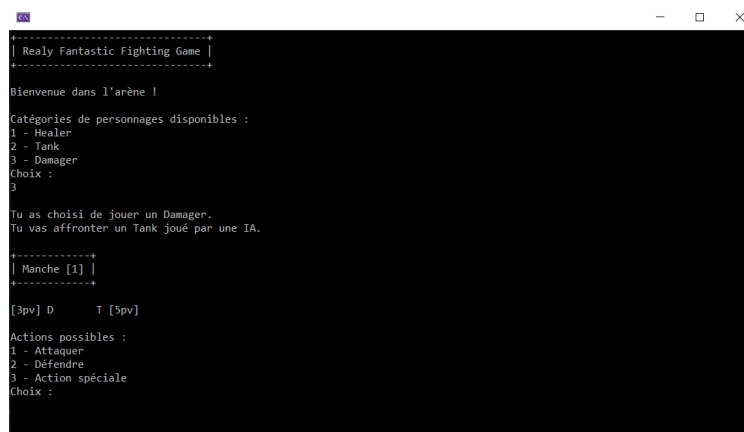

USRS45 - Défi

Réalisation d'un jeu de combat en tour à tour dans un environnement textuel

Octobre 2021

Contexte

Cette première UE vous conduira à acquérir toutes les compétences et connaissances nécessaires à l'élaboration d'un jeu en console où l'utilisateur affrontera tour par tour une IA plus ou moins coriace.



```

|-----+
| Really Fantastic Fighting Game |
|-----+

Bienvenue dans l'arène !

Catégories de personnages disponibles :
1 - Healer
2 - Tank
3 - Damager
Choix :
3

Tu as choisi de jouer un Damager.
Tu vas affronter un Tank joué par une IA.

|-----+
| Manche [1] |
|-----+

[3pv] D      T [5pv]

Actions possibles :
1 - Attaquer
2 - Défendre
3 - Action spéciale
Choix :
```

FIGURE 1 – Exemple de visuel attendu à la fin de l'UE pour la réalisation du défi.

Votre réflexion portera dans un premier temps sur les étapes logiques à mettre en place, c'est-à-dire l'algorithme du défi pour ensuite vous concentrer sur l'implémentation blocs par blocs en C#.

Fonctionnalités demandées

- L'exécution du jeu doit se faire dans une fenêtre textuelle (console) ou dans une fenêtre graphique.
- L'utilisateur doit pouvoir interagir avec le programme via des saisies clavier ou des interactions souris.
- Le programme doit intégrer une intelligence artificielle capable au minimum de faire des choix aléatoires.
- Le programme doit avoir des conditions de début et de fin associées à la victoire du joueur humain ou de l'intelligence artificielle.

Contraintes

- Travail à réaliser par groupe de 3.
- Le code source doit être propre et commenté.
- Vous devrez fournir le code source de votre défi, un readme pour expliquer à l'utilisateur comment exécuter et interagir avec votre programme et vos différentes options ainsi qu'une vidéo de présentation de groupe de votre travail avant la date indiquée sur moodle.
- La vidéo de présentation doit :
 - Durer 10 minutes.
 - Faire intervenir chaque membre du groupe.
 - Être structurée comme une présentation orale avec une introduction, une vue d'ensemble, une démonstration et une conclusion.
 - Mettre en avant vos initiatives, la réalisation des bonus, les difficultés rencontrées et les solutions mises en place.

Attention : Cette vidéo de présentation ne doit en rien être une simple revue de code où vous présentez ligne par ligne votre code source.

Règles du jeu

Vous allez implémenter un jeu de combat en tour par tour entre deux personnages. Un personnage contrôlé par l'utilisateur et un personnage contrôlé par une intelligence artificielle.

Les personnages seront choisis parmi des classes prédéfinies (voir **Classes de personnages**).

