



Technologie Objet - Projet long  
Développement d'une application Unlock  
*Fonctionnalités de l'application*

Nom des auteurs  
AUPETIT Lucien  
CAZES Noa  
DUTHOIT Thomas  
GEORGET Lucas  
JAMES Christopher  
PAOLI Baptiste  
WINTERBERGER Ilona

## Table des matières

<b>1</b>	<b>Objectif général du projet</b>	<b>3</b>
<b>2</b>	<b>Les principales fonctionnalités et leur état d'avancement</b>	<b>3</b>
<b>3</b>	<b>Le découpage de l'application en sous-systèmes</b>	<b>4</b>
<b>4</b>	<b>Le diagramme de classe</b>	<b>5</b>
<b>5</b>	<b>Les principaux choix de conception et de réalisation, les problèmes rencontrés et les solutions apportées</b>	<b>5</b>
<b>6</b>	<b>Organisation de l'équipe et mise en oeuvre des méthodes agiles</b>	<b>6</b>
<b>7</b>	<b>Conclusion et Objectifs</b>	<b>6</b>

## Table des figures

1	Diagramme de classe 1 . . . . .	5
---	---------------------------------	---

# 1 Objectif général du projet

Le but est de recréer le jeu « Unlock » sans l'aspect physique de la boîte et de son contenu. Le ou les joueur(s) pourra/pourront jouer sur une ou plusieurs énigmes proposées par l'interface graphique.

Ce produit est destiné à tout public qui souhaite se distraire avec un jeu ludique et qui nécessite réflexion, et à la différence du jeu traditionnel, il est possible de jouer n'importe où au jeu Unlock.

# 2 Les principales fonctionnalités et leur état d'avancement

Les fonctionnalités souhaitées ont été fragmentées volontairement en sous-fonctionnalités simples à réaliser ce qui a permis de mettre en avant des besoins sous-jacents.

Dans le jeu Unlock, les utilisateurs jouent une partie et interagissent avec différentes cartes. Il existe cinq types de cartes, chaque type ayant une particularité différente :

1. Histoire  
C'est la première carte (le dos d'une pièce), on la lit pour comprendre l'intrigue puis la retourner déclenche le timer,
2. Lieu  
Les cartes lieux indiquent plusieurs numéros et invitent les joueurs à fouiller le paquet afin de dévoiler les éléments à étudier,
3. Puzzle  
Les cartes puzzles se combinent entre elles pour dévoiler une nouvelle carte (une rouge avec une bleue),
4. Machine  
Les cartes machines sont des mini-jeux qui permettent d'obtenir une nouvelle carte,
5. Code  
Les cartes codes sont des énigmes et permettent d'avancer dans l'histoire une fois résolues.

Les joueurs utilisent une interface afin de mettre en relation les différents éléments de l'aventure et ainsi progresser dans les énigmes à résoudre.

Les différentes fonctionnalités ainsi que leur état d'avancement sont les suivantes :

1. créer une interface d'accueil : une interface d'accueil simple a été réalisée (itération 2), elle sera complétée en vue de l'itération 3,
2. créer une interface pour sélectionner le type de carte retournée, correspondant au plateau de cartes : interface réalisée avec les principales fonctionnalités (itérations 1 et 2), elle sera complétée en vue de l'itération 3,
3. créer un menu pour sélectionner l'histoire souhaitée : non réalisé car il est pour l'instant question de créer un seul set de cartes,
4. créer un timer : timer réalisé et l'interface graphique est en cours (itération 2),
5. mettre du son pour le timer : non réalisé,
6. mettre sur pause le jeu : réalisé car cela correspond à une des fonctionnalités du timer (itération 2),
7. saisir des codes (associés aux cartes "codes") : réalisé (itération 2),
8. demander un indice : non réalisé,
9. saisir une pénalité reçue par une carte et éventuellement automatiser ce processus de sorte que l'utilisateur n'ait pas à saisir ses propres pénalités : non réalisé,
10. obtenir de l'aide sur un objet caché : non réalisé,
11. visualiser les indices reçus : réalisé par le biais de l'interface graphique (itérations 1 et 2),

12. désactiver ou activer la musique d'ambiance : non réalisé,
13. changer la langue de l'application : non réalisé,
14. création d'une interface permettant de voir le paquet de cartes face cachées et de visualiser les cartes déjà retournées afin de se situer dans l'avancement de la résolution des énigmes : interface graphique réalisée (itérations 1 et 2), elle sera complétée en vue de l'itération 3,
15. créer notre propre histoire : non réalisé.

