Projet Midiwar

Enseignants:

Franck Gechter - Alexandre Brunoud - Badreddine Chah

Membres du groupe
Michel DIENG
Sakhewar DIOP

Sommaire:

Présentation du projet

La modélisation, dans son sens le plus large, est un processus qui vise à représenter un phénomène, un système ou une idée en suivant une norme spécifique. Son objectif principal est de simplifier la compréhension d'entités complexes. Dans le domaine de l'ingénierie, elle offre une représentation claire du système pour le concepteur et sert de base de compréhension et de communication pour les autres parties prenantes.

Parmi les nombreux langages ou méthodes de modélisation disponibles aujourd'hui, l'UML (Unified Modeling Language) se distingue par sa facilité de compréhension et son intégration de l'approche orientée objet (OO), qui est devenue essentielle. Par conséquent, la maîtrise des concepts de la modélisation UML est un défi crucial pour tout futur ingénieur, car elle peut être appliquée aussi bien à des systèmes existants qu'à des créations originales.

C'est dans cette perspective que l'apprentissage de la modélisation UML a été intégré à un projet d'équipe, offrant ainsi plus d'opportunités pour mettre en pratique les concepts appris et se préparer aux tâches qui seront courantes dans le monde professionnel.

Pour vous donner une meilleure idée de notre travail, nous commencerons par contextualiser le thème choisi, puis nous détaillerons les spécifications de la conception et le calendrier de réalisation de notre projet. Enfin, nous présenterons le projet lui-même et partagerons notre retour d'expérience.

