



CROYEZ EN L'ÂME DESCARTES

PAR
AROU LAPIN KILLIAN
BOUFARES ADAM
GAMO CHRISTIAN
GUIAO JOSHUA
SOMARE MOUHAMADOU
RAHARINOSY MAMILALA



REMERCIEMENTS

Dans un premier temps, l'ensemble du groupe tenait à remercier Julien Rossit, notre tuteur et encadrant qui nous a suivi tout au long de ce projet. Grâce à ses nombreux conseils, son expérience et son soutien, nous avons pu travailler autour de ce projet dans de bonnes conditions.

Nous souhaitons aussi adresser nos remerciement à Madame Ariane Mayer qui nous a guidé et orienté vis à vis du rendu du projet. Ses conseils, retour d'expérience et ses différentes attentes nous ont permis de concrétiser de la meilleure façon notre projet.

Nous remercions également le youtubeur Code Monkey ainsi que la communauté active d'Unity, qui nous ont permis d'appréhender de façon autonome l'utilisation de la plateforme de développement logiciel Unity.

TABLE DES MATIÈRES

04	Introduction
05	Présentation du jeu
06	Analyse du sujet : Choix du sujet
07	Analyse du sujet : Contexte et Inspirations
08	Cahier des charges : objectifs
09	Présentation de l'équipe
11	Outils et langages utilisés:Unity
12	Outils et langages utilisés: Communication
13	Conception du jeu
15	Développement du jeu : Programmation
17	Développement du jeu : Aspect Graphique
19	Résultat final
23	Conclusion du PJS4
27	Annexes
	Diagramme de Gantt
	Manuel de jeu
	Lexique
	Charte graphique
	Poster
	Résumés en anglais

INTRODUCTION

Dans le cadre de notre deuxième année de DUT Informatique, il nous a été demandé de travailler en groupe autour d'un projet commun dont le sujet serait libre à condition qu'il rentre dans le cadre de notre formation.

C'est un projet tutoré avec notamment l'accompagnement et le suivi d'un professeur de l'IUT. Les groupes sont constitués au maximum de 7 membres issus du même département de formation.

Le but de ce projet est principalement de nous apprendre à travailler en groupe, découvrir de nouvelles technologies, travailler de manière autonome et organisée en équipe, et de développer une application qui a un aspect original et innovant.

Pour répondre aux attentes du rendu final, on se doit de respecter différents critères et contraintes qui font partie du cycle de développement logiciel. Ce projet reflète une cohésion et un travail concret du groupe au fil des différentes semaines. Il permet en outre de nous mettre face à des difficultés que l'on se doit de surmonter par nous-mêmes, avec ténacité. Ce projet nous a permis de développer les compétences et les qualités de chacun.

Lors du choix de notre sujet de PJS4, nous avons eu au départ plusieurs idées très différentes, allant d'une application connectée à un réfrigérateur pour récupérer les aliments stockés et proposer des idées de plats, à une application étudiante réunissant emplois du temps, trajet et transports en commun, messagerie étudiant/professeur...

Au final, comme nous étions majoritairement composés de gamers, nous avons décidé de réaliser un jeu vidéo. Nous nous sommes donc fixé comme objectif la création d'un jeu vidéo, notre projet devant être réalisable par un groupe de 6 étudiants créant pour la première fois un jeu mais qui reste tout de même divertissant et de qualité pour les joueurs.

Ce rapport nous permettra donc de présenter et restituer des éléments du processus de développement, tels que la gestion, le contexte ou encore la réalisation de ce projet.

PRÉSENTATION DU JEU



Card Souls est un jeu 2D de stratégie et de cartes se déroulant dans un univers medieval-fantasy. Dans ce jeu, vous allez pouvoir affronter un autre joueur en 1 contre 1 et en tour par tour, où chaque joueur effectue ses actions avant de donner le tour à l'autre.

Chaque joueur a son camp : On a le camp Héros, qui représente le joueur 1, et le camp Dark, qui représente le deuxième joueur et où les personnages sont des clones maléfiques des personnages héros.

Le but est de détruire le « Nexus » du joueur adverse, qui représente les points de vie du joueur. La partie se termine donc lorsque l'un des deux Nexus est détruit.

A travers un système de cartes et de mana, les joueurs vont pouvoir invoquer des personnages unique et hauts en couleur, chacun ayant leurs propres pouvoirs et capacités. Dragon, chevalier, mage, ou autres créatures magiques, vont s'affronter dans une bataille épique au sommet ...

Inspiré de jeux mobiles et de tactical RPGs comme Hearthstone ou Fire Emblem, les joueurs les plus aguerris pourront retrouver des mécaniques semblables et un univers familier.

ANALYSE : CHOIX DU SUJET

Nous avons donc choisi de réaliser notre PJS4 autour la création d'un jeu vidéo.

Cependant, nous devions nous décider sur quel genre et quel type de jeu. Notre choix s'est rapidement porté sur un jeu 2D, la 3D étant impossible pour des amateurs comme nous et la création des graphismes aurait été une tâche colossale pour des étudiants qui débutent dans le développement de jeu vidéo.

Nous avons ensuite débattu sur le fait de créer un jeu de stratégie, inspiré de Fire Emblem et ses déplacements dans un terrain quadrillé, ou un jeu de cartes, similaire à Hearthstone.

Nous nous sommes alors rendu compte que nous pouvions combiner les deux, et créer un jeu reposant sur un concept assez original, ou en tout cas, peu exploré dans le monde du jeu vidéo.

La deuxième question que nous nous sommes donc posés concerne la création des graphismes ; qui allait s'en occuper, et comment les réaliser ? Fort heureusement, Killian a déjà de l'expérience avec la création de sprites sur Aseprite, nous avons donc décidé d'avoir un aspect graphique de type pixel art.

Enfin pour le thème, nous avons choisis le medieval - fantasy, puisque nous avons le plus de connaissances dans ce genre d'univers, par exemple avec les films du Seigneur Des Anneaux, mais aussi car c'est l'univers le plus répandu dans les jeux Tactical RPG comme Shining Force ou Tactics Ogre, et dans Hearthstone, qui est notre inspiration principale pour l'aspect jeu de cartes ; Nous voulions amener le joueur dans un univers familier et cohérent à notre type de jeu.

Concernant le nom de notre jeu vidéo, celui-ci vient d'un jeu de mot entre les cartes, qui font référence à Descartes et donc à L'UT mais aussi au système de cartes dont le jeu est basé, et Dark Souls, une série de jeux d'action-RPG japonais, réputée pour sa difficulté.



ANALYSE :

CONTEXTE & INSPIRATIONS

Le domaine du jeu vidéo étant l'un des plus vastes qui soit, il existe à peu près un jeu sur tout ce à quoi l'on puisse penser. Et le type de jeu qu'est Card Souls ne déroge bien évidemment pas à cette règle.

Plusieurs jeux existent dans le domaine du Tactical RPG, du jeu de cartes et du Gacha.

Cependant, Card Souls est un hybride de ces trois types de jeux, c'est ce qui fait son identité propre et le distingue du lot.

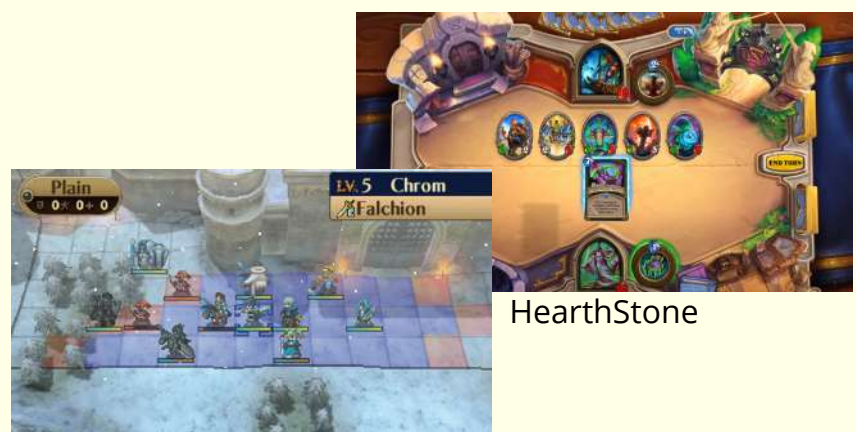
Pour créer notre jeu nous nous sommes donc inspirés d'un jeu de chacun des 3 types cités précédemment.

