

## Rapport

# PROJET JEU DE L'OIE ENSICAEN

le 13/10/2023,  
version 1.1

Cécile LU  
Winnie KAMTCHUENG  
Paul NGUYEN  
Maxime JINGEAUX, Ely  
SERAIDARIAN, Kévin HERMAN

Tuteur école : Régis CLOUARD  
Tuteur TP: Anthony ROGER



[www.ensicaen.fr](http://www.ensicaen.fr)

# TABLE DES MATIÈRES

---

<b>1. INTRODUCTION</b>	<b>4</b>
1.1. Contexte	4
1.2. Règles du Jeu	4
1.3. Rôles et fonctions dans l'équipe	6
1.3.1. Entente et collaboration au sein de l'équipe	6
<b>2. Conception du Jeu</b>	<b>6</b>
2.1. Méthodologie de conception	6
2.1.1. Diagramme des Cas D'Utilisation	6
2.1.2. Diagramme des risques	7
2.1.3. Diagramme des paquets	8
2.2. Conception du plateau de jeu	9
2.3. Fonctionnalités	9
<b>3. Développement</b>	<b>9</b>
3.1. Architecture du Jeu	9
3.2. Patron de conception	11
3.3. MVP n°1	12
3.4. MVP n°2	13
<b>4. Perspectives Futures</b>	<b>14</b>
4.1. Possibilités d'amélioration	14
4.2. Prochaines étapes	15
<b>5. Conclusion</b>	<b>16</b>
5.1. Récapitulatif des accomplissements	16
5.1.1. Accomplissement sur le modèle	16
5.1.2. Accomplissement sur la vue	16
5.2. Difficultés rencontrées	16
5.2.1. Difficultés sur la communication	16
5.2.2. Difficultés au niveau du développement	17

## TABLE DES FIGURES

---

Figure 1 Plateau du jeu de l'oie traditionnel	4
Figure 2 Diagramme des cas d'utilisation	7
Figure 3 Diagramme des paquets	8
Figure 4 Patron d'architecture MVP	10
Figure 5 patron fabrique abstrait	11
Figure 6 Classe abstraite AbstractFactoryStudent du jeu	11
Figure 7 Classe Interface case et quelques cases	12
Figure 8 MVP 1	12
Figure 9 MVP 2	14
Figure 10 Ajout d'un autre parcours initial, Master	15
Figure 11 Ajout d'un nouvel attribut	15

## TABLE DES TABLEAUX

---

Tableau 1 Description des cases	5
Figure 2 Tableau des risques	8

# 1. INTRODUCTION

## 1.1. Contexte

Le projet de jeu de l'oie a été initié dans le cadre de la création de notre entreprise dans le domaine des jeux numériques de plateau. Notre client, qui représente une célèbre école d'ingénieurs française, nous demande de concevoir un jeu pour leur activité promotionnelle. Notre jeu de l'oie se veut une interprétation moderne de ce classique, avec des éléments représentatifs de l'école d'ingénierie, des fonctionnalités interactives et une expérience utilisateur conviviale.

## 1.2. Règles du Jeu

Les règles du jeu de l'oie que nous avons développées sont conçues pour offrir une expérience de jeu amusante et engageante. Le jeu commence avec chaque joueur qui trouve son pion sur la case de départ. Les joueurs lancent ensuite un dé virtuel, dont le résultat détermine le nombre de cases qu'ils avancent sur le plateau. Les cases spéciales incluent des cases bonus qui permettent aux joueurs de relancer les dés et des cases de retour en arrière qui les font reculer de quelques cases.

L'objectif du jeu est d'atteindre la case d'arrivée en premier. Les joueurs doivent obtenir le nombre exact requis pour atteindre cette case, sans dépasser, sinon ils rebroussement chemin.

Le jeu de l'oie que nous avons créé est destiné à être joué par plusieurs joueurs en simultané, chaque joueur lançant les dés à tour de rôle. Les règles sont simples et accessibles, ce qui le rend adapté à un large public, y compris les familles et les amis qui cherchent un moyen divertissant de passer du temps ensemble.



Figure 1 Plateau du jeu de l'oie traditionnel

Cases Jeu de l'Oie standard	Cases Jeu de l'Oie ENSICAEN	Description	Numéro de cases
Ponts	Révisions Intensives	Gagne 1 point de compétence et avance jusqu'à 12	6
Oles	Kangourous	Rejoue du même résultats de dés	Multiple de 9
Dés	Dés	Au premier tour, si un joueur fait le résultat (6 + 3), il se déplace sur la case 26, et s'il fait (5 + 4), il se déplace sur la case 53	
Labyrinthe	TOEIC Raté	Le joueur recule de 12 cases	42
Puits	BDE	Le joueur est bloqué et passe son tour, jusqu'à qu'un autre joueur prenne sa place. Il sera bloqué à son tour	31
Hôte	Lendemain de Wei	Le joueur passe son tour	19
Prison	Problème Informatique	Le joueur doit attendre que la DSI règle le problème, il passe son tour jusqu'à qu'un autre joueur le libère en allant sur la case	52
Tête de Mort	Arrêt d'études	Le joueur retourne à la case Départ	58
	Examens	Le joueur doit avoir un niveau de compétences supérieur au cap. Sinon, il recule de la différence de niveau.	13, 32, 48
	Soirées	-1 points de compétences	4, 14, 23, 29, 40, 60
	Stage à l'étranger	+2 points de compétences	37, 38, 39
	Enseignements généraux	+1 points de compétences	Environ 20 cases
	Enseignements spécifiques	+1 points de compétences et +/-1 case en fonction du hard skill	Environ 15 cases