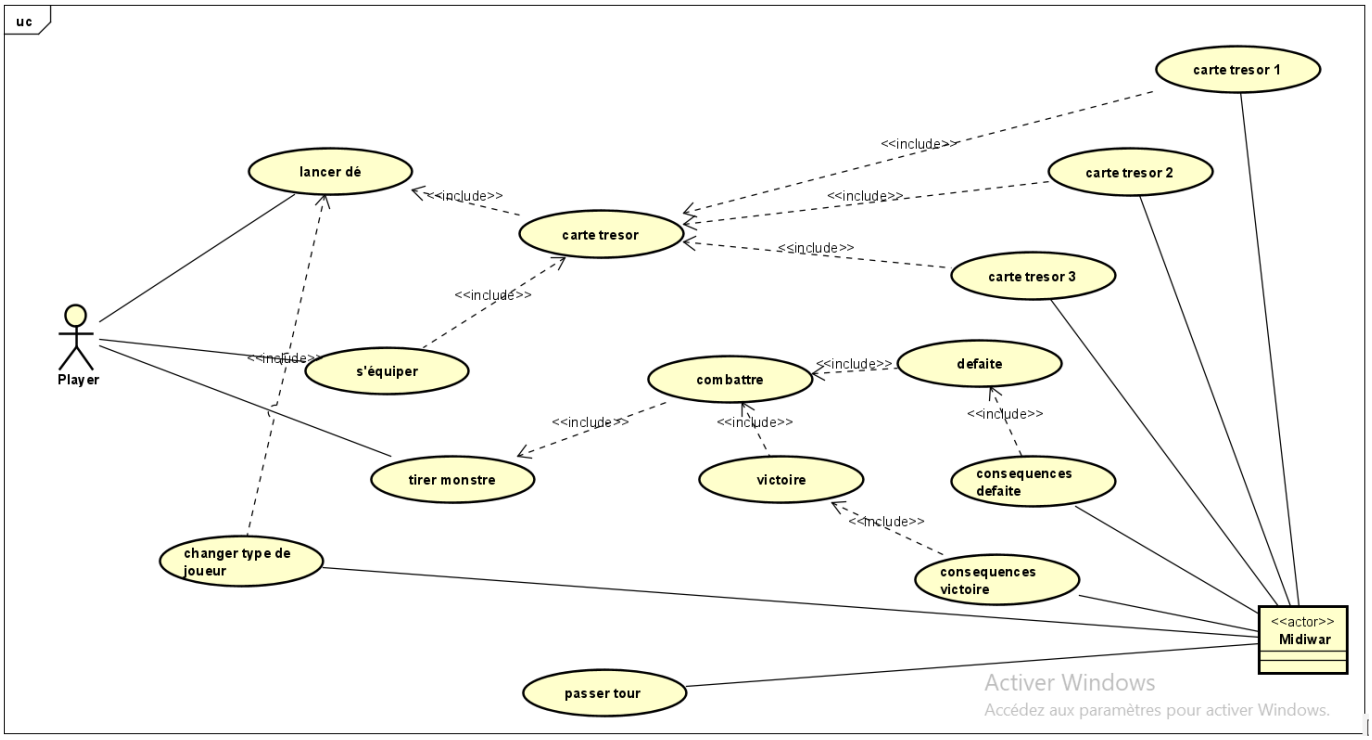
Fonctionnement du jeu

Au début du jeu, le joueur qui commencera la partie est décidé au hasard. Pour les tours suivants, l'ordre de jeu suit le sens des aiguilles d'une montre pour déterminer le joueur suivant.
Au début de la partie, tous les joueurs sont de niveau 1, et chaque joueur commence par lancer un dé pour déterminer les trésors dont il disposera au départ. Si un 1 est obtenu, le joueur commence avec 1 trésor, s'il obtient entre 2 et 5, il commence avec 2 trésors, et s'il obtient un 6, il commence avec 3 trésors. Les trésors que le joueur reçoit sont toujours ajoutés à ses trésors cachés.
Avant chaque combat, et avant de connaître le monstre auquel il fera face, le joueur peut équiper les trésors qu'il souhaite, à condition de respecter les règles sur la quantité et les types de trésors qui peuvent être équipés. Équiper un trésor implique de le retirer du groupe des trésors cachés et de le passer à celui des équipés.
Une fois que le joueur est équipé, il retourne la carte qui est au sommet de la pioche de cartes de monstres pour livrer un combat. Le combat consiste à comparer le niveau de combat des deux adversaires. Si le niveau de combat du joueur (son niveau actuel plus les bonus de tous ses trésors équipés) est supérieur au niveau de combat du monstre, le joueur est vainqueur et les avantages associés au monstre sont appliqués. Cela lui fera augmenter son niveau et/ou recevoir des trésors. Les trésors reçus sont toujours ajoutés au groupe des trésors cachés.
Si le niveau de combat du joueur n'est pas supérieur à celui du monstre, le joueur perd. Cela signifie qu'il doit faire face aux conséquences négatives associées à ce monstre. Cela peut entraîner une diminution de son niveau et/ou une perte de trésors. Si les conséquences négatives impliquent la perte de trésors, le joueur devra se défaire d'un certain nombre ou type de trésors spécifiés pour les subir. Il ne sera pas possible de passer au tour suivant sans avoir complètement subi les conséquences négatives.
Enfin, le niveau minimum d'un joueur est de 1, donc, indépendamment de ce que dit la conséquence négative, le joueur n'aura jamais un niveau négatif ou zéro. Le respect des conséquences négatives est toujours conditionné par les cartes dont dispose le joueur à tout moment. Ainsi, si la conséquence négative indique que le joueur doit se défaire de certains trésors dont il ne dispose pas, le joueur est exempté de respecter la partie de la conséquence qu'il ne peut pas assumer.
Dans le cas où il est indiqué qu'il doit perdre plus de trésors qu'il n'en a, le joueur respectera les conséquences négatives dans la mesure de ses possibilités et en fonction des trésors dont il dispose. Par exemple, si une conséquence négative indique qu'un joueur doit perdre 2 armures cachées et un trésor de main visible, et qu'il n'a qu'une seule armure cachée et aucun trésor de main visible, il ne devra se défaire que de son unique armure cachée et la conséquence négative sera considérée comme respectée. Parmi tous les trésors qui peuvent faire respecter la conséquence négative, le joueur peut choisir de se défaire de ceux qu'il souhaite. Par exemple, si la conséquence négative lui dit de se défaire d'un trésor visible, il peut décider lequel.
Il existe un type de conséquence négative qui entraîne la mort du joueur, bien qu'il lui soit permis de continuer à jouer. En l'appliquant, le joueur perd tous les trésors dont il dispose (à la fois équipés et cachés), et son niveau est fixé à 1. Si un joueur meurt, lorsqu'il reprendra son tour, il se verra à nouveau attribuer des trésors en utilisant la même méthode qu'au début de la partie.