Pour le côté T-RPG et jeu de cartes qui représentent le gameplay, nous avons pris comme base respectivement les jeux "Fire Emblem" et "HearthStone". En effet, notre terrain de jeu est basé sur un système de grille (terrain quadrillé) et le déplacement des personnages se fait sur cette grille, comme dans le jeu "FireEmblem" entre autres.

Quant à l'aspect jeu de cartes, dans "HearthStone" chaque joueur pioche au maximum une carte par tour et gagne une unité d'Energie (appelée "Mana") au début de son tour, cette même unité est nécessaire pour placer les cartes sur le terrain, cartes qui ont donc un coût en Mana. Ce même système de pioche et de gain d'énergie est retranscrit dans Card Souls avec quelques singularités.

L'aspect gacha est lui inspiré essentiellement du jeu "Seven Deadly Sins Grand Cross", lui même hybride entre un jeu de cartes et un gacha.

Le côté gacha n'étant pas encore implémenté, cette partie sera développée dans la partie "Points d'améliorations" dans la conclusion du rapport.



Fire Emblem

HearthStone

CAHIER DES CHARGES

Objectifs

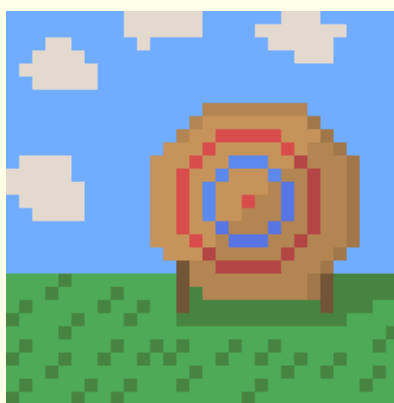
Afin d'être le plus conforme à nos attentes, nous avons mis en place un cahier des charges sur lequel repose le projet. Ce dernier nous permet de définir de manière précise les fonctionnalités auxquelles notre jeu doit répondre, et de pouvoir s'y référer en cas de doutes ou d'incompréhensions entre les membres de l'équipe.

Nous nous sommes donc mis d'accord sur les objectifs et les fonctionnalités/mécaniques primordiaux de Card Souls, et mis en place un planning de tâches à réaliser répartie sur 8 semaines.

- Apprendre de manière totalement autonome le développement de jeu vidéo 2D à l'aide d'Unity
- Créer l'univers du jeu avec le design des personnages, leurs capacités et leurs sprites
- Pouvoir jouer contre un autre joueur à travers un système de tour par tour
- Créer un système d'invocation à travers des cartes à piocher depuis un deck
- Créer un terrain de telle sorte à ce qu'il soit quadrillé et dont les déplacements se font de case en case
- Créer un système de combat où l'on peut attaquer des personnages ennemis

En plus de ces objectifs premiers, nous avons l'intention de créer un gacha pour que les joueurs puissent constituer eux-mêmes leurs decks, un système de pathfinding, des animations et des sons pour rendre le jeu plus vivant. Nous voulions aussi créer une interface simple qui permettrait à n'importe quel type de joueur, débutant ou non, de prendre rapidement les commandes et commencer à jouer.

De ces objectifs nous avons pu attribuer à chacun un rôle et donc une tâche à effectuer.



RÉPARTITION DES TÂCHES

CHRISTIAN GAMO

CHEF DE PROJET et DEVELOPPEUR UNITY

Assemblage des différentes parties du code + bugs

En tant chef de projet, il a été responsable du bon déroulement du projet et de la bonne coordination entre les différentes tâches .C'est lui qui a été chargé d'assembler le code de chacun tout en vérifiant leur compatibilité et régler d'éventuels bugs et ajustements.

MOUHAMADOU SOUMARE

DEVELOPPEUR UNITY

Développement du système de combat

Responsable du développement du système de combat et de leurs mécaniques, en allant de l'implémentation des stats des personnages comme l'attaque ou l'armure au lancement des attaques contre les ennemis. Il s'est aussi assuré de l'équilibrage des personnages.

ADAM BOUFARÈS

DEVELOPPEUR UNITY et IDEES

Développement du terrain et du système de déplacement

Responsable du développement du terrain quadrillé dans lequel les personnages se déplacent de case en case et où il peut y avoir des obstacles. Il est aussi à l'origine de plusieurs idées de mécaniques dans notre jeu en s'inspirant de ses jeux mobiles.

Designer du Teaser Vidéo, du Logo et de l'affiche du jeu.

RÉPARTITION DES TÂCHES

KILLIAN AROULAPIN

GRAPHISTE et DEVELOPPEUR UNITY

Création de l'aspect graphique et animations

Responsable de l'entièreté de l'aspect graphique, c'est lui qui a réalisé les sprites des personnages, du terrain de jeu, ainsi que toutes les animations que l'on retrouve dans le jeu.

JOSHUA GUIAO

DEVELOPPEUR UNITY et DOCUMENTATION

Système de cartes

Responsable du développement du système de cartes, c'est à dire de l'affichage de la main, de la pioche aléatoire et de l'invocation sur le terrain du personnage de la carte choisi.

MAMILALA RAHARINOSY

DEVELOPPEUR UNITY et REDACTEUR

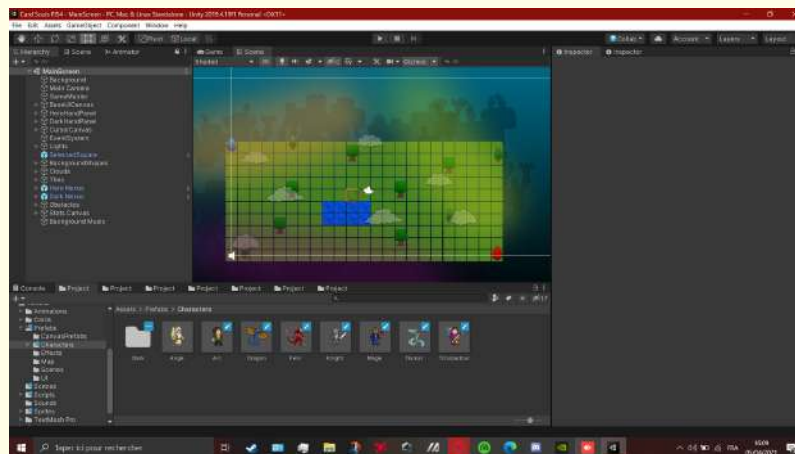
Développement de l'interface et événements

Responsable du développement de l'interface et des événements, comme le changement de tour, l'indicateur de dégâts lors d'une attaque, l'affichages des stats des personnages, et le déclenchement des effets de particules lors de la mort d'un personnage ou de la destruction du Nexus.

OUTILS ET LANGAGES UTILISÉS

Plateforme de développement : Unity et C#

Pour mener à bon port ce projet de développement jeux vidéo, nous avons préféré un outil "spécialisé" dans le domaine. On s'est après réflexion orienté vers la plateforme Unity qui convenait à nos attentes. Nous avons tous découvert son fonctionnement et maniement lors de ce projet.



