



REVELIO

DOSSIER DE PROJET

PJS²

ENCADRE PAR

JEAN FRANÇOIS BRETTE

ALEXIS GOMES MATIAS

GABRIEL GUISSSET

MAUD GELLE

ARSENE LAPOSTOLET

THOMAS TURBIDE

SIXTINE TRAVERSE





SOMMAIRE

Table des matières

Remerciements.....	1
Introduction.....	2
Presentation.....	2
Concept du jeu.....	2
Annonce du plan.....	3
Analyse.....	4
Problematique du sujet.....	4
Analyse conceptuelle de l'application.....	4
Cahier des charges.....	6
Besoins fonctionnels.....	6
Contraintes.....	6
Outils et langages.....	8
Application de bureau.....	8
Application mobile.....	10
Echanges reseau.....	11
Management et Communication.....	11
Organisation du travail.....	13
Presentation de l'application.....	17
Presentation generale.....	18
Deroulement d'une partie.....	18
Conception de l'application.....	19
Logique applicative.....	19
Programmation Android.....	21
Communication Client-Serveur.....	22
Programmation JaxaFX.....	23
Aspect graphique.....	24
Validation et tests.....	27
Logique Applicative.....	27
Application Android.....	27
Interface JavaFx.....	28

Communication Client-Serveur.....	28
Conclusion.....	29
Resultat abouti.....	29
Difficultés rencontrées.....	29
Evolution possible.....	30
Annexes.....	32
Glossaire	32
Chiffres amusants	34
Sources.....	35
Liste des cartes	38
Affiche	40
Conclusions personnelles	41
Thomas Turbide.....	41
Arsene Lapostolet	42
Sixtine Traverse.....	43
Maud Gellee.....	44
Alexis Gomes Matias.....	45
Gabriel Guisset.....	46

REMERCIEMENTS

Nous souhaitons avant tout remercier les différents acteurs nous ayant aidé et soutenus durant ce projet de PJS4. Nous voulons particulièrement remercier M. Jean-François BRETTE qui nous a suivi, aiguillé, et conseillé tout au long de ce projet, l'IUT Paris Descartes pour nous avoir fourni des lieux et du matériel pour pouvoir travailler dans de bonnes conditions, ainsi que madame Véronique BEDOS et madame Hassiba ASSISSE pour nous avoir aidés à trouver des salles et nous avoir encouragés pendant tout le long du projet. Enfin, nous tenons à remercier Ariane Mayer pour ses conseils avisés quant à notre présentation orale et notre rapport écrit.

Tout le monde



INTRODUCTION

PRESENTATION

Les jeux de sociétés, et plus particulièrement les jeux de plateau, connaissent un regain d'intérêt depuis ces dernières années. De nouveaux jeux apparaissent, ne s'adressant plus seulement aux enfants et aux adolescents, mais aux adultes en complexifiant les règles et en créant des histoires plus intéressantes et plus prenantes. Cependant, certaines personnes ont encore l'idée qu'un jeu de société classique est ennuyant par rapport aux jeux numériques que l'on peut trouver sur téléphone ou sur PCs et d'autres sont bloqués par l'installation parfois longue du plateau. Nous avons donc pensé qu'il pourrait être intéressant de combiner un jeu de société avec un jeu numérique en le dématérialisant. En combinant l'utilisation du PC (qui serait l'équivalent du plateau), des smartphones permettant l'interaction avec le joueur et en automatisant toutes les opérations ne nécessitant pas de réflexions humaines d'un jeu (tirage de cartes, déplacement d'un pion, ...), cette dématérialisation a pu être effectuée.

Ce dossier va permettre de présenter le projet mis en place par Maud Gellée, Sixtine Traverse, Arsène Lapostolet, Gabriel Guisset, Alexis Gomes Matias et Thomas Turbide. Vous pourrez donc voir comment nous avons choisi, conçu et programmé notre jeu appelé Revelio. Ce dernier a pour thème principal l'univers d'Harry Potter et celui des animaux fantastiques.

Maud Gellée

CONCEPT DU JEU

Pour ce jeu, nous nous sommes inspirés du jeu *Mysterium*, un jeu d'enquête coopérative s'apparentant au *Cluedo*. Un maître du jeu (MJ) doit associer à chaque autre joueur un ensemble d'éléments du crime, soit un meurtrier, un lieu et une arme, que l'on nommera cartes enquêtes, qu'il sera le seul à connaître. On va ensuite présenter à tous les joueurs un certain nombre de cartes enquêtes, parmi lesquelles se trouvent celles associées à chacun. Une fois que le MJ a distribué les cartes enquêtes, il a 7 tours pour faire deviner à tous les joueurs les bonnes associations parmi toutes celles proposées. Pour cela, le MJ a à disposition des cartes indices, cartes oniriques et abstraites, qui doivent être données intelligemment à chaque joueur.

Si tous les joueurs n'ont pas trouvé leurs associations au bout des 7 tours, la partie est perdue, sinon, la dernière phase de jeu démarre. On présente toutes les associations de cartes à tous les joueurs et le MJ doit alors sélectionner une desdites associations ainsi que donner une dernière fois des cartes indices pour que les joueurs se mettent d'accord et choisissent l'association correspondant effectivement au crime. Si la majorité des joueurs trouve la bonne association, la partie est gagnée, sinon tous les participants ont perdu.

Nous avons repris les règles de ce jeu, en l'intégrant dans une ambiance Harry Potter, nous avons donc recréé toutes les cartes du jeu pour qu'elles correspondent à l'univers.

Maud Gellée

ANNONCE DU PLAN

Dans ce dossier nous allons voir comment nous avons créé cette application. Nous allons d'abord analyser notre projet, expliquer comment l'idée d'adapter un jeu de société en numérique nous est venue, pourquoi nous avons choisi le Mysterium, et les limites de notre projet. Nous verrons ensuite comment nous avons séparé les différentes fonctionnalités du projet et comment nous avons créé des itérations afin de hiérarchiser les fonctionnalités principales et les secondaires.

Nous rentrerons plus dans la technique dans la partie "Outil et Langage", où nous présenterons les outils utilisés et développerons la raison de leur choix. Nous indiquerons ensuite comment nous nous sommes organisés pour travailler efficacement, en expliquant notamment la répartition du travail.

Pour faciliter la compréhension du projet, sans nécessairement faire beaucoup de parties, nous présenterons plus en détails l'application et nous continuerons ensuite en détaillant chaque partie de la conception de l'application, afin d'expliquer plus en profondeur son fonctionnement. Au fur et à mesure du développement nous avons fait de nombreux tests permettant de vérifier le bon fonctionnement de toutes les fonctionnalités pour éviter d'avancer sur une version du projet dont nous ne pouvions garantir la validité.

Nous terminerons ce dossier en rédigeant une conclusion globale, où nous vous ferons part des difficultés rencontrées, et où nous réfléchirons aux améliorations possibles. Nous rajouterons enfin une annexe, dans laquelle se trouveront un glossaire pour expliquer certains termes techniques, toutes les sources que nous avons utilisées, des exemples des cartes que nous avons créées, le diagramme de Gantt que nous avons réalisé au début du projet et finalement, vous pourrez trouver les conclusions personnelles en anglais de chaque membre du groupe.

Maud Gellée



ANALYSE

PROBLEMATIQUE DU SUJET

Pour trouver un sujet de PSJ4, nous avons longuement réfléchi aux différentes applications que nous utilisons ou que nous voudrions utiliser au quotidien et il s'est avéré que nous n'avons pas trouvé de problématique suffisamment importante qui pourrait adhérer aux contraintes d'un sujet de projet tuteuré ainsi qu'à nos exigences personnelles. C'est pourquoi nous nous sommes orientés vers la création d'un jeu qui combinerait à la fois une vue **serveur** et une vue **Android**, ce qui nous permettrait d'améliorer et d'approfondir nos connaissances acquises au sein des différents modules de l'IUT. Afin de respecter les conditions imposées, nous avons réalisé de longues recherches sur les différents types de jeux réalisables avec une séparation possible des actions utilisateur. Premièrement, nous voulions réaliser un jeu de quiz sur une télévision, inspiré du jeu "qui es-tu ?" mais nous voulions un concept plus innovant, qui n'est pas déjà existant et nous nous sommes orientés vers la numérisation d'un jeu de société. Cependant, il existe un certain nombre de jeux de société adaptés sur ordinateur et c'est pourquoi nous avons préféré choisir un jeu très peu ou pas connu du grand public afin d'en démocratiser l'accès et d'augmenter sa visibilité, c'est pourquoi nous avons choisi le Mystérium.