Une fois le combat et l'application des conséquences négatives ou positives terminés, le joueur pourra équiper des trésors et/ou se défaire de trésors. Ensuite, le joueur pourra voler un trésor à son archiennemi. Chaque joueur ne pourra le faire qu'une seule fois dans la partie. Le vol d'un trésor consiste à choisir l'un des trésors cachés de son archiennemi et à l'ajouter à sa liste de trésors cachés. Le joueur qui vole doit choisir la carte de trésors à l'aveugle, sans connaître la carte qu'il vole.
Une fois le processus hypothétique de vol de trésors terminé, si le joueur est dans un état valide en termes de trésors équipés et cachés, et en termes de respect des conséquences négatives, il passera son tour pour que le joueur suivant joue.
Joueurs sectaires
Pendant le développement des pratiques, un aspect supplémentaire sera introduit en ce qui concerne les joueurs. Si un joueur perd un combat et ne meurt pas en subissant les conséquences négatives, il a la possibilité de relancer le dé. S'il obtient un 6, le joueur devient un joueur sectaire.
La transformation en joueur sectaire implique de prendre une carte de sectaire. Cette carte indiquera un certain niveau. Ainsi, le niveau de combat d'un joueur sectaire sera le niveau de combat d'un joueur non sectaire augmenté de 20%, auquel s'ajoute le niveau indiqué sur sa carte de sectaire multiplié par le nombre de joueurs sectaires en jeu. Les joueurs sectaires se comportent également différemment lorsqu'ils sont volés, car au lieu d'un trésor caché, ils doivent céder une carte de trésor visible. Dans tous les cas, la carte perdue sera déterminée au hasard, et le joueur qui va la recevoir ne pourra pas choisir une carte spécifique.
À ce stade, de nouvelles cartes de monstres seront ajoutées qui incluent une augmentation ou une réduction du niveau de combat lorsqu'elles sont confrontées à des joueurs sectaires. Une fois qu'un joueur devient sectaire, il ne perd jamais cette condition.

Diagramme de cas d'utilisation

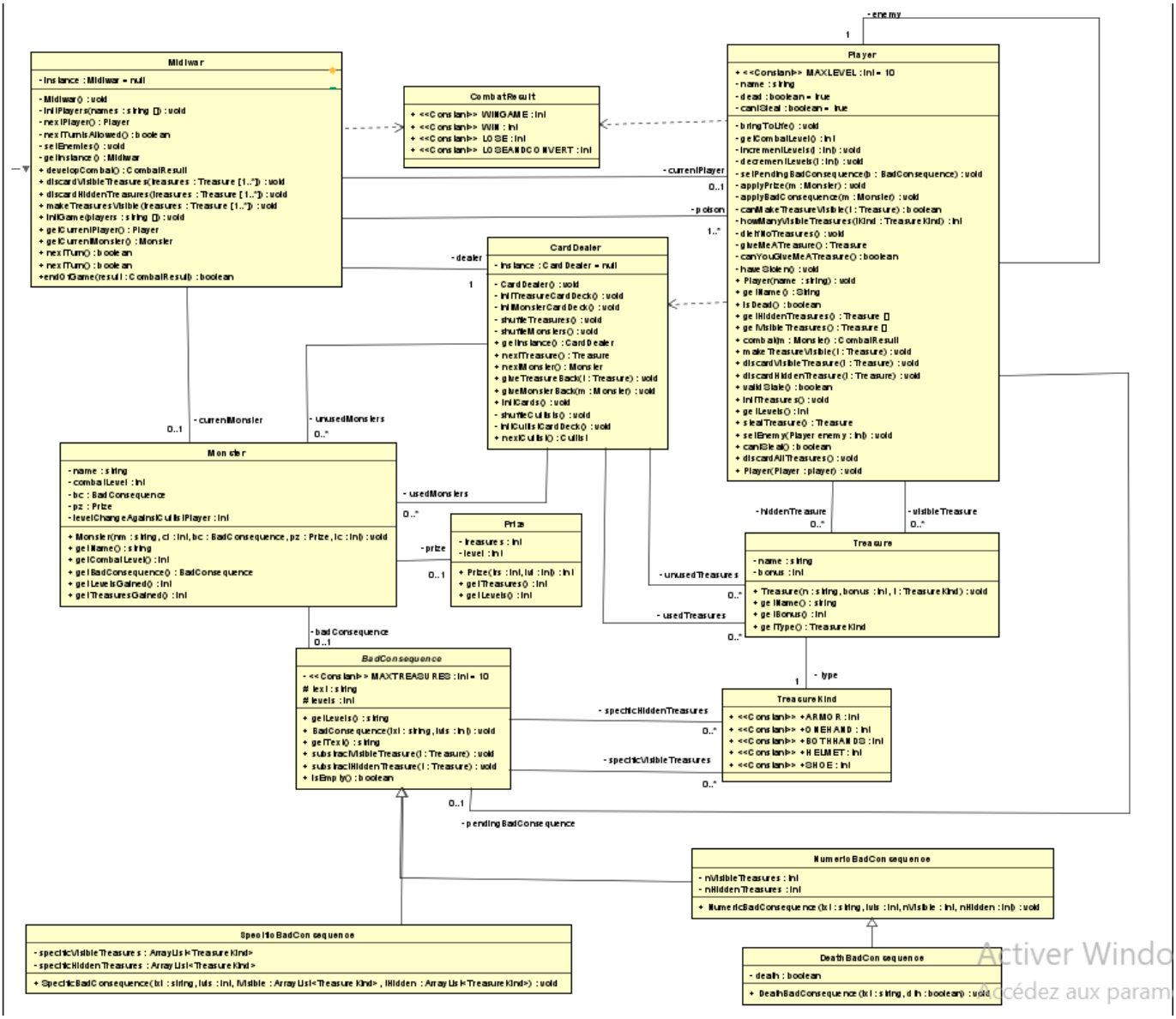
Au regard des fonctionnalités qui ont été définis dans la partie précédente, et des différentes contraintes sur ses fonctionnalités, nous avons donc obtenu le diagramme de cas d'utilisation suivant:



