

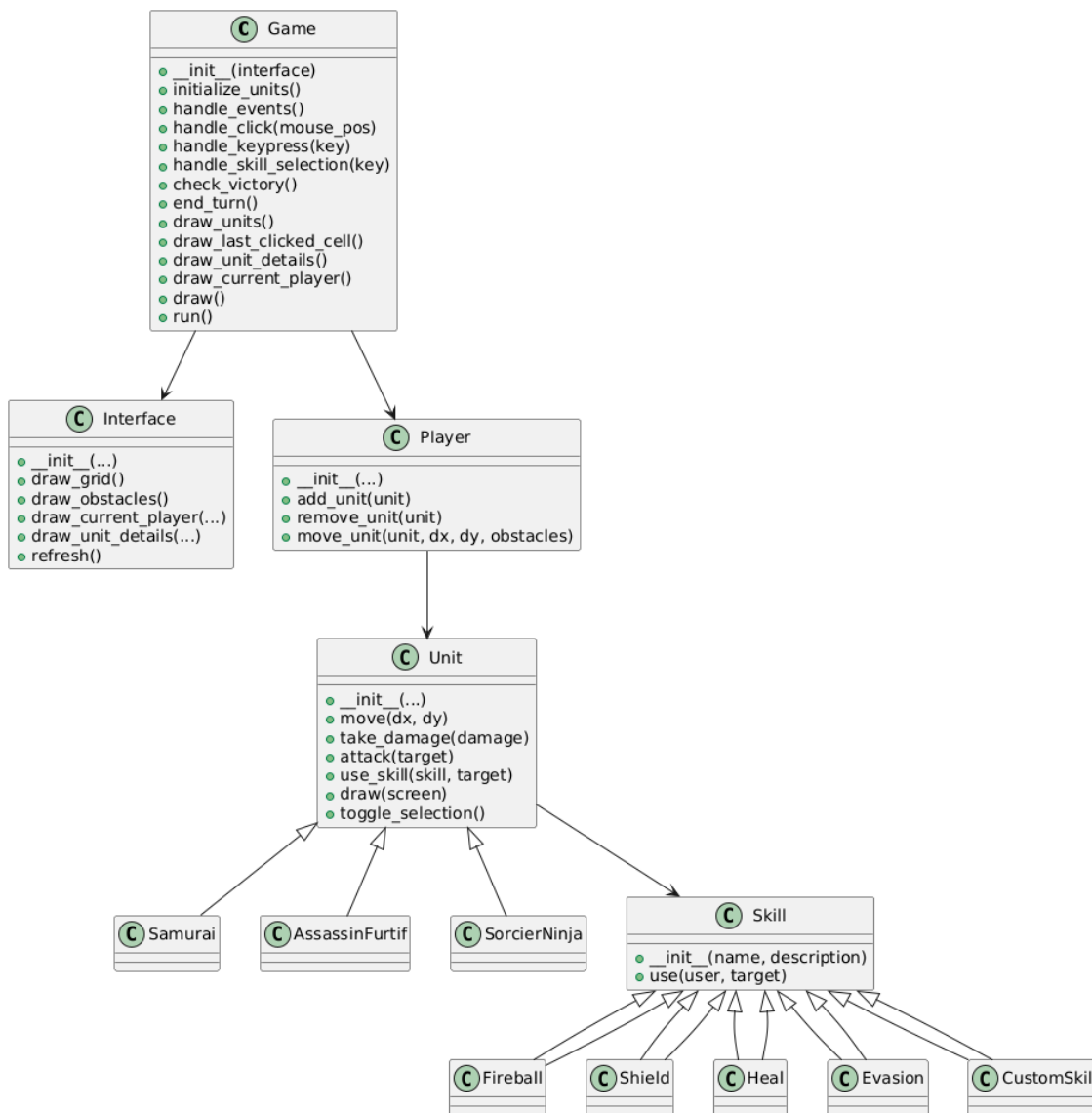
Rapport et Diagramme de Classes pour le Jeu de Combat Ninja

1. Diagramme de Classes

Le diagramme de classes pour le jeu présente une structure claire et organisée :

- La classe abstraite Unit regroupe les fonctionnalités communes pour toutes les unités.
- Trois classes dérivent de Unit : Samurai, AssassinFurtif et SorcierNinja. Elles représentent les différents types d'unités avec des compétences spécifiques.
- Player est une classe qui représente chaque joueur. Elle gère les unités qui lui appartiennent.
- Skill est la classe de base pour toutes les compétences. Les classes dérivées comme Fireball, Heal et Shield définissent des effets particuliers pour chaque compétence.

Le diagramme inclut des relations d'héritage pour montrer la spécialisation des classes et des relations d'association pour modéliser les interactions entre les joueurs, les unités et les compétences.



2. Justification des Relations

Les relations de notre diagramme sont justifiées comme suit :

- **Héritage** : La classe abstraite Unit permet de regrouper les attributs et méthodes communs à toutes les unités (déplacement, attaque, utilisation de compétences). Cela évite la duplication de code.
- **Spécialisation** : Les classes Samurai, AssassinFurtif et SorcierNinja héritent de Unit et ajoutent des compétences spécifiques, rendant chaque type d'unité unique.
- **Association** : Chaque Player contrôle un ensemble d'unités. Les unités sont aussi associées à des compétences via la classe Skill. Cela permet d'organiser les actions que chaque unité peut exécuter.

- Composition : Les unités sont étroitement liées à leur joueur. Si un joueur est éliminé, ses unités n'ont plus de raison d'exister.

3. Répartition du Travail

Notre équipe était composée de deux membres :

- Membre 1 : Responsable de l'interface graphique, incluant la gestion de la grille, l'affichage des unités et des obstacles avec Pygame.
- Membre 2 : Responsable des fonctionnalités du jeu, incluant le développement des classes Game, Player, Unit, ainsi que des compétences via la classe Skill et ses sous-classes.

4. Fonctionnalités Supplémentaires

En plus des fonctionnalités de base, nous avons ajouté :

- Une interface graphique utilisant Pygame pour afficher la grille et les unités de manière visuelle.
- Un système de compétences variées telles que Fireball pour infliger des dégâts, ou Heal pour soigner.
- La gestion des obstacles pour rendre les déplacements plus stratégiques.
- Un système de jeu au tour par tour pour alterner les actions entre les deux joueurs.

5. Conclusion

Le diagramme de classes et l'implémentation du jeu respectent les principes de conception orientée objet. La classe abstraite Unit permet une centralisation des comportements communs, tandis que l'héritage et les relations d'association assurent une architecture claire et extensible. Ce projet nous a permis d'acquérir des compétences techniques en programmation objet et en développement d'une interface graphique.