Unity 3D est un moteur de jeu développé par la société Unity Technologie. Il est régulièrement mis à jour, gratuit et à la disposition de tous. Celui-ci fonctionne avec le langage C# et JavaScript. Cette plateforme de développement a de nombreux intérêts, comme la prise en main simple et une communauté active. En joignant ces qualités, nous avons pu travailler efficacement sur Unity.

Pour un développement commun et progressif, nous avons utilisé Collaborate. C'est une extension d'Unity permettant une collaboration fluide via une sauvegarde synchronisée dans le cloud et partagée au reste du groupe. Il est similaire à Git dans le fait que il y a un système de conflits au cas où plusieurs personnes modifieraient le même fichier et un système de "branches" pour pouvoir développer des fonctionnalités indépendamment des autres développeurs..



OUTILS ET LANGAGES UTILISÉS

Communication et documents : Discord, WhatsApp et Google Drive

La communication au sein du groupe s'est faite via WhatsApp, Discord, et Google Drive. Ce projet à été le fruit d'une collaboration dans le temps.

Nous avons utilisé Discord pour pouvoir organiser des réunions en vocal lorsque l'on ne pouvait se rendre à l'IUT, nous y avons notamment créé notre "QG" via un serveur avec différents salons dédiés. Par exemple, il y avait des salons pour la bibliographie, des salons pour les images, des salons pour les différents tutos ... Cette organisation du serveur nous a permis que les données échangées aient un flux de transition. L'application de communication Discord nous a permis de constamment être réunis et affronter le projet de manière groupé au moment nécessaire.

Sur Whatsapp, nous échangeons de façon plus libre dans le sens où l'on prenait du recul quant à la réalisation de Card Souls. Sur Google Drive, nous partagions perpétuellement les documents nécessaires et trop lourds pour être conservé sur Discord.

Au final, ces 3 applications ont permis la circulation de l'information, grâce à cela, nous avons pu avancer avec l'aide commune et partagée de toute l'équipe.



CONCEPTION DU JEU

La programmation du projet s'est donc divisé en 6 parties :

- L'aspect graphique
- Le terrain et le système de déplacement
- Le système de combat
- Le système de cartes
- Les évènements (changement de tour, indicateur de dégâts, affichage des stats du personnage...)

Pour accompagner notre travail, nous avons notamment mis en place un cahier des charges, un diagramme de gantt et des moyens de communication pratiques sur Discord et WhatsApp.

Tout d'abord, il nous a fallu énormément de temps pour apprendre le développement sur Unity et la programmation avec le langage C# : l'équipe a dû se former de manière autonome pour ainsi faire face à un tel projet. N'ayant aucune l'expérience sur Unity et C#, l'adaptation a été assez hasardeuse et compliqué, mais avec l'aide de tutoriels sur Youtube, nous avons su rebondir et commencer le développement du jeu.

Pour ce faire, il a d'abord fallu programmer les personnages en premier, car c'est autour d'eux que toutes les fonctionnalités se reposent, surtout au niveau de leurs stats et de leurs capacités. Nous avons donc établi une liste d'idées de personnages collant à un univers medieval-fantasy, puis nous avons créer les sprites et le code permettant d'implémenter leurs stats. Pour ce faire, Christian, le chef de projet, a demandé à ceux qui avait des idées pour les personnages, particulièrement Adam, de fournir une description détaillé du personnage, de leur nature, apparence, stats, pour que Killian puissent dessiner un sprite à la hauteur de nos attentes. Nous avons alors chacun pu développer en avance l'implémentations des personnages avant de recevoir les sprites en eux-même.

CONCEPTION DU JEU

Ensuite, chacun a pu s'atteler au rôle qu'il lui a été attribué;

Adam et Mouhamadou ont travaillé en étroite collaboration pour s'assurer que les personnages se déplacent correctement sur le terrain pour ensuite attaquer un ennemi, surtout lorsque un personnage à distance attaque un autre au niveau de la portée d'attaque.

Mamilala a programmé l'interface et les évènements, donc le changement de tour, l'affichage des dégâts, et a donc surtout collaboré avec Killian pour avoir des sprites pour l'interface et que les effets lors de la mort d'un personnage ou du Nexus corresponde bien à la charte graphique. Il a également collaboré avec Mouhamadou pour s'assurer que l'indicateur de dégâts apparaisse bien au moment de l'attaque et qu'il était correct au niveau du calcul.

Joshua a quant à lui développer le système de cartes, donc la pioche aléatoire de personnages, l'affichage de la carte et des stats du personnage, et le placement du personnage sur le terrain. Il a donc collaboré avec Adam pour s'assurer du placement à la bonne case du personnage.

Killian a donc réaliser l'entièreté des sprites et des animations, sur Aseprite et sur Unity, et a donc collaboré avec tout le monde, surtout avec Mohamadou pour les animations pour qu'elles se lancent au moment de l'attaque du personnage.

Christian, étant chef de projet, s'est assuré du bon déroulement du projet en s'assurant qu'il y ait une bonne cohésion d'équipe et a assemblé le code final via Unity Collaborate; Il a pu régler les nombreux bugs et vérifier que les différents codes produit par les membres du groupe soit bien compatible entre eux, et s'il y avait des conflits ou un non fonctionnement, renvoyer le code au membre concerné pour qu'il le corrige ou le corriger lui même.

DÉVELOPPEMENT DU JEU: PROGRAMMATION

La programmation de notre jeu relève de la programmation objet mais est assez différente du Java.

Tout d'abord, l'aspect visuel et le code programmé sont séparés : Le visuel se retrouve sur ce que l'on appelle une "scène" où l'on va ajouter les personnages et le terrain. Cette scène donc représente ce que l'on verra sur l'écran, il y a une "caméra" pour contrôler ce que le joueur voit, et elle contient les différents objets du jeu, donc les personnages, les obstacles ou l'interface. Les différents écrans que voit le joueur sont donc sous la forme de scènes sur Unity. Lorsque le jeu démarre, ces scènes sont modifiées dynamiquement en fonction

Ensuite, les objets sur Unity qui interagissent donc avec cette scène apparaissent sous la forme de "Prefabs" : Sa nature peut varier, mais dans notre jeu ils commencent généralement sous la forme d'un sprite qui est ajouté sur la scène auquel on va rajouter des "scripts", qui contiennent le code pur et dur. Il y a ensuite des objets simples qui contiennent uniquement du script.

Enfin, il est important de parler du Canvas : en effet, c'est là où est contenu l'ensemble de l'interface; Il est séparé du reste des objets présents sur la scène, et va permettre à l'interface de rester responsive face à des changements d'écrans.

DÉVELOPPEMENT DU JEU: PROGRAMMATION

On a donc comme structure du code les dossiers :

- Scenes : Elle contient les différents écrans du jeu
- Scripts : Elle est le coeur du projet, elle contient toute la partie programmation dont le code qui va composer les prefabs. Elle est divisé en plusieurs sous-dossier en fonction de quelle système de jeu est concerné.
- Prefabs : Elle contient les objets Unity interagissant avec la scène, donc les personnages, les cases du terrain, les effets de particule lors de la mort d'un personnage ou du Nexus, et la template de la carte affiché sur la main du joueur.
- Sprites : Elle contient les sprites de l'ensemble du jeu : Il est divisé en deux dossiers pour les personnages et pour l'interface.
- Animations : Elle contient les animations des personnages mais aussi celle de la caméra lorsqu'il y a une attaque
- Sounds : Elle contient les sons du jeu. Ici par manque de temps nous avons mis que la musique d'arrière plan, correspondant à une bataille épique.
- Cards : Elle contient les objets scripts de cartes qui contiennent chacun un prefab de personnage, leur coût en mana et leur nom, et les decks Héros et Sombre dans lequel les deux joueurs vont piocher.

