

Rapport de projet

Animatch



BY ILLUSION

Adrien Coureau
Josselin Priet
Guillaume Rat
Hector Thubert
Guilhem Dardonville



Table des matières

1	Cahier des charges	3
1.1	Introduction	3
1.2	Cahier des charges fonctionnel	5
1.2.1	Origine et nature du projet	5
1.2.2	Objet de l'étude	6
1.2.3	État de l'art	7
1.2.4	L'entreprise	9
1.2.5	L'équipe	10
1.2.6	Répartition des tâches	13
1.2.7	Avancement et planification	14
1.3	Conclusion	15
2	Cahier des charges technique	16
3	Tâches collectives	18
4	Tâches individuelles	19
4.1	Adrien Coureau	19
4.1.1	Gestion du groupe	19
4.1.2	GitHub	19
4.1.3	Architecture du projet	20
4.1.4	Graphismes	21
4.1.5	Musiques et sons	23
4.1.6	Animations	23
4.1.7	Menu, sélection	23



4.2	Josselin Priet	24
4.2.1	Algorithme	24
4.2.2	UML	25
4.2.3	Code	26
4.3	Guillaume Rat	30
4.3.1	Communication	30
4.3.2	Multijoueur	30
4.3.3	Launcher	31
4.3.4	Suite Multijoueur	32
4.3.5	Affichage de la Partie	33
4.3.6	Choix des Cartes	33
4.3.7	Améliorations du Code	34
4.3.8	Conclusion	35
4.4	Hector Thubert	37
4.4.1	Conception des animaux	37
4.4.2	Le site Internet	37
4.4.3	Partie Code	39
4.5	Guilhem Dardonville	41
4.5.1	Correction linguistique	41
4.5.2	Système de la barre de vie	41
4.5.3	Le système des cartes	42
4.5.4	Autre	43
5	Récit de la réalisation	45
5.1	Nos peines	45
5.2	Nos joies	46



TABLE DES MATIÈRES

5.3	Avis personnel des membres de l'équipe	47
5.3.1	Adrien Coureau	47
5.3.2	Josselin Priet	48
5.3.3	Hector Thubert	48
5.3.4	Guillaume Rat	50
5.3.5	Guilhem Dardonville	50
6	Annexes	51



1 Cahier des charges

1.1 Introduction

Animatch est le tout premier produit de l'entreprise indépendante ILLUSION. Animatch est un jeu de duel stratégique et de création de deck à destruction de terrain dans lequel deux joueurs s'affrontent. Il sera disponible sur la plateforme Windows mais une déportation vers Android et iOS est également possible.

Le jeu est en deux dimensions, et chaque joueur prend tour à tour le contrôle d'un groupe d'animaux dont l'objectif est d'éliminer le groupe adverse. Pour ce faire, avant le début de la partie, chaque joueur construit son propre deck de cartes, qu'il va pouvoir débloquer progressivement au cours de son expérience sur le jeu.

Chaque carte représente une action qui a un coût associé au coût de la carte. Les cartes sont utilisées pour éliminer les animaux adverses (par des tirs de missiles, des coups au corps-à-corps, etc...), pour détruire le terrain, pour faire tomber son adversaire dans le vide, ou encore pour se déplacer, ce qui offre des possibilités infinies de stratégies à l'utilisateur. Une partie se termine lorsque l'un des deux joueurs a réussi à éliminer le groupe d'animaux adverse.

En somme, Animatch peut être rapproché d'un mélange entre le jeu *Worms* (1995) et le jeu *Clash Royale* (2016). Il est accessible à tout public, amusant, et surtout, divertissant. Le multijoueur sera donc basé sur du "Matchmaking", un mécanisme de formation automatique de partie. Un joueur appuie sur le bouton pour jouer puis attend qu'un autre joueur le fasse



aussi, ils entrent alors dans la partie et se retrouvent l'un contre l'autre.

Le jeu sera également doté d'une intelligence artificielle que l'utilisateur pourra affronter s'il décide de jouer seul. Celle-ci aura la capacité de faire des coups stratégiques pour battre le joueur. Le bot se formera une intelligence à la fois par le choix des cartes jouées, par la position des actions (combat), et par le calcul des trajectoires ("Path Finder") afin d'évaluer les gains et les pertes grâce à un algorithme poussé.

La création de ce produit permet de faire acquérir de nombreuses compétences informatiques utiles à de futurs ingénieurs en informatique, membres du groupe de travail, mais aussi de créer un contenu ludique et amusant, réutilisable de multiples fois.

Animatch sera développé intégralement par notre équipe en utilisant principalement un moteur de jeu open source. La création de ce jeu nécessitera un matériel informatique ainsi qu'une licence pour pouvoir coder dans des conditions optimales. C'est pourquoi le budget de l'entreprise s'élèvera aux environs de 65 000 € (salaire de l'équipe compris). Le code, les graphismes, le son et la musique seront cependant entièrement réalisés par l'entreprise.

Ce cahier des charges sert à expliquer comment le jeu sera construit jusqu'à sa phase finale, depuis les concepts qui définissent le jeu jusqu'aux rôles de chaque membre de l'équipe. Il évoquera aussi les étapes et la planification du calendrier afin de respecter l'échéance souhaitée par l'école EPITA.



1.2 Cahier des charges fonctionnel

1.2.1 Origine et nature du projet

Pour que notre projet se déroule avec le moins d'obstacles possible et que nous soyons enthousiastes à l'idée de travailler dessus, nous sommes parvenus à un accord sur un jeu à partir de jeux de cartes.

Après avoir recueilli l'avis de l'ensemble des membres de notre entreprise, nous avons statué sur un mélange de deux types de jeux. Le jeu se reportera donc sur un style tactique en "deckbuilding" (construction stratégique de jeu de cartes), où l'importance de la création d'un deck (main de cartes) équilibré entre les différentes notions que comportera le jeu est primordiale afin de battre l'adversaire. Le déroulement de la partie fonctionne sur le principe de combat en tour par tour. L'idée serait que chacun joue ses tours sans que l'adversaire puisse répondre afin de créer une mécanique de jeu stratégique et intéressante, dépendante du placement du joueur et de ses capacités pour effectuer (ou pas) des actions. Cela crée alors une diversité conséquente de choix, qui dépendrait du deck créé au préalable.

Ce mélange de caractéristiques de jeux nous permettra de créer une dimension stratégique au jeu nécessaire afin que chaque joueur puisse acquérir un niveau plus élevé dans le jeu. Pour créer des decks optimisés, il faudra prendre en compte que les cartes ne sont pas toutes débloquées dès le départ ; il faudra les débloquent au fur et à mesure. Plus l'utilisateur jouera, plus il aura de possibilités pour complexifier son deck.



1.2.2 Objet de l'étude

L'entreprise s'est fixée pour objectif majeur de développer un jeu extrêmement divertissant, conçu pour captiver une grande variété de joueurs, quel que soit leur âge. L'ambition est de transformer chaque instant de jeu, partagé entre amis, en famille ou même seul (avec l'intelligence artificielle) en une expérience ludique et stratégique. Si cela est possible, le projet sera même disponible sur téléphone pour permettre au joueur de s'amuser dans toutes circonstances.

À travers ce projet, l'entreprise a l'intention de mettre en avant ses compétences en matière de créativité et de complexité, en témoignant d'un travail collaboratif efficace. Animatch constituera une véritable stimulation pour les joueurs, les poussant à élaborer des stratégies ingénieuses et à exercer leur réflexion pour parvenir à la victoire.

L'élaboration du jeu Animatch, s'il permet de divertir les joueurs, démontrera également la capacité de notre entreprise à travailler en équipe pour aboutir à un projet concret de jeu vidéo. A titre individuel, cette conception permettra à chaque membre d'apprendre le travail collectif, de se dépasser pour atteindre la qualité, et de se perfectionner en apprenant à utiliser Notion ou des plate-formes de partage de fichiers comme GitLab ou GitHub.



1.2.3 État de l'art

Animatch représente un mélange de plusieurs genres de jeux distincts. Trouver un jeu au style identique est donc très compliqué. Cependant, notre concept de base a été inspiré par le jeu *Worms* (1995) développé par Team17. L'idée d'un affrontement entre deux camps, avec un terrain destructible et de nombreuses possibilités de combat (avec des stratégies adaptables en fonction des choix de mouvement, d'attaque ou de dissimulation) forme la base de notre projet qui a suscité l'enthousiasme de tous.

Notre intention était de ne pas se limiter à un seul genre de jeu, mais plutôt de créer un mélange stratégique alliant divertissement et complexité tactique. Notre concept stratégique s'articule autour de la création de decks, similaire à *Clash Royale* (2016) ou *Hearthstone* (2014), permettant aux joueurs de choisir le style de jeu qu'ils souhaitent développer. Ainsi, un joueur peut opter pour un deck orienté vers l'offensive, axé sur la mobilité ou pour un deck défensif, selon ses préférences.

À cela s'ajoute un élément de collection : les cartes ne sont pas toutes débloquées au début, et il faudra les récupérer en jouant, permettant ainsi aux joueurs de progresser, au fil du temps, dans le jeu. Notre approche de la collection s'inspire du modèle de *Clash Royale*. Plus les joueurs s'investissent, plus ils accumulent des récompenses, ce qui leur permet d'améliorer, de débloquer et finalement d'utiliser différentes cartes.

Dans notre projet nous nous inspirons du concept de *Clash Royale*. Ce jeu a été lancé en 2016 sur les plate-formes mobiles, telles que les téléphones et les tablettes. Le succès de ce jeu, développé par une entreprise déjà renommée dans le monde du jeu mobile (Supercell), réside dans son système de jeu



unique. L'acquisition progressive de cartes, le passage d'arène en arène au fur et à mesure des victoires, procure aux joueurs une satisfaction qui les incite à continuer à jouer. En suivant ce principe, Supercell a régulièrement mis à jour le jeu en ajoutant de nouvelles cartes, arènes et compétitions, ce qui a permis de fidéliser un grand nombre de joueurs. Même après sept ans d'existence, le jeu continue de fonctionner avec succès. C'est sur ce modèle similaire que nous avons l'intention de développer Animatch, en mettant l'accent sur le système de cartes, notamment au début de l'expérience de jeu.

