

Rapport

MYENSICAENLIFE

Le 14 décembre 2021,
Version 2.0

JOLLIVET Romaric
NAPOLI Dorian
EL KILALI Hamza
PINSEL Tony
PATRY Alan

FESSARD Arthur
LE BEC Fabien
ROBIN Geoffroy



Table des matières

| | |
|--|-----------|
| 1. PRESENTATION DU PROJET | 4 |
| 1.1 Equipe | 4 |
| 1.2 Projet | 4 |
| 2. ANALYSE DES RISQUES | 4 |
| 3. DIAGRAMME DE CAS D'UTILISATION | 5 |
| 4. DIAGRAMME DE SEQUENCE | 5 |
| 5. DIAGRAMME DE PAQUETS | 6 |
| 6. MODELISATION | 7 |
| 6.1 Décorateur | 7 |
| 6.2 Stratégie | 8 |
| 6.3 Proxy | 8 |
| 6.4 Conclusion sur les patrons | 9 |
| 6.5 Autre implémentation | 9 |
| 7. CLASSES DES JOUEURS | 9 |
| 8. CASES DU PLATEAU | 10 |
| 8.1 Cases Enseignements | 10 |
| 8.2 Cases Examen | 11 |
| 8.3 Cases Antisèche | 11 |
| 8.4 Cases Soirée | 11 |
| 8.5 Cases Redoublement | 12 |
| 8.6 Cases Jeudi | 12 |
| 9. AFFICHAGE DES RESULTATS | 12 |
| 10. DIFFICULTES RENCONTREES | 13 |
| 11. CONCLUSION | 13 |

Table des figures

| | |
|---|----|
| Figure 1 Diagramme de cas d'utilisation | 5 |
| Figure 2 Diagramme de séquence | 5 |
| Figure 3 Diagramme de paquets | 6 |
| Figure 4 Patron de conception proxy | 8 |
| Figure 5 Enseignement | 10 |
| Figure 6 Examen | 11 |
| Figure 7 Antiseche | 11 |
| Figure 8 Soirée | 11 |
| Figure 9 Redoublement | 12 |
| Figure 10 Jeudi | 12 |

1. Présentation du projet

1.1 Equipe

Ce projet a été réalisé par une équipe de 8 personnes comprenant Romaric JOLLIVET, en tant que chef de projet, Hamza EL KILALI en tant qu'architecte et Arthur FESSARD en tant que responsable de version. Le reste de l'équipe avait un rôle de développeur (que les trois personnes citées précédemment avaient aussi).

1.2 Projet

Ce projet a pour but de produire une version alternative du jeu de société très populaire, « le jeu de l'oie ». Nous devions le revisiter pour en faire un jeu qui représentait un parcours scolaire à l'ENSICAEN. C'est pourquoi nous avons décidé de nommer notre jeu : « my ensicaen life ». Tout ceci devait être codé en Java, tout en utilisant le Framework JavaFX pour l'interface graphique. Le but de ce projet est de nous faire appliquer ce que nous avons appris lors de nos cours de génie logiciel. Nous disposons de 6 séances de deux heures pour venir à bout de ce projet.

Nous détaillons dans ce rapport notre démarche aussi bien que le travail que nous avons réalisé.

2. Analyse des risques

| Risque | Impact | Gravité | Action préventive | Probabilité |
|--|---|---------------|--|-------------|
| Echec de la démonstration | Délai de livraison du projet | Grave | Réaliser de multiples tests | Faible |
| Confinement dû à la situation sanitaire | Réduction de l'efficacité du travail | Significative | Travailler avec des outils adaptés au travail à distance | Moyenne |
| Changement de la structure du site de l'ENSICAEN | Impossibilité d'afficher les résultats du jeu | Très grave | S'arranger avec le service en charge du site | Forte |