Alexis Gomes Matias

ANALYSE CONCEPTUELLE DE L'APPLICATION

Une fois le sujet trouvé, il nous a fallu comprendre et nous approprier les règles du jeu de société afin de pouvoir en extraire les règles et les différents principes de fonctionnement de notre application. C'est pour cela que nous avons consacré la première séance à la réalisation d'une partie témoin et à la rédaction du principe de fonctionnement général d'une partie. De plus, il nous a été nécessaire de compartimenter les différentes actions réalisées au sein d'une partie pour définir les actions à faire réaliser sur le client mobile et les actions à réaliser sur la vue **serveur**. L'ergonomie et la nécessité d'une rapide prise en main de notre application a nécessité un certain nombre de réflexions afin de pouvoir ouvrir notre jeu à tous les types d'utilisateurs et que tout le monde puisse jouer sans avoir de connaissances en informatique. C'est pour cela que les utilisateurs n'auront qu'à utiliser un **QR** code afin de se connecter au **serveur** et que nous avons réalisé une interface sur l'application mobile la plus simple possible.

Alexis Gomes Matias

NOTE SUR LE DROIT D'AUTEUR

Il est important de parler du droit d'auteur dans le cadre de ce projet s'inspirant grandement d'un jeu de société existant et thématique sur une œuvre ou ensemble d'œuvres, aux-

quelles nous avons soutiré des histoires, musiques et différents aspects visuels, et dont une grande partie repose sur des productions graphiques.

En effet, notre projet s'inspire du jeu de plateau *Mystérium*, cité précédemment. Dans l'optique de connaître la faisabilité légale de notre jeu, nous nous sommes renseignés sur le droit d'auteur qui régit les jeux de société en général. Comme toute œuvre originale et issue de l'esprit, le jeu de société peut être protégé par le droit d'auteur. Parmi les éléments protégés, seules les règles et les principes du jeu nous intéressent, puisqu'il s'agit des seuls éléments véritablement empruntés au jeu *Mystérium*. Si les règles que nous avons définies divergent légèrement de celles du jeu de plateau, l'inspiration est indéniable, et l'essence des règles originelles se retrouve indéniablement dans nos règles adaptées. Cela implique alors que notre jeu ne peut être considéré comme original à 100%, de par notre emprunt aux règles du jeu de plateau *Mystérium*.

Nous avons vu qu'à l'évidence, notre jeu était inspiré d'une œuvre soumise au droit d'auteur, mais il ne s'agit pas de la seule œuvre sous ce droit sur laquelle est fondé notre projet. En réalité, notre jeu de société numérique est thématiqué autour des œuvres de J.K. Rowling, qui ne sont pas libres de droit. Elles sont utilisées pour l'histoire du jeu, les musiques ainsi que les visuels de notre application. Cependant, même si l'utilisation des images initiales issues de ces œuvres auraient posé un réel problème, le droit d'auteur ne s'applique pas dans le cas de collages et montages photo. Il s'avère effectivement que la plupart des images issues des œuvres d'Harry Potter et des Animaux Fantastiques ont été utilisées au sein de compositions personnelles et originales. Néanmoins, toutes les images n'ont pas subi ce traitement et les musiques sont utilisées dans leur forme originelle.

Dans nos multiples sources secondaires, on retrouve aussi bien des polices d'écriture que des dessins réalisés par des internautes. Ces éléments ne sont en majorité pas libres de droit mais nous créditons les auteurs à la fin de notre rapport.

Avec cette majorité écrasante de sources sous-droit, qu'est-ce que cela signifie pour notre projet ? Il se trouve qu'après renseignements, les projets effectués dans un cadre universitaire et pédagogique, notés ou non, n'ont pas la nécessité de respecter le droit d'auteur, ce qui n'est évidemment vrai que si le projet reste dans ce cadre et n'est pas réutilisé en dehors de ce dernier. Ce qui nous amène à la conclusion que l'utilisation d'images, de musiques et d'histoires que nous jugeons de qualité a été inévitable pour la production de notre projet. En effet, la création totale de nos visuels aurait été un travail trop conséquent et ne sélectionner que des images, vidéos et audios libres de droit aurait été trop lent et un handicap pour la bonne réalisation du projet. En outre, le cadre universitaire du projet ne nous limite pas aux sources libres de droits, mais il est clair qu'un possible usage commercial serait impossible en l'état.

Thomas Turbide

CAHIER DES CHARGES

BESOINS FONCTIONNELS

Afin d'être conforme à nos attentes, nous avons mis en place un cahier des charges sur lequel repose le projet. Ce dernier nous permet de définir de manière précise les fonctionnalités et contraintes auxquelles notre application doit répondre, et de pouvoir s'y référer en cas de doutes ou d'incompréhensions entre les membres de l'équipe.

Le maître mot de ce projet, aussi bien du côté PC que du côté mobile, est l'ergonomie. Notre projet visant à être multigénérationnel, familial, et plus agréable à jouer qu'un jeu plateau, c'est l'axe principal auquel nous nous référons. L'ergonomie conditionne toutes nos phases de développement, de la partie logique à l'interface. A partir de cela et des règles du jeu en découlent des fonctionnalités phares que notre application doit posséder pour remplir son but.

Parmi ces fonctionnalités du projet, on retrouve les fonctionnalités inhérentes au jeu et ses règles. On pourra citer les principales interactions comme la connexion au plateau, la distribution de cartes indices par le MJ aux autres joueurs, le vote des joueurs pour une carte ou une combinaison de cartes et la consultation des combinaisons de cartes associées aux joueurs par le MJ. De plus, nous voulions une distribution de cartes aussi simple, aussi rapide et aussi aléatoire qu'avec un jeu de plateau classique, ce que nous avons pu réaliser au sein de notre application avec un objet java d'aléatoire et une distribution automatique des cartes aux joueurs.

D'autre part, d'autres fonctionnalités non-relées au jeu et à ses règles nous ont semblé importantes. En effet, il nous nous paraissait primordial de pouvoir consulter les règles du jeu depuis son téléphone, pour que tout nouveau joueur qui arrive dans une partie puisse s'informer un minimum sur le jeu, son objectif et son déroulement. Au-delà de ceci, la volonté d'intégration de musiques dans le jeu nous a amené à développer un menu d'option afin de pouvoir contrôler ces dernières, autant sur leur volume que leur présence ou non.

Alexis Gomes Matias

CONTRAINTES

Outre les besoins fonctionnels liés à l'utilisation de notre application, il demeure un certain nombre de contraintes initiales à respecter, que ce soit pour le bon déroulement du

projet ou encore pour le bon déploiement de notre application sur les divers appareils utilisateurs.

En effet, afin de pouvoir maximiser le nombre d'utilisateurs pouvant utiliser notre application, une des principales problématiques a été de trouver la version d'**Android** sur laquelle développer notre application mobile pour que chacun puisse l'installer sur son téléphone sans rencontrer de problème à l'installation. C'est pourquoi, après mûre réflexion, nous avons décidé de la déployer sur l'**API 21** associée à la version 5 d'**Android** (Lollipop). Celle-ci couvre 85% des détenteurs de smartphone **Android**, ce qui constitue une large majorité d'appareils. En outre, elle dispose de fonctionnalités intéressantes et nécessaires pour la bonne réalisation de l'application et les fonctionnalités présentes n'étaient pas un frein au bon développement de l'application. L'utilisation d'un niveau d'**API** plus élevé n'aurait apporté que peu de fonctionnalités, et aurait restreint l'accessibilité de l'application au plus grand nombre d'utilisateurs.

Par ailleurs, pour anticiper les problèmes techniques et pour réaliser dans les meilleures conditions possibles notre projet, nous nous sommes fixés des contraintes dès le début de celui-ci. Effectivement, nous avons fait le choix d'utiliser le langage **Java**, grâce à sa dimension multifonctionnelle qui nous permet aussi bien de réaliser les connexions réseau, l'application graphique et l'application mobile. De plus, ce langage est un des langages fondamentaux en programmation que nous maîtrisons depuis plus d'un an, ce qui nous a permis d'avoir des bases solides sur lesquelles commencer notre projet. D'un autre côté, nous avons établi un développement agile qui consiste à découper notre projet par étapes de deux semaines, chaque étape permettant de faire avancer notre application. Chacune de nos étapes était une fondation de l'étape suivante, ce qui nous permettait à la fin de celles-ci de nous organiser par rapport à notre avancement précédent.

Enfin, compte tenu de la nature de notre projet, nous avons dû nous imposer un certain nombre de critères visuels pour les utilisateurs finaux. En effet, un jeu vidéo se doit d'être esthétique afin de ne pas repousser les utilisateurs de celui-ci. C'est pourquoi nous nous sommes concertés afin de proposer une charte graphique finale qui se veut la plus thématique, ergonomique et visuellement agréable possible, afin de créer une immersion du joueur dans notre application. Pour finir, il est au demeurant évident que le délai imposé par le cadre (Projet Tutoré du Semestre 4) dans lequel s'effectue la conception et le développement de ce projet est une des contraintes principales et non négligeable de ce dernier. Cette contrainte est la plus grande limite quant à l'ambition porté au projet.

