UE - Défi

Implémentation en C# d'une version allégée de MiniVilles

Octobre 2021

Présentation :

Au travers du défi de cette deuxième Unité d'Enseignement, vous allez travailler à la mise en place d'un jeu en tour par tour associant la gestion de cartes et de dés. L'objectif du jeu est d'être le premier joueur à réunir une somme d'argent fixée. Les mécanismes de base sont ceux du jeu de société Minivilles.
Vous ne serez pas guidés dans vos implémentations pour ce dernier travail de l'UE. Ce document contient les fonctionnalités attendues, les règles du jeu et quelques indications à partir desquelles vous allez devoir travailler en autonomie. C'est un travail conséquent, organisez-vous bien, répartissez-vous le travail quand cela vous semble possible et communiquez entre vous.

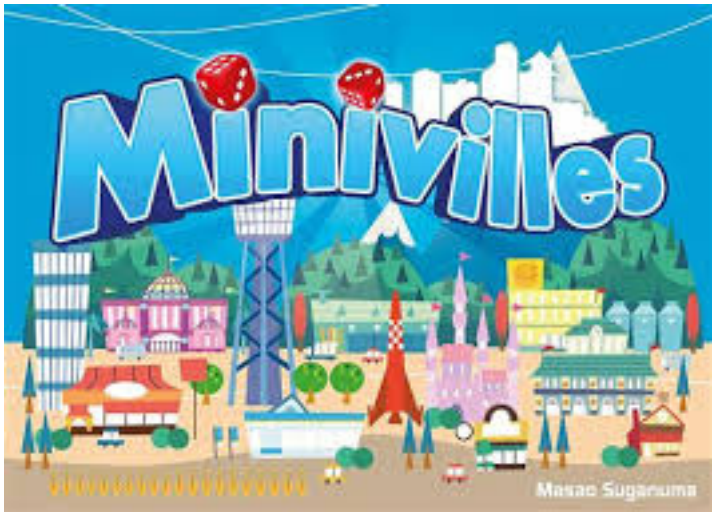

FIGURE 1 – Visuel du jeu de société Minivilles

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 groupes de 3.

→ Le code source doit être propre et commenté.

→ Vous devrez fournir un lien vers un dépôt git contenant l'ensemble des **fichiers nécessaires à l'exécution de votre jeu, les différentes évolution de votre diagramme de classes et un readme** permettant d'expliquer à l'utilisateur comment exécuter et interagir avec votre programme. La version disponible à la date finale indiquée sur moodle fera foi pour l'évaluation.
Un unique dépôt de votre travail le dernier jour ne correspond pas à l'utilisation de l'outil de gestion de versions git qui est attendu de vous. Ce comportement sera donc pénalisé.

→ Vous devrez également fournir **une vidéo de 10 minutes** de présentation de votre travail, des points clés de votre développement, de vos initiatives et difficultés. Introduisez votre travail en vous appuyant sur votre diagramme de classes et présentez en détail les améliorations développées. Votre vidéo devra également contenir une démonstration de l'exécution du jeu, vous êtes libre de choisir les aspects que vous souhaitez montrer. Les membres du groupe devront se partager équitablement le temps de parole.

Règles du jeu :

Chaque joueur commence la partie avec 2 cartes, un champ de blé et une boulangerie, ainsi que 3 pièces.

FIGURE 2 – Visuel du jeu de société Minivilles pour les cartes Champs de blé et Boulangerie.

Comprendre les cartes :

⋆ La couleur de la carte correspond au moment où elle peut être activée : Une carte bleue s'activera durant le tour de n'importe quel joueur. Une carte verte ne pourra s'activer que durant le tour du joueur qui la possède. Une carte rouge ne s'activera que durant le tour de l'autre joueur.
Exemple : ***Champs de blé*** *est une carte bleue qui pourra s'activer durant le tour de n'importe quel joueur et dont les bénéfices iront au joueur qui la possède.*

⋆ Le chiffre tout en haut correspond à la valeur du dé qu'il faut obtenir pour que la carte soit activée.
Exemple : ***Boulangerie*** *est une carte verte, elle ne sera activée que lorsque le joueur qui la possède obtient lui-même un 2 ou un 3 lors de son lancer de dé.*