3. Diagramme de cas d'utilisation

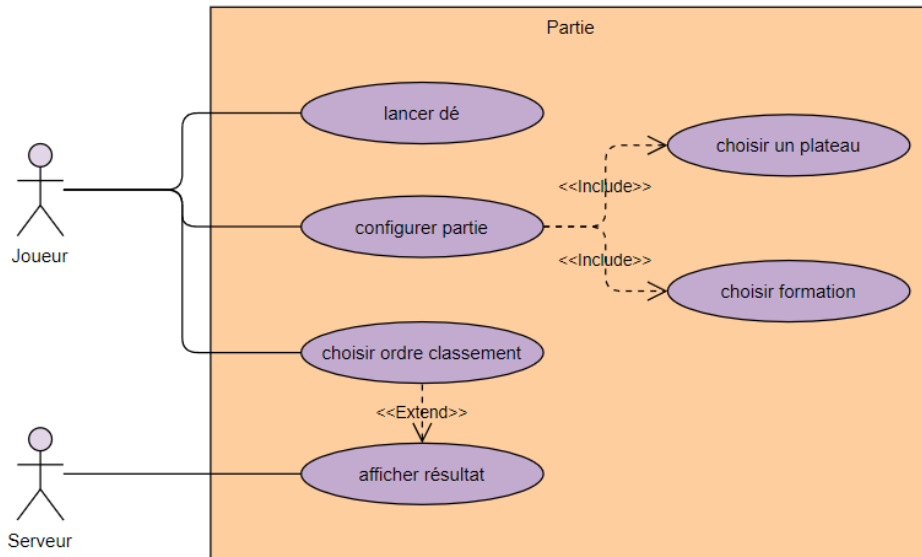


Figure 1 Diagramme de cas d'utilisation

4. Diagramme de séquence

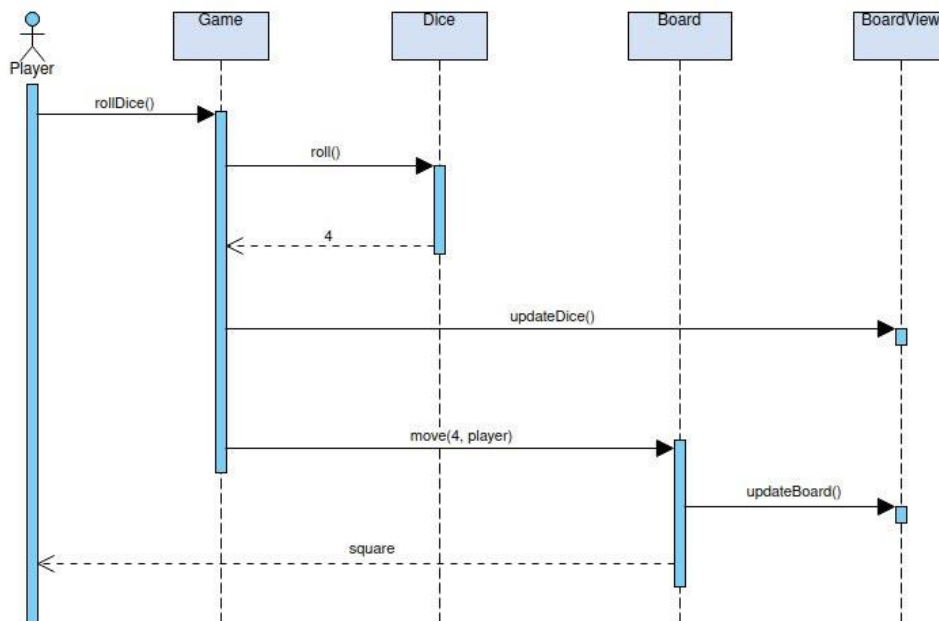


Figure 2 Diagramme de séquence

5. Diagramme de paquets

Dans le cadre de ce projet, nous avons décidé d'utiliser le patron d'architecture Modèle-Vue-Présentateur, principalement car ce concept permettait de séparer le modèle du reste des composants, la présentation affichant ensuite les données dans la vue. C'est également une architecture utilisée par le framework JavaFX.