Thomas Turbide

OUTILS ET LANGAGES

APPLICATION DE BUREAU

Java

Nous avons choisi **Java** car il s'agit d'un des langages de programmation que nous avons le plus étudié et le plus utilisé au sein de notre DUT. De plus, c'est un langage que nous maîtrisons tous correctement, ce qui est un avantage considérable pour notre projet. Il s'agit d'un langage moderne, performant et très utilisé ; c'est le deuxième langage le plus utilisé au monde selon PYPL.

Rank	Change	Language	Share	Trend
1	↑	Python	26.42 %	+5.
2	↓	Java	21.2 %	-1.
3	↑	Javascript	8.21 %	-0.
4	↑	C#	7.57 %	-0.
5	↓↓	PHP	7.34 %	-1.
6		C/C++	6.23 %	-0.
7		R	4.13 %	-0.
8		Objective-C	3.04 %	-0.
9		Swift	2.56 %	-0.
10		Matlab	1.98 %	-0.

Classement des langages utilisés dans l'industrie selon PYPL

Java est utilisé dans tous les domaines, du web aux applications de bureau, en passant par les applications mobiles et embarquées. Il nous a donc semblé pertinent de consolider nos compétences avec Java, de plus nous allons découvrir un environnement encore inconnu pour

beaucoup d'entre nous, comme nous n'avions jamais eu l'occasion d'en faire au cours du DUT, la programmation événementielle et graphique avec JavaFX. De plus, nous avons préféré Java à Kotlin pour la programmation **Android**, afin de faciliter les échanges entre les PC et les téléphones et pour que les membres de l'équipe puissent facilement passer de la programmation **Android** et la programmation sur l'ordinateur.

JavaFX

Java offre plusieurs bibliothèques graphiques : **SWT**, **AWT**, **Swing** et **JavaFX**. **SWT** et **AWT** sont très anciennes et possèdent beaucoup d'implémentations natives, ce qui aurait pu poser des problèmes pour la portabilité de notre application. De plus, **JavaFX** est la bibliothèque actuellement officiellement supportée par Oracle, car elle est plus moderne que **Swing**. En outre, elle est mieux conçue, et possède de très nombreuses fonctionnalités, comme des contrôles graphiques plus modernes et variés, la lecture audio et vidéo, ainsi que la possibilité d'utiliser le langage **CSS** pour définir des styles. Enfin, certains des membres du groupe avaient déjà eu l'occasion de travailler avec **JavaFX** dans le cadre de projets personnels. C'était aussi l'occasion pour d'autres de découvrir cette bibliothèque et son utilisation dans un projet conséquent, c'est pourquoi notre choix s'est porté sur **JavaFX**.

Eclipse

Eclipse est un environnement de développement intégré open source que nous avons utilisé tout au long de notre DUT pour développer et tester nos programmes **Java**. Nous sommes familiers avec son utilisation et les fonctionnalités les plus puissantes qu'il propose comme les **refactorings** de code, les tests **unitaires JUnit**, le débogage pas à pas, etc ... Ainsi, forts de ces connaissances, nous travaillons rapidement et efficacement dans cet environnement, il nous a donc paru évident de continuer à l'utiliser pour ce projet.

Dépendances

Au cours de ce projet, nous avons utilisé de nombreuses bibliothèques externes, notre projet en dépend donc. Voici la liste des dépendances de notre projet, et ce à quoi elles servent :

Google ZXing : Bibliothèque permettant de générer des codes **QR**, que nous utilisons pour que les joueurs ne soient pas obligés de renseigner l'adresse **IP** affichée par le PC, mais puissent simplement scanner le **QR** code pour se connecter.

Pelican & JAI : Bibliothèques de traitement d'image pour intégrer le code **QR** à notre charte graphique.

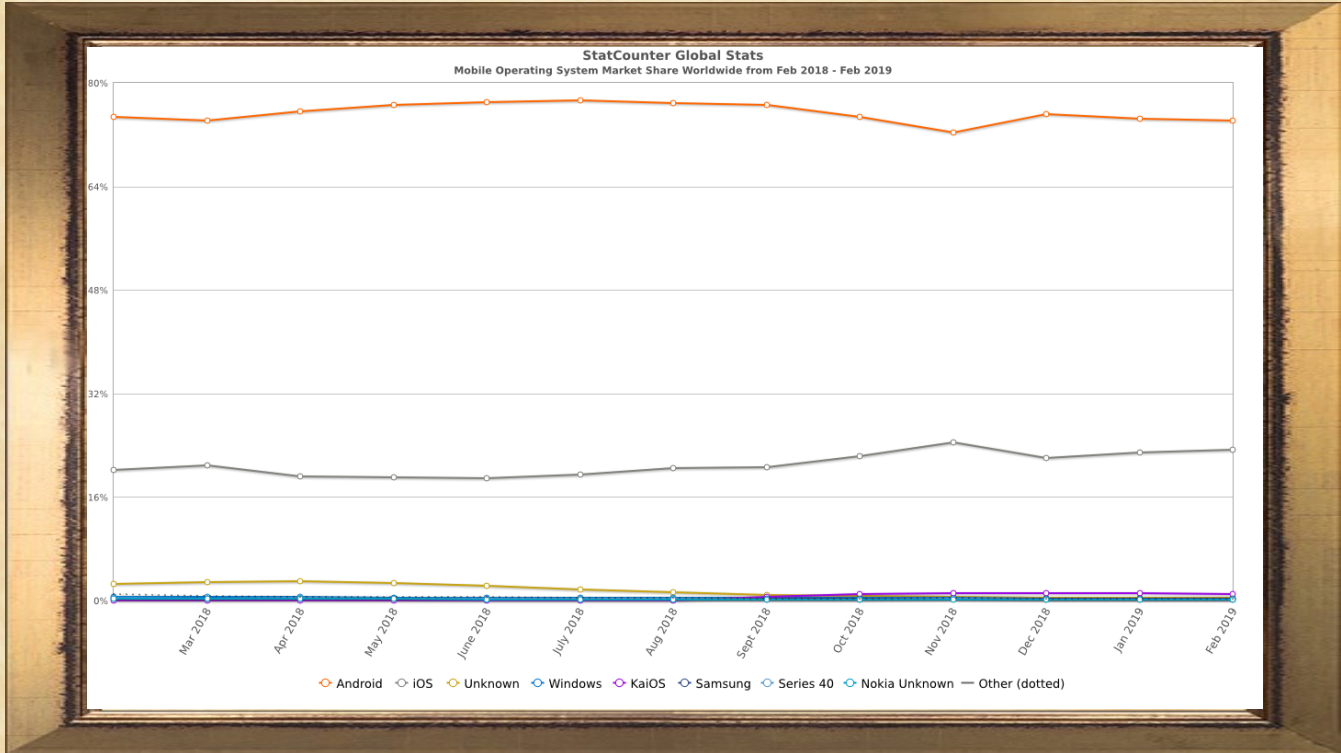
ControlsFX : Bibliothèque fournissant des contrôles graphiques supplémentaires pour **JavaFX** tels que l'apparition d'une fenêtre explicative en cas d'erreurs lors de l'exécution du programme.

JSON.simple : Bibliothèque de traitement de données au format **JSON**, permettant de faciliter la communication client - **serveur** car les données sont communiquées au format **JSON**.

Arsène Lapostolet

APPLICATION MOBILE

Android



Parts marché des différents OS mobiles d'après

Android est le système d'exploitation mobile le plus utilisé au monde avec plus de 70% de parts de marché. Cette plateforme nous a donc semblé comme un des choix les plus judicieux pour mener à bien notre projet. De plus, nous avons étudié le développement d'application **Android** au cours du module Applications Mobiles du DUT et un développement d'application iOS aurait impliqué l'utilisation de machines Apple qu'aucun des membres du groupe n'avait.

Pour le langage de développement sur **Android**, nous avons choisi **Java**, pour les mêmes raisons qu'évoquées précédemment.

Android Studio

Android Studio est l'environnement de développement intégré pour **Android** soutenu officiellement par Google. Il est basé sur la plateforme **IntelliJ**, une alternative à Eclipse. Nous avons choisi de l'utiliser car il est très puissant et possède de nombreuses fonctionnalités avancées comme le débogage pas à pas sur un appareil, la virtualisation d'appareil **Android**, la gestion de dépendance via Gradle et de nombreux **refactorings** de code. De plus nous avons acquis un peu d'expérience sur **Android Studio** au cours du module Applications Mobiles du DUT et réutiliser cette expérience pour consolider nos compétences en programmation mobile s'est avéré comme la solution la plus efficace pour réaliser l'application que nous nous étions imposés.