Dans notre jeu, comme il n'y a malheureusement pas de menu, il n'y a qu'une seule scène. Cette scène contient donc la Canvas qui contient l'interface, le terrain composé de prefabs de cases et d'obstacles, le curseur du jeu, et contiendra les personnages qui seront invoqués sur le terrain.

Ensuite, pour entrer dans la technique, chaque prefab de personnage hérite d'un script "Unit.cs" qui dicte les stats, les déplacements sur le terrain et les attaques, et chaque prefab de cases du terrain hérite de "Tile.cs" qui dicte les interactions entre le terrain et les personnages et le placement d'un personnage nouvellement invoqué. Enfin, il existe un script "GameMaster.cs" qui gère les événements et va relier tout les scripts entre eux et permettre de lier les différents systèmes.

DEVELOPPEMENT DU JEU: ASPECT GRAPHIQUE

Avant de pouvoir commencer à travailler sur les graphismes, nous avons dû nous décider sur le choix d'une charte graphique qui convenait à tout le monde. Cette étape fut assez rapide puisque nous avions tous la même idée de ce à quoi un T-RPG ressemblait. Notre objectif dans le choix de la charte graphique était de créer une ambiance familière aux joueurs, de les accueillir dans un univers qu'ils connaissent. Nous avons donc décidé de partir sur un thème Medieval Fantasy, puisqu'il s'agit du standard lorsque l'on parle de RPG.

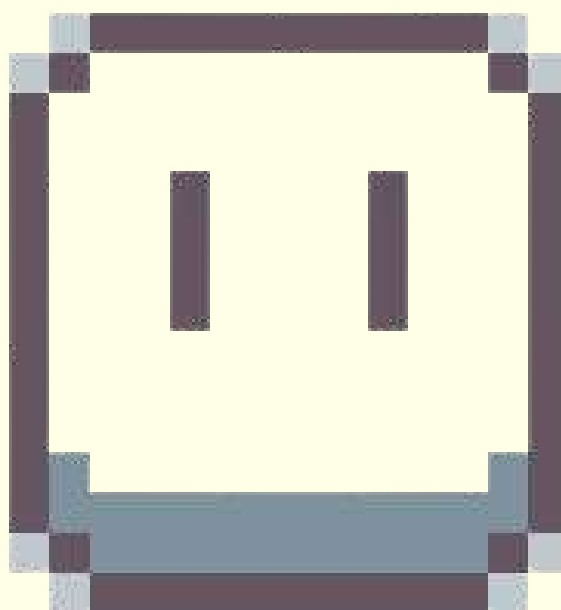
Après avoir choisi le thème des graphismes, nous nous sommes concertés sur leur forme, à quoi ressemblerait le jeu. Encore une fois, le débat fut court puisque nous étions tous d'accord pour faire un jeu en pixel art. Nos raisons derrière ce choix sont assez simples : il était beaucoup plus simple de faire un jeu en pixel art plutôt qu'en dessin, et l'un des membres de notre équipe avait déjà de l'expérience dans le domaine.

Une fois la charte graphique et la forme décidée, il fallait réaliser les sprites (les modèles des personnages), la carte du jeu, et les animations des personnages, afin de permettre un support visuel complet lors d'une partie. Pour la création des sprites, le logiciel Aseprite a été utilisé, puisque la personne en charge des graphismes avait déjà de l'expérience avec. Lors de la création des personnages, ces derniers étaient découpés en plusieurs parties (torse, tête, bras, ect...) afin de faciliter la création des sprites, mais surtout en prévision de leur animation. Il a aussi été décidé que plus un personnage est rare, plus il serait gros, ainsi, les unités les plus communes faisaient du 32x32 pixels, et les plus rares du 80x80 pixels, les dimensions des sprites pouvant donc varier entre ces valeurs.

DEVELOPPEMENT DU JEU: ASPECT GRAPHIQUE

La carte du jeu a aussi été réalisée avec Aseprite, une case sur laquelle un personnage peut se situer fait du 32x32 pixels, puisque cela permet d'obtenir un résultat visuel satisfaisant, tout en étant assez rapide à produire. La carte en elle-même représente une forêt, pour rester en accord avec le thème Medieval Fantasy.

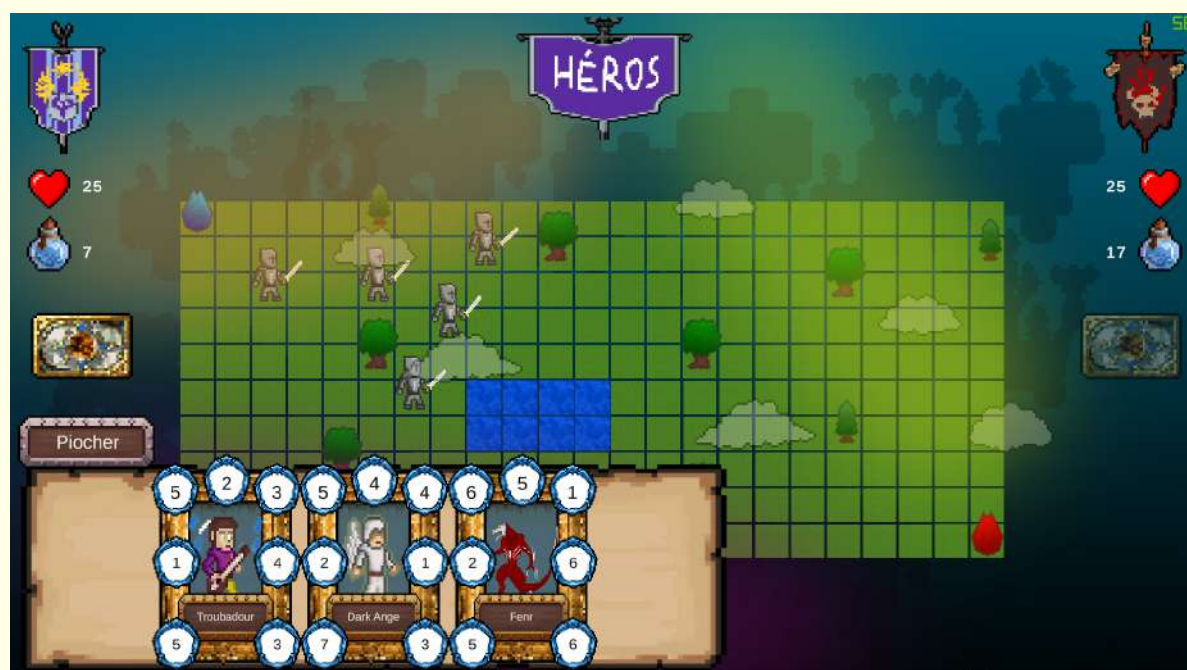
Enfin, les animations des personnages furent réalisées sur Unity, le logiciel permettant d'animer des objets avec du rigging. L'animation par rigging consiste à faire bouger différentes parties d'une image, par exemple un bras, sans changer l'image en elle-même. Cela permet de réaliser des animations certes limitées, puisque les sprites en eux-mêmes ne changent pas, mais pour un jeu en 2D, le rigging a permis d'obtenir un excellent résultat. Grâce à cette méthode, chaque personnage possède deux animations : tout d'abord une animation Idle, c'est-à-dire quand le personnage ne fait rien, afin de le rendre plus vivant quand le joueur ne le sélectionne pas, et une animation d'attaque, lorsqu'ils doivent se battre contre d'autres unités.



RESULTAT FINAL



Ecran lors du lancement du jeu : Ici c'est le tour du premier joueur. On peut voir la main à gauche et à droite les points de vie du Nexus et les points de mana respectif des deux joueurs.



Lorsque l'on clique sur l'icône de la carte, le contenu de la main du joueur en question apparaît. il peut piocher une fois par tour et invoquer un personnage en choisissant une carte

RESULTAT FINAL



Ici le joueur choisit sur quelle case il va invoquer le personnage qu'il a choisi



On peut voir ici le changement de jouer et la main du deuxième joueur.

