



Projet **CY-Fighters**

CLASSE PRE-ING1 • 2021-2022

DESCRIPTION

Le but de ce projet est de réaliser un programme qui permet à 1 ou 2 joueurs de faire s'affronter des équipes de plusieurs combattants au tour par tour. Il permettra donc d'afficher à chaque étape, l'état du jeu (caractéristiques des combattants, attaques possibles, ...)

Ce projet mettra l'accent sur les structures, l'ergonomie de l'affichage et la lecture de fichiers.

CAHIER DES CHARGES MINIMAL

- Chacun des 2 joueurs (ou ordinateur) possède une équipe de plusieurs combattants. A vous de décider si les équipes sont forcément de la même taille ou non. Voir plus bas pour les propriétés d'une équipe
- Au début du lancement du programme, chaque joueur peut créer son équipe en choisissant l'un après l'autre un combattant parmi une liste qui devra être construite à partir d'un fichier lu depuis le disque dur
- Chaque combattant possède plusieurs caractéristiques qui fourniront les bases pour calculer les dégâts à appliquer, ceux reçus et/ou évités, le moment où un combattant peut attaquer ou non. Voir plus bas pour les propriétés de base d'un combattant
- Un combat dure jusqu'à ce que l'une des 2 équipes soit vaincue (il ne reste aucun combattant de cette équipe avec des points de vie strictement supérieurs à 0)
- Chaque combattant d'une équipe peut attaquer l'adversaire de son choix en utilisant, soit une frappe classique, soit faire appel à une technique spéciale. A vous de décider du nombre de techniques spéciales que chaque combattant possède. Voir plus bas pour la description d'une technique spéciale
- Dans le cas d'une partie avec 1 joueur contre l'ordinateur, ce dernier devra attaquer de manière plus ou moins cohérente : pour cela vous programmerez différentes difficultés :
 - **noob** : l'ordinateur n'utilise aucune technique spéciale et attaque les combattants au hasard
 - **facile** : l'ordinateur n'utilise aucune technique spéciale et attaque systématiquement le combattant ayant le moins de points de vie
 - **moyen** : l'ordinateur utilise ses techniques spéciales dès qu'il le peut et cible toujours le plus faible (allié ou ennemi)
- Ce projet doit mettre l'accent sur l'ergonomie de l'interface graphique pour afficher toutes les informations nécessaires aux choix des joueurs

BONUS POSSIBLES

- Définir des équipes prédéfinies, équilibrées, chargées au démarrage, permettant de faire des parties rapides
- Un éditeur en ligne de commande pour charger, modifier et sauvegarder différents combattants, équipes, techniques, ...
- Utilisation d'un terminal Linux pour afficher des couleurs, des émoticônes, afin que l'interface soit plus conviviale
- Implémenter un mode **difficile** contre l'ordinateur, avec un algorithme de votre choix qui tentera de maximiser les dégâts sur la durée
- Vous pouvez implémenter d'autres améliorations de votre choix

RESSOURCES UTILES

GitHub

- Site web : <https://github.com>

Raid : Shadow Legends

- Ordre des tours de jeu : https://www.serial-gamers.fr/raid-shadow-legends/guide/#Compteur_de_tours

Les séquences d'échappement d'un terminal

- Wikipédia : https://en.wikipedia.org/wiki/ANSI_escape_code

EQUIPES

- Une équipe contient un nombre fini de combattants ainsi qu'un nom pour pouvoir l'identifier
- Le nom de l'équipe doit être entré par le joueur en début de partie (ou posséder une valeur par défaut comme « Equipe 1 » par exemple)