Animatch s'inspire aussi de 2 grands axes de *Worms* : l'idée du duel en 2D et celle d'un terrain destructible. *Worms* doit en effet son succès à son style de combat qui, pour l'époque, en 1995, changeait beaucoup par rapport aux jeux vidéo classiques. De plus, il se démarque par l'option de détruire le terrain (grâce au lanciers de grenade, de lance-roquettes ou autres), qui faisait et fait toujours le charme du jeu. C'est donc ces 2 principales idées que Animatch prendrait en compte afin de créer un gameplay amusant et différent des jeux déjà existants.



1.2.4 L'entreprise

Fondée au début du mois d'octobre 2023 au cours d'une session en sciences humaines, ILLUSION a vu le jour grâce à la détermination de deux membres fondateurs, M. Coureau et M. Priet. La vision partagée par ces membres a ensuite attiré l'expertise de M. Thubert, M. Rat et M. Dardonville, qui ont rejoint le projet avec enthousiasme.

ILLUSION est composée de cinq membres entièrement investis dans la réussite de cette entreprise. Chacun d'entre eux apporte une contribution unique à l'équipe, combinant leurs compétences et leur passion. Cette unité et cette diversité de talents sont les fondements sur lesquels repose ILLUSION et sont ce qui mènera ce projet à la réussite.

L'entreprise tire son nom de la capacité des jeux vidéo à créer une ILLUSION d'immersion totale pour le joueur, l'amenant à croire qu'il se trouve réellement dans un monde fictif. Le choix de ce nom a été déterminé à la suite d'un vote des membres de l'entreprise.

ILLUSION se spécialise dans le développement de jeux vidéo et utilise une combinaison de logiciels pour mener à bien son premier projet. L'entreprise s'appuie sur Notion pour centraliser, stocker et organiser les données, et Discord pour communiquer simplement. Elle utilise le moteur de jeu Unity, qui offre des capacités de développement en langage C# (C sharp), et est optimal pour le rendu 2D et pour le calcul des éléments de physique et pour la création d'animations nécessaires à la conception d'un jeu.

ILLUSION n'a pas encore conçu de projet, celui ci est le premier et l'équipe mettra tout en œuvre pour le concrétiser avec succès.



1.2.5 L'équipe

Adrien, chef de projet, 18 ans, actuellement élève à EPITA, passionné d'informatique depuis petit. L'univers de la création l'attire beaucoup (dessin, sculpture, dessin digital, production de musique). Il a commencé à créer ses premiers jeux avec Scratch puis Python en 5ème, puis en terminale un jeu de A à Z en Pygame (code, art, son), ce qui l'a ensuite amené à prendre cette voie dans laquelle il souhaite exprimer sa créativité.



Dans ce projet, Adrien sera utile pour la conception théorique (équilibrage du jeu), la conception graphique et sonore, et la programmation. Il saura aussi mener à bien le travail collectif de l'équipe en tant que chef de projet.



Josselin Priet, 17 ans, élève à EPITA, a commencé la programmation par Scratch et Unity. Au lycée, Josselin choisit les spécialités NSI, Mathématiques et HGGSP. Il a choisi cette dernière pour sa passion pour le journalisme. Durant son année de terminale, il a été à l'origine de 6 projets différents. Il est donc aussi passionné par la création de projets informatiques, que ce soit l'idée, l'algorithme, la partie code ou encore la partie débogage.

Dans ce projet, Josselin sera utile pour la partie algorithme, la conversion de concepts théoriques en programme et l'intelligence artificielle.



Guillaume Rat, actuellement élève à Epita, a suivi les spécialités Mathématiques, Physique et NSI durant le lycée. Il est passionné par la programmation et l'informatique en général, cela vient en partie de son père. Il en apprend sur l'informatique en le regardant travailler sur des réseaux informatiques. L'idée de créer un projet de 0 et de le voir complété est un bel objectif qu'il a toujours voulu réaliser mais par manque de moyens et de temps il n'en a pas encore eu l'occasion. C'est pourquoi il est impatient de pouvoir commencer à travailler sur ce projet.



Durant ce projet, Guillaume sera utile pour le stockage des données, la gestion de réseau du jeu mais aussi la création du site internet.



Hector Thubert, 18 ans, actuellement élève à Epita, passionné par la robotique, crée ses premiers robots grâce à des modules Arduino. Il a également créé plusieurs jeux vidéo dans différents langages durant ses années de lycée. Il choisit ainsi les spécialités Mathématiques, Physique et NSI. Cette dernière lui a permis de développer des compétences de programmation en Python et d'approfondir ses connaissances en informatique. Étant très curieux, cela l'aide à développer son sens de l'autonomie et son esprit d'initiative.

Dans ce projet, Hector sera utile pour créer la dynamique au sein du groupe, pour la partie graphisme mais aussi pour la création du site internet.



Guilhem Dardonville, actuellement élève à Epita, a suivi les spécialités NSI, Mathématiques et Physique au cours du lycée. Ces spécialités lui ont permis d'en découvrir davantage sur le monde actuel et son fonctionnement. Pour lui, les travaux de groupe sont importants pour la cohésion d'équipe et la répartition des tâches. C'est pourquoi, réaliser un projet de 0 est une expérience très intéressante et remplie de surprises qui ne peuvent qu'apporter du bien à la maturité du groupe.



Dans ce projet, Guilhem sera utile pour développer la physique du jeu, mais aussi pour la partie algorithmique et l'implémentation des concepts dans le jeu.



1.2.6 Répartition des tâches

<i>Tache</i> \ <i>Poste</i>	Adrien	Josselin	Guillaume	Hector	Guilhem
Cahier des charges	R				S
Site internet	S			R	
Conceptualisation		R			S
Graphismes	R			S	
Son / Musique	R			S	
Physique			S		R
Réseau / Serveurs			R		S
Intelligence Artificielle		S		R	
Menu	S			R	
Traduction			R		S
Équilibrage	S	R			

R = Responsable

S = Suppléant



1.2.7 Avancement et planification

(Prévision approximative)

<i>Tche</i> \ <i>Soutenance</i>	22 janvier (1)	18 mars (2)	17 juin (3)
Cahier des charges	90%	95%	100%
Site internet	60%	80%	100%
Conceptualisation	80%	90%	100%
Graphismes	40%	70%	100%
Son/Musique	60%	80%	100%
Physique	20%	60%	100%
Réseau/Serveurs	30%	50%	100%
Intelligence artificielle	60%	80%	100%
Menu	20%	50%	100%
Traduction	0%	0%	100%
Équilibrage	10%	40%	100%



1.3 Conclusion

Ainsi, Animatch est un projet de jeu vidéo mené par l'entreprise ILLUSION. Ce jeu allie des éléments de jeux de cartes et de combat en 2D sur un terrain destructible, conçu pour Windows avec une possible extension vers Android et iOS. L'objectif est de fournir un divertissement amusant pour les joueurs et d'enrichir les compétences et connaissances en informatique de l'équipe.

Le cahier des charges guidera le développement en définissant les concepts, les objectifs, les contraintes temporelles et budgétaires. Inspiré par *Worms* et *Clash Royale*, Animatch promet une expérience immersive en multijoueur ou en solo contre une intelligence artificielle.

L'entreprise ILLUSION est déterminée à réussir ce premier projet et compte sur l'engagement et les compétences de chaque membre de son équipe. Elle se spécialise dans le développement de jeux vidéo en utilisant des outils tels que Notion, Discord, et le moteur de jeu Unity.

Le projet Animatch est un défi passionnant pour l'entreprise, et nous sommes convaincus que le résultat final sera un jeu divertissant et innovant. Nous sommes impatients de voir ce projet prendre forme et de le présenter au public. Ce cahier des charges servira de guide tout au long du processus de développement.

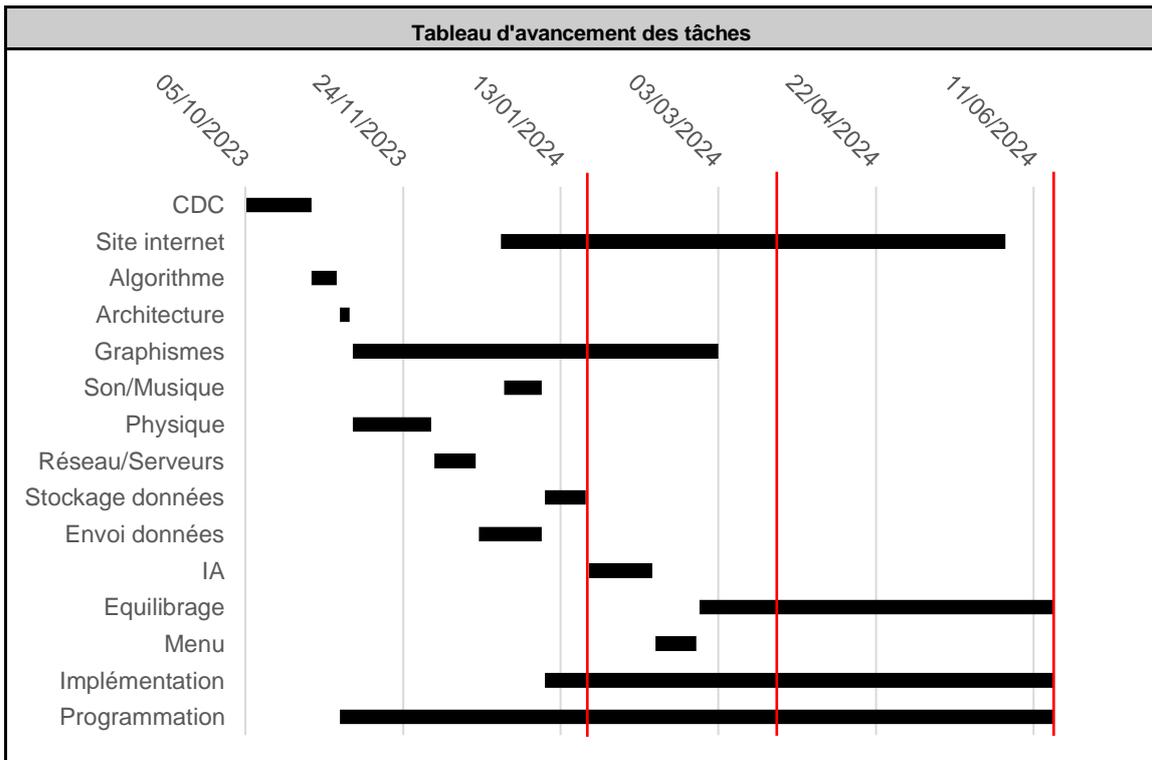
Nom du groupe :	ILLUSION
Nom du projet :	Animatch