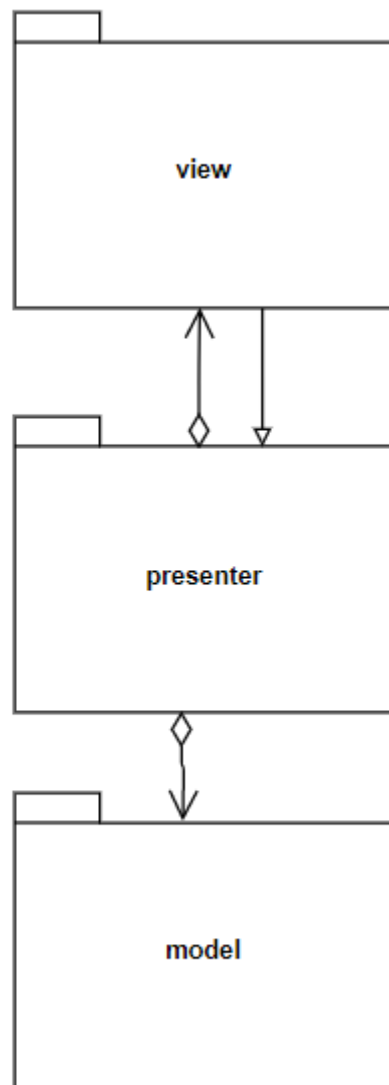
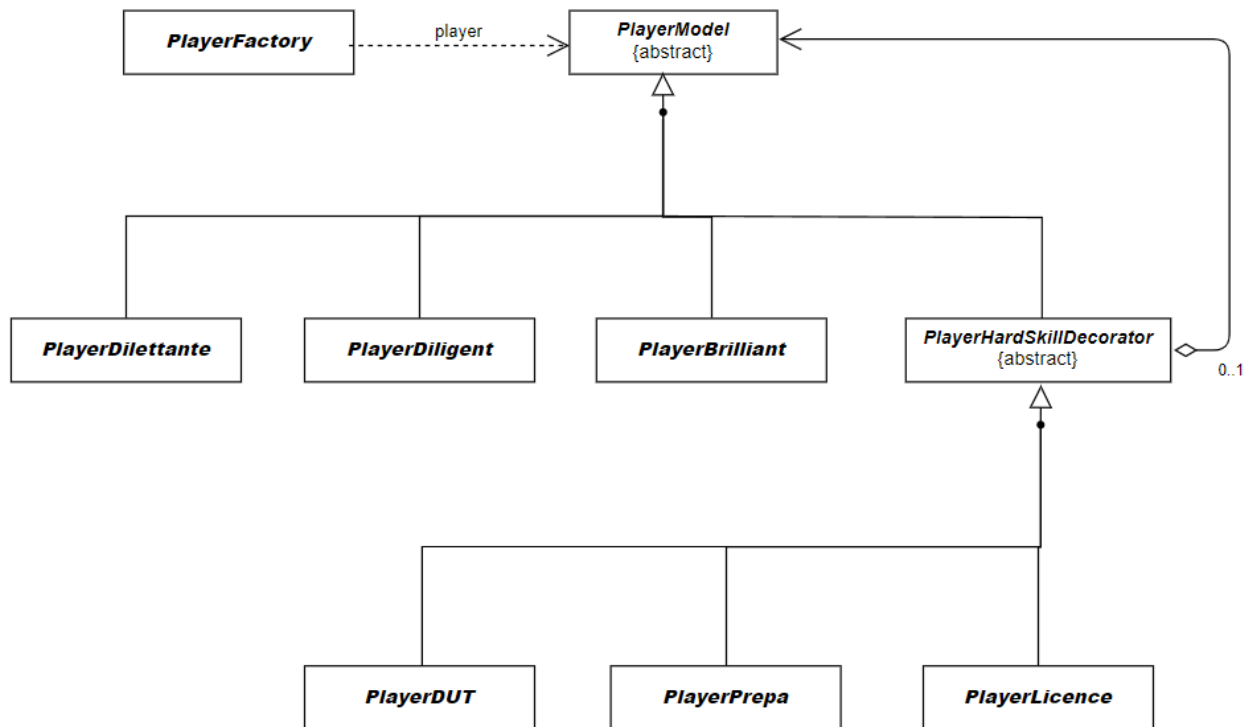


Figure 3 Diagramme de paquets

6. Modélisation

Nous avons fait l'usage de plusieurs patrons de conceptions lors de la réalisation de ce projet afin de réaliser un code propre respectant les règles SOLID de génie logiciel. Nous allons détailler ci-dessous les différents patrons de conception utilisés.