Tableau 1 Description des cases

## 1.3. Rôles et fonctions dans l'équipe

Notre équipe de projet se compose de membres clés ayant des rôles spécifiques pour mener à bien notre projet de jeu de l'oie :

- **Chef de Projet et Développeur** : Cécile LU, assure la gestion globale du projet tout en contribuant activement au développement du jeu.
- **Release Manager et Développeur** : Paul Nguyen gère les versions et les mises à jour du jeu tout en contribuant au développement du jeu.
- **Architecte et Développeur** : Kévin Herman qui est responsable de la conception de l'architecture du jeu et de son implémentation.
- **Architecte et développeuse** : Winnie Kamtchueng se concentre sur la conception de composants clés du jeu, notamment la définition des personnages, en plus de son rôle de développeur.
- **Deux Développeurs sur la Vue** : Maxime Jingeaux et Ely Seraidarian se spécialisent dans le développement de l'interface utilisateur et des éléments visuels du jeu.

Chacun de ces rôles contribue de manière essentielle à la réalisation de notre projet.

### 1.3.1. Entente et collaboration au sein de l'équipe

Le début de notre projet a été marqué par des difficultés liées à la communication au sein de l'équipe. Nous avons rapidement réparti les tâches sans avoir défini de bases solides, ce qui a compliqué notre avancement. Une des erreurs que nous avons commises était de nous focaliser sur les détails spécifiques avant d'établir une vision générale du projet.

Dès que nous avons réalisé que la communication était au cœur de notre problème, nous avons réagi rapidement. Nous avons tenu des discussions approfondies au sein de l'équipe et nous avons échangé avec le coach du projet pour identifier les problèmes et trouver des solutions. Nous avons convenu qu'il était essentiel de travailler sur la communication et de définir des objectifs précis pour le projet.

## 2. Conception du Jeu

### 2.1. Méthodologie de conception

#### 2.1.1. Diagramme des Cas D'Utilisation

Le diagramme des cas d'utilisation est un outil essentiel de notre processus de conception. Il nous a permis de visualiser et de planifier les interactions entre les acteurs et le système, ainsi que de définir les

fonctionnalités clés de notre jeu de l'oie. Voici un aperçu des principaux éléments de notre diagramme des cas d'utilisation :

#### Acteurs :

1. **Joueurs** : Les acteurs principaux du jeu, ils interagissent directement avec le système pour jouer et avancer sur le plateau.

#### Cas d'Utilisation :

1. **Commencer une partie** : Les joueurs peuvent lancer une nouvelle partie de jeu.
2. **Lancer les dés** : Les joueurs peuvent lancer les dés pour déterminer leur avancement sur le plateau.
3. **Choisir la formation initiale** : Les joueurs peuvent sélectionner leur formation initiale, ce qui peut influencer le déroulement du jeu.
4. **Choisir la langue** : Les joueurs ont la possibilité de choisir la langue dans laquelle ils souhaitent jouer, pour une expérience personnalisée.