Noms des membre :			
Nom :	Prénom :	Login :	Classe :
Coureau	Adrien	adrien.coureau	D2
Priet	Josselin	josselin.priet	D2
Rat	Guillaume	guillaume.rat	D2
Hector	Thubert	hector.thubert	D2
Dardonville	Guilhem	guilhem.dardonville	D2

Type de jeu :				
Action/Aventure	Battle Royale	Beat them all	Combat	Simulation
FPS	MMORPG	MOBA	Party Games	Survival Horror
Plateforme	Puzzles	Reflexion	Rogue Like	TPS
RPG	RTS	Sandbox	Shoot them up	Course
Autre :	Tactique, Tour par tour, Deckbuilding			

Caractéristiques générales du jeu :				
IA :	Errer	Attaquer	S'échapper	"Path Finder"
Multijoueurs :	Coopé	Battle (2-4)	Massif	
Réseau :	P2P	Lan	Online	
Caractéristiques graphiques :				
Dimension :	2D	3D	Autres :	
Particularités graphiques :	Stéréoscopie	AR	VR	
graphiques :	Perso	Custom	Existant	
Précisions :				
Caractéristiques sonores :				
Musique :	Perso	Custom	Existant	
FX :	Perso	Custom	Existant	
Précisions :				
Autres caractéristiques :				
Site Web :	Perso	Custom	Préfabriqué	

Répartition des tâches : deux personnes par tâche: (R)esponsable & (S)uppléant					
Tâches	adrien.coureau	josselin.priet	guillaume.rat	hector.thubert	guilhem.dardonville
CDC	R	S	S	S	S
Site internet			S	R	
Algorithme		R			S
Architecture	R	S			
Graphismes	R			S	
Son/Musique	R	S			
Physique	S				R
Réseau/Serveurs			R		S
Stockage données			R		S
Envoi données		S	R		
IA		R		S	
Equilibrage	S	R			
Menu	S			R	
Implémentation		R			S
Programmation	R	R	R	R	R

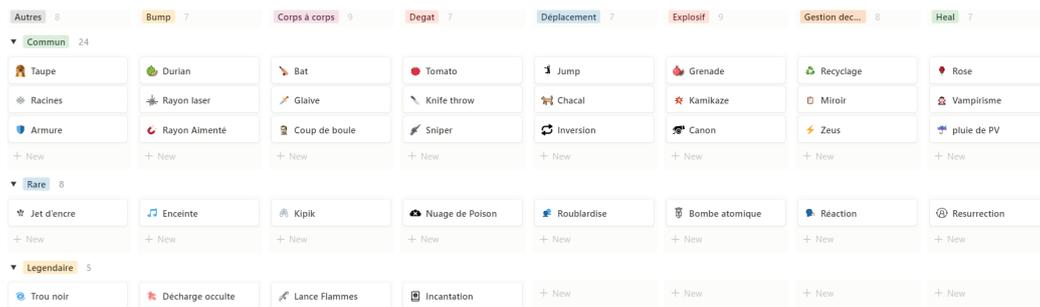




3 Tâches collectives

Suivant la création du cahier des charges fonctionnel, de nombreuses tâches collectives ont été nécessaires afin de conceptualiser le jeu, précédant la programmation. Nous avons commencé la modélisation d'Animatch sur le site internet Notion, en créant 37 cartes, 10 animaux et le déroulement d'une partie.

Les caractéristiques de chaque carte et de chaque animal y sont également présentes : nom, coût, description, distance, dégâts et taille de l'explosion pour les cartes ; et nom, Animax, passif et points de vie pour les animaux.



Cette étape fut cruciale pour avoir une idée précise du projet et le lancer sur la bonne voie. Nous avons ensuite réparti chaque membre du groupe principalement sur des tâches précises :

- Adrien sur les graphismes et la musique
- Josselin sur la programmation
- Guillaume sur le multijoueur
- Hector sur le site internet et sur la programmation
- Guilhem sur la programmation



4 Tâches individuelles

4.1 Adrien Coureau

4.1.1 Gestion du groupe

En tant que chef de groupe du projet Animatch, la gestion des membres du groupe est très importante afin de s'assurer que le travail est réalisé dans les délais prévus.

J'ai été élu chef de groupe à l'unanimité car j'avais la vision la plus précise du projet en tête. N'ayant jamais eu à diriger une équipe auparavant, il m'a fallu gagner en autorité pour motiver l'ensemble des membres. J'ai su progresser dans ce domaine, ce qui m'a pris une grande partie de mon temps de travail. Grâce au serveur Discord créé par Guillaume, j'ai pu organiser des appels de groupe, rédiger des messages d'annonce et prendre des décisions pour mener à bien le projet et répartir les tâches convenablement. J'ai passé du temps à organiser des appels et des réunions pour se motiver mutuellement et constater le progrès effectué.

4.1.2 GitHub

Nous avons utilisé la plateforme GitHub ainsi que le logiciel GitHub Desktop pour échanger facilement nos fichiers. J'ai créé le répertoire et veillé à ce que la version Unity du projet et des ressources soit légères et simples à exploiter. Depuis sa création, le projet a donc reçu plus d'un millier de "commits" qui concernent le jeu en lui-même, les graphismes, le site internet, le cahier des charges ou encore le rapport de projet.



4.1.3 Architecture du projet

Le choix le plus compliqué a été celui du moteur de jeu. Nous hésitions entre Godot et Unity, chacun possédant ses avantages et inconvénients.

Dans un premier temps, nous avons opté pour Godot car il semblait simple à prendre en main et qu'il était open-source. Cependant, au bout d'une dizaine d'heures de programmation, nous avons rencontré deux problèmes majeurs :

- Il est très difficile de trouver des ressources en ligne pour nous aider face à des problèmes de programmation.
- L'exportation sur mobile d'un jeu codé sur Godot spécifiquement en C# est impossible à ce jour. Cela a été rédhibitoire car nous voulons porter notre jeu sur téléphone et tablette à terme, pouvant se jouer sans clavier pour attirer un plus large public et le rendre plus accessible à tout moment.

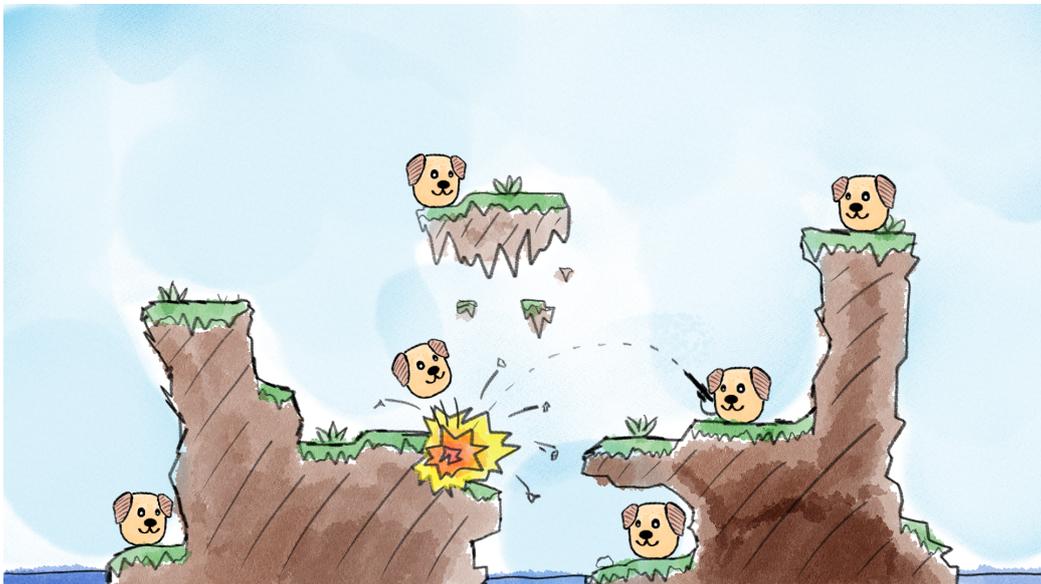
Nous avons donc opté pour Unity, qui n'avait aucun de ces deux problèmes et dont l'intégration de C# est plus intuitive. J'ai donc modélisé l'architecture des fichiers, les prémices du projet et les paramètres de base d'Animatch.

4.1.4 Graphismes

La majorité de mon temps fut dédiée à la conceptualisation et à la réalisation des graphismes du jeu Animatch. En effet, la totalité des ressources graphiques est faite par mes soins à l'aide d'une tablette graphique, la XP-PEN Deco Pro Gen2, et du logiciel Clip Studio Paint. J'ai également utilisé PaintDotNet pour des ajustements mineurs.

J'ai beaucoup progressé avec l'utilisation de cette tablette graphique achetée notamment pour le jeu, et j'ai perfectionné mon style et ma charte graphique.