ECHANGES RESEAU

Sockets TCP / IP

Pour la communication réseau entre l'application mobile et l'application de bureau, nous avons plusieurs possibilités :

- Bluetooth
- Sockets TCP/IP
- Protocole HTTP (APIs REST et Websockets)

Nous avons éliminé rapidement le **Bluetooth**, car il nécessite une carte réseau sans fil sur l'ordinateur hôte, ce que nous avons considéré trop contraignant. Ensuite, un dialogue via le protocole HTTP nous a semblé trop lourd par rapport aux échanges que nous devons faire. C'est pourquoi nous avons opté pour les **Sockets TCP/IP** du package *java.net*, qui permettent une très bonne flexibilité des échanges et nous permettent de définir nous-même notre protocole de communication réseau. De plus, nous avons déjà eu l'occasion de les utiliser au cours du DUT, dans des modules comme Applications **Serveur Java** et Applications Réflexives **Java**.

JSON

Afin de transmettre des données de façon structurée sur le réseau, nous étions à la recherche d'un format d'échange de données en texte. Deux candidats s'offraient à nous : **XML** et **JSON**. Nous avons choisi **JSON** car il est plus lisible et moins verbeux que **XML**, de plus la manipulation du **JSON** dans le code **Java** est plus concise.

Arsène Lapostolet

MANAGEMENT ET COMMUNICATION

Trello

Trello est un outil de gestion de projet fonctionnant avec des listes et des cartes. Nous l'avons utilisé en tant que **backlog** pour appliquer la méthode agile au développement de notre projet. En effet, nous avons rempli une liste des tâches à effectuer, en détaillant le plus possible. Ensuite, nous avons chaque jour fait le point pour savoir quelles tâches allaient être effectuées et les avons attribuées à chacun en plaçant les cartes de ces tâches dans une liste correspondant à la prochaine séance. Ainsi, les tâches terminées étaient laissées dans le jour où elles avaient été terminées, et les tâches toujours en cours étaient reportées sur la séance suivante. Cette planification agile nous a permis de prendre rapidement des décisions et de gagner du temps de travail sur du temps d'organisation. Cet outil est gratuit et très flexible, de plus nous avons appris à l'utiliser pour appliquer la méthode agile dans le module Méthodes de Production d'Application.

Messenger

Facebook Messenger est l'application de messagerie instantanée que nous utilisons au quotidien pour communiquer à distance. C'est pourquoi nous l'avons naturellement choisie pour la communication au sein de ce projet. Nous apprécions cette application car elle propose une interface moderne, de nombreuses fonctionnalités intégrées au chat, comme par exemple les sondages, de plus, la fiabilité de ses **serveurs** est irréprochable.

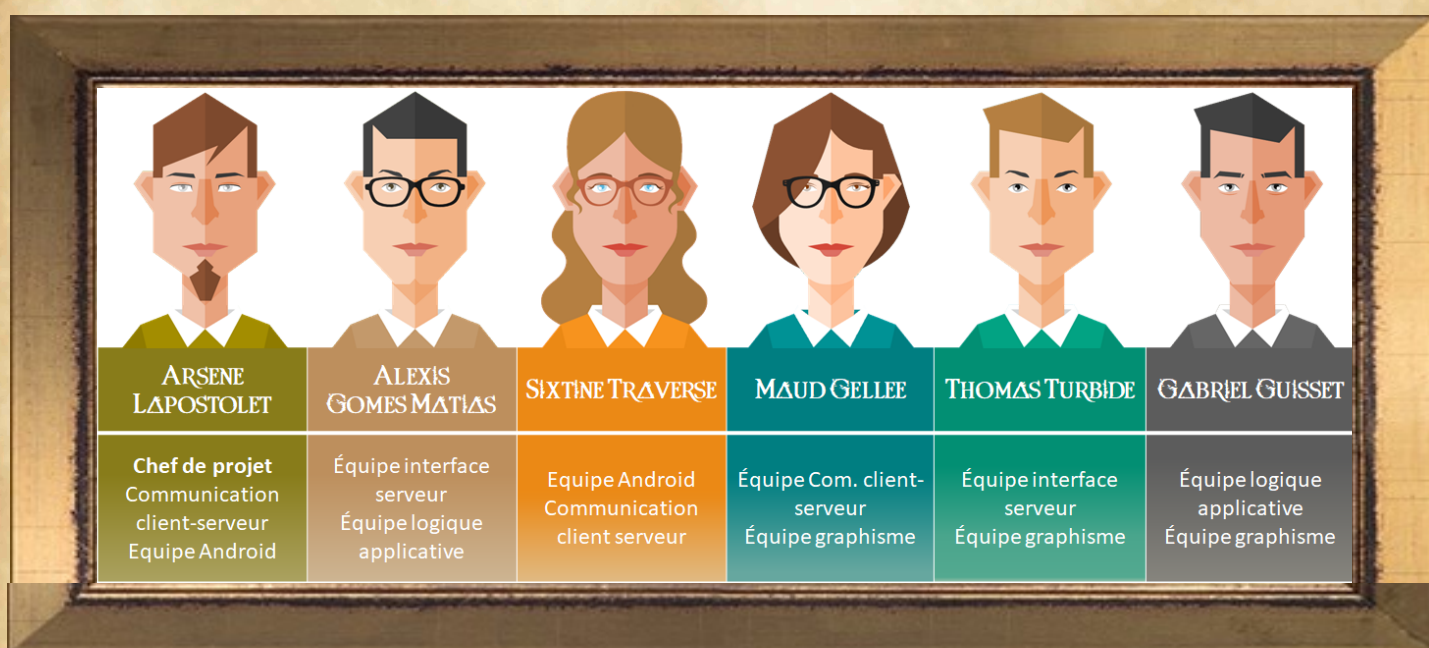
Google Agenda

Nous avons utilisé Google agenda pour nous organiser dans nos réunions. Cette plateforme collaborative de planification d'emploi du temps nous a permis de planifier facilement les plages horaires durant lesquelles nous allons nous réunir, ainsi que d'indiquer un lieu de travail. De plus cela permettait aux membres de notifier les autres de leurs absences de façon claire, permettant de mieux organiser les tâches.

Arsène Lapostolet

ORGANISATION DU TRAVAIL

Lors de notre première semaine de travail, nous avons commencé par définir les principales parties qui représentaient des pôles importants du projet. Ensuite, chacun a choisi de se positionner sur les aspects qui l'intéressaient. Nous étions donc 3 personnes par tâches, ce qui n'était pas une bonne idée vu la complexité et la quantité de travail pour chacune d'entre elles. Monsieur Brette nous a donc proposé de revoir notre organisation, lors d'une réunion avec ce dernier. Suite à cette discussion, nous nous sommes chacun attribué une tâche principale ainsi qu'une secondaire sur les parties pôles importants. Cette réunion nous a été très bénéfique et nous a permis d'être plus efficaces pour les séances suivantes









					
ARSENE LAPOSTOLET	ALEXIS GOMES MATIAS	SIXTINE TRAVERSE	MAUD GELLEE	THOMAS TURBIDE	GABRIEL GUISSSET
Chef de projet Communication client-serveur Equipe Android	Équipe interface serveur Équipe logique applicative	Equipe Android Communication client serveur	Équipe Com. client-serveur Équipe graphisme	Équipe interface serveur Équipe graphisme	Équipe logique applicative Équipe graphisme

Tableau de répartition des tâches

Ainsi, afin d'être le plus efficaces possible dans notre production, nous nous sommes séparés en binômes afin de gérer les différents aspects nécessaires au programme (la logique applicative, l'interface du **serveur**, l'interface Android, la communication client-**serveur** et les graphismes).