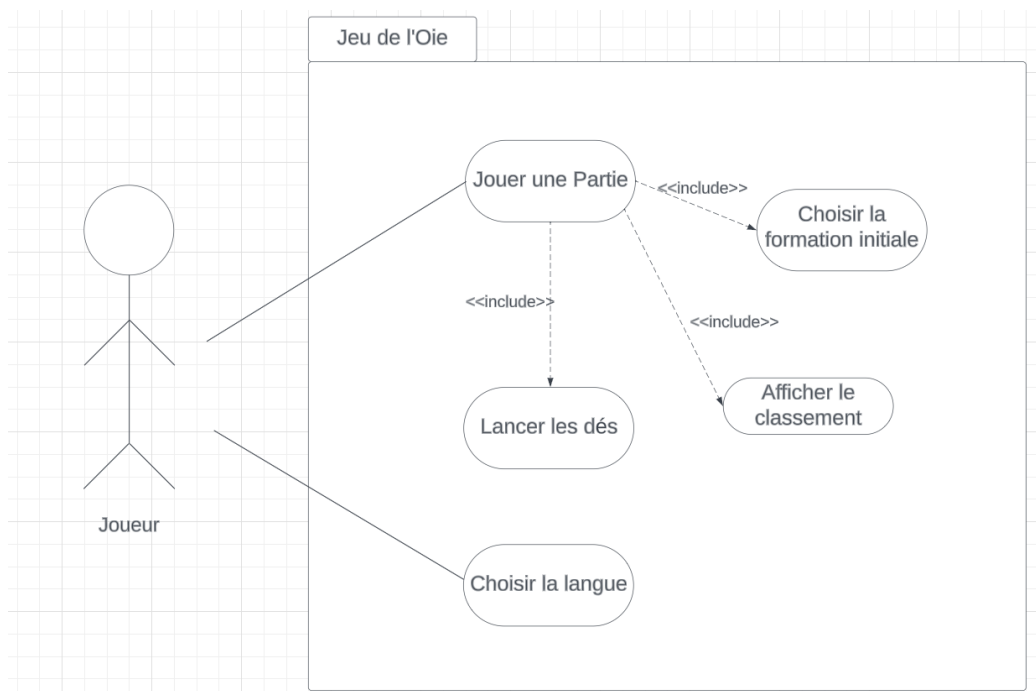


Figure 2 Diagramme des cas d'utilisation

#### 2.1.2. Diagramme des risques

La gestion des risques a été au cœur de notre démarche de projet pour garantir la réussite de notre jeu de l'oie. Le diagramme des risques que nous avons élaborés nous a permis d'identifier, d'évaluer et de prioriser les défis potentiels qui pourraient surgir tout au long du projet. Voici un aperçu de notre approche de gestion des risques :

Type de risque	Description	Gravité (1-4)	Fréquence (1-4)	Criticité (1-16)	Stratégie retenue
Technologique	Manque de pratique sur les langages et technologies utilisés	2	3	6	Mitigation : Formation des développeurs
Délais	Mauvaise planification et mauvaise estimation des délais	3	1	3	Mitigation : Bonne encapsulation des parties du projet. Acceptation active : Concentration du travail sur les fonctionnalités essentielles
Intrinsèque	Affectation vague des tâches	3	2	6	Mitigation : Communication active entre les membres du groupe
Fonctionnel	Projet final non conforme aux attentes	4	1	4	Mitigation : Communication fréquente avec le client
Humain	Absence ou retard d'un ou plusieurs membres de l'équipe	1.5	1	1.5	Acceptation passive

Figure 2 Tableau des risques

Le risque technologique fut bien estimé car le manque d'expérience de l'équipe sur le JavaFX, entre autres, a retardé l'avancée du projet. En ce qui concerne les délais, nous n'avons pas réussi à appliquer l'acceptation active. En effet, des fonctionnalités secondaires ont été implémentées avant certaines fonctionnalités primordiales. Nous avons aussi eu des problèmes d'affectation des tâches lors des premières itérations du projet dû à un manque de communication qui a été résolu. Notre équipe a connu quelques retards mineurs sans grande répercussion sur le projet.

### 2.1.3. Diagramme des paquets

Nous pouvons reconnaître l'architecture MVP dans le diagramme des paquets.

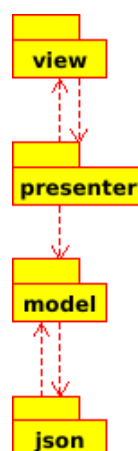


Figure 3 Diagramme des paquets



## 2.2. Conception du plateau de jeu

Le plateau de jeu a été réalisé en JavaFX par la simple réalisation d'une grille à partir de la création de différents carrés. Au même titre qu'un Jeu de l'Oie classique le plateau est composé de 64 cases avec une numérotation des cases en spirale. Cette tâche s'est avérée plus difficile que prévu car nous n'avions pas anticipé comment calculer cette numérotation, heureusement après réflexion avec un papier et un crayon nous avons réussi à trouver l'algorithme qui nous permet de définir cette numérotation et nous avons pu compléter la grille entièrement.

Ensuite, le plateau est composé de plusieurs cases différentes ayant des effets précis, il était ainsi important de pouvoir distinguer aisément ces différentes cases. Pour cela, nous avons utilisé une coloration différente pour chaque case ayant un effet différent.

Nous avons créé un package json qui communique avec le package model. Ce package contient toutes les méthodes relatives au chargement du plateau à partir d'un fichier json, comme demandé dans les exigences minimales. Pour cela, nous avons utilisé la bibliothèque Jackson qui lit le fichier json et remplit un tableau de 64 cases. Nous avons donc mis à disposition des joueurs, dans un menu déroulant dans l'écran de login, 2 plateaux prédéfinis dans 2 fichiers json (une configuration de base correspondant le plus fidèlement possible au parcours d'un étudiant de l'ENSICAEN, et une plus aléatoire, pour plus de fun). Dans le futur, nous avons la possibilité d'en créer d'autres.

## 2.3. Fonctionnalités

Nous avons ensuite implémenté plusieurs fonctionnalités afin de répondre aux exigences du client. Par exemple, nous avons commencé par ajouter un bouton pour recommencer la partie à tout moment pour éviter de devoir fermer l'application lorsque nous voulions recommencer une partie.

