

Rapport

Les éléments traités :

Lors de ce projet différents aspects ont été traités afin de réaliser les jeux Dames, Butin et Safari. Dans un premier temps, on a identifié les opportunités de factorisation du code. Cela a mené à la création d'une classe de base Jeu, qui encapsule les éléments communs tels que la gestion du damier et le cycle de jeu. Chaque jeu spécifique a ensuite été développé comme une sous-classe de Jeu, exploitant ainsi l'héritage pour étendre ou modifier le comportement de base. Nous avons porté une attention particulière à identifier et corriger les erreurs dans notre code. Cela a impliqué de s'assurer que le programme reste stable même en cas de problèmes inattendus, améliorant ainsi sa fiabilité.

Nous avons aussi mis en place des mécanismes pour gérer correctement la création et la suppression des éléments dans le jeu, ce qui est crucial pour éviter les problèmes de mémoire et assurer la cohérence des données, surtout dans les situations où plusieurs éléments interagissent ou changent. Pour rendre les jeux plus intuitif, nous avons développé une interface graphique. Cela a rendu les jeux plus accessibles et a fourni une interaction plus intuitive avec le programme.

Les problèmes connues :

- Problème au niveau de la fonction `AnimalCapture()` dans le jeu Safari, impactant le calcul pour déterminer si un animal est capturé ou non.
- Problème d'affichage des barrières dans l'interface graphique du jeu. Bien que les barrières existent et fonctionnent correctement dans le jeu, elles ne sont pas correctement représentées visuellement.

Diagramme UML :