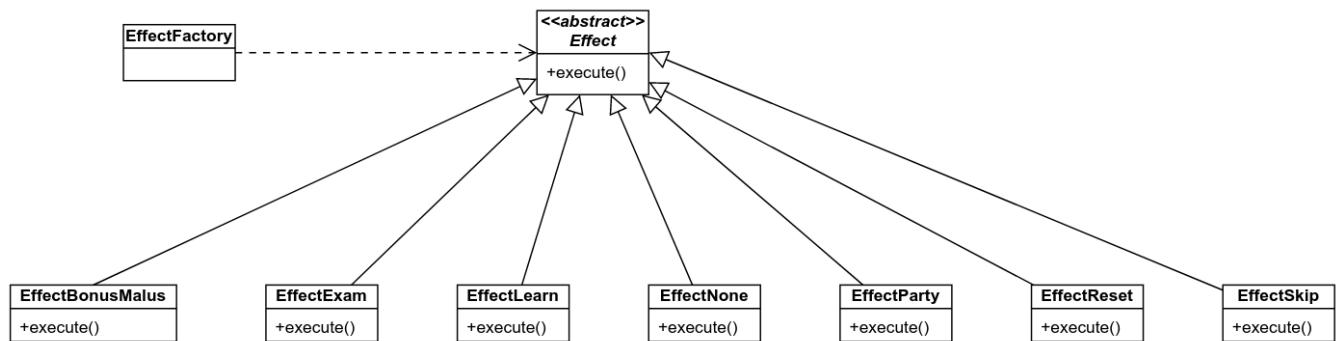
6.1 Décorateur



Les joueurs devant s'octroyer deux attributs simultanés : un softskill et un hardskill, il nous a paru intéressant d'avoir recours à un décorateur afin d'éviter des combinaisons binomiales de classe tout en tirant profit des différentes capacités liées à ces attributs dans le cas où nous voudrions en spécifier ou en ajouter un par exemple. Nous avons donc créé une classe abstraite initiale "PlayerModel" contenant toutes les fonctions de joueur puis nous les spécialisons dans les sous-classes de softskill et de hardskill. Il est intéressant de noter que nous avons choisi de faire un décorateur sur les hardskills mais il aurait été aussi bien possible de le faire sur les softskills.

En outre, nous avons créé une classe annexe PlayerFactory dépendante de PlayerModel et qui a pour fonction la gestion de la création des joueurs, cela afin de gérer l'aspect aléatoire de l'attribution du softskill et d'assurer le principe de responsabilité unique.

6.2 Stratégie



Afin d'implémenter les effets des cases du plateau, nous avons décidé d'utiliser le patron « Stratégie ». Ce patron permet de choisir dynamiquement quelle stratégie utiliser pour un type d'objet donné, ce qui est parfait pour les cases de notre plateau. On a par exemple la stratégie « **EffectBonusMalus** » qui permet de faire avancer ou reculer le joueur d'un certain nombre de cases. Il permet d'éviter d'avoir à tester le type de chaque case pour savoir quel effet appliquer.