On peut bien se rendre compte que chaque joueur, représenté par l'acteur Player, interagit avec le système de manière dynamique et stratégique. Au début de la partie, le joueur lance un dé pour déterminer les trésors initiaux et peut les équiper avant d'affronter un monstre tiré de la pioche. Les combats se résolvent en comparant les niveaux de combat du joueur et du monstre, influencés par les trésors équipés. Les conséquences positives ou négatives sont ensuite appliquées, affectant le niveau et les trésors du joueur. Les règles strictes concernant l'équipement et la gestion des conséquences ajoutent une dimension tactique au jeu. De plus, l'introduction des joueurs sectaires offre une variante intrigante, modifiant les règles traditionnelles lors des combats et des vols de trésors. Ces interactions englobent un ensemble complexe de décisions stratégiques, de risques calculés et d'ajustements continus, créant une expérience de jeu riche et immersive pour chaque joueur.

Diagramme de classe

Compte tenu du langage de développement : le Java. Et du fait que celui-ci est un langage Orienté objet, nous avons donc décider de partir par la suite sur un diagramme de classe afin de mieux structurer notre jeu. Conformément à la mise en contexte du projet, le diagramme qui en découle est le suivant :



Le code du jeu "Midiwar" est structuré autour de plusieurs classes clés, chacune jouant un rôle spécifique dans la modélisation des composants du jeu. La classe Player représente les joueurs du jeu, avec des méthodes pour gérer l'équipement des trésors, les combats contre les monstres, et la gestion des conséquences positives et négatives. Les classes Monster, Prize, et

BadConsequence modélisent respectivement les monstres du jeu, les récompenses gagnées en cas de victoire, et les conséquences néfastes en cas de défaite. Les classes NumericBadConsequence et SpecificBadConsequence étendent BadConsequence en définissant des conséquences spécifiques, telles que la perte numérique de trésors ou la perte de types spécifiques de trésors. La classe DeathBadConsequence gère les conséquences de mort pour les joueurs. Enfin, la classe Treasure représente les trésors, avec une énumération TreasureKind définissant les différents types de trésors. Ces classes interagissent de manière cohérente pour simuler le flux du jeu, des combats aux conséquences, en passant par l'équipement et la gestion des niveaux. Cette structuration offre une base solide pour la logique du jeu et permet une extension facile avec l'introduction des joueurs sectaires, qui sont modélisés à travers des mécanismes spécifiques liés à leur transformation et à leurs interactions uniques avec les trésors.