RESULTAT FINAL



Affichage des stats de l'un des personnages. Le panneau apparaît lorsque l'on fait un clic droit sur un personnage.



Ici on peut voir les indicateurs de dégâts suite à une attaque.



Mort d'un personnage

RESULTAT FINAL



Destruction du nexus et fin de la partie : Victoire du joueur 1

CONCLUSION

Points d'amélioration

Bien que la majorité des objectifs que l'on s'était fixés aient été atteints, il est évident que, comme tout développement dans le domaine du jeu vidéo, certaines parties puissent être sujettes à diverses améliorations et extensions.

Dans cette partie, nous allons évoquer les points du projets qui nous semblent favorables à une amélioration et/ou implémentation.

-L'aspect économique du jeu : Le Gacha

Quand nous avons eu l'idée du jeu Card Souls, nous étions partis sur un hybride entre un jeu de cartes, un tactical RPG pour le coté gameplay et un jeu dit "Gacha" pour la partie gestion de cartes et obtention de celles-ci.

En effet, dans la version actuelle de notre jeu, chaque joueur possède l'intégralité des cartes du jeu. Hors, pour implémenter un système de rivalité entre les joueurs et initier l'envie de jouer certaines cartes chez l'utilisateur, nous pensons qu'un système d'obtention des cartes plus ou moins rares devrait être établi. Le joueur utiliserait alors une ressource rare obtainable en jouant, pour ouvrir ce que l'on appelle des "boosters" ou "packs" dans le monde des jeux de cartes. Ce procédé est même utilisé dans la vraie vie pour les jeux de cartes à jouer et à collectionner tels que "YuGiOh" ou même "Pokémon". La seule différence entre les boosters de la vraie vie et les packs présents dans les gachas c'est la manière d'obtention. Dans un gacha ce n'est pas de l'argent réel qui est utilisé mais une ressource obtainable dans le jeu en jouant ou en effectuant différentes missions quotidiennement. Cependant, cette même ressource peut elle même être achetée avec de la monnaie réelle : cela constitue la stratégie économique des jeux gachas. Il est tout a fait possible de jouer et de progresser sans payer, mais payer permet d'aller plus vite dans l'obtention des différentes cartes à jouer.

CONCLUSION

Points d'amélioration

-Le système de déplacements : Le PathFinding A*

Du côté du gameplay, cette version de Card Souls a des bases solides, seul bémol, le système de déplacement, qui bien que fonctionnel, n'est pas très optimisé.

Pour aller d'une case A à une case B le personnage, situé sur la case A, se déplace sur le terrain quadrillé horizontalement vers la colonne de la case B, puis verticalement vers la ligne de cette dernière.

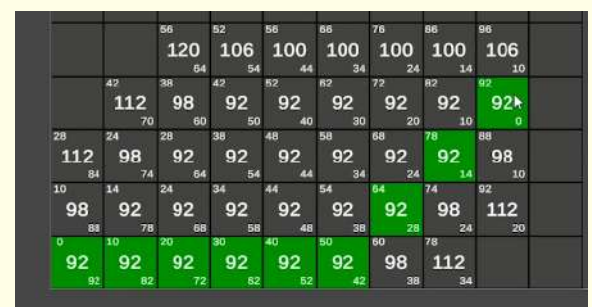
Ici, deux problèmes se posent :

- Le personnage ne prend pas le chemin le plus court pour aller à sa destination.
- Le personnage traverse les obstacles inaccessibles tels que l'eau ou les arbres.

C'est pour cela que nous avons codé un algorithme basé sur le très connu A* (très utilisé en graphes) pour optimiser le chemin que va emprunter le personnage et pour prendre en compte les obstacles et les contourner.

Cet algorithme est déjà codé et fonctionnel, mais pas encore implémenté, nous devons juste faire quelques ajustements pour corréliser le code actuel du jeu et l'algorithme de Path Finding.

Donc, bien que l'algorithme ne soit pas encore dans le jeu final, ce point d'amélioration peut être considéré comme réglé.



Optimisation du chemin
avec A*

CONCLUSION

Points d'amélioration

- Le menu

C'est un élément indispensable pour notre jeu vidéo. Il permettrait en effet au joueur de choisir le mode de jeu auquel il souhaite avant que la partie ne commence. Nous sommes conscient que dans notre rendu final, le joueur commençait forcément une partie 1 contre 1 et l'ajout d'un menu en cas d'ajout d'autres modes de jeu est judicieux. Il permettrait aussi à l'utilisateur de changer de langue ou de couper les sons ou la musique et bien d'autres fonctionnalités auxquelles nous n'avons pas encore réfléchi.

- l'UX

Comme dit précédemment, l'expérience utilisateur est limitée uniquement au lancement d'une partie 1 contre 1. Nous voulons améliorer l'expérience utilisateur en documentant les cartes, disant à quoi servent telle ou telle carte. Nous souhaitons aussi ajouter un tutoriel lors d'un premier lancement du jeu, permettant au joueur de mieux s'appropriier les bases de notre jeu. Enfin, comme dit précédemment, le menu est un ajout considérable pour l'amélioration de l'UX, permettant à l'utilisateur de naviguer comme il le souhaite dans notre application.

- Ajout de bruitages

Si l'on tend l'oreille lors d'une partie de jeu, vous remarquerez qu'il n'y a pas de son lors des déplacements ou des attaques des personnages. Encore une fois nous tenons à souligner que nous nous sommes attardés uniquement sur l'essentiel du jeu, et l'ajout de bruitage n'était pas nécessaire pour nous. C'est donc pour cela que nous comptons ajouter des bruitages de déplacement et d'environnement pour dynamiser les parties.

- Les cartes à effet

Nous comptons enfin ajouter des cartes à effet sur le terrain et les personnages. On parle ici de cartes qui permet par exemple de créer un mur sur le terrain ou d'électrocution d'un personnage. Cela rajoute un enjeu supplémentaire car les joueurs peuvent désormais améliorer leur stratégie de défense et d'attaque.

