

Copter Battle



UNIVERSITÉ ÉVRY
PARIS-SACLAY

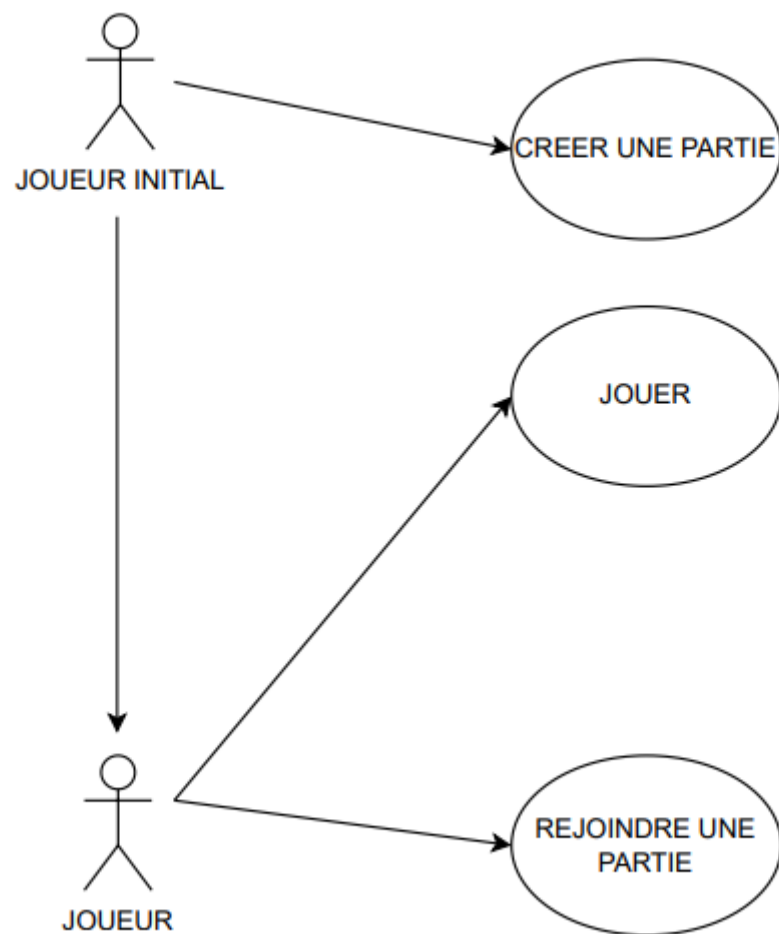
Yacine OUILLEDIRNE

Kylian RIGAULT

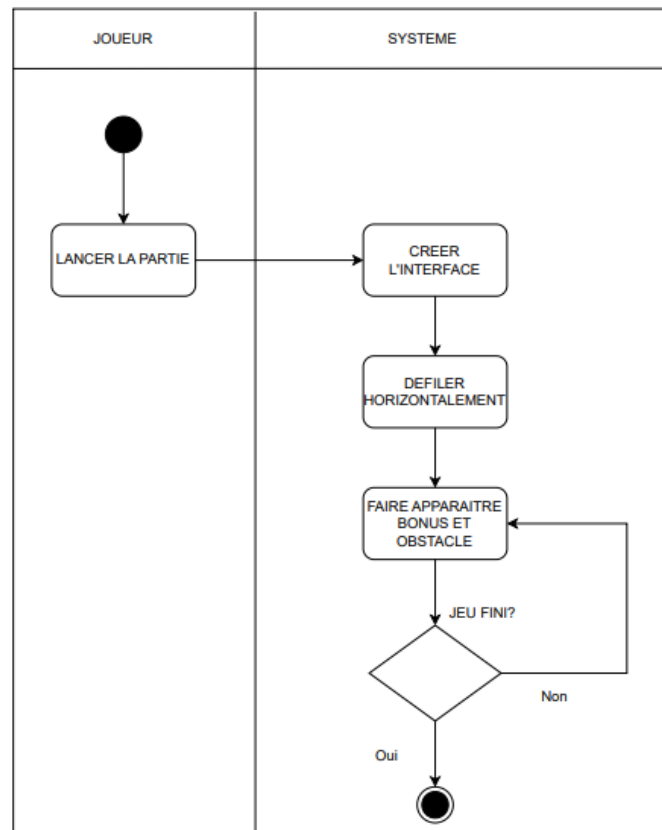
Chanfi MOHAMED

Yanis HADJ-CHERIF

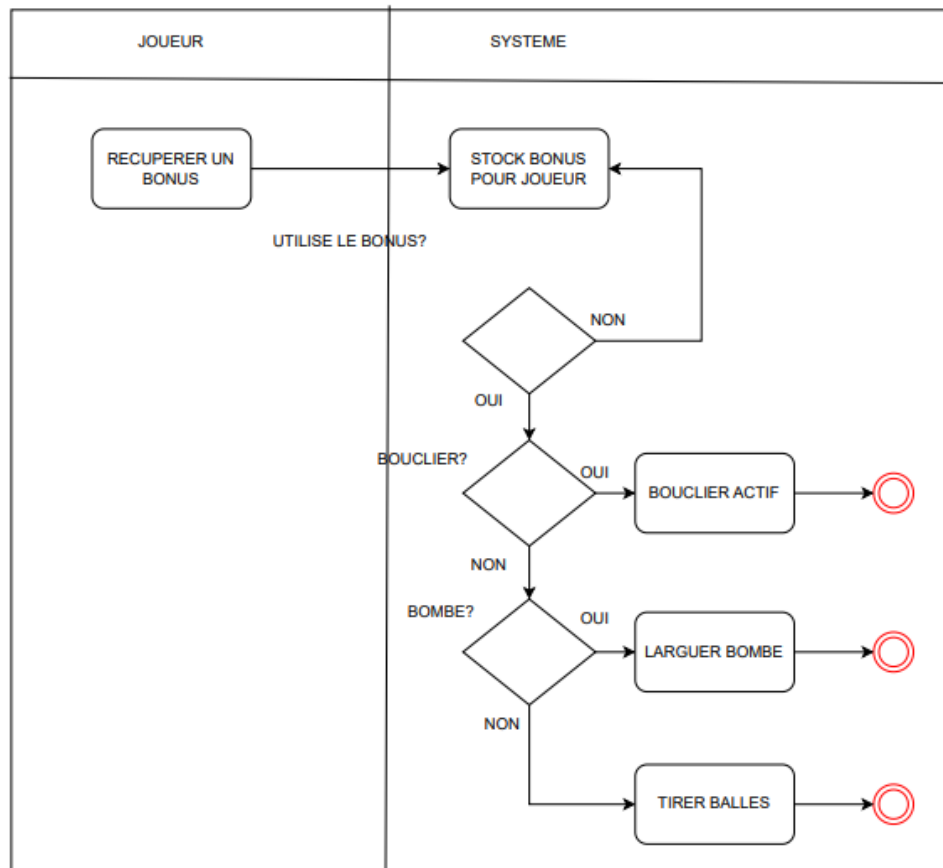
Modélisation