Diagramme de séquences

Les diagrammes de séquences suivants vont nous permettre d'obtenir une représentation visuelle des interactions dynamiques entre les composants du système, offrant ainsi une compréhension approfondie du flux d'exécution et des échanges d'informations.

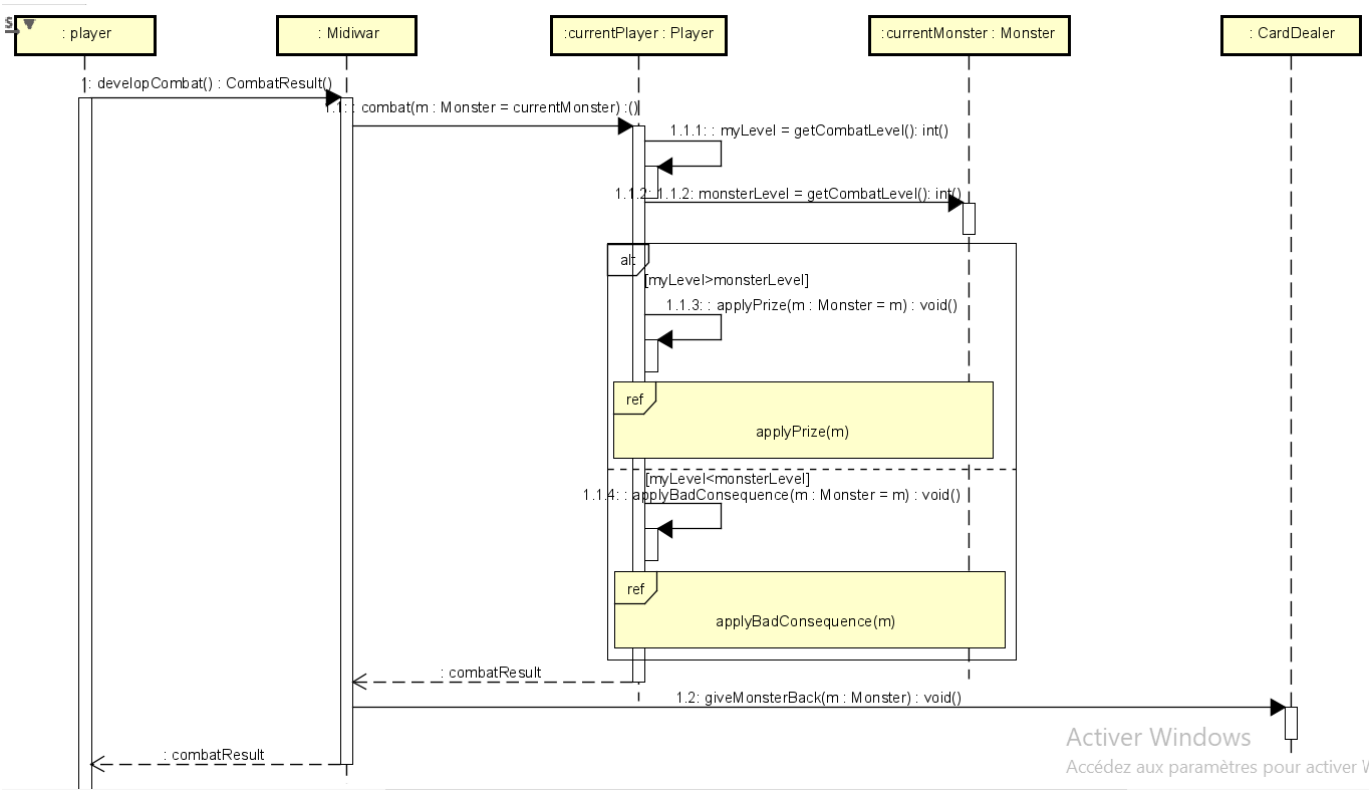


Diagramme de séquence Combat

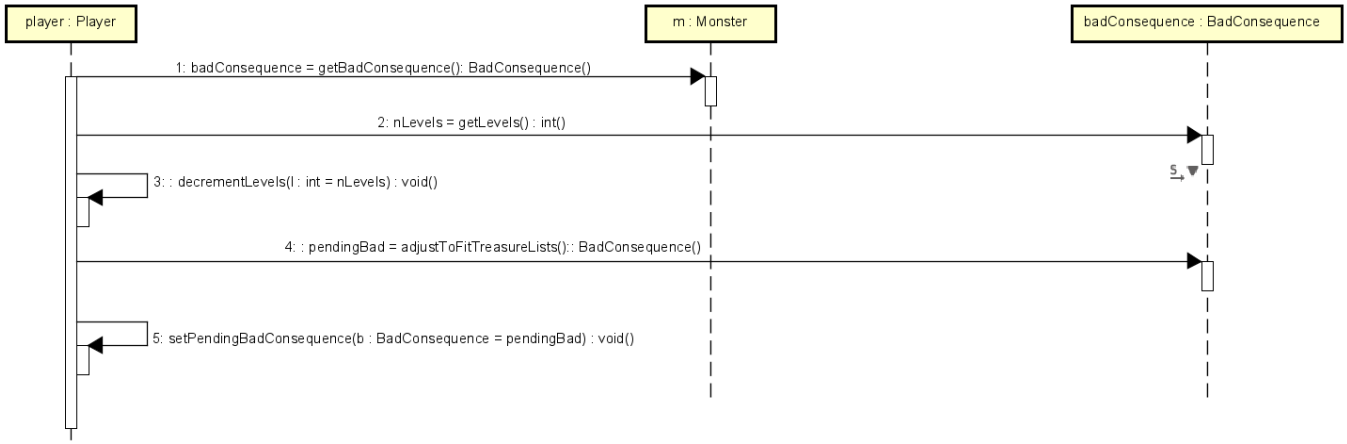


Diagramme de séquence applyBadConsequence

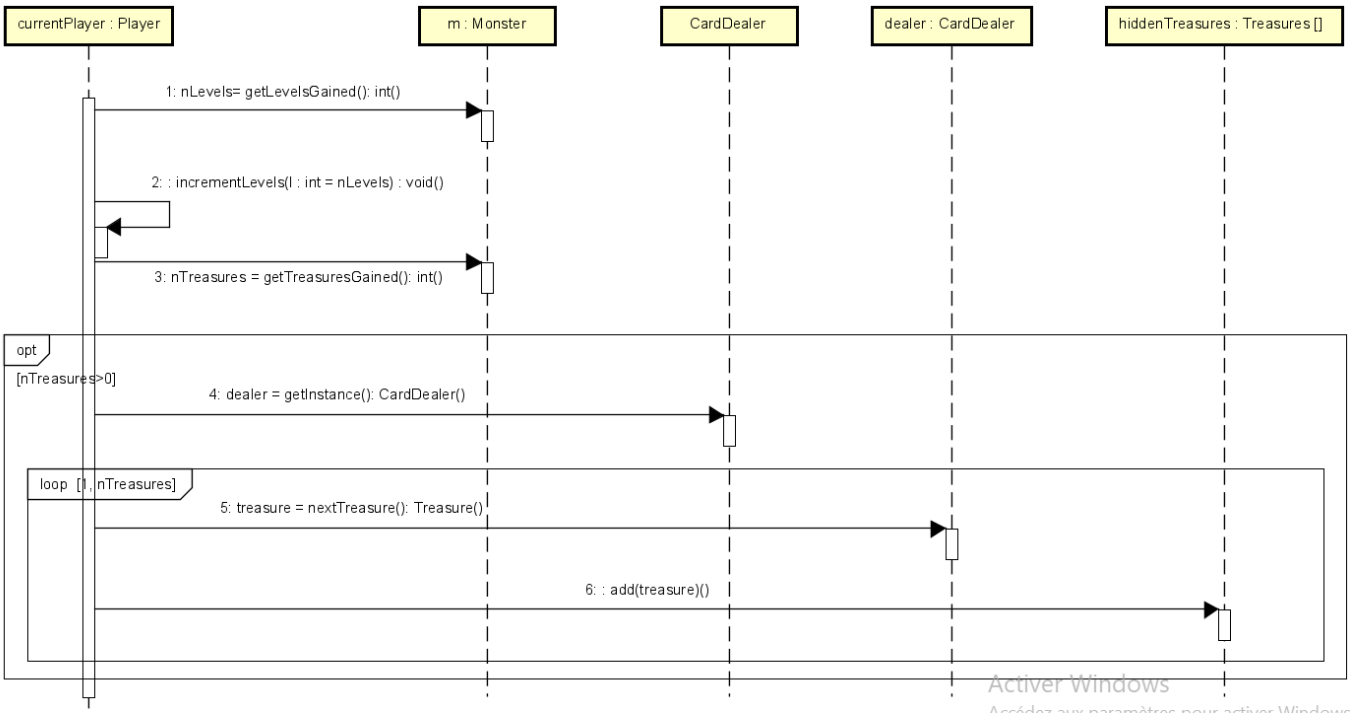