6.3 Proxy

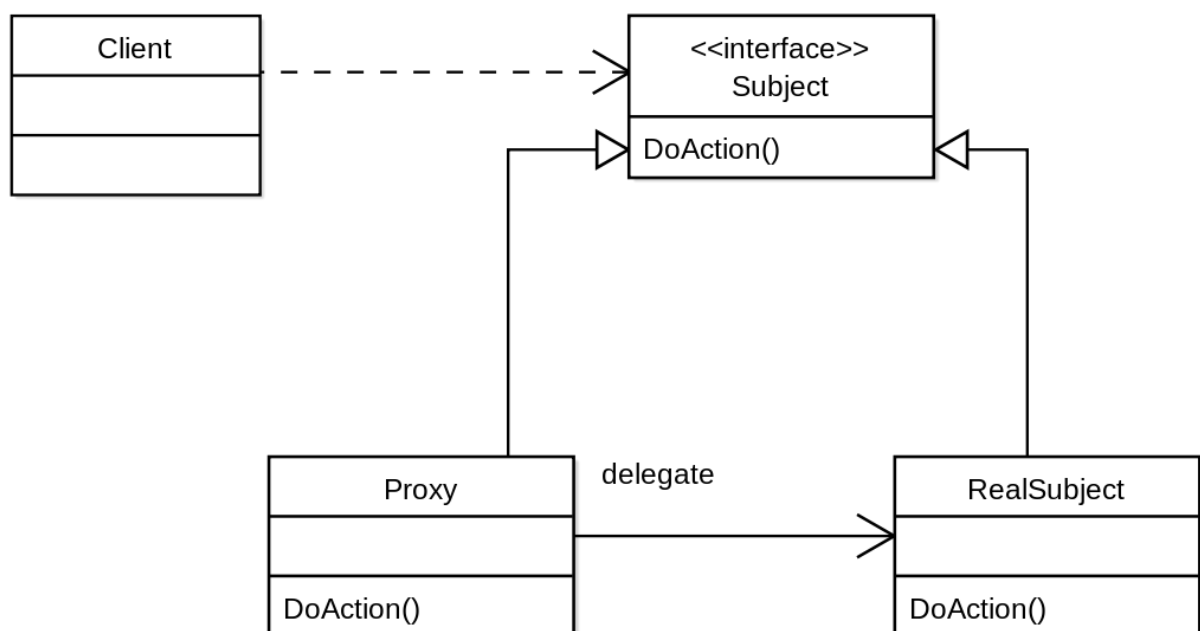


Figure 4 Patron de conception proxy

Nous avons utilisé un Proxy pour contrôler l'accès aux méthodes de la classe « ServerModel » qui dialogue directement avec le serveur de l'ENSICAEN afin de récupérer les salaires moyens annuels après l'obtention du diplôme.

6.4 Autre implémentation

Nous avons aussi utilisé des interfaces pour un meilleur niveau d'abstraction ce qui permet de faciliter le travail entre les développeurs gérant la partie présentation, vue et modèle.

Ceci permet une bonne adaptabilité car les différents développeurs peuvent développer des classes reposant sur des interfaces.

6.5 Conclusion sur les patrons

Nous avons donc utilisé 3 patrons de conception sur les 23 patrons existants, il aurait sûrement été possible de rajouter certains patrons à notre projet dans un laps de temps plus long. Ces derniers nous ont semblés cependant très utiles pour rendre le code propre, organisé et facilement modulable de par leur haute adaptabilité et leur niveau d'abstraction.

7. Classes des joueurs

Comme demandé, nous avons implémenté différentes classes pour les joueurs.

Ainsi, un joueur en début de partie se voit attribuer un SoftSkill de manière aléatoire entre les trois différents skills suivant : Brillant, Dilettante et Diligent.

Le SoftSkill Brillant permet au joueur de se voir attribuer un bonus de 1 sur le résultat de chacun de ses lancers de dé.

Le SoftSkill Dilettant condamne le joueur à se voir retirer un malus de 1 sur le résultat de chacun de ses lancers de dé.

Quant-au SoftSkill Diligent, il n'accorde ni malus ni bonus au joueur.

Au début de chaque partie, le joueur doit choisir son HardSkill en fonction des études qu'il a effectué avant d'intégrer l'ENSICAEN. Il a ici le choix entre trois Skills : Prepa, DUT ou licence. Ces derniers peuvent être bénéfiques ou handicapants dépendamment des cases.

