

Projet 3: TErmifighter

Mickaël HORN



29 novembre 2021

Openclassrooms

Openclassrooms.com

Table des matières

[Présentation du projet et objectifs 2](#_Toc88575445)

[Fonctionnement du jeu 3](#_Toc88575446)

[Diagramme de classe 6](#_Toc88575447)

[Conclusion 7](#_Toc88575448)

# Présentation du projet et objectifs

TermiFighter est un projet de réalisation d’un jeu vidéo en Swift basé sur le terminal.

C’est un jeu de combat à deux joueurs ou chacun va choisir trois personnages parmi une liste déjà établie afin d’attaquer son adversaire et remporter la partie.

Il est réalisé sur xCode et prêt à compiler en se lançant dans un terminal.

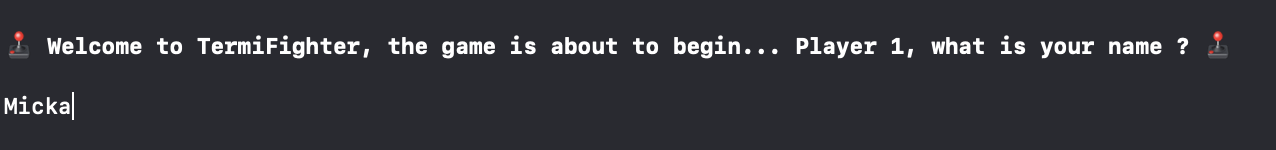
J’ai pris la liberté de rajouter des Emojis pour rendre le programme plus accueillant et plus clair.

Les objectifs à réaliser sont les suivants :

* Chaque joueur doit avoir un pseudonyme et 3 personnages
* Chaque personnage à : Nom/Points de vie/Arme
* Obligation de renommer les personnages
* Un coffre aléatoire lors du tour d’un personnage avec sa nouvelle arme à l’intérieur
* Afficher les statistiques de fin de partie : Gagnant/Nombre de tour/Liste des personnages.

# Fonctionnement du jeu

Il se joue à deux, chaque joueur choisis son pseudo.



Ensuite, 3 personnages parmi une liste déjà implémentée.



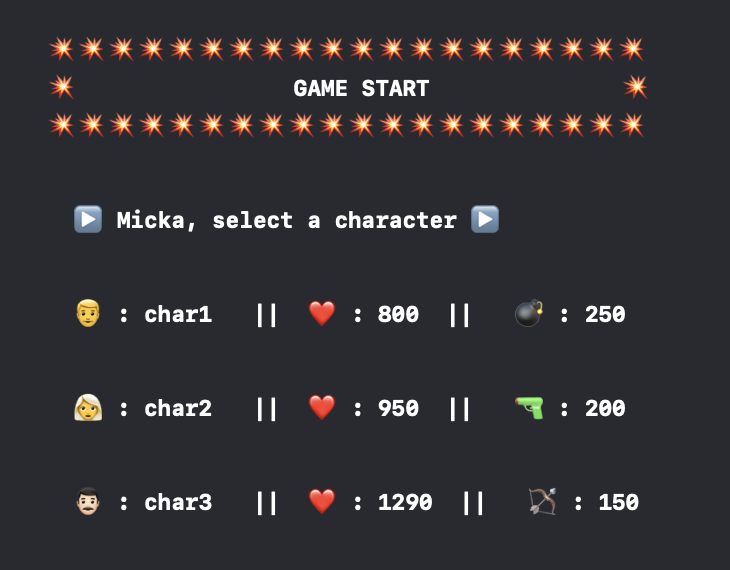
Les joueurs doivent renommer leurs personnages, le nouveau nom choisi doit être différent de ceux déjà existants.

Une image contenant texte

Description générée automatiquement

Une fois les pseudos et les personnages choisis, la partie peut commencer.

Le joueur 1 ouvre le bal, choisissant un de ses personnages afin d’attaquer un ennemi ou bien de soigner un allié.



Lorsqu’un personnage est choisi, un coffre apparaît de façon aléatoire lors de son tour, contenant sa nouvelle arme.



Si un coup fatal est porté, le personnage tué est retiré de la liste des personnages disponibles.