⋆ L'effet écrit tout en bas correspond aux bénéfices accordés au joueur qui possède la carte lorsque celle-ci s'activera.
Exemple : Lorsqu'un joueur obtient 1 à son lancer de dé, tous les joueurs qui possède un ou des ***Champs de blé*** *gagneront une pièce pour chaque Champs de blé qu'ils possèdent.*

Un tour de jeu s'organise comme suit :

1. Le joueur A lance le dé.
2. Le joueur B regarde s'il a des cartes bleues ou rouges qui s'activent et il en applique les effets
3. Le joueur A regarde s'il a des cartes bleues ou vertes qui s'activent et il en applique les effets
4. Le joueur A peut acheter une nouvelle carte et l'ajouter à sa ville. Il est possible d'avoir plusieurs fois la même carte, les effets s'additionnent.

Une fois le tour du joueur A terminé, c'est au tour du joueur B de réaliser les mêmes actions.

Fin de partie :

Le jeu se termine dès que l'un au moins des deux joueurs possède 20 pièces.

Les cartes du jeu :

Les cartes sont toutes disponibles en 6 exemplaires chacune. Les joueurs peuvent acheter au maximum une carte par tour en dépensant le prix indiqué sur la carte. Ils choisissent parmi la liste des cartes celle qu'ils souhaitent acheter.

Les cartes sont listées sous le format :

[Valeur d'activation] Couleur - Nom : Effet - Coût de la carte

- ⋆ [1] Bleu - **Champs de blé** : Recevez 1 pièce - 1$
- ⋆ [1] Bleu - **Ferme** : Recevez 1 pièce - 2$
- ⋆ [2] Vert - **Boulangerie** : Recevez 2 pièces - 1$
- ⋆ [3] Rouge - **Café** : Recevez 1 pièce du joueur qui a lancé le dé - 2$
- ⋆ [4] Vert - **Superette** : Recevez 3 pièces - 2$
- ⋆ [5] Bleu - **Forêt** : Recevez 1 pièce - 2$
- ⋆ [5] Rouge - **Restaurant** : Recevez 2 pièces du joueur qui a lancé le dé - 4$
- ⋆ [6] Bleu - **Stade** : Recevez 4 pièce - 6$

Indications :

1. Votre programme devra contenir au minimum les classes suivantes
   - ⋆ **Die** - Génère et manipule un dé
   - ⋆ **Player**
   - ⋆ **Cards**
   - ⋆ **Piles** - Génère et manipule des piles de cartes
   - ⋆ **Game**

⋆ **Program**

2. Le code source de Program.cs vous est fourni, il n'attend aucune modification.
3. Le code source de CardInfo.cs vous est fourni, il n'attend aucune modification. Vous n'êtes pas tenu de l'utiliser si vous n'en avez pas besoin. Il vous apporte une structure de données permettant de stocker les statistiques associées à une carte. Ce n'est pas une classe.
4. Commencer par réaliser **un diagramme de classes** en organisant les classes les unes par rapport aux autres (Quelle classe fait appel au constructeur de quelle classe ? Où sont les instances de telle ou telle classe ? Quelle classe aura besoin de quelle classe ? ...).
5. Commencer ensuite l'implémentation par les classes qui ne font appel à aucune autre et remonter ensuite de proche en proche jusqu'à terminer par la classe qui supervise le plus d'instances.

Améliorations :

Pour compléter votre travail, voici une liste d'améliorations. **Il est attendu de vous que vous en réalisiez au moins 2**, celles de votre choix. Le travail n'est pas guidé, vous êtes autonomes sur l'implémentation.

1. Jouer avec deux dés et ajouter des cartes qui s'activent entre 7 et 12.
2. Modifier certaines cartes pour qu'elles puissent s'activer pour deux valeurs de dé successives (comme la boulangerie dans le jeu initial).
3. Proposer une stratégie pour l'IA qui ne repose pas uniquement sur un choix aléatoire.
4. Proposer ce jeu dans une fenêtre graphique.
5. Proposer un choix sur les conditions de fin de partie (20 pièces pour une partie standard, 10 pour une partie rapide et 30 pour une version longue). Ajouter également une condition de fin de partie "expert" pour laquelle la partie prendra fin uniquement lorsqu'un joueur aura en sa possession un exemplaire de chaque carte et 20 pièces.