De plus, une autre exigence demandée était la possibilité de changer la langue du jeu. Cette fonctionnalité nous a occupé pendant un petit moment car il demandait de changer tous les textes que nous avons utilisés en utilisant un bundle de messages. Il a également fallu créer le menu déroulant permettant de changer la langue en FXML.

Une des requêtes du client présente dans le cahier des charges était de récupérer le montant du salaire moyen brut d'embauche à la sortie d'école. Pour cela, nous utilisons la bibliothèque Jsoup qui va récupérer les éléments html de la page à l'adresse :

<https://www.ensicaen.fr/formations/insertion-professionnelle/>.

Pour calculer le salaire de chaque joueur à la fin de la partie, notre choix a été de calculer la moyenne des points de compétence de tous les joueurs. Cela permet de ne pas avoir de valeurs trop absurdes lorsque les parties durent trop longtemps. On utilise ensuite cette valeur de la manière suivante : le salaire correspond au salaire de base, plus ou moins une proportion du salaire moyen qui dépend du score de compétence du joueur par rapport à la moyenne.

## 3. Développement

### 3.1. Architecture du Jeu

#### Architecture Modèle-Vue-Présentation (MVP) pour Notre Jeu de l'Oie

Notre jeu de l'oie repose sur un modèle d'architecture bien défini appelé Modèle-Vue-Présentation (MVP). Cette architecture a été choisie pour organiser efficacement le développement de notre jeu, en séparant clairement les différentes responsabilités.

**Modèle (Model):** Le modèle gère la logique métier de notre jeu. Il prend en charge la gestion des données du jeu, les règles du jeu, les calculs, et la gestion des événements. C'est le cœur fonctionnel de notre application.

**Vue (View):** La vue est responsable de l'interface utilisateur. Elle présente les éléments visuels du jeu aux joueurs et recueille leurs interactions. Cela inclut l'affichage du plateau de jeu, des pions, et tout ce que les joueurs voient et avec quoi ils interagissent.

**Présentation (Presenter):** La présentation agit comme un médiateur entre le modèle et la vue. Elle assure la communication entre les deux, transmettant les mises à jour de l'état du jeu. Elle gère la logique de présentation et s'assure que l'interface utilisateur réagit aux actions des joueurs.

En utilisant le modèle MVP, nous avons créé une structure claire pour notre jeu. Chaque composant a un rôle spécifique, ce qui simplifie le développement, améliore la maintenance et facilite les tests. Cette architecture permet une expérience utilisateur cohérente et une évolutivité efficace de notre jeu, tout en maintenant une séparation claire entre la logique du jeu et l'interface utilisateur.

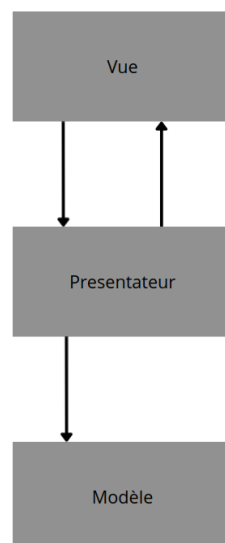


Figure 4 Patron d'architecture MVP

## 3.2. Patron de conception

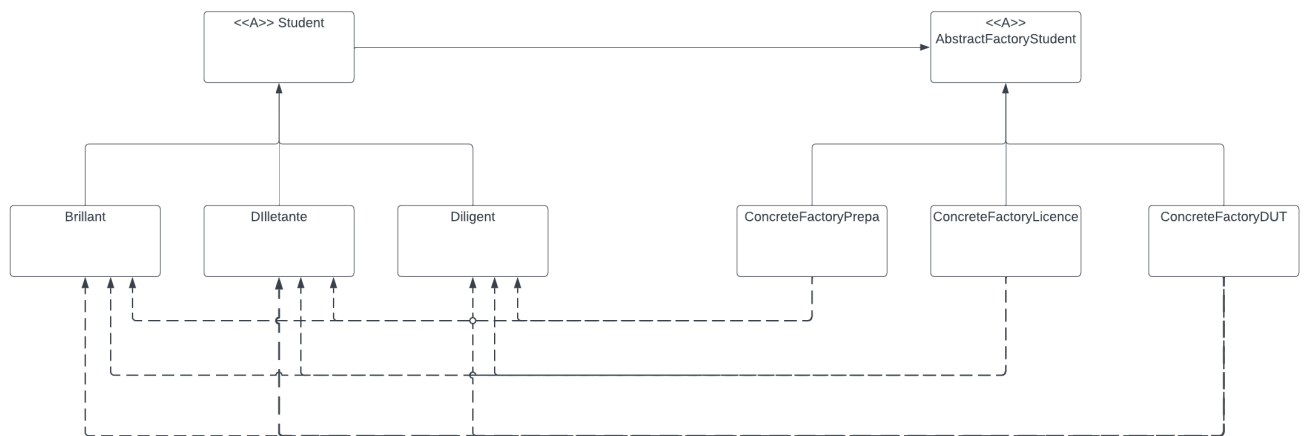


Figure 5 patron fabrique abstraite

L'énoncé du projet précise que la création des joueurs doit être basée sur le choix de leur formation initiale (Prépa, DUT, Licence), et qu'un attribut (dilettante, diligent, brillant) doit être attribué de manière aléatoire. Deux modèles de conception, la fabrique abstraite et le décorateur, semblaient adaptés pour résoudre ce problème. Nous avons opté pour la fabrique abstraite pour la raison suivante : un étudiant ne peut avoir qu'un seul attribut à la fois, tandis que le décorateur aurait la possibilité d'avoir plusieurs attributs, ce qui n'était pas conforme à nos besoins.