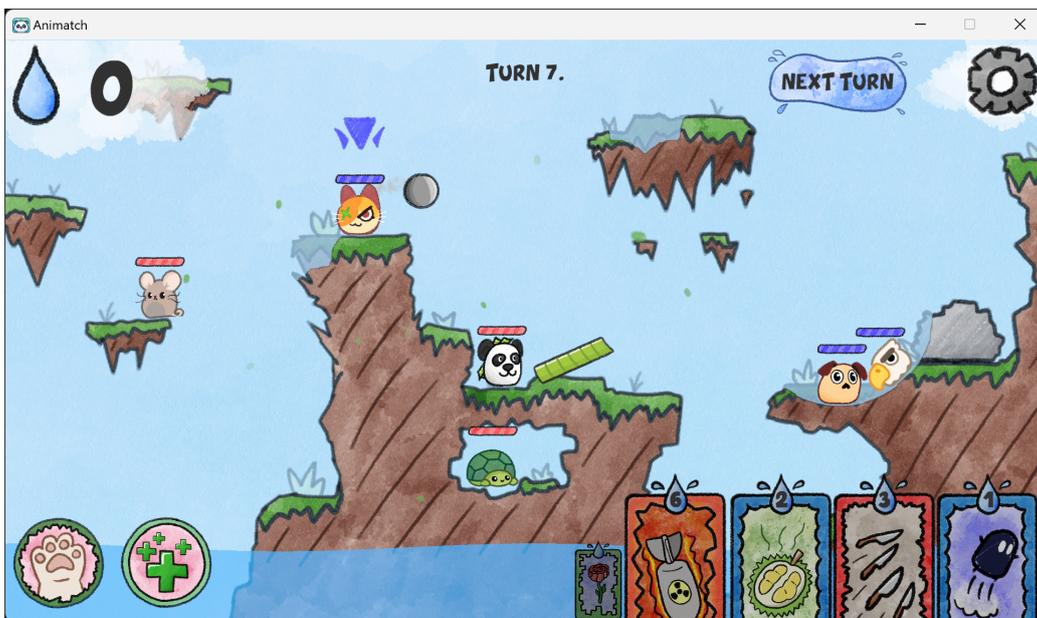
Après être passé par plusieurs phases comme celle du dessin en vecteurs, j'ai choisi de réaliser le projet avec un style "croquis", qui représente bien l'univers de notre jeu, plutôt enfantin et accessible. J'ai donc commencé par dessiner un premier prototype du jeu :





4 TÂCHES INDIVIDUELLES

Ensuite, j'ai donc créé les éléments un à un : Animaux, Animax, passifs, cartes, projectiles des cartes, icônes, logos, interface, ATH, boutons, maps et autres. Cela fait à ce jour 90 éléments au total, dont 6 animaux, 8 cartes et une map. Chaque élément me prenait de 30 minutes à plusieurs heures comme la map et le menu. Je voulais faire respecter l'univers et la cohérence d'Animatch, c'est-à-dire un jeu simple, accessible mais aux mécaniques profondes. La cohérence du jeu passait aussi par le fait que tous les éléments présents dans le jeu sans exception sont l'oeuvre d'un travail perfectionniste à mon image. Voici à quoi ressemble le jeu final :





4.1.5 Musiques et sons

Étant compositeur de musiques digitales depuis plus de 5 ans, j'ai composé la soundtrack d'Animatch sur FL Studio. Je suis surtout fier de la musique du menu qui représente exactement l'image d'Animatch que j'ai en tête, en lui donnant un petit air simple et légèrement dissonant pour créer un contraste intéressant. Les 5 musiques du jeu sont accompagnées par des bruitages et sons que j'ai également produit avec ce logiciel, ma bouche et divers objets. J'ai intégré ces musiques et ces sons dans le fonctionnement du jeu.

4.1.6 Animations

Pour donner vie au jeu, une étape cruciale selon moi a été la création d'effets visuels pour fluidifier l'expérience de jeu. J'ai ajouté de nombreuses particules comme des feuilles qui tombent sur l'écran, des traînées de projectiles et des explosions style "cartoon". J'ai également créé un effet de transition au style d'Animatch pour renforcer le mouvement. Finalement, j'ai ajouté de l'eau qui se déplace pour donner un effet de danger et faire vivre l'île sur laquelle sont placés les animaux.

4.1.7 Menu, sélection

J'ai créé le menu d'Animatch ainsi que ses graphismes, un travail dont je suis fier car il vient de l'idée originale de conceptualiser le menu sous la forme d'un arbre. J'ai également participé avec Hector à la création de la sélection des animaux et des cartes pour permettre bien plus de possibilités. Appuyer sur les boutons du menu nous fait rentrer dans le feuillage de la branche pour davantage d'immersion (cf. Annexes).



4.2 Josselin Priet

4.2.1 Algorithme

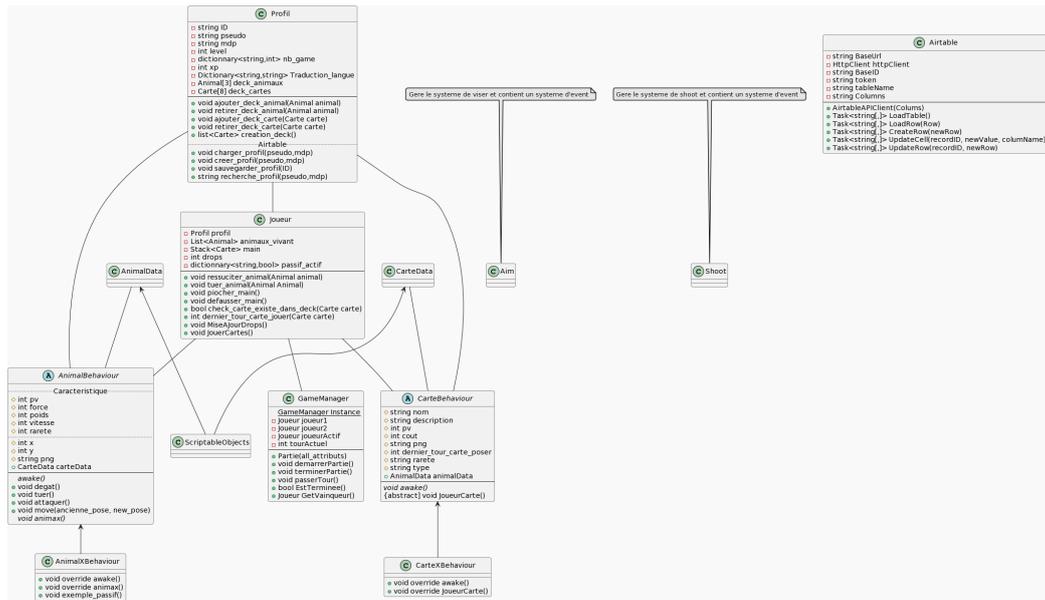
Conception de la base des animaux, cartes, projectiles, maps, etc : La participation à l'élaboration des structures de données nécessaires pour représenter les différents éléments du jeu, notamment les animaux, les cartes, les projectiles et les cartes.

- Conception de la gestion : des animaux, des cartes, des projectiles et des maps.
- Conception du système de tour : Travail effectué sur la mise en place du système de tour, permettant aux joueurs d'interagir avec les éléments du jeu de manière séquentielle et ordonnée.
- Optimisation : Contribution à l'optimisation du code et des performances du jeu pour garantir une expérience fluide et réactive.
- Commentaires dans le code : Veiller à ce que le code soit bien documenté en ajoutant des commentaires explicatifs pour faciliter la compréhension et la maintenance.
- Organisation des fichiers de scripts : Organisation logique et structurée des fichiers de scripts pour une gestion efficace du code source.



4.2.2 UML

Réalisation un diagramme UML pour représenter la structure et les relations entre les différentes classes et composants du jeu, offrant ainsi une vue d'ensemble claire et précise de l'architecture du projet.





4.2.3 Code

- Scriptable Object : Utilisation des Scriptable Objects pour stocker et gérer les données de jeu de manière modulaire et réutilisable.
- Classe : Mise en place de classes pour encapsuler le comportement et les attributs des différents éléments du jeu, favorisant ainsi la modularité du code telle que "TortueBehaviour" ou "TomateBehaviour", etc.
- Classe abstraite : Utilisation de classes abstraites pour définir des comportements que d'autres classes peuvent étendre et personnaliser selon leurs besoins spécifiques tels que "AnimalBehaviour", "CarteBehaviour", etc.
- Les comportements : Définition et implémentation de comportements pour les différents éléments du jeu, tels que les mouvements des animaux et les effets des projectiles.
- Héritage : Utilisation de l'héritage pour partager et réutiliser du code entre les différentes classes, favorisant la cohérence et l'efficacité du développement.
- Game Objects : Manipulation des Game Objects dans Unity pour représenter et interagir avec les éléments du jeu, en utilisant les composants et les scripts pour définir leur comportement et leur apparence.
- API : Exploration et utilisation de l'API Airtable pour sauvegarder les



données des joueurs, enrichissant ainsi l'expérience des joueurs.

- Singleton : Mise en place de Singletons pour garantir qu'une seule instance de certaines classes soit disponible dans tout le jeu, assurant ainsi la cohérence et la gestion des ressources.
- Système de collision : J'ai implémenté un système de collision pour gérer les interactions entre les différents objets du jeu, assurant ainsi un comportement réaliste et cohérent.
- Système de Rigidbody : J'ai intégré un système de Rigidbody pour gérer la physique des objets du jeu, leur permettant de réagir aux forces extérieures telles que la gravité et les collisions.
- Système d'affichage : J'ai développé un système d'affichage pour rendre les éléments du jeu visibles à l'écran, assurant ainsi une présentation cohérente et esthétique.
- Uniformisation des éléments : J'ai uniformisé les éléments du jeu en utilisant des Scriptable Objects pour centraliser les données et les comportements, facilitant la gestion et la modification des éléments du jeu.
- Ajout du GameManager : J'ai créé et codé majoritairement le fichier GameManager, qui centralise les actions globales d'une partie, comme le système de spawn, le système de tour, etc.