Projet de jeu : Revelio

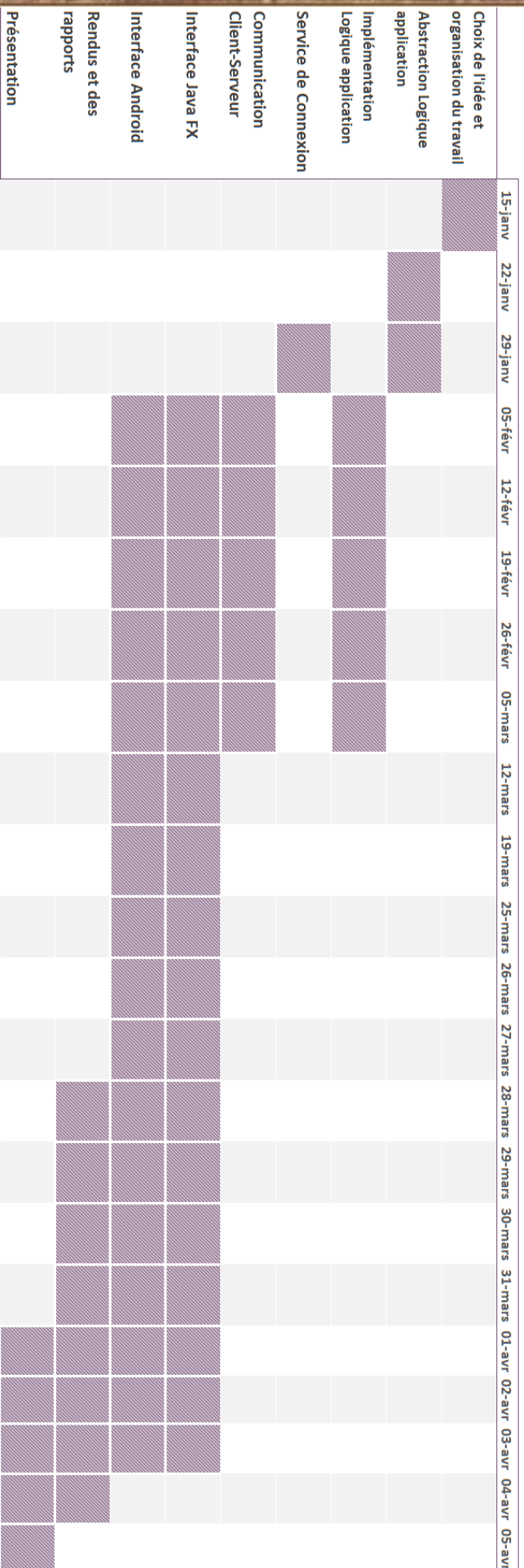
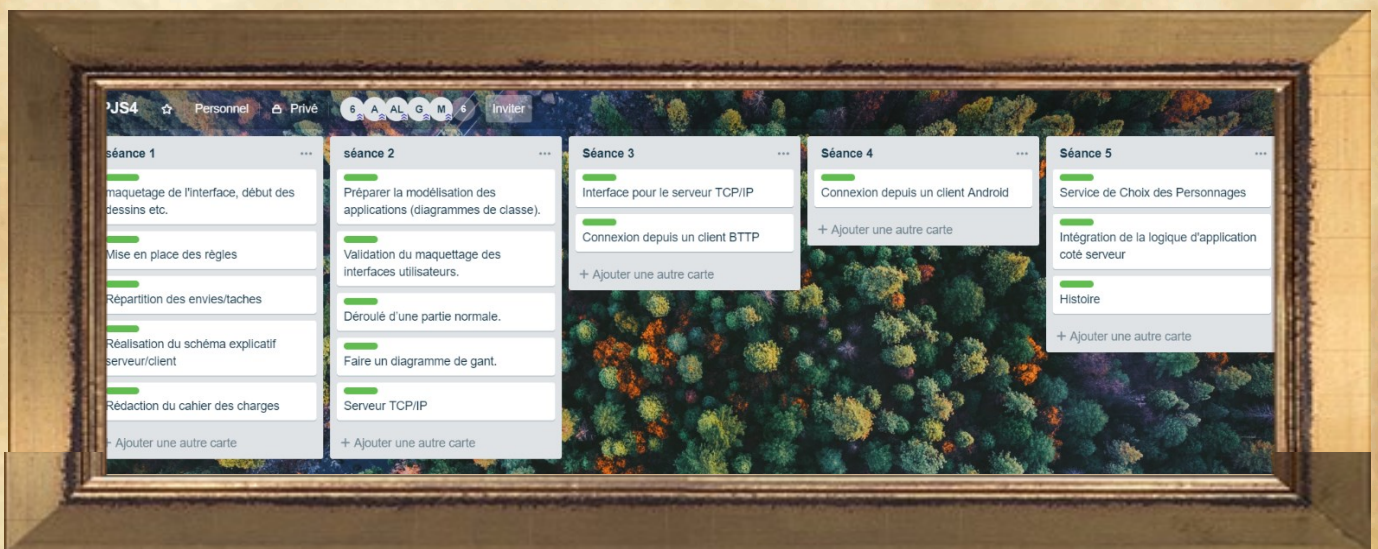


Diagramme de Gantt du Projet

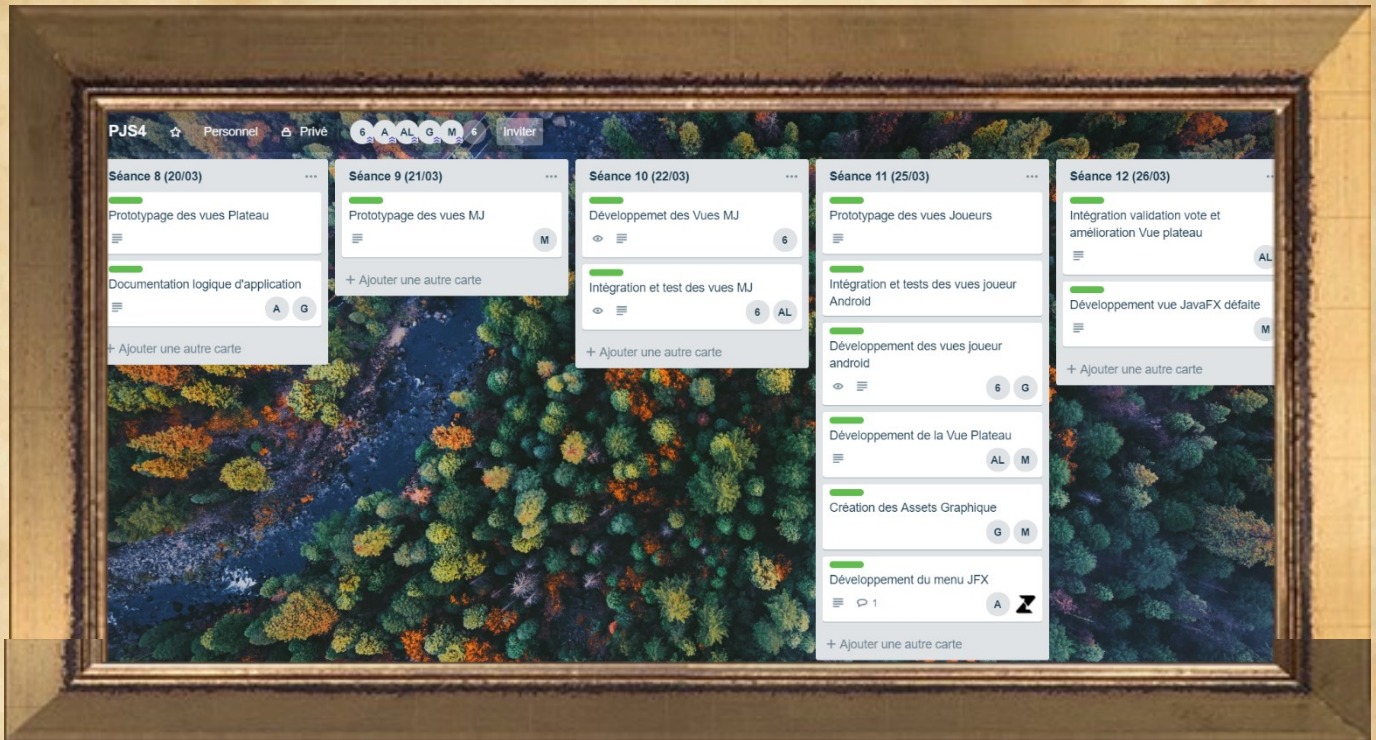
Là encore, nous, et notre tuteur, avons estimé plus judicieux de travailler en méthode agile, qui consiste en la définition de différentes itérations (version jouables) du jeu, ainsi nous limitons les risques de nous perdre parmi toutes les fonctionnalités possibles du jeu que nous pouvions développer. Cela nous a permis de nous fixer une version minimale à atteindre, mais aussi de réellement se concentrer sur les aspects importants du projet. La première itération avait pour objet d'avoir un jeu jouable, des interfaces graphiques fonctionnelles et une sélection d'histoires (association de cartes personnage, arme et lieux) aléatoires pour chaque joueur. Le second rendu exigeait en plus d'avoir des interfaces graphiques travaillées mais aussi que le maître du jeu puisse choisir les combinaisons de cartes associées à chaque joueur. Enfin, la dernière itération devait prendre en compte l'ajout de la mécanique des points de clairvoyance (voir dans la conclusion les améliorations possibles pour plus d'explication), la possibilité de choisir le nombre de joueurs et enfin la mise en place du système du **QR** code, permettant aux joueurs de le scanner afin de se connecter au **serveur**.