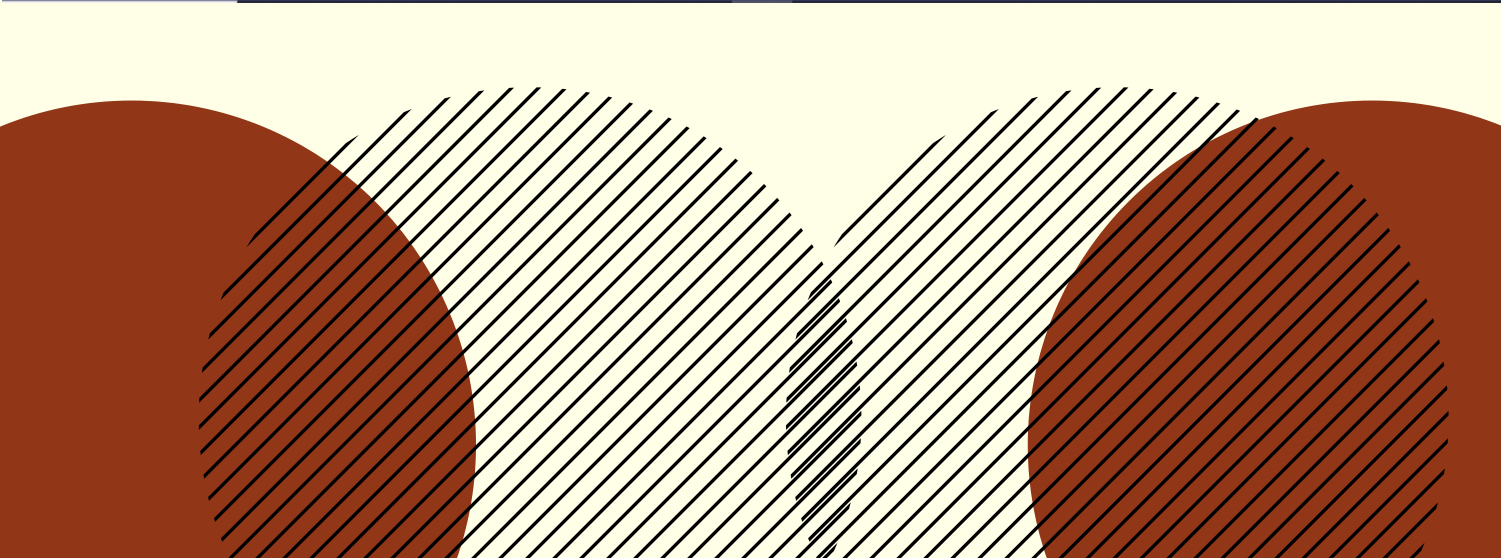
CONCLUSION

Conclusion générale

Au terme de ce projet, notre sentiment est positif. Malgré les problèmes que nous avons pu rencontrer tel que l'impossibilité d'aller au bout de nos idées, nous avons pu acquérir de nombreuses connaissances et compétences nouvelles. Tout au long de ce PJS4, nous avons été amenés à concevoir et implémenter une application menant à la création d'un jeu vidéo de cartes sur Unity. Il nous a permis de consolider nos compétences et développer au mieux notre autonomie et notre capacité de travail en équipe. C'est une expérience plus que bénéfique qui a permis de mettre en exergue les connaissances apprises plutôt dans la formation mais aussi de nouvelles connaissances sur de nouveaux support de développement tels qu'Unity, un aspect du développement logiciel non étudié dans notre DUT. Nous pourrions, en outre, utiliser les connaissances acquises lors de nos études. Les difficultés rencontrées ont malheureusement ralenti notre projet et nous n'avons pas pu mettre en place toutes les fonctionnalités que nous souhaitions, notamment le pathfinding ou encore le menu qui aurait amélioré l'expérience utilisateur. Ce projet a demandé beaucoup de documentation et cela fut à la fois une source de nouvelles connaissances, mais aussi un frein à notre projet. Enfin, la situation sanitaire et le protocole qui en découle ne favorisent pas le travail en équipe, surtout lorsque celle-ci est "naissante", deux membres de l'équipe sont tombés malades au même moment, ils avaient des symptômes du Covid-19, en ajoutant que Mamilala est à Madagascar depuis fin décembre. Ces conditions ont ralenti à leurs manières notre avancée dans le projet.

Toutefois, les fonctionnalités offertes par le jeu vidéo sont multiples, la plus importante pour nous étant le déplacement et l'interaction des différents sprites sur la map...

Pour conclure, nous avons pu produire un projet utilisable et nous en sommes assez fiers, qui comportent une simulation de partie complète, avec des caractères alliées et ennemies, un système de combat et de cartes "piochables". Il y a encore beaucoup d'améliorations à faire et nous en sommes conscient, ceux-ci pourront être effectués par chacun de nous de manière personnelle en guise d'accomplissement personnel.

[illegible]

ANNEXES : MANUEL DE JEU

Principe du jeu

JOUEUR CONTRE JOUEUR
TOUR PAR TOUR

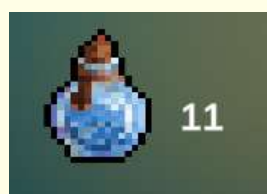


2 CAMPS : HERO ET DARK

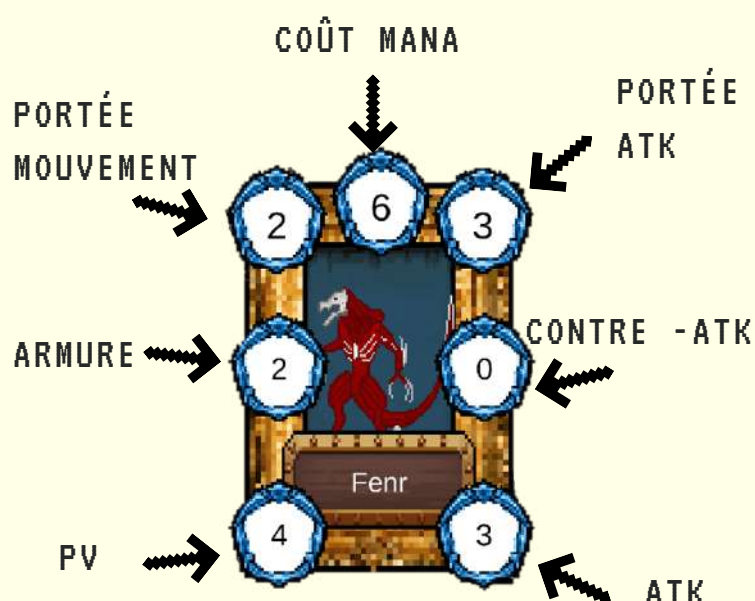


BUT : DÉTRUIRE LE NEXUS ADVERSE
EN INVOQUANT DES PERSONNAGES

Système de cartes et les personnages



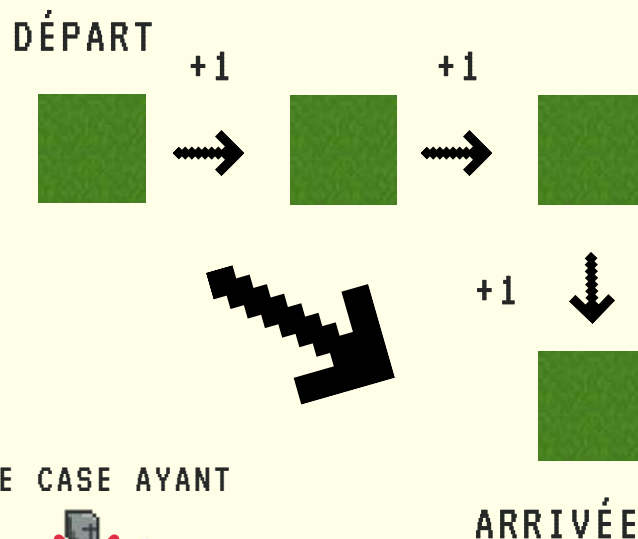
INVOQUER UN
PERSONNAGE NÉCESSITE
DE LA MANA



DES PERSONNAGES UNIQUES AYANT DES STATS DIFFÉRENTS

ANNEXES : MANUEL DE JEU

Terrain et système de déplacement



UN PERSONNAGE NE PEUT SE DÉPLACER SUR UNE CASE AYANT
UN OBSTACLE OU UN AUTRE PERSONNAGE

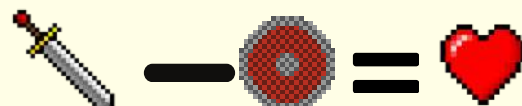


Système de combat



LES STATS DES PERSONNAGES QUI SONT
PERTINENTS :

- LES DÉGÂTS D'ATTAQUE,
- LE NOMBRE DE POINTS DE VIE,
- LE NOMBRE DE POINTS D'ARMURE
- LES DÉGÂTS DE CONTRE-ATTAQUE,
- LA PORTÉE D'ATTAQUE.



UN PERSONNAGE AYANT 2 POINTS DE PORTÉE PEUT DONC ATTAQUER UN
PERSONNAGE SITUÉ 2 CASES PLUS LOIN EN FACE DE LUI, OU UNE CASE EN
DIAGONALE.

ANNEXES

Lexique

Tactical RPG: "Un tactical RPG (abrégé T-RPG) est un jeu de rôle tactique. Dans ce genre de jeu vidéo, le gameplay est basé sur les décisions tactiques que le joueur doit prendre au cours des combats."

