## CAHIER DES CHARGES

# Présentation du Projet :

Dans le cadre de l’UE LIFAP4 (Conception et développement d'applications), nous avons dû former des groupes de 2 à 4 afin de se lancer dans le développement d’une application, et plus particulièrement dans un jeu de combat tour par tour inspiré des différents Pokémons.

Notre groupe est composé de : BONNET Corentin, GOUBIN Fabien, BOUCHUT William et CHALOYARD Lucas.

Nous aurons 4h de TP supervisés par semaine où nous pourrons travailler sur les ordinateurs de l’université ou nos ordinateurs personnels, et nous pourrons aussi demander l’assistance des professeurs présents dans le cas où nous sommes bloqués ou si nous avons besoin de quelques indications quant à la démarche à suivre.

# Description de la demande :

#### Résultats attendus :

Nous devons rendre une archive avec une arborescence classique de développement logiciel contenant tous les fichiers nécessaires à la compilation et l’exécution de l’application. Cette dernière devra contenir deux modules IHM différents, l’un permettant un affichage texte et l’autre un affichage graphique.

Le code source rendu devra respecter les règles de programmation.

L’archive contiendra donc :

• Un README.txt (à la racine de l’archive)

• Le code source (dans le dossier src)

• Un Makefile (à la racine de l’archive)

• Les librairies incluses (autre que SDL)

• Les assets de notre application (dans le dossier data)

• Les exécutables (dans le dossier bin)

• La documentation de notre projet (dans le dossier doc)

Dans le dossier documentation nous retrouverons également, la présentation orale, le diagramme de Gantt et des modules en plus de la documentation du code.

#### Fonctionnalité du jeu :

Le jeu consistera en un jeu de combat stratégique tour par tour de combat de monstres extrait de la série Pokémon. Ce jeu permettra à n’importe quel joueur de s’entraîner ou de simuler certains combats en duel avec 6 Pokémons chacun. Les combats pourront se faire en 1 vs 1 ou bien en 2 vs 2. Débutant ou expérimenté, ce jeu est fait pour tous. Pour cela nous souhaiterions développer une IA de différents niveaux afin que n’importe quel joueur puisse s’entraîner et constater son amélioration au fil des combats ainsi que des fonctions LAN (Local Area Network) dans le but de permettre aux joueurs de pouvoir se défier entre eux dans le cadre de combat plus divertissant que des combats contre des IA. Le joueur se promènera sur une petite carte, pouvant sauvegarder son avancement à tout moment. Ce dernier fera des combats contre des dresseurs contrôlés par l’ordinateur. Il aura comme choix lors d’un combat, une grande variété de Pokémons qu’il débloquera en fonction de son niveau, niveau qu’il pourra augmenter en gagnant des combats. Pour jouer contre un nouvel adversaire, le joueur devra déjà gagner le précédent, ce qui le fera évoluer dans la hiérarchie.

**Combat :**

* Menu qui gère les attaques (sélectionner quelle attaque lancer)
* Menu pour changer de Pokémon au combat
* Menu pour choisir et utiliser des objets
* Interaction (perte de vie après attaque, état physique)
* Affichage graphique du combat (affichage classique Pokémon)
* Plusieurs types de combat : en duo, avec IA, autre joueur en LAN ou sur même ordi ou encore duo mais avec un seul joueur donc de la même équipe Pokémon

**Equipe :**

* Composer son équipe en fonction des Pokémons débloqués selon le niveau du joueur.
* Changer l’ordre des Pokémons dans l’équipe, et changer en cours de combat.
* Garder un historique des équipes précédentes ainsi que de l’équipe favorite.

**IA (Différents niveaux de difficultés) :**

* Facile (prend juste en compte la vie adversaire).
* Moyen (prend en compte les types des attaques et du Pokémon adverse).
* Difficile (prend en compte les statistiques du Pokémon adverse).

**Réseau :**

* Créer/Rejoindre des lobbys (permet de faciliter le Matchmaking).
* Echanger les informations de départ (équipe Pokemon) puis seulement de simples informations (quelle attaque a été effectué, changement ou objet).
* Affiche les statistiques de fin de bataille.

**Carte :**

* Pouvoir se déplacer sur une petite carte avec son personnage et pouvoir accéder au combat, à la sauvegarde ainsi qu’a la boutique.
* Trouver une manière de faire originale tout en gardant les graphismes Pokemon

**Tutoriel :**

* Un bouton au menu principal permettant d’ouvrir deux-trois pages d’explications sur les combats (faiblesse/résistance face aux types, les différences entre physique et spécial etc…).