Cependant le déroulement du projet nous a amené à apporter des modifications à ces itérations. En effet, nous nous sommes rendu compte que certains points cités précédemment n'étaient pas si importants, comme la sélection des histoires aléatoires. En effet, nous avons trouvé une alternative plus intéressante et simple à mettre en place (chaque joueur choisit chacun son tour le personnage qu'il incarnera). Nos différentes versions prévues pour le jeu ont donc évolué au cours de ces semaines de travail.

Au début de chaque séance nous faisons un petit récapitulatif de ce que nous avons fait, de ce qui nous restait à faire et de ce que nous allions faire durant la séance. Cela nous a permis d'être organisé et efficace mais aussi de mettre en commun l'avancée de chaque équipe. De plus, nous avons décidé de mettre en place un Trello, un outil d'organisation de travail collaboratif, afin de définir les tâches à réaliser pour chaque séance. Nous avons déjà pu utiliser cet outil lors d'anciens projets, et nous le trouvions très simple à prendre en main et à utiliser, ainsi qu'en adéquation avec la méthodologie agile avec laquelle nous travaillons.



Trello comprenant les tâches à faire des séances 1 à 5



Trello comprenant les tâches à faire des séances 8 à 12

Dès le début du projet nous avons établi différents outils de travail et de communication. Nous avons créé un groupe de travail et des projets Git (via FramaGit) pour chaque équipe. Ces projets étaient un projet “Client” pour tout ce qui est communication réseau, un projet “**Android**” pour le développement côté Android et enfin un projet “Serveur” pour l’équipe développant le serveur et l’interface graphique du plateau. Ainsi, chaque équipe a pu travailler aussi bien ensemble qu’à distance grâce à cet outil. Une fois la partie réseau développée et fonctionnelle, nous l’avons intégrée à la partie Android, afin de pouvoir commencer les échanges entre le serveur et les portables des joueurs. De plus, cette partie nous a permis de réaliser l’ensemble des tests de notre application. Un nouveau projet Git “Client-Android” a été créé sur lequel la suite du développement Android, prenant en compte les communications avec le plateau, a pu se poursuivre. Par ailleurs, une équipe a créé un client BTTP qui permet de simuler des joueurs se connectant au serveur. Cela nous a permis de gagner du temps lorsque nous avons simplement besoin de réaliser des tests sur le serveur, et la communication avec un client.

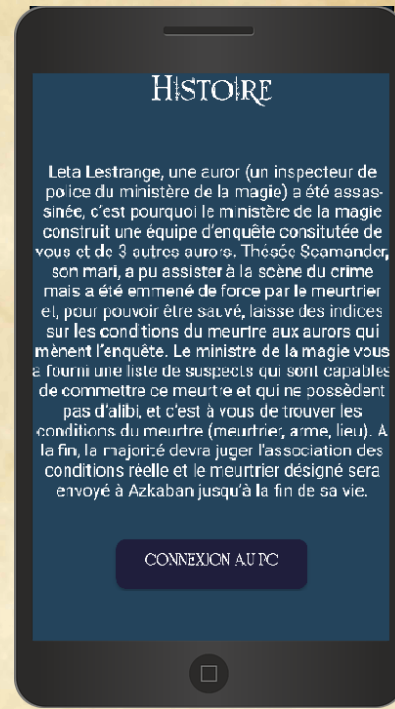
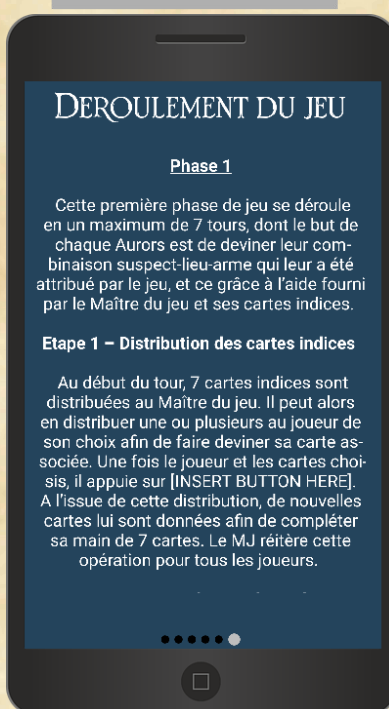
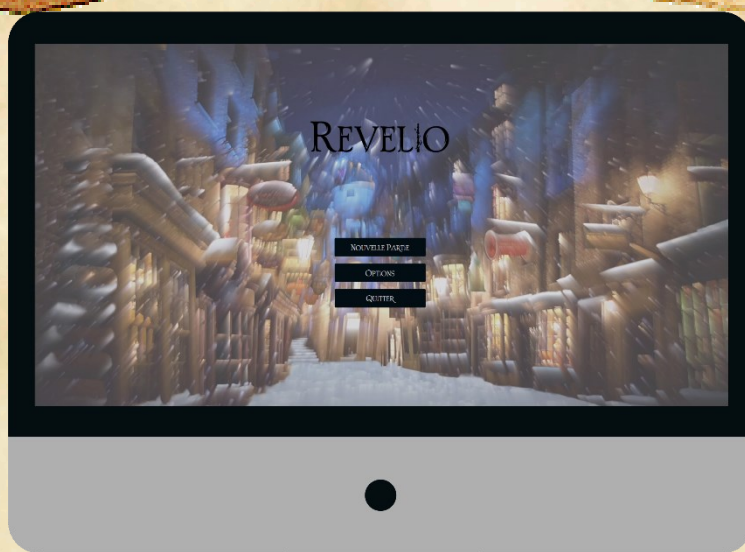
Par ailleurs, nous avons utilisé le logiciel **Fork** afin de simplifier les opérations à réaliser sur **Git**.

De plus, nous nous sommes servis d’office 365, de façon à pouvoir rédiger nos comptes rendus chaque semaine, de sorte à ce que toutes nos ressources soient accessibles pour tout le groupe, où que l’on soit. Nous y avons partagé nos images, notre cahier des charges, les règles du jeu et d’autres documents écrits.

Pour finir, même si nous étions assignés à une tâche bien précise durant le projet, chacun a pu apporter son aide au sein des différentes équipes lorsque cela était nécessaire.

Sixtine Traverse

PRESENTATION DE L'APPLICATION



Captures d'écran des applications

PRESENTATION GENERALE

Notre application est l'adaptation numérique d'un jeu de société : le mystérium. Celui-ci est un jeu coopératif dans lequel une équipe de 5 joueurs travaille et réfléchit ensemble à la résolution d'une enquête de police autour du meurtre d'une comtesse. L'application réalisée reprend cette idée générale de partie en l'adaptant avec l'histoire que nous avons créée :

“Leta Lestrage, une auror (un inspecteur de police du ministère de la magie) a été assassinée, c'est pourquoi le ministère de la magie construit une équipe d'enquête constituée de vous et de 3 autres aurors. Thésée Scamander, son mari, a pu assister à la scène du crime mais a été emmené de force par le meurtrier et, pour pouvoir être sauvé, laisse des indices sur les conditions du meurtre aux aurors qui mènent l'enquête. Le ministre de la magie vous a fourni une liste de suspects qui sont capables de commettre ce meurtre et qui ne possèdent pas d'alibi, et c'est à vous de trouver les conditions du meurtre (meurtrier, arme, lieu). A la fin, la majorité devra juger l'association des conditions réelle et le meurtrier désigné sera envoyé à Azkaban jusqu'à la fin de sa vie.”

Pour jouer à notre application, il faudra 5 utilisateurs, qui auront chacun un téléphone **Android** pour réaliser les différentes actions liées au fonctionnement du jeu et d'un ordinateur qui jouera le rôle d'affichage global du jeu.

Alexis Gomes Matias

DEROULEMENT D UNE PARTIE

Pour lancer une partie, il faut d'abord lancer le serveur, c'est à dire l'application installée sur l'ordinateur. Un utilisateur doit sélectionner le bouton nouvelle partie et tous les utilisateurs peuvent soit scanner le **QR** code qui s'affichera à l'écran, soit écrire directement l'adresse **IP** affichée au-dessus dudit **QR** code. Une fois les 5 joueurs connectés, le premier qui se sera connecté deviendra le MJ (maître du jeu, soit le rôle de Thésée Scamander) et les 4 autres joueurs devront choisir, les uns après les autres, le personnage qu'ils incarneront. Une fois que tous les rôles ont été répartis, la partie commence et le serveur attribue à tous les joueurs une association de meurtriers, lieux et armes, et les transmettra au MJ. Ce dernier aura à sa disposition les cartes indices et les distribuera durant les 7 tours suivant pour faire deviner au joueurs leurs associations.