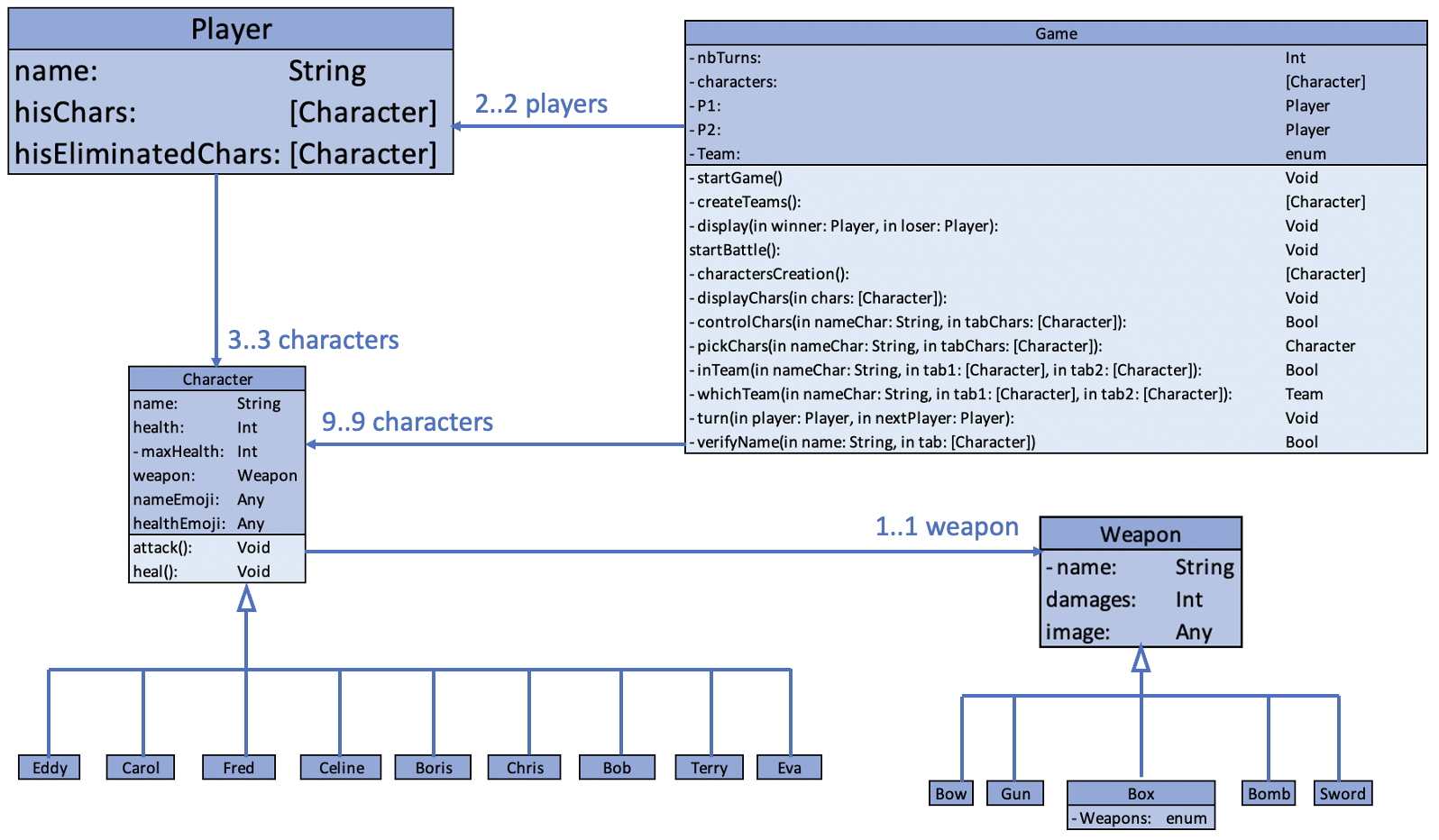
Une image contenant texte, écran

Description générée automatiquement

La partie se termine quand un joueur ne possède plus de personnage et l’écran de fin affiche les statistiques de la partie (nom du gagnant, nombre de tour, liste des personnages des deux joueurs).



# Diagramme de classe



# Conclusion

Ce projet a été une excellente occasion pour moi de mettre en pratique ce que j’ai appris et m’a également responsabilisé à rendre un projet en temps et en heure.

J’ai rencontré divers problèmes lors de sa réalisation, spécialement dans l’utilisation de la fonction « readLine() » sur laquelle j’ai passé pas mal de temps mais j’ai su trouver des solutions et produire un jeu fonctionnel.

J’ai dû également faire preuve d’autonomie, de réflexion et de patience et je me sens prêt à aller de l’avant.

Cela m’encourage beaucoup pour le futur et je suis impatient de réaliser des applications plus « graphique » et continuer à gagner de l’expérience en Swift.