### 3 Le découpage de l'application en sous-systèmes

L'application a été découpée en sous-systèmes selon l'architecture suivante :

1. Le jeu en lui-même
  - Carte.java  
Interface qui permet par la suite de définir tous les types de cartes présentés précédemment,
  - CarteCodes.java,
  - CarteInteraction.java,
  - CarteMachine.java,
  - CarteLieu.java,
  - CarteObjet.java,
  - CartePenalite.java.
  - Plateau.java  
Interface qui permet de mettre en place les différentes actions possibles sur le plateau avec les cartes,
  - Jeu.java  
Une réalisation de Plateau.java qui permet de définir un jeu quelconque,
  - Tuto.java  
Une réalisation de Plateau.java qui permet de définir le tutoriel de ce jeu.
  - Les exceptions
    - CaseOccupeeException.java  
Est levée lorsqu'une carte essaye d'être posée sur une case du plateau déjà prise,
    - CombinaisonInvalideException  
Est levée lorsqu'une combinaison est invalide (Bleu et Bleu, Rouge et Rouge, Rouge et Gris...).
2. L'interface graphique (dépendante des classes précédentes)
  - Les classes principales
    - UnlockSwing.java  
La classe qui permet la représentation du plateau de cartes du jeu,
    - UnlockSwingTuto.java  
La classe qui permet la représentation du plateau de cartes du jeu du tutoriel,
    - FenetreAcceuil.java  
La classe qui permet la représentation de la fenêtre d'accueil,
    - FentreChrono.java  
La classe qui permet la représentation du chronomètre ainsi que des fonctionnalités telles que demander un indice, saisir un code, obtenir des pénalités etc.
  - Les actions associées à des événements
    - ActionNouvellePartie.java  
Permet de définir l'action engendrée lors de l'évènement "appui sur l'item du menu Nouvelle Partie",
    - ActionQuitter.java  
Permet de définir l'action engendrée lors de l'évènement "appui sur l'item du menu Quitter",

- ActionSaisirCode.java  
Permet de définir l'action engendrée lors de l'évènement "appuie sur l'item du menu Saisir code".
- Les images de cartes

## 4 Le diagramme de classe

La figure 1 représente le diagramme de classe de notre application, réalisé uniquement avec les informations utiles pour comprendre l'architecture globale.