- Ajout du système de spawn : J'ai codé la majorité du système permettant de poser les animaux, constituant ainsi la phase de départ lors d'une partie d'Animatch.
- Système de destruction du terrain : J'ai ainsi pu créer et perfectionner l'un des éléments majeurs d'Animatch, la destruction de terrain, en utilisant des bibliothèques telles que "D2Destructible", permettant ainsi une réalisation plus rapide et simplifiée, et donc de tenir les délais.
- Ajout des 8 cartes présentes dans le jeu : J'ai créé les cartes Tomate, Bat, Bombe, Saut, Rose, Durian, Couteaux et Canon en utilisant des objets scriptables et des classes abstraites déjà vues précédemment.
- Ajout des 6 animaux : J'ai également créé les 6 animaux en suivant à la lettre l'UML déjà réalisé plus tôt. Je me suis aussi occupé de leur physique, de la gestion des collisions et de leur clignement d'œil régulier.
- Ajout des Animax et passifs : J'ai implémenté les Animax et passifs de chaque animal (soit 12 au total), ce qui a permis d'augmenter la stratégie et les possibilités pour les utilisateurs.
- Ajout des projectiles et de leurs effets : En utilisant l'UML et les classes, j'ai pu uniformiser un processus de création de projectiles, permettant ainsi leur création de manière rapide et simple.



- Rectification des bugs : J'ai également pu régler la plupart des bugs liés à mes créations.

- Ajout du système de visée : J'ai développé le système de visée malgré la présence de bugs au début de sa création. J'ai dû modifier l'algorithme consacré et j'ai séparé les actions en trois phases :

- Phase d'attente de clic sur l'animal
- Phase de visée avec les pointillés
- Phase de tir utilisant la classe ProjectileBehaviour

- Ajout du système de visée au corps à corps : De même que pour la visée à distance, j'ai réutilisé la même fragmentation pour gérer ce système.



4.3 Guillaume Rat

4.3.1 Communication

Pour améliorer la communication au sein de l'équipe de développement, nous avons instauré un serveur Discord dédié.

Ce canal favorise l'échange d'idées, le partage de ressources et la coordination des efforts de manière plus efficace que les conversations individuelles. Cette initiative a considérablement optimisé la collaboration et la productivité de l'équipe. Auparavant, nous ne disposions que d'un seul fil de discussion, ce qui risquait d'entraver la transmission d'informations et de noyer les éléments importants.

C'est pourquoi la mise en place de ce serveur Discord s'est avérée essentielle, offrant à chaque membre de l'équipe un espace dédié pour partager des informations cruciales.

En outre, une section spécifique a été réservée au gestionnaire de versions Git, permettant ainsi à chacun de suivre en temps réel les progrès réalisés par ses pairs, évitant ainsi toute perte de travail.

4.3.2 Multijoueur

Des progrès significatifs ont été accomplis dans l'intégration du mode multijoueur grâce à l'utilisation des outils Photon et PUN 2.

Ces technologies ont jeté les bases nécessaires pour le multijoueur, malgré quelques défis techniques persistants. Actuellement, la fonctionnalité multijoueur n'est pas encore opérationnelle en raison de problèmes tels que la connexion au serveur qui n'est pas encore totalement fonctionnelle.

Néanmoins, une infrastructure de base a été mise en place, permettant



la synchronisation des personnages et des projectiles grâce à l'ajout de composants Photon. Par ailleurs, des travaux ont été entrepris sur la gestion des sessions de jeu, explorant des mécanismes d'appariement des joueurs pour garantir une expérience multijoueur fluide et équilibrée. L'objectif est que les joueurs puissent simplement appuyer sur "Jouer" et être automatiquement dirigés vers une partie appropriée. La suite du développement du multijoueur serait donc de se concentrer sur la relation, serveur / joueur afin de pouvoir faire fonctionner le mode.

Dans la continuité de ceci, dès que la relation au serveur sera effectuée, la difficulté du multijoueur sera passée et nous pourrions ajouter plus d'options reliées à cela.

Une idée serait d'intégrer un système de messages afin de pouvoir communiquer dans la partie, cela étant réalisable avec les outils utilisés. D'autres options pourraient être envisagées si la relation serveur est réussie sous peu de temps, après quoi cela serait donc considéré comme un bonus.

4.3.3 Launcher

Dans le cadre de notre stratégie de développement, un Launcher provisoire a été conçu pour faciliter l'accès aux parties multijoueurs.

Bien qu'il ne soit pas encore entièrement fonctionnel, ce Launcher permet aux joueurs de se connecter à des "salles" de jeu pour participer à des sessions multijoueurs.

Son objectif principal est de diviser les joueurs en fonction de leurs préférences de jeu en les redirigeant vers des parties correspondantes. Actuellement, il permet aux joueurs, après avoir cliqué sur "Se connecter", de re-



joindre une partie existante ou d'en créer une.

Ainsi, le joueur est assuré de se connecter à une partie, même s'il choisit de créer une salle et d'attendre qu'un autre joueur la rejoigne.

Ce Launcher constitue un élément essentiel de notre infrastructure multijoueur en cours de développement, visant à offrir une expérience utilisateur fluide et enrichissante à l'avenir.

4.3.4 Suite Multijoueur

Après avoir évalué les défis associés à l'intégration du multijoueur avec Photon j'ai créé un mode 2 joueurs car le multijoueur par l'utilisation de Photon était trop complexe à mettre en place. Plusieurs problèmes techniques persistaient, notamment des affichages qui ne fonctionnaient pas correctement et la création des joueurs ainsi que leurs decks qui ne se synchronisaient pas correctement entre les clients.

Cette décision a nécessité des modifications substantielles du code, initialement conçu pour le jeu contre un ordinateur, ce qui a demandé un investissement de temps considérable.

Ce mode 2 joueurs permet à deux joueurs de s'affronter dans une même instance, offrant ainsi la possibilité de jouer une véritable partie d'Animatch. Pour cela, il a été nécessaire de gérer simultanément les deux joueurs et de définir clairement les priorités de jeu, c'est-à-dire qui joue à quel moment.

J'ai également mis en place un système de gestion des decks pour chaque joueur, garantissant que chacun dispose de cartes distinctes. Ainsi, lorsque leur tour arrive, les joueurs peuvent jouer leurs propres cartes sans interférer avec le deck de leur adversaire.



4.3.5 Affichage de la Partie

J'ai développé un affichage spécifique pour les parties en mode 2 joueurs, qui intègre un système de tour sophistiqué. Ce système indique de manière claire et précise quel joueur doit jouer à quel moment, assurant ainsi une alternance fluide entre les participants. Chaque tour est visuellement marqué, permettant aux joueurs de suivre facilement le déroulement de la partie sans confusion.

En plus de ce système de tour, j'ai ajouté des textes explicatifs détaillés pour chaque animal, décrivant leurs capacités spéciales appelées Animax, ainsi que leurs compétences passives. Ces explications fournissent aux joueurs des informations essentielles sur les compétences uniques de leurs animaux, ce qui améliore leur compréhension des mécanismes de jeu et leur permet de mieux élaborer des stratégies. Cette documentation en jeu aide les joueurs à tirer pleinement parti des capacités de leurs animaux, enrichissant ainsi l'expérience de jeu et rendant chaque partie plus engageante et stratégique.

4.3.6 Choix des Cartes

J'ai également mis en place une fonctionnalité d'assistance pour la création du choix des cartes, qui est applicable aussi bien au mode solo qu'au mode 2 joueurs. Cette fonctionnalité permet aux joueurs de personnaliser leur expérience de jeu en sélectionnant les cartes qu'ils souhaitent utiliser dans leur deck.

Pour le mode solo, les joueurs peuvent choisir parmi une variété de cartes disponibles pour créer un deck qui correspond à leur style de jeu et à leurs stratégies préférées. Ils peuvent ainsi affiner leurs compétences et expérimen-



ter différentes combinaisons de cartes pour découvrir ce qui fonctionne le mieux contre les bots.

Pour le mode 2 joueurs, cette fonctionnalité devient encore plus cruciale. Chaque joueur a la possibilité de créer un deck unique et personnalisé, garantissant que les deux decks sont distincts et équilibrés. Pendant la partie, le système veille à ce que chaque joueur utilise uniquement les cartes de son propre deck, évitant toute confusion ou interférence avec le deck de l'adversaire. Ce niveau de personnalisation enrichit l'expérience de jeu en offrant une plus grande diversité de stratégies et en permettant des affrontements plus dynamiques et intéressants entre les joueurs. (cf. Annexes).

4.3.7 Améliorations du Code

Le code a été considérablement amélioré grâce à l'introduction de nouvelles classes, ce qui a permis une gestion beaucoup plus efficace et structurée des informations de la partie. Ces nouvelles classes ont été conçues pour gérer divers aspects du jeu, allant de la détermination du vainqueur à la gestion des différents scénarios de jeu.

L'une des améliorations clés a été la capacité à distinguer clairement entre les différents modes de jeu, que le joueur soit en train d'affronter un bot ou un autre joueur humain. Ces classes permettent de gérer les règles spécifiques et les conditions de victoire adaptées à chaque type de partie. Par exemple, elles facilitent la mise à jour en temps réel des scores, la vérification des conditions de victoire et la gestion des états de jeu intermédiaires.

De plus, ces classes améliorent la modularité et la maintenabilité du code. En encapsulant les fonctionnalités dans des classes distinctes, il devient plus



facile de mettre à jour et d'ajouter de nouvelles fonctionnalités sans perturber le reste du système. Cela inclut la gestion des interactions entre les joueurs, la synchronisation des actions et des événements de jeu, ainsi que le suivi des statistiques de chaque joueur.