Une fois les 7 tours écoulés la partie est perdue si tous les joueurs n'ont pas deviné leur association de cartes (ou histoire). Néanmoins, s'ils l'ont tous devinée, ils doivent finalement deviner laquelle des 4 histoire s'est réellement déroulée, grâce aux derniers indices que le MJ leur distribuera. Si la majorité des joueurs (soit 3 aurors) trouve la bonne histoire, la partie est gagnée, sinon, elle est perdue pour tous les joueurs.

Maud Gellée

CONCEPTION DE L'APPLICATION

LOGIQUE APPLICATIVE

Afin de concevoir l'organisation des données et leur logique, nous avons d'abord dû organiser les données utilisées par notre programme. Nous avons fait en sorte que toute notre application respecte les principes SOLID afin de créer un code stable et facilement évolutif (afin de rajouter des types de cartes, ...). Dans cette partie, nous avons réalisé les opérations atomiques permettant le bon fonctionnement du jeu sans prise en compte de la connexion réseau et des interfaces **Android** et **JavaFx**, le déroulement d'une partie peut donc être réalisé grâce à cette portion de code.

Ce code s'organise en différents **paquetages** :

- Un **paquetage** général nommé Jeu gère la partie, c'est-à-dire les différentes opérations atomiques propres au bon fonctionnement du jeu (c'est la **classe** essentiellement utilisée par le **serveur**) et les interfaces des différents objets de notre programme (cartes, maitre du jeu, aurors, ...).
- Un second **paquetage**, nommé Joueur, définit les implémentations des différents types de joueurs et leurs interactions.
- Un troisième, appelé Carte, permet de définir les différents types de cartes du jeu.

Alexis Gomes Matias

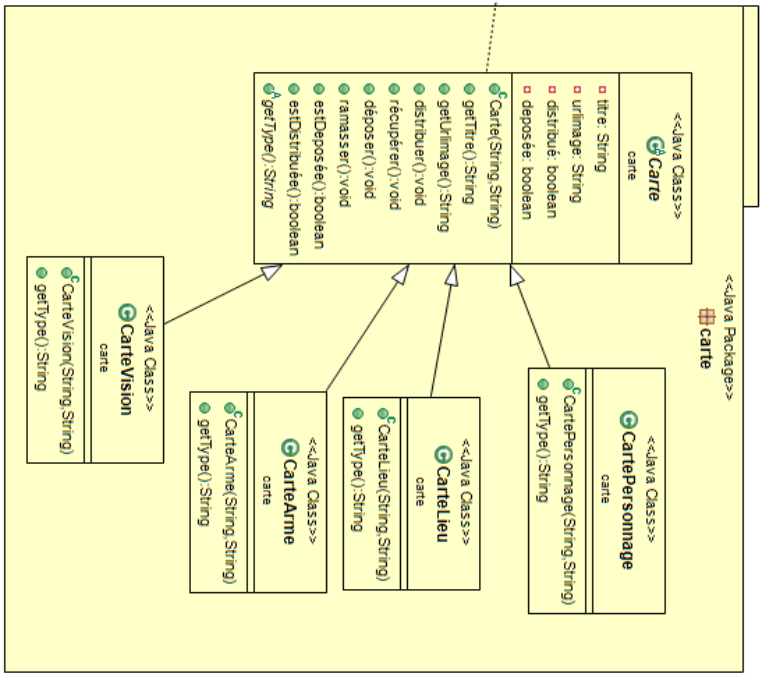
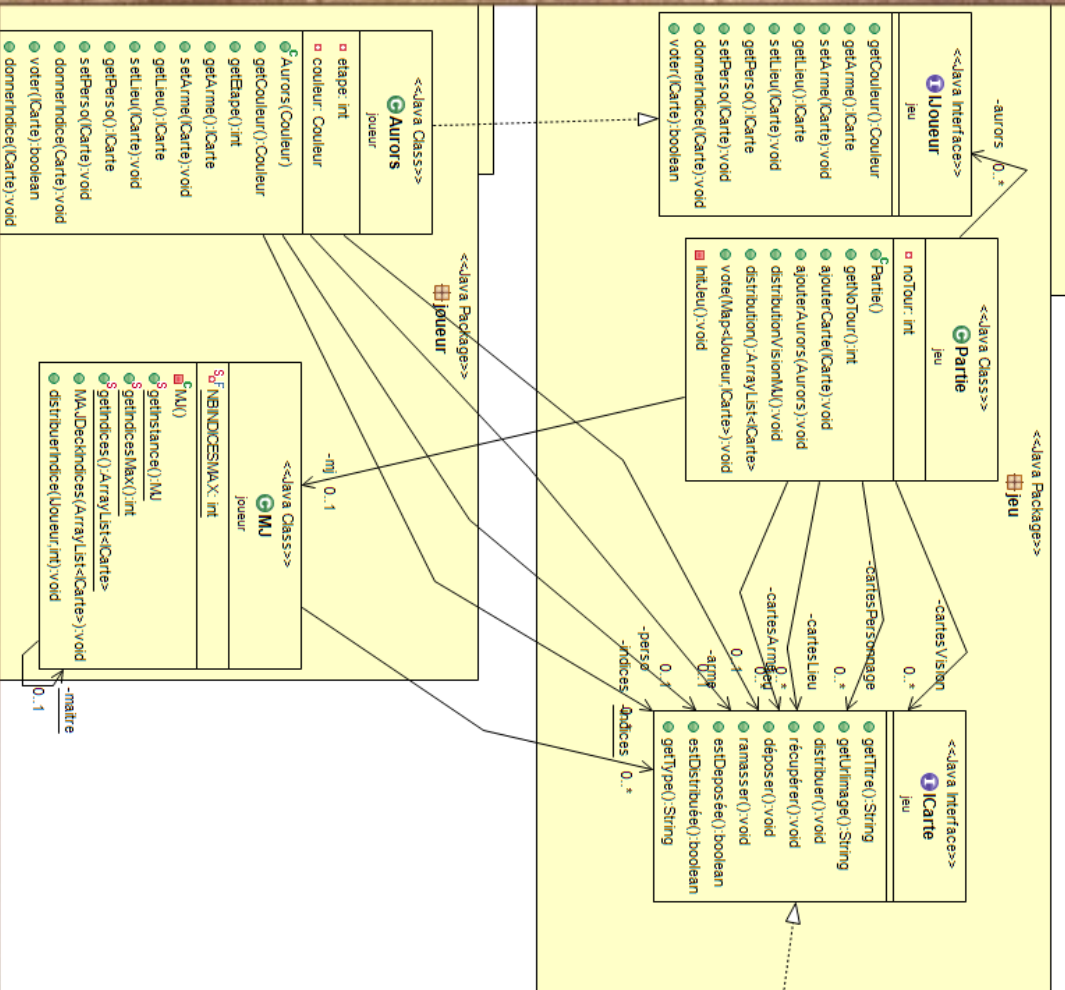
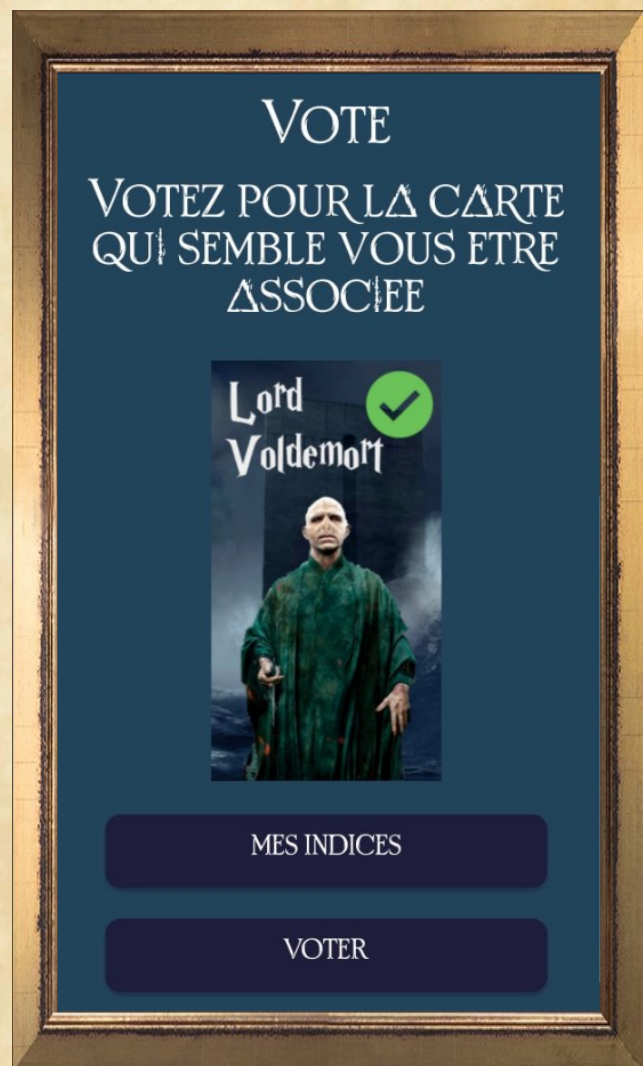
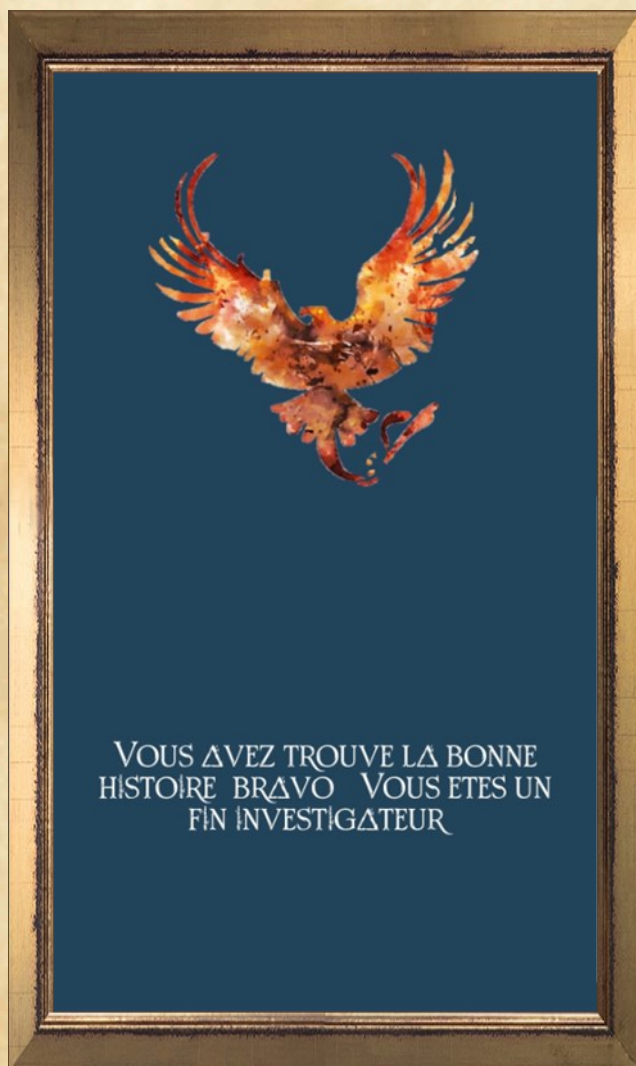