Dans notre implémentation, l'utilisateur entre la formation initiale souhaitée, et c'est la fabrique abstraite qui prend en charge ce processus. Elle crée l'étudiant et attribue l'attribut de manière aléatoire.

Chaque étudiant est responsable de sa propre gestion, il connaît le numéro de la case où il se trouve, sa compétence, son attribut, ainsi que d'autres informations telles que s'il a des problèmes en informatique, s'il est membre du BDE (Bureau des Étudiants) et s'il participe au WEI (Week-End d'Intégration).

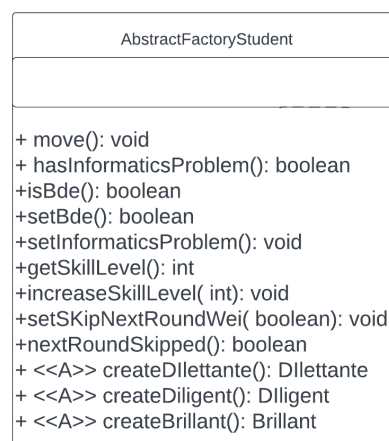


Figure 6 Classe abstraite AbstractFactoryStudent du jeu

Une case du jeu se charge en général d'appliquer son effet sur le joueur. Ainsi, toutes les cases partagent la même structure.

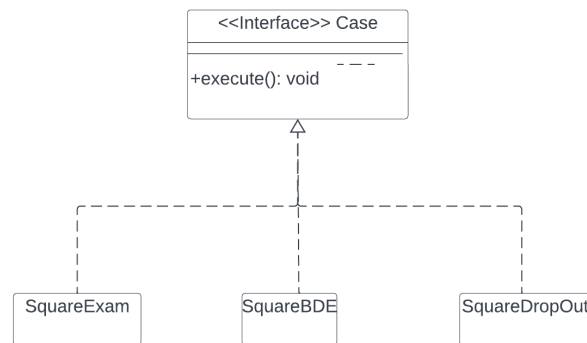


Figure 7 : Classe Interface case et quelques cases

### 3.3. MVP n°1

Dans notre première itération, connue sous le nom de MVP 1, notre objectif principal était de créer un socle de fonctionnalités de base pour notre jeu de l'oie. Nous avons suivi une approche itérative, en commençant par la définition d'un ensemble de tâches précises qui correspondaient aux fonctionnalités clés de cette itération.

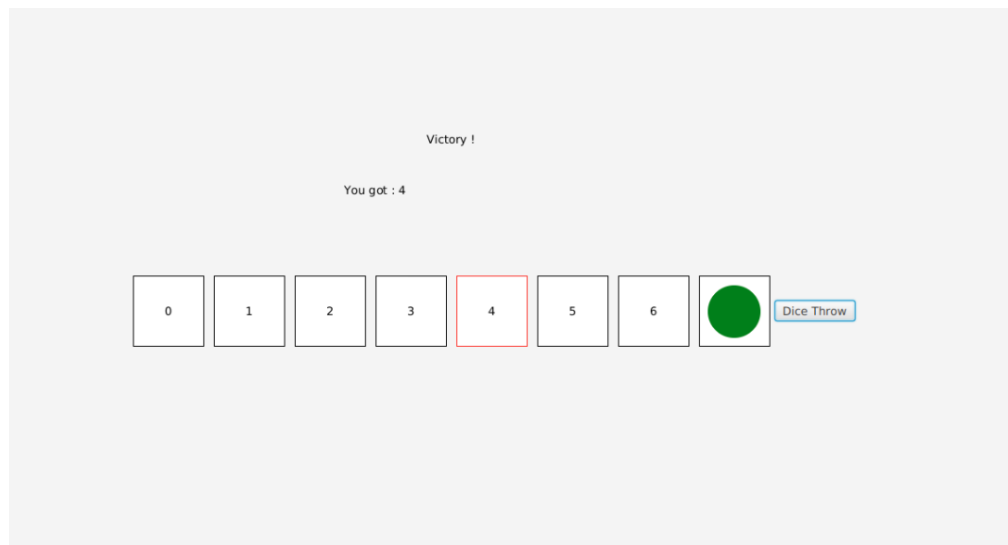
Les tâches sont les suivantes :

- Création d'un plateau minimal
- Création de case sans effets sur le joueur
- Création d'un personnage type sans soft skills
- Création d'un dés
- Ajouter le déplacement des joueurs en fonction de la valeur du dés
- Gérer la fin de partie

Dans notre MVP1, nous avons concentré nos efforts sur un ensemble de fonctionnalités essentielles. Nous avons codé huit cases de jeu de base, avec un seul personnage qui pouvait avancer d'une valeur aléatoire générée par un dé à six faces. Nous avons également introduit la notion de "round" (tour) pour suivre la progression des joueurs.