A chaque tour, les personnages pourront effectuer une action d'attaque, une action de défense ou encore utiliser leur capacité spéciale (voir **Actions possibles**). Les actions sont ensuite révélées simultanément.

Le jeu se poursuit tant que les deux personnages ont encore des points de vie et s'arrête dès que l'un ou les deux personnages perdent leur dernier point de vie.

Les classes de personnages

Damager :



Points de vie : ♥♥♥

Force d'attaque : ♦♦

Capacité spéciale - Rage : Inflige en retour les dégâts qui lui sont infligés durant ce tour. Les dégâts sont quand même subis par le Damager.

Healer :



Points de vie : ♥♥♥♥

Force d'attaque : ♦

Capacité spéciale - Soin : Récupère un point de vie (+♥).

Tank :



Points de vie : ♥♥♥♥♥

Force d'attaque : ♦

Capacité spéciale - Attaque puissante : Correspond à une attaque durant laquelle le Tank sacrifie un de ses points de vie pour augmenter sa force d'attaque de 1 et ce uniquement durant le tour en cours. ($-\heartsuit \Rightarrow +\diamond$)

Les actions possibles

Infliger à l'adversaire autant de dégâts que sa force d'attaque.

Attaquer :

*Exemple : Un Healer qui choisit l'action **Attaquer** inflige à son adversaire 1 dégât du fait de sa force d'attaque (◇) alors qu'un Damager infligera 2 dégâts du fait de sa force d'attaque (◇◇).*

Permet de se prémunir des dégâts infligés par l'adversaire.
*En cas de défense contre une attaque puissante d'un Tank, l'action **Défendre** ne permet de se prémunir que d'un seul dégât et conduit donc à prendre une blessure.*

Défendre :

*Exemple : Si un Healer qui choisit l'action **Défendre** contre un Damager qui attaque alors aucun des deux personnages ne perd ne point de vie. Cependant si un Damager défend face à une action spéciale "Attaque puissante" d'un Tank alors le Damager prévient une blessure mais est quand même touché par l'Attaque Puissante et perd un point de vie. Le Tank perd également un point de vie pour avoir activé son Attaque Puissante.*

Action spéciale :

Voir les capacités spéciales de chaque personnage.

L'intelligence artificielle

Une intelligence artificielle simple procédant par choix aléatoires de son personnage et de ses actions suffit pour la réalisation du jeu de base.

Implémentation

L'implémentation du jeu se fera au fur et à mesure de la progression dans l'UE. Vous commencerez par réfléchir à l'algorithme général du jeu qui vous permettra d'identifier des blocs de sens. Certains blocs seront plus simples que les autres et pourront être implémentés en premiers. Les derniers blocs seront implémenter en fin d'UE quand vous aurez à votre disposition tous les outils théoriques pour le faire.

[Bonus] Pour aller plus loin

Les fonctionnalités suivantes sont facultatives. Assurez-vous d'avoir un jeu de base fonctionnel avant de vous aventurer à les mettre en place. Ces améliorations vous permettront d'approfondir les notions vues durant cette première UE et de vous préparer aux concepts de l'UE prochaine. Elles vous permettront également de mesurer votre progression et votre habilité à traduire une situation en langage humain, en algorithme dans un premier temps puis en programme dans un deuxième temps.

1. Ajouter une catégorie de personnage avec des statistiques et une capacité spéciale de votre choix
2. Ajouter des niveaux de difficultés à l'intelligence artificielle pour lui permettre d'avoir de meilleures stratégies.
3. Ajouter un mode de simulation IA contre IA pour tester l'équilibrage des différents personnages et remplir le tableau suivant sur 100 simulations pour chaque case.

	Damager	Healer	Tank
Damager		ex : 65%	
Healer			
Tank			

Interprétation de la case Damager/Healer : Sur 100 simulations aléatoires, la classe Damager gagne 65% des fois contre la classe Healer.

4. Proposer une interface graphique pour le jeu de combat.