En résumé, l'introduction de ces nouvelles classes a non seulement rationalisé le processus de détermination du vainqueur et la gestion des différents scénarios de jeu, mais a également renforcé la robustesse et la flexibilité du code, permettant une évolution continue et une amélioration de l'expérience de jeu globale.

4.3.8 Conclusion

En rétrospective, ce projet a été une expérience profondément enrichissante et formatrice. La mise en place du serveur Discord dédié a transformé la communication au sein de notre équipe, permettant une collaboration plus fluide et une meilleure coordination des efforts. Cela a non seulement optimisé notre productivité, mais a également renforcé notre esprit d'équipe.

L'intégration initiale des outils Photon et PUN 2 pour le mode multijoueur a été un véritable défi. Malgré des efforts soutenus, les problèmes techniques persistants tels que les affichages défectueux et les difficultés à synchroniser les joueurs et leurs decks ont rendu cette tâche particulièrement complexe. Cette situation a été une source de frustration et de déception, car notre objectif était de créer une expérience multijoueur en ligne aboutie.

Cependant, cette contrainte m'a conduit à explorer d'autres solutions, et j'ai réussi à développer un mode 2 joueurs local. Ce processus a nécessité une refonte significative du code, initialement conçu pour des parties joueur



4 TÂCHES INDIVIDUELLES

contre ordinateur. Grâce à ces efforts, j'ai pu mettre en place un système de tour clair et ajouter des textes explicatifs pour les capacités spéciales des animaux (Animax) et leurs passifs, enrichissant ainsi l'expérience de jeu.

La création d'une fonctionnalité d'aide pour le choix des cartes, applicable au mode solo comme au mode 2 joueurs, a permis de personnaliser encore davantage l'expérience de jeu. De plus, l'introduction de nouvelles classes dans le code a amélioré la gestion des informations de la partie, facilitant la détermination des vainqueurs et la gestion des différents scénarios.

En conclusion, bien que le regret de ne pas avoir pu réaliser un véritable mode multijoueur en ligne soit présent, je suis fier des solutions alternatives que j'ai mises en place. Elles ont permis de proposer une expérience de jeu riche et personnalisée. Cette expérience m'a non seulement permis de surmonter des obstacles techniques, mais elle a également renforcé mes compétences en développement et en gestion de projets complexes. Je suis déterminé à continuer à améliorer ce projet et à explorer de nouvelles possibilités pour offrir une expérience de jeu encore plus aboutie à l'avenir.



4.4 Hector Thubert

4.4.1 Conception des animaux

Durant cette période, j'ai continué de confectionner le site internet, mais j'ai aussi apporté de l'aide à Josselin et Guilhem sur l'avancement du code du jeu vidéo. Je vais donc vous parler de mon implication durant les semaines passées sur Animatch.

J'ai aidé à la création d'animaux grâce au "moule" fait par Josselin mais aussi de leur pouvoir. Cela m'a permis de rentrer un peu plus dans le code du jeu et pouvoir donner ma pâte au jeu vidéo

4.4.2 Le site Internet

La conception initiale de celui-ci avait été réalisée avant la première soutenance. Ce sur quoi j'ai vraiment travaillé est le design, c'est-à-dire une refonte totale du CSS du code en ajoutant des reliefs au logo et une barre située en haut du site pour pouvoir naviguer entre chaque page.

Améliorations apportées au positionnement des éléments HTML :

J'ai optimisé le positionnement des éléments HTML dans une page web en utilisant diverses propriétés CSS telles que display, flexbox, align-items, justify-content, text-align, et margin, pour pouvoir positionner chaque chose utile aux bons endroits.

Enrichissement par des animations avec jQuery : J'ai enrichi l'expérience sur le site en intégrant des animations à différents éléments HTML grâce à jQuery. Ces animations ont été appliquées aux liens de la barre de navigation et aux boutons pour animer la page.

Personnalisation des polices de caractères : J'ai incorporé une po-



lice de caractères personnalisée dans le CSS via @font-face, ce qui a permis d'apporter une touche différent à la typographie du site.

Centrage optimal des items : Pour assurer une présentation visuellement équilibrée, j'ai fait un centrage précis de chaque item tant horizontalement que verticalement sur chaque page, en utilisant les fonctionnalités de Flexbox pour obtenir les bons résultats.

Disposition harmonieuse des éléments HTML : J'ai fait une mise en page fluide en positionnant les éléments HTML tels que les images et le texte, les disposants soient : côte à côte ou en superposition, à l'aide de plusieurs techniques de positionnement et de mise en page.

J'ai dû mener un sondage au sein de notre groupe pour connaître tous les logiciels qu'ils utilisaient afin de les intégrer dans la page source du site web. De plus, j'ai créé un bouton "DOWNLOAD" qui télécharge une version .exe de notre jeu ainsi que le cahier des charges technique et fonctionnel.

Il est également important de mentionner que la page la plus récente du site web est 'ANIMAUX'. Cette page m'a demandé plus de temps que les autres, car je souhaitais que les personnages présents sur cette page soient animés. J'ai donc dû trouver une solution en JavaScript pour animer ces éléments.

Durant mon année de première, en spécialité NSI, j'ai pu découvrir le HTML et le CSS. J'ai donc été motivé pour mettre à profit mes compétences dans ce domaine. Ce projet de site web m'a offert l'opportunité d'améliorer mes connaissances. En effet, j'ai appris le code JavaScript uniquement pour ce projet, mais je sais que cela me sera utile pour le futur.



4.4.3 Partie Code

Durant les dernières semaines pour la réalisation d'Animatch, je me suis penché sur la conception du **menu** de notre jeu et du **mode solo**

Le Menu a été réalisé en collaboration avec Adrien, qui s'est occupé de tous les graphismes. J'ai donc commencé par instancier un AudioManager pour pouvoir régler le volume des musiques du jeu. Par la suite, après de nombreuses séances de débogage, je me suis attaqué à la scène des options du jeu. Cette scène Unity permet de modifier le volume des musiques de notre jeu grâce à un slider, qui a été très difficile à mettre en place.

Ensuite, sur le menu, j'ai ajouté plusieurs fonctionnalités comme la possibilité de choisir sa propre équipe d'animaux. Cela se fait grâce à de multiples prefabs qui se mettent à jour en appuyant sur des boutons. Cependant, après la réalisation de cette fonctionnalité, Guillaume, qui travaille sur le multi-joueur, m'a indiqué qu'il fallait que je refasse l'interface pour permettre de choisir les animaux des deux équipes séparément. J'ai donc créé une nouvelle scène pour sélectionner également les animaux de l'équipe numéro 2. Après avoir corrigé plusieurs bugs, l'interface pour choisir ses animaux fonctionne désormais correctement. Côté code, j'ai dû regarder de nombreux tutoriels en ligne pour comprendre comment enregistrer une équipe d'animaux. J'ai finalement trouvé la solution en utilisant la fonction PlayerPrefs qui permet de stocker des informations.

Sur le menu, j'ai également ajouté avec Guillaume une nouvelle interface permettant de choisir ses cartes dans le jeu. Cette interface a pris moins de temps à développer que celle pour les animaux, car j'ai simplement dû copier



l'interface du choix des animaux. Cependant, au niveau du code, cela n'a pas été aussi simple. Nous avons dû complètement revoir le système de cartes déjà implémenté pour qu'il soit compatible avec notre système de sauvegarde. (cf. Annexes)

Le Mode Solo de notre jeu était le premier mode implémenté dans Animatch. Au départ, il ne comportait pas de bot, et les joueurs pouvaient utiliser des cartes avec n'importe quel animal, qu'il fasse partie de leur équipe ou non. Par la suite, j'ai donc intégré un "bot" dans le jeu. Ce bot n'était pas très compliqué à créer : il devait simplement pouvoir faire apparaître des animaux à des positions cohérentes et tirer.

La première étape, qui consistait à gérer le "spawn" des animaux a été très rapide, j'ai pu facilement relier mon code au jeu. Cependant, la fonctionnalité de tir du bot a été beaucoup plus compliquée à mettre en place.

Premièrement, il fallait déterminer quelle carte le bot allait utiliser pour tirer. L'équipe d'Animatch a décidé que le bot tirerait uniquement des tomates. Le premier script que j'avais écrit ne fonctionnait pas comme prévu. En fait, il fonctionnait mais ne pouvait pas s'intégrer correctement avec le code déjà en place.

Mon nouveau script comprenait les éléments suivants : une fonction GetNearest pour trouver le personnage le plus proche du bot, une fonction pour déterminer la position de cette cible par rapport au bot, et une fonction pour tirer soit à gauche, soit à droite en fonction de cette position relative.

Après de longues heures de réglages et de peaufinage, le bot était enfin prêt. Je l'ai alors présenté à l'équipe d'Animatch, qui m'a demandé d'apporter encore quelques modifications.



4.5 Guilhem Dardonville

4.5.1 Correction linguistique

Mon implication dans le projet s'est distinguée par ma contribution à la correction des erreurs de français, un aspect essentiel pour assurer la qualité et la crédibilité de notre travail. En examinant attentivement le cahier des charges fonctionnel, la présentation de notre jeu vidéo aux investisseurs et le contenu du site web, j'ai identifié et corrigé les fautes linguistiques, les incohérences et les formulations ambiguës. Cette démarche méticuleuse a permis d'assurer la clarté et la précision de nos communications écrites, renforçant ainsi la crédibilité de notre projet aux yeux de nos collaborateurs et des parties prenantes externes.

Grâce à cette attention particulière portée aux détails linguistiques, nous avons pu transmettre nos idées de manière efficace et professionnelle, renforçant ainsi la compréhension et l'adhésion à notre vision commune.