Sur le plan architectural, dans le modèle (Model), nous avons intégré ces huit cases de jeu, ainsi que la logique pour gérer le déplacement du personnage et la génération des valeurs de dé. La vue (View) a été développée pour afficher les huit cases et gérer le déplacement du personnage, tandis que le présentateur (Presenter) a coordonné les interactions entre le modèle et la vue, en veillant à ce que le déplacement se déroule correctement.

De plus, nous avons inclus des cases de départ et d'arrivée dans le jeu, et si le personnage dépassait la valeur requise pour atteindre la case d'arrivée, il était automatiquement placé sur cette case. La case numérotée 4 colorée en rouge correspond à un bonus qui fait avancer le joueur d'une case supplémentaire.



*Figure 8 MVP 1*

### 3.4. MVP n°2

L'itération MVP2 représente une étape clé de notre projet de jeu de l'oie, où nous avons introduit un ensemble de fonctionnalités visant à enrichir l'expérience de jeu. Ces ajouts ont été conçus pour rendre le jeu plus interactif, multijoueur et personnalisable, tout en élargissant les possibilités offertes aux joueurs.

**Jeu Multijoueur en Local (Max 2) :** L'une des principales fonctionnalités introduites dans cette itération est la possibilité de jouer en mode multijoueur local, avec un maximum de deux joueurs. Cela permet aux amis et aux membres de la famille de jouer ensemble, ajoutant une dimension sociale au jeu.

**Changement de Langue :** Nous avons inclus la possibilité de choisir la langue dans laquelle les joueurs souhaitent jouer, offrant ainsi une expérience plus accessible et personnalisée.

**Ajout des Softskills et Choix des Hardskills :** Les joueurs peuvent désormais acquérir des compétences (soft skills) de manière aléatoire, ce qui apporte un élément de surprise à chaque partie. De plus, ils ont la possibilité de choisir leurs compétences principales (hardskills), ce qui influence leur progression dans le jeu.

**Fichier JSON :** Pour une gestion plus efficace des données du jeu, nous avons mis en place une structure de fichiers au format JSON pour stocker les informations relatives au plateau de jeu, aux compétences et aux cartes d'événements.

**Ajout de Toutes les Cases à Effets :** Nous avons implémenté toutes les cases à effets, y compris celles qui apportent des défis ou des avantages aux joueurs, ajoutant ainsi de la stratégie au jeu.

**Plateau 64 Cases :** Nous avons étendu le plateau de jeu à 64 cases pour offrir une expérience de jeu plus variée. Les joueurs devront désormais faire preuve de stratégie pour naviguer sur ce plateau plus vaste.

**Ajout d'un Exécutable :** Nous avons simplifié l'accès au logiciel en introduisant un exécutable dédié, éliminant ainsi le besoin d'ouvrir le jeu manuellement depuis un IDE. Les utilisateurs bénéficieront d'une utilisation plus aisée et rapide du programme.

L'itération MVP2 a renforcé notre jeu de l'oie en ajoutant des fonctionnalités essentielles qui le rendent plus interactif, personnalisé et varié. Ces améliorations contribuent à une expérience de jeu plus complète pour nos joueurs.