Collectif, "Tactical RPG", Wikipedia,

Gacha: "Gacha est le diminutif de Gachapon, des distributeurs de petits goodies (mini figurines, porte-clés) souvent à l'effigie d'une licence."

Vinh, "Dans le merveilleux monde des gacha games japonais", Japan Vrac

Sprite: "Un sprite, ou lutin, est dans le jeu vidéo un élément graphique qui peut se déplacer sur l'écran."

Collectif, "Sprite (jeu vidéo) ", Wikipedia

Gamer: "Gamer est un anglicisme désignant un individu qui joue de manière assidue aux jeux vidéo."

Collectif, "Gamer", Larousse

Unity: "Unity est un moteur de jeu multiplateforme (smartphone, ordinateur, consoles de jeux vidéo et Web) développé par Unity Technologies."

Collectif, "Unity (moteur de jeu), Wikipedia

Aseprite: "Aseprite est un logiciel performant d'édition de sprite et d'animation pour le pixel art"

"Aseprite – logiciel pixel art et animation de sprite", THE_GREAT_PEARLER

Jeu 2D: "Les jeux 2D utilisent des graphismes plats, appelés sprites, sans dimension tridimensionnelle. Ces graphismes sont dessinés sur l'écran sous la forme d'images plates ; la caméra (orthographique) n'a pas de perspective"

"Les caractéristiques des jeux vidéo 2D", Unity

ANNEXES

Lexique

Google Drive : "service de stockage lancé par Google en 2012. "

Collectif, "Google Drive : comment utiliser et télécharger l'app de stockage de fichiers", Le journal du Net

Discord : "Discord est un logiciel propriétaire gratuit de VoIP et de messagerie instantanée. Il fonctionne sur les systèmes d'exploitation Windows, macOS, Linux, Android, iOS ainsi que sur les navigateurs web."

Collectif, "Discord (logiciel)", Wikipedia

Cahier des charges : "document contractuel à respecter lors d'un projet."

"Cahier des charges", Le Journal Du Net

ANNEXES : CHARTRE GRAPHIQUE



ANNEXES : CHARTRE GRAPHIQUE



ANNEXES

Poster

L'idée de base dans la conception du poster était de mettre en scène un duel entre deux personnages, vu que notre jeu est un 1vs1. À défaut de pouvoir représenter deux véritables utilisateurs nous avons décidé de représenter ces derniers avec deux personnages à l'apparence humanoïde de notre jeu.

La mise en scène voulue ici est le fait que ces deux personnages "invoquent" deux monstres non-humanoïdes en plaçant des cartes sur le terrain, comme le ferait un véritable joueur pendant une partie. Le fond de l'affiche représente une carte géographique dans un univers d'heroic fantasy, pour coller avec le thème du jeu.

Finalement, nous avons placé le logo du jeu Card Souls au centre de l'affiche pour attirer l'attention, logo qui lui-même a été réalisé sous PhotoShop de façon à coller encore une fois à cet aspect "fantaisie".

Pour conclure, nous avons rajouté en bas une barre noire divisée en deux parties, la première dans laquelle nous avons indiqué les langages principaux utilisés et la deuxième comportant les photos de nos enseignants qui ont encadré le projet ainsi que les têtes de chacun d'entre nous, le tout en PixelArt pour ne pas trahir l'aspect graphique de l'affiche et du jeu en général. Sans oublier, bien évidemment, la devise de notre jeu "Croyez en l'âme Descartes" jeu de mot on ne peut plus parlant !



RESUMÉ EN ANGLAIS

Christian

The main focus of our project was to create a fun video game, with some complex mechanics but still beginner-friendly. At first we had trouble deciding on whether or not we would make a strategy game or a card game, similar to those on mobile phones, but we quickly settled on a mix of both, a sort of hybrid between a Tactical RPG and a card game, which is pretty original for a video game, or at least not very common. We also decided to make our game in a medieval fantasy setting, since we knew this kind of universe the most, and to pixel stylized, since Killian had some prior experience in making sprites. Thus, Card Souls was born.

During this project, I was assigned to the role of project leader. As a project leader, I assigned to everyone their roles depending on their knowledge and their capacities. I made sure that everyone worked cohesively, in a group, and that nobody worked alone on their side without contacting the others. I also learned how to develop games on Unity and C#. I was also tasked to assemble the whole project, by assembling the different pieces of programming made by the other members and making sure that everything worked perfectly. Thus I was the one to have the final project. I had to check on the various bugs the game had, and correct them if I could. If not, I would contact the author to make him correct his code, and make him talk to the others if necessary.

I was able to do the assembling part with the Unity Collaborate tool, which allows several developers to work on the same project simultaneously. It is similar to Git as it has a system of branches which allows the developers to work on functionalities separately from the other developers and their own code. It also has a “conflict system”, which detects when a piece of code is incompatible with the rest when it is imported into the final project.

This gave me some major headache sometimes because conflicting code is very hard to fix, but I was able to fix those every time it happened, by contacting the one who worked on this code to make sure I wasn't doing any mistakes and checking each conflict one by one and fixing them by redoing the code. Another difficulty I encountered was the random bugs on the Unity platform, as sometimes everything ceases to work and you can no longer import code nor modify it. I'm not sure if it was because we were using Unity Collaborate, but this was extremely annoying and stressful as I had to recreate a blank project from scratch and reimport everything each time. In the end we managed to have a working game, and a pretty fun one too, so I'm satisfied.

In conclusion, managing this project as the project leader really pushed me to my limits, as I had to learn how to manage a team for the first time in my life, and since I am an introvert, I was pretty anxious at first when I told everyone what to do and when I made sure everyone worked in harmony as a team, because I didn't really know what I was doing myself. But I managed to adapt and I think I did a pretty good job as a leader. This project really allowed me to be more confident in myself but also to trust others. However, I don't think I'll be a project leader later as a job since it is still a bit too stressful for me with all the weight of the responsibilities. But who knows, maybe one day.

RESUMÉ EN ANGLAIS

Adam

To briefly sum up my impression about this project and its realisation I would say that it was, undoubtedly the funniest and most rewarding project I have ever done throughout my two years of study spent at the institute. In fact, having the opportunity to freely choose our topic was very pleasant. Thus, we established the needs of our project and created a specification that suited us. Although, in the professional world, we will have to fulfill the clients' requests, it is undeniable that this freedom of choice was very refreshing, motivating and makes the work all the more fun. Actually, making a video game is, for many of us, and for our generation in general, a childhood dream.

When it comes to the distribution of tasks, I was personally in charge of the grid-system and the movement of the characters on this grid, these steps were crucial and very important in the creation of the game.

I first created the field as a grid with inaccessible cells within it. In the code, this is represented by a node tree, some of which are called "Unwalkable". In the final game, these nodes will be unreachable areas such as rocks or water (river, etc...).

For the movement system, I started with a "dumb" movement system, meaning that the character does not take into account the optimization of the path, he moves from a cell A to a cell B just horizontally then vertically. That's why I decided to implement a PathFinding system to optimize the paths taken and taking into account the different obstacles. This algorithm is commonly used in mathematics, it is the A* Algorithm. However, although functional and ready to be used, it is not implemented in the version of the game we presented.

In addition, I was also in charge of making the game teaser, the logo and the poster.

Finally, besides the subjective side, this project was the most complete and the most difficult to realize. The discovery of a brand new environment (Unity) and the use of the C# language were challenges that we were able to overcome, of course not without difficulty.

Teamwork has always been very important throughout these two years of DUT but it was even more important for a project of such a scale as the PJS4.

Hard work, cooperation and perseverance were the key words during the realisation of this project.

Thank you very much for giving us the opportunity to do it!

At the end of this enriching experience I became convinced that software development is what I want to do in the future!