Le HardSkill Prepa octroie un bonus d'une case au joueur s'il s'arrête sur une case « enseignement maths ». A contrario, il condamne le joueur à recevoir un malus d'une case si celui-ci se trouve sur une case « enseignement info ».

Le HardSkill DUT octroie un bonus d'une case au joueur s'il s'arrête sur une case « enseignement info ». A contrario, il condamne le joueur à recevoir un malus d'une case si celui-ci se trouve sur une case « enseignement maths ».

Le HardSkill Licence quant à lui n'accorde aucun malus ni bonus au joueur.

8. Cases du plateau

À l'instar du « jeu de l'oie », certaines cases du plateau sont particulières et permettent le déclenchement de certains événements.

8.1 Cases Enseignements

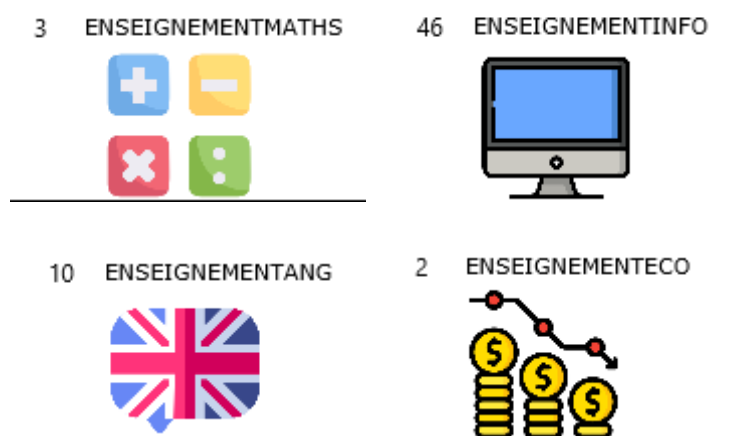


Figure 5 Enseignement

Ces cases permettent aux joueurs d'augmenter leur niveau de compétence.

Elles sont au nombre de 4 : « Info, Maths, Anglais, Economie »

Chaque cours incrémente le niveau de compétence de 1. Ce niveau sera utile sur les cases examens comme nous allons le voir ci-dessous, et aussi lors du calcul du salaire après l'obtention de diplôme.

Nous avons décidé de définir un niveau de compétence global et non pas spécifique par matière afin de simplifier le jeu.

De plus, comme nous l'avons vu dans le point précédent, en fonction du HardSkill, ces cases peuvent aussi accorder un bonus ou un malus au joueur.

8.2 Cases Examen

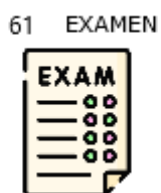


Figure 6 Examen

Ces cases, comme leur nom l'indique, sont des cases où on simule un passage d'examen. Pour pouvoir le réussir, il faut que le joueur ait un niveau de compétence défini, qui augmente à chaque examen passé. Si le joueur ne réussit pas l'examen, il recule alors de 6 cases.

8.3 Cases Antiséche



Figure 7 Antiseiche

Ces cases bonus permettent de faire avancer le joueur de 6 cases.

8.4 Cases Soirée



Figure 8 Soirée

Ces cases malus interdisent au joueur de jouer pendant un tour, et décrémentent le niveau de compétence du joueur de 1.

8.5 Cases Redoublement

23 REDOUBLEMENT



Figure 9 Redoublement

Ces cases malus obligent le joueur à recommencer la partie de zéro.

8.6 Cases Jeudi

43 JEUDI



Figure 10 Jeudi

Ces cases malus interdisent au joueur de jouer pendant un tour.

9. Affichage des résultats

À la fin de la partie, les résultats sont affichés sous deux formes.

La première est l’affichage en fonction de la case sur laquelle se trouve chaque joueur en fin de partie, le classement se fait bien sûr par ordre décroissant du numéro de case de chaque joueur.