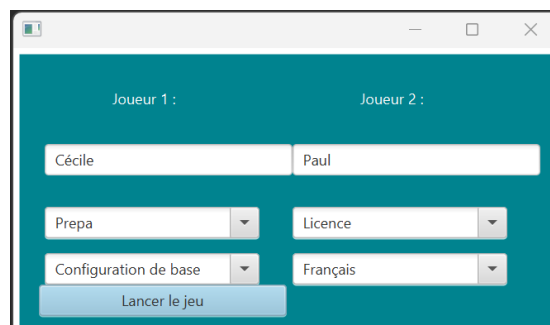
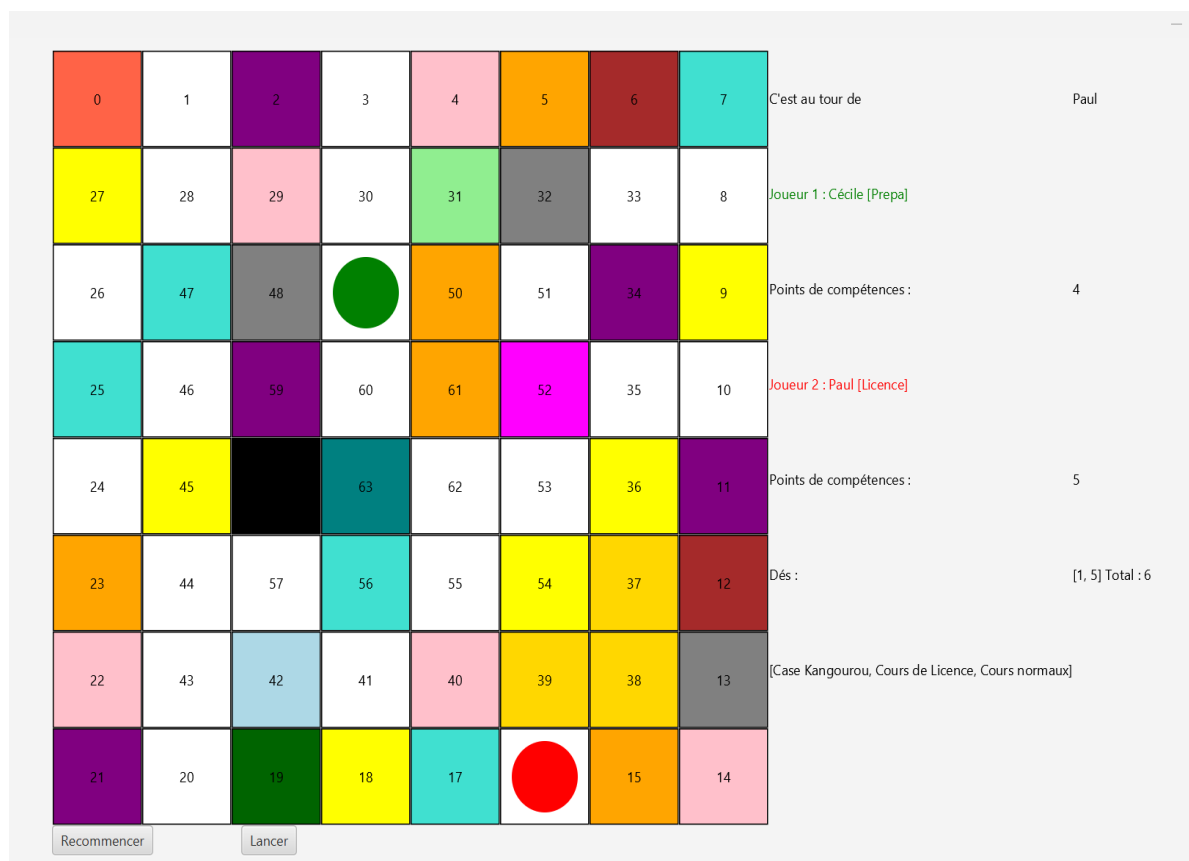


Figure 9 MVP 2

## 4. Perspectives Futures

### 4.1. Possibilités d'amélioration

Nous avons utilisé comme énoncé plus haut le patron abstrait, il est vrai qu'il satisfait la contrainte de n'avoir qu'un seul attribut mais il n'est pas totalement ouvert aux extensions et fermé aux modifications. Si nous voulons un nouveau parcours initial par exemple "Master" il n'y aura qu'à créer une autre classe et ça ne gênera rien.

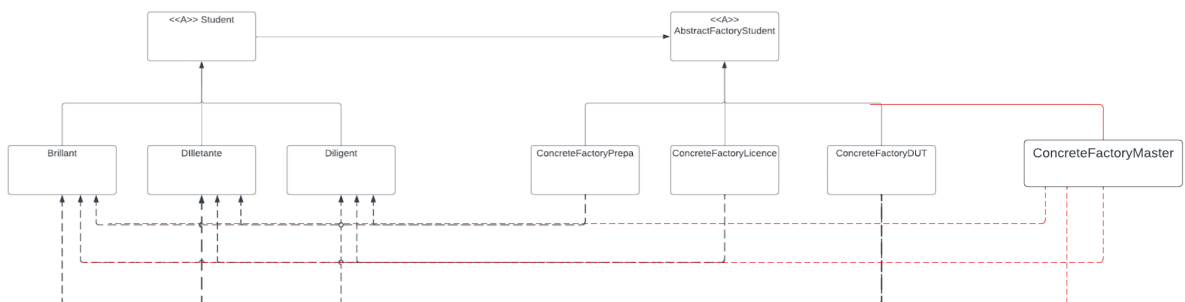


Figure 10 Ajout d'un autre parcours initial, Master

Mais s'il fallait ajouter un nouvel attribut en plus, par exemple 'Courageux', nous serions alors obligés de modifier le code des autres classes. Toutefois les autres critères du principe SOLID sont bien respectés.

