat contre un boss.

10. Bibliographie

Work Cited

"Mii — Wikipédia." *Wikipédia*, <https://fr.wikipedia.org/wiki/Mii>. Accessed 28 April 2024.