La deuxième est l’affichage en fonction du salaire en fin de partie. Pour calculer le salaire, nous utilisons, comme nous l’avons dit précédemment, le niveau de compétence de chaque joueur.

Ainsi, on stocke en mémoire le plus haut niveau de compétence parmi les joueurs en fin de partie. Le joueur qui possède ce niveau de compétence obtient un salaire deux fois plus élevé que le salaire moyen annuel en sortie de l’ENSICAEN. Ensuite, pour calculer le salaire des autres joueurs, nous calculons le pourcentage de différence entre le niveau de compétence de ce joueur et le niveau de compétence le plus élevé.

10. Difficultés rencontrées

La première difficulté que nous avons rencontrée a été de devoir travailler en équipe. En effet, travailler avec une équipe de 8 personnes s'est avéré plus compliqué que prévu initialement. La bonne cohésion entre les différents membres de l'équipe fût assez difficile, car la communication manquait pour assurer une répartition efficace et permanente des tâches. De plus, nous avons dû suivre les évolutions fréquentes du projet de par notre choix de ne pas spécialiser les membres sur un domaine spécifique ce qui ajoutait un travail supplémentaire à chaque nouvelle séance.

En outre, nous avons aussi rencontré une difficulté liée à la limite de temps dédiée à ce projet. Cela ne nous a malheureusement pas permis d'implémenter toutes les fonctionnalités attendues du jeu, notamment la lecture de cartes diverses à partir d'un fichier JSON ou le développement de tests unitaires exhaustifs. Nous aurions aussi apprécié améliorer certains détails comme le style graphique si le temps nous le permettait.

Un autre incident imprévu fut le départ soudain d'un des membres de notre équipe, ce qui a également impacté considérablement notre efficacité.

11. Conclusion

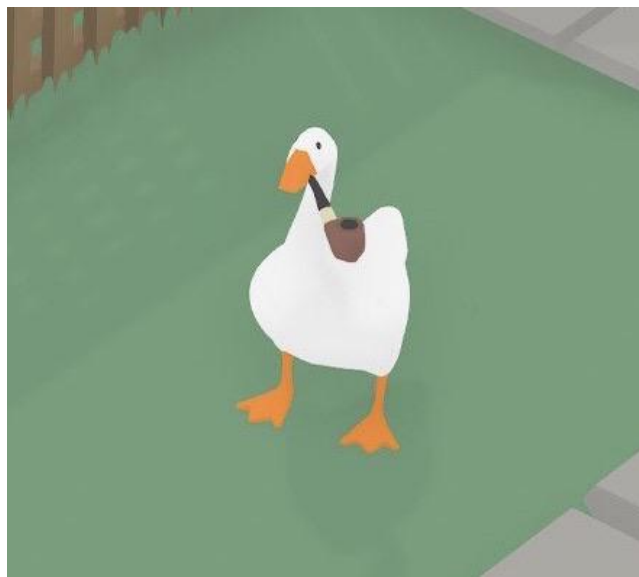
Ce projet nous a permis dans un premier temps d'améliorer notre aptitude à travailler en équipe. En effet, il a fallu que nous réussissions à nous partager efficacement le travail pour pouvoir avancer le plus efficacement possible, tout en s'entraînant.

Il nous a aussi permis de nous entraîner au développement en langage Java et de nous familiariser avec le framework JavaFX. Nous avons également pu mettre en œuvre la plupart des compétences et des informations que nous avons acquises tout au long de notre deuxième année telle que nos compétences apprises en « Génie Logiciel et patron de conception » afin de structurer notre projet et de rendre notre code le plus compréhensible, lisible et extensible possible.

Bien que le temps ait été limité, ce projet a donc été enrichissant à mener à son terme, mettant en œuvre différentes compétences de notre formation tout en restant ludique et mettant en lumière l'importance de la gestion de projet dans notre futur métier d'ingénieur.



Merci



Ecole Publique d'Ingénieurs en 3 ans

6 boulevard Maréchal Juin, CS 45053
14050 CAEN cedex 04