**Pokedex :**

* Menu permettant de voir tous les Pokémons disponibles, les stats, les attaques qu’ils peuvent avoir etc…

# Contraintes :

* La durée de ce projet sera d’approximativement 2 mois et demi, la date maximale de la remise de l’archive étant le Lundi 6 Mai à 18h.
* Nous avons aussi une démo à présenter le Mardi 26 Mars, il nous est donc nécessaire d’avoir un minimum de code source fonctionnel afin de présenter quelque unes de nos principales fonctionnalités ainsi que si possible les deux versions de l’IHM.
* Le jeu sera développé en C/C++ sous Linux/MacOS.
* Les librairies utilisées seront SFML et une librairie texte pour une version alpha.
* Le code respectera le standard des réglés de programmation : code indenté, variable ayant du sens, …
* Le code sera géré et archivé sur GitLab.
* La documentation du code sera générée par Doxygen.
* Un diagramme des classes sera mis à jour régulièrement afin d’avoir une vision d’ensemble du code source.
* L’équipe de développement utiliseras les outils de debug et de profiling : Gdb et Valgrind.

# Déroulement du Projet :

La mise en commun du travail de chaque étudiant se fera à travers l'interface de GitLab afin que chacun puisse travailler en local depuis chez soi.

Notre planification s'appuie sur un projet Gantt afin de mieux répartir les tâches et de respecter les directives de l'UE.

* **Tâche 1** : Rédaction du cahier des charges.

*Durée : 2 semaines.*

Tâche 1.1 : Définir le diagramme des classes.

Membres impliqués : tous.

Tâche 1.2 : Définir le diagramme de Gantt.

Membres impliqués : tous.

* **Tâche 2** : Développement d’un 1er prototype en mode texte.

*Durée : 3 semaines*

Tâche 2.1 : Écriture et test du module Attaque.

Membres impliqués : BOUCHUT William.

Nécessaire au module Pokémon, afin de pouvoir gérer les attaques de chaque Pokémon.

Tâche 2.2 : Écriture et test du module Pokémon.

Membres impliqués : GOUBIN Fabien.

Nécessaire à tout combat du jeu, le « Pokémon » sera le combattant.

Tâche 2.3 : Écriture et test du module Personnage.

Membres impliqués : BONNET Corentin.

Ce module représentera le personnage jouer par le joueur.

Tâche 2.4 : Écriture et test du module Objet et Sac.

Membres impliqués : CHALOYARD Lucas.

Permet la gestion de objets utilisables durant un combat.

Tâche 2.5 : Écriture et test du module Combat.

Membres impliqués : tous.

Gérera le principe du jeu, le combat, cela utilisera toutes les classes écrites et testées précédemment.

Tâche 2.6 : Écriture et test du module AffichageTxt.

Membres impliqués : tous.

Permet l’affichage console du jeu, cela utilisera toutes les classes précédentes et nous permettra de visualiser facilement les problèmes qu’on rencontrera afin de pouvoir optimiser le jeu.

Tâche 2.7 : Test et debug

Membres impliqués : tous

Tâche 2.8 : Première démo.

Membres impliqués : tous

* **Tâche 3 :** Développement d’un 2e prototype en mode graphique (SFML)

*Durée : 2 semaines*

Tâche 3.1 : Apprentissage et pratique de SFML.

Membres impliqués : tous

Tâche 3.2 : Affichage d’image et test de performance :

Membres impliqués : tous

Test de différentes méthodes d’affichages afin d’optimiser au maximum notre module d’affichage en mode graphique.

Tâche 3.3 : Développement de la boucle de jeu en SFML dans le module AffichageGraphique.

Membres impliqués : tous

Permettra la gestion graphique du jeu, les assets utilisés seront principalement extraits de la série de jeu « Pokémon » et certains seront créés par nous-même.