RESUMÉ EN ANGLAIS

Killian

From the very beginning, I was very excited about working on this project, I was the one who proposed the idea of creating a T-RPG to the rest of the group after all. The possibility to work on a video game is something very appealing to me, since I want to work in the field after I get my degrees, and I really liked the creative liberty that came with the project, especially since I worked on the graphics.

When we decided on everyone's roles, I was chosen to bring the visuals to the game since I was the most experienced with making pixel art. While it may not be a role as hard as coding for the game, it did not come without difficulties. As the only one working on the graphics, I had to planned how the characters, the terrain, the esthetics of the game would be before starting to work on them. A good example of this would be the size of the characters, as some units would be stronger than other, I had to make it visually explicit. To allow the player to immediately understand where the bigger threats were, I needed to make the stronger units stand apart from the rest, while not make them stand out too much, or else they would feel like they are not part of the game. In order to do that, I decided to scale the resolution of the sprites to the strength of the units, going from 32 x 32 pixels to 80 x 80 pixels. This allowed all the characters to feel natural on the battleground, while making them distinct enough to let the player understand what is happening during the game.

Another very important part of my job was the animation of the characters. In order to make them feel more alive, making the units move was very important. I really liked animating the characters, as I had never done it before. I think it was a great opportunity to learn something new and applying into a game at the same time. However, there were some problems I had to face during the animation process, especially with the way Unity works. Since I animated all the sprites with Unity, I had to import all the characters I created into the project, but that simple task proved to be quite difficult. For some reason, with the default settings, all images imported in Unity became blurred, and I struggled for way too long in order to resolve this. In the end, there were still some minor errors in the imported sprites, like the face of a character became "greener" because Unity decided spray his eye color all over his face. But those errors being small and unnoticeable in the game, I decided to focus on the animation, since I couldn't solve them. The animation system in Unity is quite limited but very simple to understand, and I think I managed to do a decent job with bringing life into those units by giving them an idle animation and an attack one.

I am really happy I got to work on this project, I learned a lot about how the graphics designers had to think in order to incorporate all the sprites to the game, and that it wasn't just about making the sprites. This firsthand experience is something I really liked, especially since I want to work in the video game industry.

RESUMÉ EN ANGLAIS

Mouhamadou

The PJ54 project is a tutored project carried out as part of our last year of DUT, that is to say with the supervision of a referent teacher throughout the process. It is done entirely for the purpose of learning and personal development. In particular, it will allow students to better understand teamwork and use the knowledge acquired during these two years of DUT. To achieve our objectives for this project, we had to set up specifications and a precise distribution of tasks. We all worked on the Unity development platform and at the same time, we all had a secondary function, mine was that of writer.

My missions in relation to the development of the game mainly revolve around setting up a combat system and the various mechanics around it. So in Card Souls, I was manipulating the implementation of stats for characters, their moves in attack position and damage done to others.

Card Souls being a 2D game, the main point to develop remains the animations, so that is why I often had to communicate with our graphic designer at Sprite in order to "order" images that can be used to create this impression of movement and fighting spirit around the characters.

In terms of code, in particular, I have implemented functions that determine the attack radius, the damage inflicted and the number of possible hits since Card Souls is a turn-based game.

The problems I encountered are particularly related to the implementation, indeed, there are many mechanics such as counterattacks, chaining. The second is that I must have worked with Adam on a constant basis because our parts are closely related. He works on the movements and the environment and I on the system of interaction between the opposing characters. The third difficulty, but not the least, was the learning of video game development on the Unity platform, indeed no member of the team has never created a 2d game let alone from scratch. The C# language used on Unity was not taught during the two years of training, so we had to do some self-learning.

The way I solved these different problems is simple, I persisted and constantly look for the loophole in my way of working. I also helped myself a lot from the Unity community which is very active. As the group was united, we also helped each other from within.

In the end, I managed to put in place the necessary combat system for our game. The characters have a range of attack and skills that are diverse and unique to each.

In conclusion, the project has been very interesting to build over the weeks. The teamwork was more than productive and we were able to achieve our main goals. Despite the fact that there are still improvements to be made, our feeling is positive and these can be done by each of us in a personal way as a personal accomplishment.

RESUMÉ EN ANGLAIS

Joshua

In our project Card Souls where you must summon heroes or evil clones to destroy the enemy Nexus, I dedicate my part in the card system, I mainly deal with the draw of cards to put them in the hand and to place it in the summoning zone.

To begin this part, we started by creating the game board, creating the different hands: the hero hand then the dark hand and in the center of that the summoning area and the button that will allow you to draw the cards to summon.

We then created the different cards. In the first version we created 2 cards to begin to understand the mechanics of the game and also how Unity Engine works because it was the first time we used this engine, we differentiated them with different colors, then we configured them so that these cards can be placed in the hands of each player, for that we configured the button with a new script that will allow to draw a certain number of cards in both sides. Then in the second version we created a button for each side so now each player can now draw cards directly into their hand and they will not share the button and they will summon their own characters according to their side.

Let us move on to the invocation of the cards. To invoke a card, we have to place them into the invocation area but when we select a card, we can place them anywhere we want but it makes nonsense. In order to solve this problem, we will use an event trigger called Drag Drop, this event allows you to drag an object and drop it on a desired location, so we must create this event on all the cards created and create a new script that will only allow to drop the selected card only in the invocation area, so they will not place themselves anywhere. In the latest version of our game the invocation area is now a grid, we can now place them anywhere as we want as long that is in the summoning area and will not stay in a card but the character will come to life to beat the enemy Nexus.

Here is how the card system of our game Card Souls works

The conception of a video game was quite fun, and the discovery of the Unity engine is very interesting. Moreover, the realization of this project in self-learning was very complicated because we had no knowledge in the creation of video game, and it was a challenge that I think to have personally achieved, with the help of several tutorial seen on the internet and the help of my project partners, I thank them for their reliance for the realization of this project.

RESUMÉ EN ANGLAIS

Nicolas

For our last year of DUT Informatique, we were asked to work together on a free subject. As we were searching for our subject, we thought that making a video game was a good idea because we are mostly gamers. That's why we created Card Souls, a medieval-fantasy, tactical RPG, strategy and card game

My team is composed of Christian Gamo our project manager, Killian Aroulapin our graphist, Adam Boufares, Mouhamadou Soumare and Joshua Guiao our programmers

In the group, everyone had different tasks to do, but my job was to create the user interface: I programmed all interactions with buttons, the events like changing turn, the damage display, the camera shaking when attack, and with Killian I added the particles effects when the characters or the Nexus dies.

I first made a template of what the game could look like before Killian gave me his work. I tried to assign a function to every button that I created in the Unity layout and it was pretty easy

Next, for the camera that shakes when attacking and the damage display, I had to contact Mouhamadou to synchronize our work together, we did it with Unity Collaborate and it was very difficult to work together far from each other.

Then Killian gave me his sprites and assets pretty lately so I had to speed up to get every graphic content at its right place.

The main difficulty that I had was that I did all of this with a hazardous internet connection which means that sometimes I couldn't follow tutorials online. I should have done a menu but because of the time, I preferred to focus on the game itself.

It was also difficult to combine each script written by the different team workers because everyone has its own way of coding.

Finally, I can say that it was very difficult to get into Unity because of the new environment and the new programming language.

In conclusion, I can say that this project was very interesting by discovering a new programming language. My programming skills were improved and I can now develop an app with unity in C#. I also enforced my teamwork skills. This project was very difficult because of the sanitary situation but with good communication we were able to work in team, and I think that this is the best way to learn how to work in team. This project helped me understand that even if I'm good at coding, I don't really like to read different scripts written by different persons and then combine them. I think I prefer when I lead the team and that's why I chose MIRAGE to add some company management skills for my scholarship project.