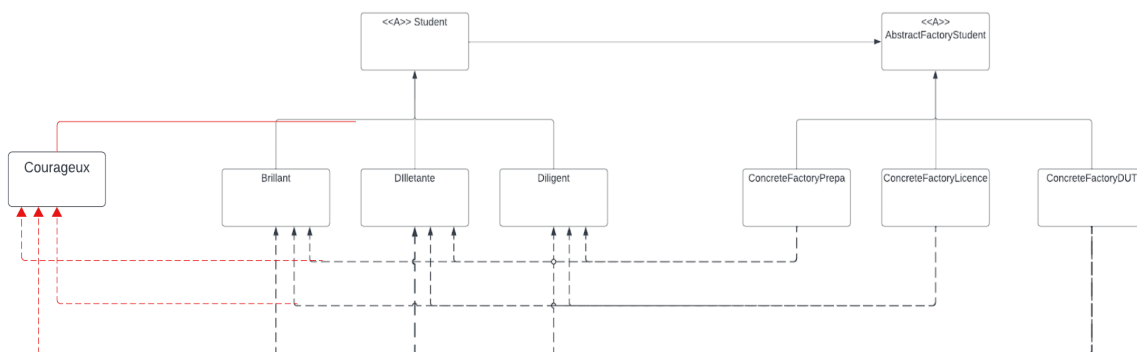


Figure 11 Ajout d'un nouvel attribut

Dans ce cas, nous voyons mieux l'intérêt d'un Décorateur plus qu'une Fabrique abstraite puisqu'un décorateur validerait le critère "open-closed" du principe SOLID.

## 4.2. Prochaines étapes

Nous considérons notre jeu de l'oie actuel comme une étape importante de notre projet, mais nous avons également identifié des axes d'amélioration potentiels. Voici quelques-unes de nos réflexions sur les prochaines étapes :

- Amélioration de l'Interface visuelle : Nous envisageons une mise à jour de l'interface utilisateur pour la rendre plus attrayante et intuitive, offrant une expérience visuelle améliorée.
- Personnalisation avancée : Nous cherchons à étendre les options de personnalisation pour les joueurs, en permettant des réglages plus fins des compétences "softskills" et "hardskills".
- Intégration de 4 joueurs : Nous explorons la possibilité d'ajouter un mode de jeu pour quatre joueurs, offrant une expérience encore plus conviviale pour les groupes d'amis.

Ces réflexions guideront notre travail futur pour améliorer l'expérience de jeu tout en répondant aux besoins et aux attentes des joueurs.

# 5. Conclusion

## 5.1. Récapitulatif des accomplissements

### 5.1.1. Accomplissement sur le modèle

Notre projet a abouti à la création d'un jeu de l'oie complet et interactif. Nous avons intégré des dés fonctionnels, la personnalisation des joueurs, des cases à effets variées, étendu le plateau de jeu, et ajouté la prise en charge de deux langues. Le modèle MVP a été efficacement utilisé pour structurer notre code de manière modulaire.

### 5.1.2. Accomplissement sur la vue

Notre interface graphique finale permet de visualiser le plateau de jeu, ainsi que les différentes informations nécessaires à la compréhension et au bon déroulé d'une partie, comme le score des dés ou le nom des joueurs. Nous avons également ajouté la possibilité de lancer une nouvelle partie d'un simple clic de bouton, et le choix de la langue directement dans la fenêtre de login, dans laquelle les joueurs peuvent entrer leurs noms.



## 5.2. Difficultés rencontrées

### 5.2.1. Difficultés sur la communication

Comme évoqué précédemment, l'une des premières difficultés que nous avons rencontrées était liée à la communication au sein de notre équipe de développement. Les rôles et les tâches n'étaient pas clairement définis, et cela a entraîné des défis au début de notre projet.

Cependant, nous avons rapidement reconnu l'importance d'une communication efficace pour le succès de notre projet. Nous avons organisé des réunions régulières pour discuter des objectifs, des problèmes et des idées. Ces réunions ont permis de clarifier les attentes de chacun et de mettre en place un processus de travail plus collaboratif.

Avec le temps, notre collaboration s'est nettement améliorée. Nous avons commencé à travailler de manière plus efficace, en nous appuyant sur une approche plus structurée. La mise en place de réunions régulières et d'une communication ouverte a renforcé notre coordination.

En fin de compte, malgré les difficultés initiales, notre projet nous a offert des leçons importantes sur la communication, la collaboration et la définition d'objectifs. Grâce à notre capacité d'adaptation et à notre engagement collectif, nous avons surmonté les obstacles et livré un jeu de l'oie fonctionnel, tout en développant des compétences essentielles pour notre avenir professionnel.

### 5.2.2. Difficultés au niveau du développement

Notre expérience de développement du jeu de l'oie en tant qu'étudiants a été enrichissante, mais elle n'a pas été sans difficultés. En tant qu'étudiant, l'un de nos principaux défis a été de mettre en pratique les principes de gestion de projet dans un environnement concret.

L'une des zones où nous avons rencontré des obstacles significatifs était la conception de l'interface graphique. Notre choix de JavaFX pour l'interface utilisateur représentait un défi en raison de notre expérience limitée. Cependant, cette difficulté nous a incités à nous perfectionner dans ce domaine.

De plus, l'utilisation de Git pour la gestion de versions a été une nouvelle expérience pour nous. Comprendre les concepts de gestion de code source et d'intégration collaborative a demandé du temps et des efforts supplémentaires.

Malgré ces difficultés, ces expériences nous ont permis d'acquérir des compétences essentielles en gestion de projet, en développement d'interface utilisateur et en gestion de versions, renforçant ainsi notre préparation pour des défis futurs.