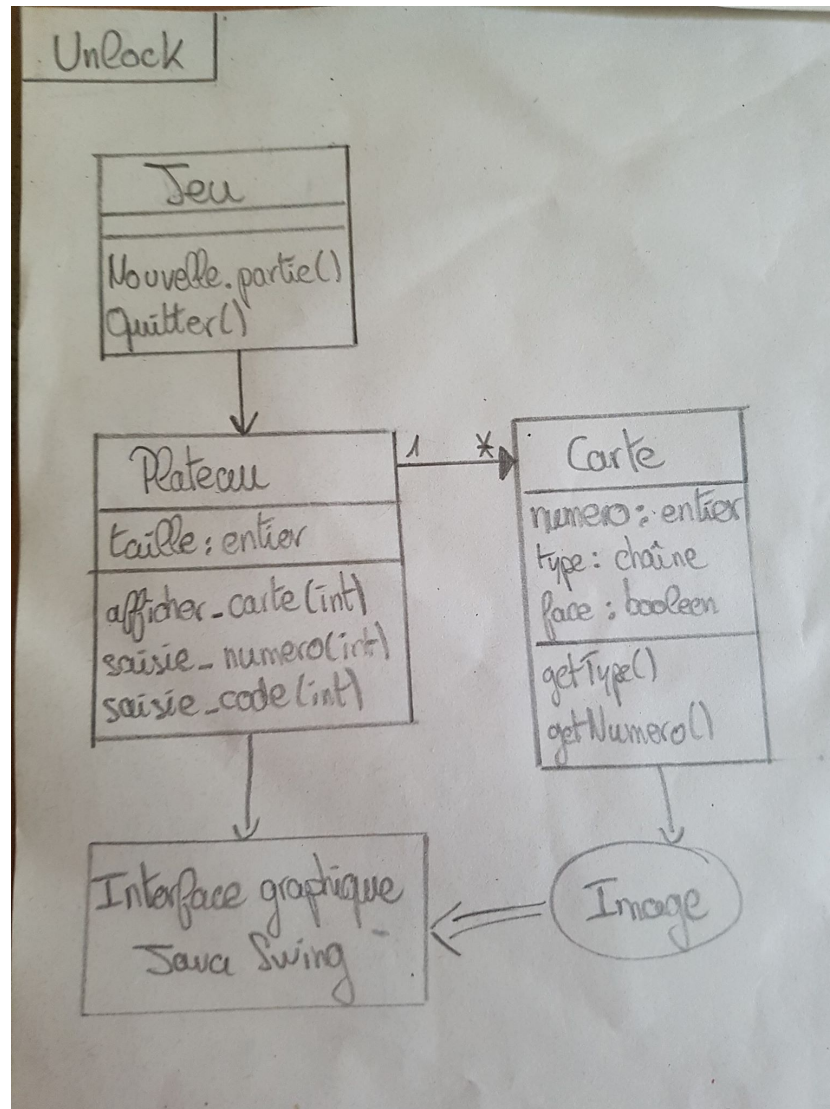


FIGURE 1 – Diagramme de classe 1

## 5 Les principaux choix de conception et de réalisation, les problèmes rencontrés et les solutions apportées

Le principaux choix de conception et de réalisation adoptés ont été les suivants :

1. Représenter les cartes grâce à une interface Carte.java,
2. Les cartes ne sont pas désignées grâce à un élément d'une énumération (par exemple `enum Carte {Un, Deux, Vide}`) mais par des entiers,
3. Utilisation d'un booléen pour savoir si une carte a déjà été découverte ou non,
4. Utilisation de la bibliothèque Swing pour l'interface graphique,
5. L'ensemble des cartes présentes sur le plateau est stocké dans un tableau à deux entrées,
6. Dans l'interface graphique, pour saisir le code, cela se fait grâce à une option du menu de la fenêtre.

Les problèmes rencontrés et les solutions apportées sont les suivants :

1. La taille des images a posé problème lors du placement de ces dernières sur une organisation de la fenêtre de type *GridLayout* : elles ont alors été redimensionnées,
2. Le passage de la représentation des cartes sous la forme d'un tableau dont les éléments sont de type énumération ou de type entier a posé problème : ce problème venait de l'utilisation du type *int* à certains endroits et *Integer* à d'autres.

## 6 Organisation de l'équipe et mise en oeuvre des méthodes agiles

Tout d'abord, nous avons discuté ensemble des différentes fonctionnalités voulues pour notre application. Ensuite, la rédaction du rapport sur les fonctionnalités a permis de nous fixer des objectifs précis.

Il a été décidé que 4 personnes se chargeraient de la programmation du jeu en lui-même, 2 personnes de la programmation de l'interface graphique et une personne des user stories, des diagrammes UML, et du compte Trello. On a aussi mené des discussions au moment de faire des choix importants quant à la réalisation du code de façon à prendre en compte l'avis de chaque membre de l'équipe. Chaque "équipe" a pris soin de partitionner le travail en tâches élémentaires, de déterminer lesquelles étaient prioritaires tout en permettant de livrer tôt de la valeur métier (en estimant l'effort que représente chaque fonctionnalité du backlog), et enfin de se les répartir.

C'est ainsi que nous avons pu *livrer tôt de la valeur métier tout en limitant le risque* (en répondant à, d'abord de petites fonctionnalités, puis en implémentant des fonctionnalités plus importantes). On a alors affiné et modifié le backlog de notre produit au cours de l'avancement de notre projet, notamment en réajustant certaines fonctionnalités : nous n'avons pas hésité à remettre en cause notre travail en changeant de direction concernant les choix de conception.

## 7 Conclusion et Objectifs

Lors de cette itération nous livrons un jeu qui présente les fonctionnalités de base du jeu Unlock et les fonctionnalités plus longues à mettre en place sont entrain d'être implémentées et seront livrées lors de l'itération 3.