4.5.2 Système de la barre de vie

Un élément crucial de notre jeu est la barre de vie, qui permet aux joueurs de surveiller l'état de santé de leur personnage. En collaboration avec les graphismes réalisés par Adrien, nous avons réussi à créer une barre de vie distincte et esthétiquement adaptée à notre univers de jeu. En utilisant les fonctionnalités offertes par Unity, notamment l'option "canvas", nous avons pu intégrer la barre de vie de manière fluide et intuitive à l'interface utilisateur, lui permettant de s'afficher de manière visible et non intrusive à l'écran. En outre, nous avons mis en place un système de "slider" qui permet à la barre de vie de se déplacer de manière dynamique en fonction des points de



vie de l'animal contrôlé par le joueur.

Cette fonctionnalité ajoute une dimension immersive au gameplay, offrant une expérience visuelle riche et interactive tout en permettant aux joueurs de surveiller facilement l'état de santé de leur personnage pendant les différentes phases du jeu.

Dans le but de renforcer la dimension compétitive et de favoriser la clarté visuelle dans notre jeu, nous avons aussi introduit une innovation majeure : la distinction des barres de vie par couleur, bleue et rouge, pour différencier les deux équipes. Cette décision stratégique permettra aux joueurs de visualiser rapidement l'état de santé de leurs coéquipiers et de leurs adversaires. La barre de vie bleue représentera l'équipe des joueurs, tandis que la barre de vie rouge symbolisera les adversaires. Cette distinction visuelle facilitera la stratégie et l'interaction entre les membres de chaque équipe, tout en renforçant l'immersion dans l'univers du jeu.

4.5.3 Le système des cartes

Dans ce projet, je me suis aussi occupé du système des cartes ainsi que du piochage de ces dernières. Il m'a fallu consacrer beaucoup de temps à cette fonctionnalité indispensable pour que notre jeu ait une dimension stratégique et variée. Le système des cartes que le joueur va utiliser pendant une partie est assez complexe. Il comprend de nombreux petits détails, comme l'éclaircissement de la carte lorsque l'on passe la souris dessus, ou encore le fait que l'on puisse visualiser la carte que nous avons sélectionnée en la mettant en avant, ce qui la fait ressortir. (cf. Annexes).

Le système de piochage des cartes était aussi très complexe à insérer dans



le jeu, mais son usage était indispensable. Le joueur, possédant un deck de 8 cartes qu'il sélectionne en début de partie, aura donc 4 cartes devant lui durant toute la partie. Après qu'il ait joué une carte, celle-ci va se positionner hors écran de jeu et la carte suivante va venir la remplacer. Ce système de piochage utilise les positions des cartes et on a en quelque sorte un deck virtuel où la carte dont les coordonnées remplissent certaines conditions va être la carte suivante que le joueur va piocher. De plus, on peut apercevoir la prochaine carte que le joueur va piocher à gauche de sa main de cartes.

Enfin, l'ajustement de ce système a pu résoudre un problème que nous avons eu pendant la création de notre jeu, qui est le mouvement de la caméra. Il faut savoir que la caméra peut bouger, c'est-à-dire que lorsque l'on fait "clique droit" et qu'on glisse la souris, on peut apercevoir le terrain de jeu global que l'on peut utiliser dans la partie. Cette fonctionnalité a donc créé des problèmes au départ, puisque lorsque l'on bougeait la caméra, nous ne pouvions plus utiliser de cartes. Ce système de cartes nous a donc permis de résoudre ce problème.

4.5.4 Autre

J'ai également contribué à la recherche pour l'implémentation du mode multijoueur. Bien que cette contribution ait été modeste, j'ai participé à la collecte d'informations, à des tests et à la proposition de pistes de réflexion pour enrichir cette fonctionnalité afin de permettre d'aboutir au résultat recherché.

Il était aussi important de faire attention aux petits détails lors de la partie. Par exemple, le fait que certains boutons soient visibles à l'écran ou



4 TÂCHES INDIVIDUELLES

non en fonction du joueur qui joue, ou encore que le joueur ne puisse pas sélectionner une carte alors que ce n'est pas son tour.

J'ai aussi implémenté, avec les graphismes qu'Adrien m'a fournis, un curseur qui indique à l'écran l'animal qui joue ce tour. De la même façon que les barres de vies, chaque équipe a sa propre couleur (soit bleu, soit rouge). (cf. Annexes).

Pour garantir la qualité de notre jeu, j'ai joué un rôle actif dans l'identification et le signalement des bugs rencontrés. Cette démarche proactive permet d'améliorer la stabilité et les performances du jeu, offrant ainsi aux futurs joueurs une expérience de jeu plus fluide et immersive.

De plus, j'ai pris soin de favoriser une atmosphère de travail positive et collaborative au sein de notre équipe. En encourageant la communication ouverte, le respect mutuel et la résolution constructive des conflits, j'ai contribué à maintenir une dynamique de groupe harmonieuse et productive.

Enfin, mon rôle était aussi de corriger toutes les fautes linguistiques et de donner des conseils sur les choix des mots pour le menu du jeu ou le site internet, par exemple.



5 Récit de la réalisation

Ce projet a été une activité enrichissante pour tous les membres du groupe. Chacun a pu découvrir les points clés pour faire ce travail en équipe tout en faisant vivre ce groupe pour que ce projet aboutisse en temps et en heure.

5.1 Nos peines

Cependant plusieurs choses ne se sont pas passées comme prévu. Tout d'abord, le lancement du projet a été un casse-tête. Nous sommes partis en Décembre sur la plate-forme Godot et avons débuté dessus les premières lignes de code. Mais au bout de quelques semaines, nous avons décidé de changer de plate-forme de création pour celle d'Unity car elle était plus adaptée pour notre style de jeu. Ce changement a dû nécessiter la suppression de notre code et une adaptation rapide.

De plus, nous avons légèrement procrastiné au fil de cette année à propos de l'avancement du code, donc nous avons commencé le projet avec un peu de retard à cause des graphismes nécessaires, ce qui ne nous a pas permis d'avoir tout ce que l'on envisageait au terme du projet, comme les 25 cartes prévues et un système de niveau et de déblocage des cartes.

Enfin, vers les dernières semaines, la tension montait de plus en plus au sein de notre équipe ce qui a fragilisé la cohésion du groupe. Nous avons su régler ce problème qui a été un frein à l'avancement du projet et à la bonne entente au sein du groupe.



5.2 Nos joies

Ce projet nous a tout d'abord permis d'apprendre à nous connaître. Chaque membre de notre équipe est unique, apportant des perspectives variées et une diversité d'idées précieuses pour la réalisation de ce jeu. Grâce à cette diversité, nous avons pu explorer une multitude d'approches créatives et innovantes. La création de notre jeu vidéo a ainsi été une occasion non seulement de mettre en commun nos compétences techniques, mais aussi de découvrir les forces et les talents individuels de chacun. Cette connaissance mutuelle a été la clé pour former une cohésion de groupe solide et harmonieuse.

En particulier, lors de la dernière ligne droite, nous avons su tirer parti de nos complémentarités. Durant toute la période de développement, nous avons mis en place une organisation efficace qui a permis à chacun de mettre en avant ses qualités et compétences spécifiques là où elles étaient le plus utiles. Cette répartition des tâches basée sur les forces de chacun a été déterminante pour l'avancement rapide et réussi de notre projet. En travaillant ainsi de manière coordonnée et en valorisant les atouts individuels, nous avons non seulement progressé de manière significative, mais nous avons également renforcé notre esprit d'équipe et notre capacité à collaborer de manière efficace et harmonieuse.

Aujourd'hui notre équipe est très fière d'avoir réalisée ce jeu vidéo et est pressée de le faire découvrir aux futurs joueurs.



5.3 Avis personnel des membres de l'équipe

5.3.1 Adrien Coureau

Animatch a été une expérience très forte en émotions et en charge de travail. Durant la dernière semaine, étant très perfectionniste, je me sentais obligé de repasser sur le travail déjà effectué par les autres membres pour le compléter ou corriger des petits détails que je trouvais gênants. Je n'ai selon moi pas assez délégué le travail aux autres, étant le chef de groupe et le directeur artistique de l'équipe. Cela m'a valu de travailler 12 heures par jour sur Animatch pendant 7 jours consécutifs juste avant la troisième et dernière soutenance. Cette période a été pour moi la plus difficile du projet mais de loin celle où l'on a vu le plus de progrès se faire.

En tant que chef, je me suis trouvé trop passif, parfois laissant les tensions régner au sein du groupe, me sentant impuissant face à la situation. J'aurais dû davantage m'investir dans le renforcement de la cohésion d'équipe.

Passer plus de temps sur la réflexion du jeu aurait également été d'une grande utilité, mais la contrainte de devoir faire le cahier des charges très tôt dans l'année a certainement eu une conséquence importante sur la direction et la tournure qu'a pris Animatch.

Cependant, les points cités ci-dessus sont mes seuls regrets. Animatch a été à la fois une expérience enrichissante mais aussi formatrice pour entraîner mon art, mon style graphique, mes connaissances sur Unity et mes capacités de chef de projet. Je suis content du résultat final, mais aussi conscient que l'on aurait pu aller plus loin dans le projet et dans son élaboration.

Cela m'a donné envie d'envisager l'idée d'un autre projet de jeu vidéo, étant donné que je sais manipuler tous les domaines de ce type de création.



5.3.2 Josselin Priet

Ce projet m'a aidé à mieux comprendre les notions apprises en programmation en les utilisant concrètement.

Il m'a également permis d'apprendre la conception d'un jeu sur Unity. J'ai pu m'exercer à la conception d'un algorithme visant à être le plus compréhensible, simple et général, tout en remplissant les fonctionnalités nécessaires à la conception d'Animatch.

En bref, Animatch a été une expérience enrichissante !

5.3.3 Hector Thubert

Mes joies :

Le développement de notre projet de jeu en C# avec Unity a été une aventure riche en défis et en satisfactions. Chaque jour était une occasion de découvrir et de résoudre de nouveaux défis, cumulant peu à peu en une expérience enrichissante.