COMBATTANTS

- Les combattants sont des personnages prédéfinis du jeu (donc stockés dans des fichiers) et qui contiennent déjà leurs valeurs de propriétés, ainsi que la liste de leurs techniques spéciales.
- C'est à vous de définir le format et le nombre de ces fichiers
- Séparer les fichiers des combattants et ceux des techniques spéciales pourrait être un bon moyen d'organiser votre code
- Les propriétés de chaque combattant sont les suivantes :
 - **points de vie courant** : si négatif ou nul, le combattant est mis K.O. et ne peut plus participer jusqu'à la fin du combat
 - **points de vie maximum** : pour limiter les points de vie courant en cas d'applications de soins
 - **attaque** : permet de calculer les dégâts de base infligés
 - **défense** : permet de calculer le pourcentage de réduction des dégâts
 - **agilité** : permet de calculer le pourcentage de chances d'éviter tout ou partie de l'attaque
 - **vitesse** : permet de calculer le moment où le personnage va jouer son prochain tour. La célèbre franchise de jeu *Raid:Shadow Legends* utilise un système similaire
 - **compétences spéciales** : plusieurs actions à utiliser au lieu d'attaquer, et qui permettront, soit d'affecter un état spécial bénéfique à un allié, ou au contraire affaiblir un des combattants adverses
 - **effets spéciaux temporaires** : plusieurs effets (positifs ou négatifs) qui découlent de l'utilisation d'une technique spéciale. Un effet spécial n'est rien d'autre qu'une technique spéciale qui a été appliquée à ce combattant précédemment

TECHNIQUES SPECIALES

- Une technique spéciale, est une action effectuée à la place de frapper un adversaire, mais dans le but de faire beaucoup plus de dégâts, ou d'affaiblir l'ennemi, ou soigner l'un de ses alliés, ou toute autre action qui permettrait de prendre un avantage stratégique
 - Chaque technique spéciale ne peut s'utiliser en boucle : il faut attendre plusieurs tours avant de pouvoir la réutiliser à nouveau : c'est le temps de recharge qui se compte en nombre de tours de jeu
 - On peut imaginer des techniques spéciales qui vont augmenter ou diminuer les caractéristiques de la cible pendant quelques tours (attaque, défense, ...), ou bien appliquer des dégâts ou des soins pendant quelques tours : il faudra donc que chaque technique contienne le nombre de tours pendant lequel l'effet s'applique
- On peut donc déjà spécifier que chaque technique spéciale contient les propriétés suivantes :
- **nom** : pour l'identifier
 - **valeur** : valeur à appliquer à une propriété. La propriété affectée dépend de la technique spéciale elle-même. L'opération aussi (ajout, soustraction, multiplication, ...)
 - **description** : petit texte pour résumer le comportement de la technique spéciale
 - **nombre de tours actifs** : nombre de tours de jeu pendant lequel l'effet spécial restera actif. Si la technique est appliquée immédiatement, alors cette propriété n'est pas utilisée
 - **nombre de tours de recharge** : nombres de tours de jeu avant de pouvoir se servir à nouveau de cette technique

EXEMPLE D'INTERFACE GRAPHIQUE

- Dans l'image ci-dessous, on peut voir un exemple non-exhaustif d'interface graphique dans lequel on peut visualiser l'ensemble des informations nécessaires aux choix stratégiques du joueur

The screenshot shows a game interface with the following elements and annotations:

- Team 1 (EQUIPE 1):**
 - SASUKE |1| (3 health icons)
 - SAKURA |2| (2 health icons)
 - NARUTO |3| (3 health icons)
- Team 2 (EQUIPE 2):**
 - KIBA (3 health icons)
 - SHINO
 - HINATA (3 health icons)
- Special Techniques (TECHNIQUES SPECIALES):**
 - [1] Double Kunai (durée: 3 tours)
 - Ø Permet de frapper 2 fois par tour
 - [-] Ecran de fumée (recharge: 2/durée: 4 tours)
 - Ø Permet d'augmenter l'esquive de 75%
 - [3] Barrière de glace (durée: 3 tours)
 - Ø Permet d'augmenter la défense de 50%

Annotations in the image:

- Nom de l'équipe adverse (points to EQUIPE 1)
- Nom du combattant (points to SASUKE, SAKURA, NARUTO)
- Cible déjà vaincue (points to SAKURA)
- Numéro de la cible (points to |1|, |2|, |3|)
- Barre de vie (points to health icons)
- Effets spéciaux en cours (points to the top of the team status)
- Indicateur du prochain tour (points to the right side of the team status)
- Rechargement et durée de la technique (points to the recharge and duration info for the smoke screen)
- Technique non sélectionnable (points to the smoke screen technique)
- Numéro de la technique spéciale (points to [1], [-], [3])
- Nom et description (points to the technique names and descriptions)