Diagramme de séquence applyPrize

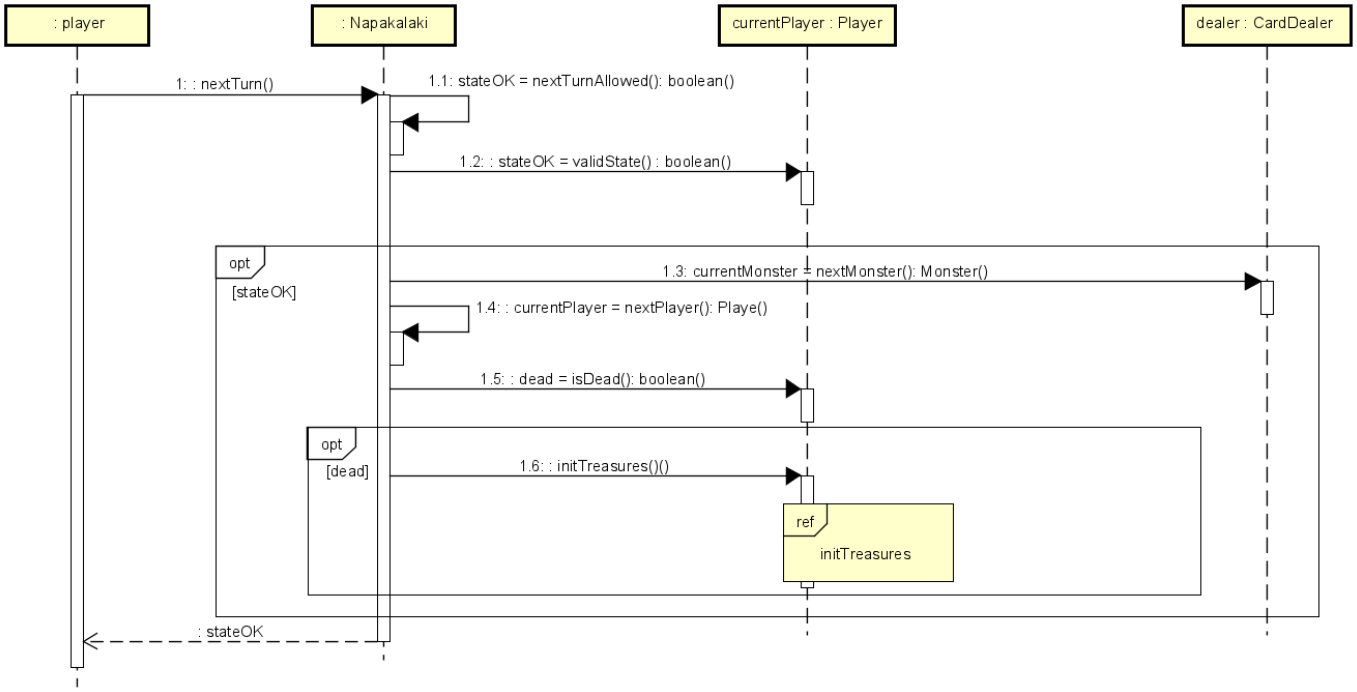


Diagramme de séquence nextTurn

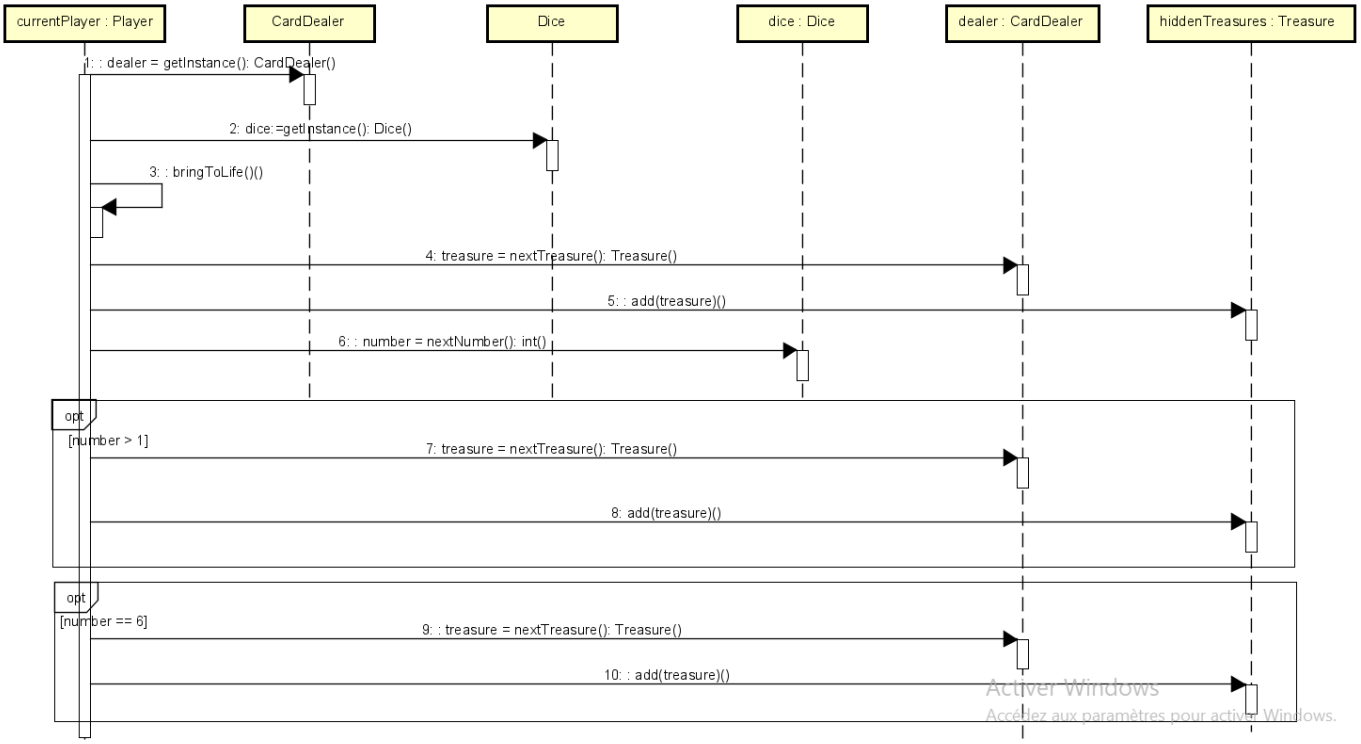


Diagramme de séquence initTreasure

Les diagrammes de séquences permettent de comprendre comment interagissent les joueurs avec les différentes classes pour mener des combats, ajuster leur équipement, et répondre aux défis du jeu. Ainsi, les différentes classes du système travaillent en tandem pour offrir une expérience de jeu riche, où les choix des joueurs et les résultats des combats influencent directement leur progression dans le jeu.

Conclusion

La conception et le développement de Midiwar ont constitué une expérience pratique enrichissante en programmation orientée objet. La mise en œuvre des fonctionnalités clés, telles que la gestion des joueurs, des monstres et des trésors, a permis de créer une structure solide. Les diagrammes de classes et de séquences ont été des outils essentiels pour visualiser les interactions du système.

Cependant, des défis ont émergé, notamment la gestion des états des joueurs et les interactions complexes entre les trésors. L'ajout des joueurs sectaires a introduit une complexité supplémentaire. Malgré un planning équilibré, des retards ont été rencontrés dans la résolution de problèmes liés aux règles du jeu.

En conclusion, Midiwar a été un projet stimulant, renforçant la compréhension de la programmation orientée objet. Les difficultés ont été surmontées avec une approche méthodique.