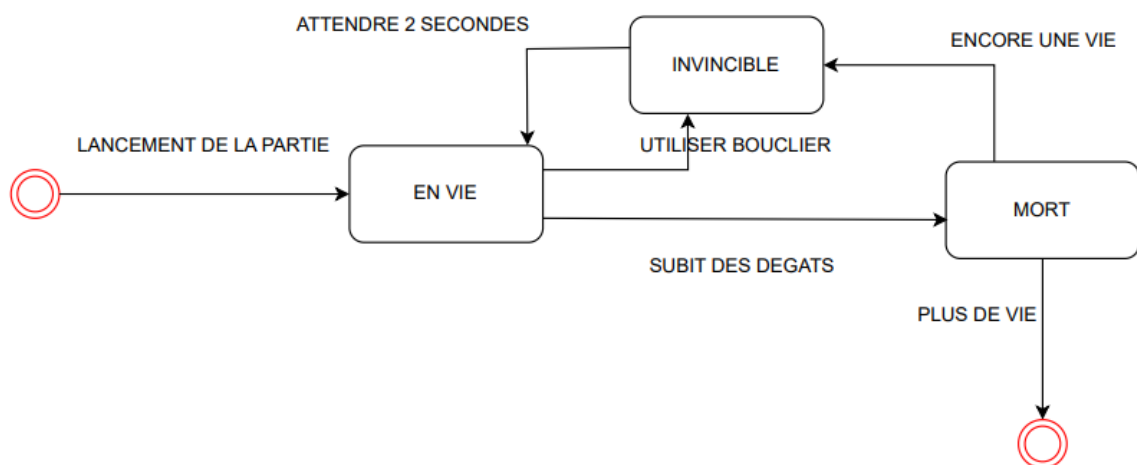
Le diagramme de cas d'utilisation présente les acteurs principaux qui auront un impact sur ce projet selon les interactions possibles. On peut ainsi distinguer que 2 types de joueurs sont identifiés : celui qui crée la partie et celui qui rejoint.