L'un des moments les plus mémorables de ce projet a été la résolution d'un problème majeur lors de la conception de la fonctionnalité de sélection de cartes. Ce système était crucial pour le gameplay, permettant aux joueurs de choisir différentes cartes avec des pouvoirs spéciaux. Le bug en question était que les cartes ne se chargeaient pas correctement et ne répondaient pas aux actions des joueurs.

Guillaume et moi avons consacré plus de deux heures à ce problème, décortiquant chaque ligne de code et explorant toutes les options possibles. La frustration de ne pas trouver la réponse était énervante.

Mais au bout de ces deux heures, nous avons enfin identifié la cause du



problème : une erreur de synchronisation dans la mise à jour des états des cartes. En réajustant la logique de notre code, nous avons réussi à corriger l'erreur. Voir le système enfin fonctionner après tant d'efforts a été très satisfaisant.

Mes peines :

Le développement n'est pas seulement fait de réussites. L'une des plus grandes peines de ce projet a été l'incapacité à implémenter une fonctionnalité souhaitée : les paramètres du jeu pendant une partie.

Mon objectif initial était de créer une scène dédiée aux options, permettant aux joueurs de modifier la musique du jeu. Cependant, la mise en œuvre de cette scène s'est avérée beaucoup plus complexe que prévu. J'ai passé de nombreuses heures à essayer de corriger des bugs persistants qui empêchaient cette scène de fonctionner correctement. Le problème principal était lié à la transition entre les scènes et la sauvegarde de la partie en cours, qui ne se faisaient pas correctement.

Malgré tous mes efforts, je n'ai pas pu résoudre ces problèmes à temps. En conséquence, j'ai dû abandonner l'idée de la scène et réduire la fonctionnalité à une simple interface accessible en jeu, qui s'activait et se désactivait.

En conclusion, ce projet a été une montagne russe d'émotions. Les moments de joie et de satisfaction, comme la résolution du bug de la sélection de cartes, et des moments de peines comme l'abandon de la scène des paramètres. Chacune de ces expériences m'a beaucoup appris sur le développement de jeux et la gestion des projets, et a renforcé ma passion pour le codage et la création.



5.3.4 Guillaume Rat

En rétrospective, ce projet a été une expérience profondément enrichissante et formatrice.

Même si nous regrettons de ne pas avoir pu réaliser un véritable mode multijoueur en ligne, je suis fier des solutions alternatives que nous avons mises en place. Ces solutions ont permis de proposer une expérience de jeu riche et personnalisée. Cette aventure m'a non seulement permise de surmonter des obstacles techniques, mais m'a également renforcé mes compétences en développement et en gestion de projets complexes. Je suis déterminé à continuer à améliorer ce projet et à explorer de nouvelles possibilités pour offrir une expérience de jeu encore plus aboutie à l'avenir.

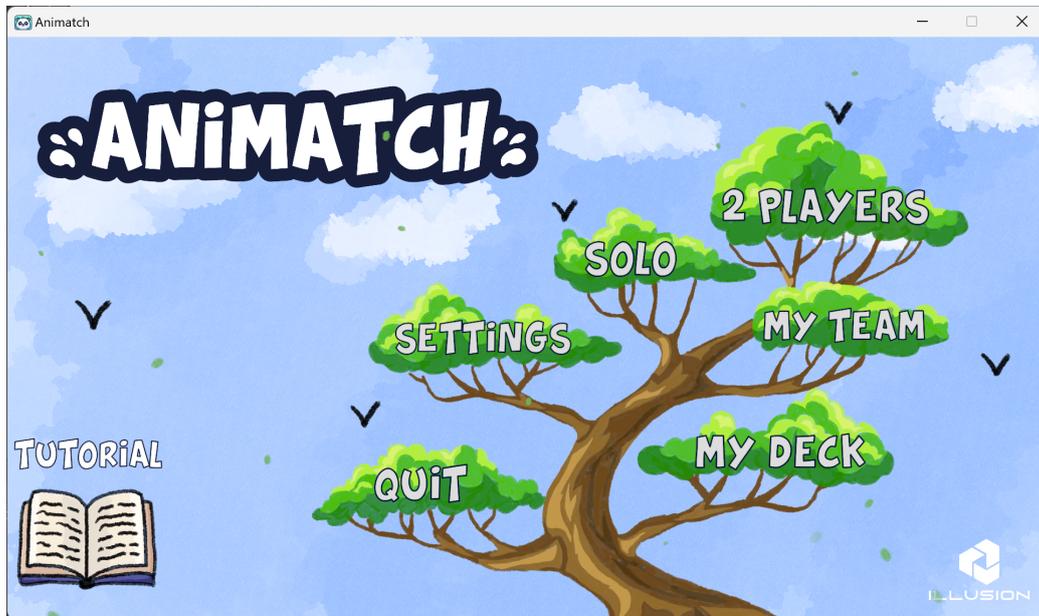
5.3.5 Guilhem Dardonville

Ce projet m'a apporté beaucoup de choses. Tout d'abord de la fierté d'avoir réalisé mon premier véritable jeu vidéo. Coder ce jeu m'a appris de nouvelles choses, notamment la cohésion de groupe qui est indispensable ainsi que l'organisation. Je serais prêt à sortir une deuxième version du jeu sur téléphone et avec de nouveaux animaux et cartes. Et si possible, reprendre ce qu'a débuté Guillaume sur le multijoueur et en avoir un opérationnel. Ce projet m'a permis d'en apprendre plus sur mes camarades et sur la façon de coder. Je pense que ce projet a été bénéfique pour tout le groupe et je suis sûr que nous serons motivés pour sortir une autre version du jeu.

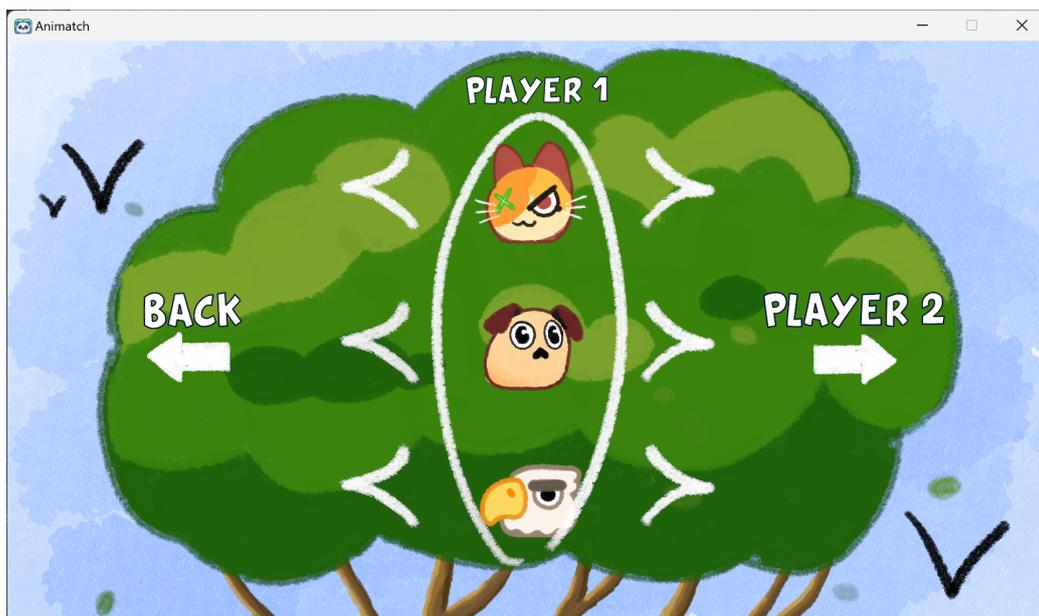


6 Annexes

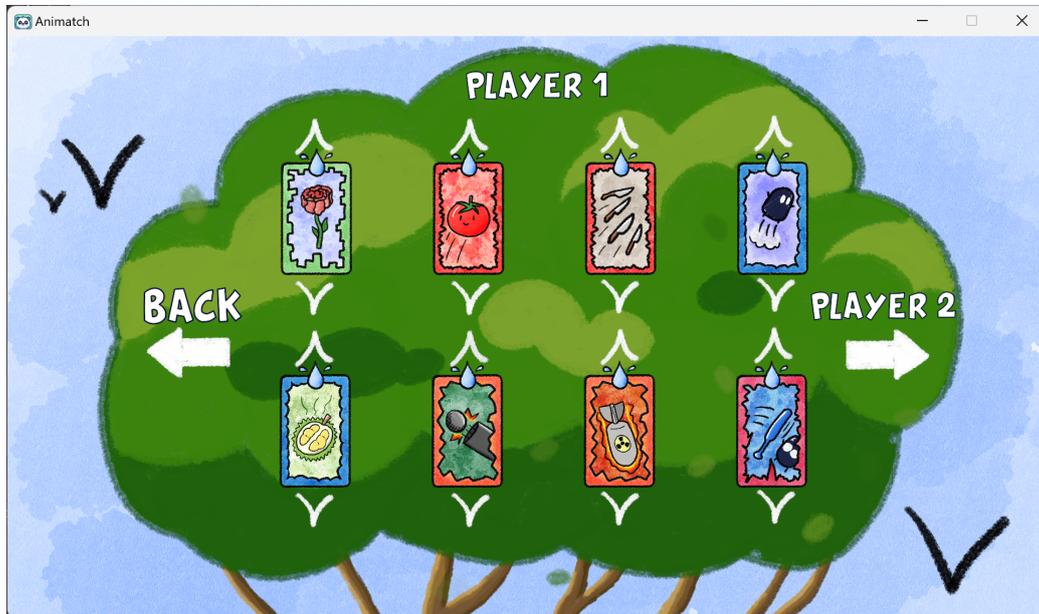
Menu d'Animatch :



Écran de sélection des animaux :



Écran de sélection des cartes :



Exemples de jeu :