Tâche 3.4 : Test et debug

Membres impliqués : tous

Tâche 3.5 : Intégrer à la 1ère démo

Membres impliqués : tous

* **Tâche 4 :** Développement d’un 3e prototype en mode graphique avec la musique et un mode multi-joueurs en local (sur le même ordinateur)

*Durée : 2 semaines*

Tâche 4.1 : Modification de la boucle de jeu afin de gérer la musique

Membres impliqués : BOUCHUT William, GOUBIN Fabien

Tâche 4.2 : Modification de la boucle de jeu afin de gérer le mode multi-joueurs

Membres impliqués : CHALOYARD Lucas, BONNET Corentin

Tâche 4.3 : Test et debug

Membres impliqués : tous

* **Tâche 5 :** Développement d’un 4e prototype avec ajout d’un mode multi-joueurs en réseau (seulement en LAN) et ajout de quelques fonctionnalités de jeu.

*Durée : 2 semaines*

Tâche 5.1 : Exploration et compréhension des fonctionnalités réseau de SFML

Membres impliqués : tous

Tâche 5.2 : Modification de la boucle de jeu afin de gérer le mode multi-joueurs en LAN

Membres impliqués : CHALOYARD Lucas, GOUBIN Fabien

Permettra d’éviter pour deux joueurs de jouer sur un seul et même ordinateur, cela rendra les fonctionnalités JcJ (« Joueur contre Joueur ») plus agréable.

Tâche 5.3 : Ajout de quelques fonctionnalités de jeu supplémentaires

Membres impliqués : BOUCHUT William, BONNET Corentin

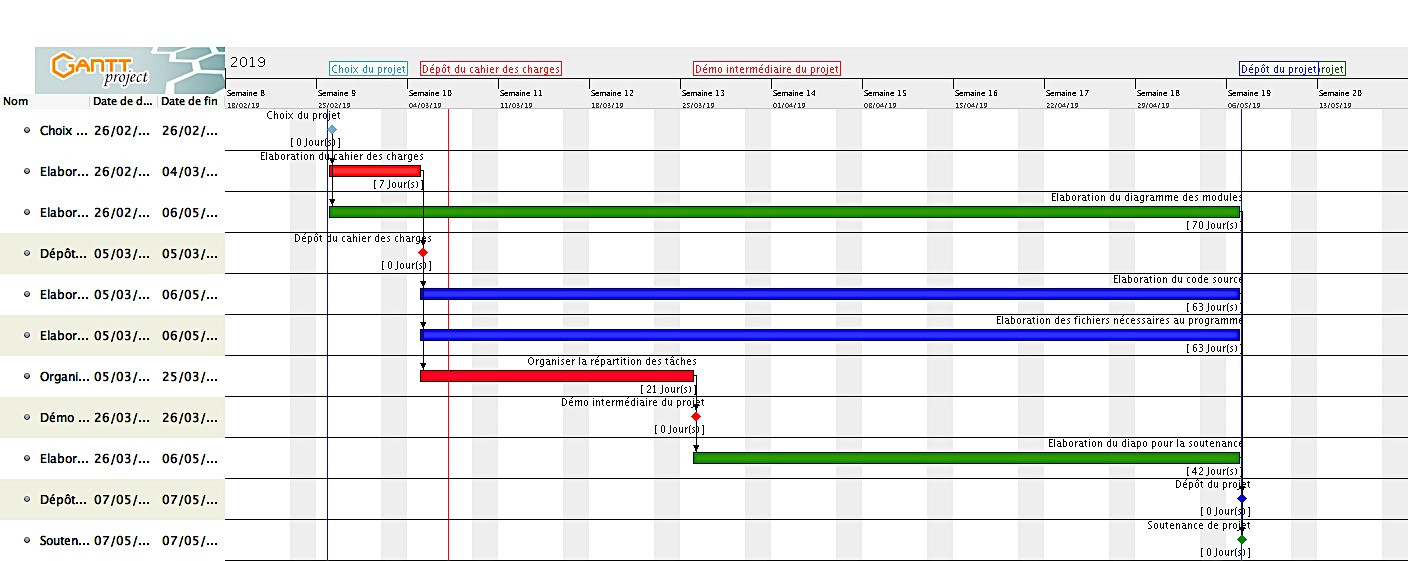
Création des modules Map, CarreDeMap et Dresseur, cela nous permettra de rajouter des fonctionnalités supplémentaires à notre jeu, afin de sortir un peu de la dimension JcJ (« Joueur contre Joueur »).

Tâche 5.4 : Test et debug

Membres impliqués : tous

# Diagramme de Gantt

#### Le diagramme de Gantt ne détaille pas l’élaboration du code source, il nous permet plutôt de gérer le travail supplémentaire au développement du jeu, cela concernant la soutenance, l’archive, etc…



# Diagramme des classes