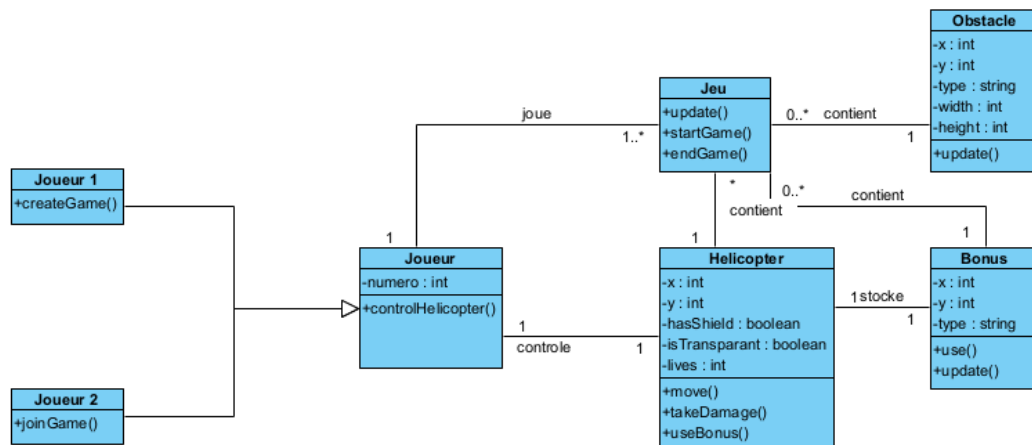
Le diagramme d'activité détaille le lancement du jeu, initié par le joueur, où l'on peut apercevoir une boucle qui permet l'apparition des objets et des obstacles tant que le jeu n'est pas fini.



Ce diagramme décrit la gestion des bonus où le système stocke et identifie le bonus lors de son activation.



Ce diagramme d'état illustre le cycle de vie des hélicoptères durant une partie passant par plusieurs états selon les actions subies.



Ce diagramme de classe représente la structure du jeu où les deux types de joueurs héritent de la classe joueur. La classe Jeu coordonne les interactions entre les autres classes.

Affectations/Planning

On découpera ce projet en **4** grandes parties :

1. Modélisation
2. Planification
3. Conception du jeu
4. Intégration, tests et corrections

Les taches de ces parties seront distribuées de manière équivalente afin que chacun ait le même montant de taches.

Modélisation

Cette première étape consiste à réaliser la modélisation UML du projet afin de définir clairement l'architecture globale du jeu avant de commencer le développement. Elle

permet d'identifier les différentes classes, leurs rôles, ainsi que les interactions entre les éléments du système, garantissant ainsi une conception cohérente et structurée.

Taches et nombre de membre nécessaire pour la réalisation :

- Identification des diagrammes pertinents à faire : travail collectif.
- Réalisation des diagrammes : travail collectif.
- Cohérence des diagrammes et leur utilité : travail collectif.

Planification

Cette phase a pour objectif d'organiser le travail du groupe en répartissant les tâches entre les membres et en définissant un ordre logique de réalisation des différentes fonctionnalités. Elle permet d'anticiper les dépendances techniques et d'assurer une progression efficace du projet.

Taches et nombre de membre nécessaire pour la réalisation :

- Découpage du projet en grandes parties : travail collectif.
- Répartition des taches dans le groupe : travail collectif.

Conception du jeu

Cette partie correspond au développement de l'ensemble des fonctionnalités du jeu. Elle comprend la mise en place du moteur du jeu, la gestion des déplacements, des obstacles, des bonus, des interactions entre joueurs, ainsi que l'affichage visuel et la logique complète du gameplay.

Taches et nombre de membre nécessaire pour la réalisation :

- Création de la fenêtre du jeu et de la boucle principale : 1 membre.
- Développement de la classe Hélicoptère et de ses déplacements : 1 membre.
- Gestion des limites de l'écran et du scrolling horizontal : 1 membre.
- Création des obstacles et génération aléatoire : 1 membre.
- Mise en place du système de collisions : 1 membre.
- Développement des bonus (bouclier, rafale de tir, bombe) : 1 membre.
- Gestion des interactions entre les deux joueurs : 1 membre.

- Implémentation des tirs et des impacts : 1 membre.
- Ajout des animations et effets visuels : 1 membre.
- Mise en place du HUD (vie, informations joueur) : 1 membre.
- Développement de la logique complète du jeu (vies, dégâts, fin de partie, redémarrage) : 1 membre.

Intégration, tests et corrections

La dernière étape consiste à tester l'intégralité du jeu dans différentes situations afin de détecter les éventuels dysfonctionnements. Elle inclut la correction des bugs, l'optimisation des performances, le nettoyage du code et la préparation du projet pour le rendu final.

Taches et nombre de membre nécessaire pour la réalisation :

- Intégration de tous les modules : travail collectif.
- Test du jeu : travail collectif.
- Corrections des bugs : travail collectif.