Diagramme de classes UML

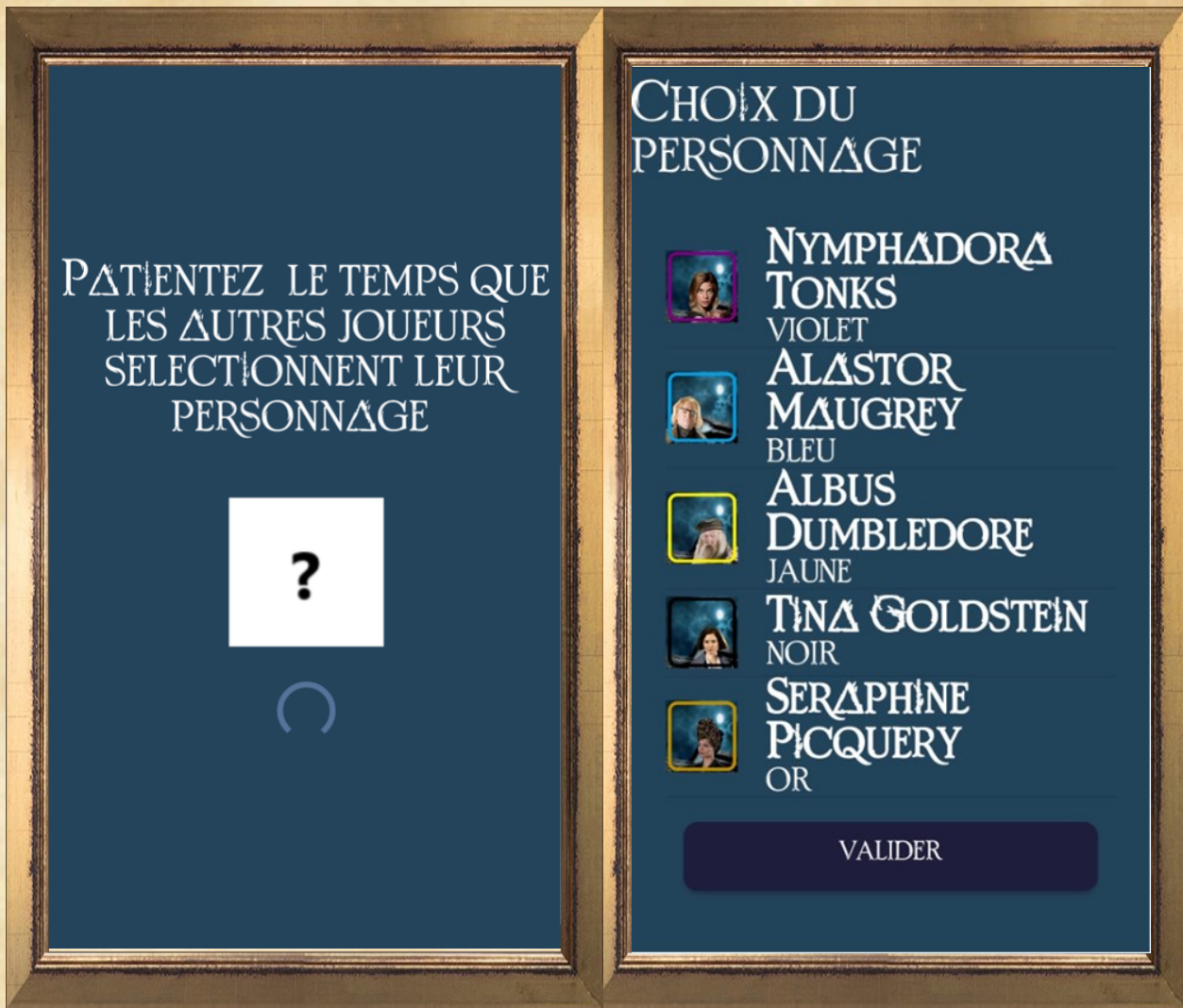
PROGRAMMATION ANDROID

Le côté **Android** est principalement composé d'**activités** et de fichiers **XML** permettant de gérer l'interface graphique de l'application. Cependant, il comprend aussi des **classes java** qui gèrent la création de certains objets logiques comme des Aurors, une Bibliothèque stockant tous les aurors de la partie, des cartes cliquables ou encore la création de sliders et de listes. Le composant majeur de notre application est d'ailleurs le slider, qui permet aux joueurs de visualiser leurs cartes. Nous avons donc créé des **SliderAdapter** afin de styliser ces éléments de la manière dont nous le souhaitions. Par exemple, nous avons rendu chaque carte du slider cliquable de façon à ce que le maître du jeu (MJ) puisse distribuer des cartes aux autres joueurs en les sélectionnant, ou encore que ces joueurs puissent voter pour la carte qu'ils pensent être la leur. Pour rendre une carte cliquable, il a donc fallu que l'équipe graphique réalise deux styles pour chaque carte : un style non cliqué et un style cliqué.

On a donc un **paquetage** "activities" où chaque activité (au total 15) correspond aux différentes phases du jeu. La particularité de la partie **Android**, est que, durant une partie, nous avons 5 joueurs, mais 4 d'entre eux correspondent à des aurors, et un au MJ. C'est pourquoi nous avons dû développer des vues différentes entre le maître du jeu et les aurors. Chaque téléphone verra donc ses données de jeu évoluer différemment les par rapport aux autres, puisque chacun d'entre eux possède sa propre identité.



Captures d'écran de l'application



Captures d'écran de l'application

Toutes les données qui s'échangent entre les téléphones, que ce soit des distributions de cartes, des votes (etc.), doivent passer par le **serveur** qui les traite et les achemine jusqu'au téléphone destinataire (MJ ou auron). Ainsi, le **serveur** peut réaliser ses opérations logiques, comme supprimer les cartes distribuées de la pioche, ajouter les cartes indices pour chaque auron, faire passer un auron à l'étape suivante etc.

La communication se fait grâce à l'API que l'équipe **Client-Serveur** a développé, qui est décrite ci-dessous.

Sixtine Traverse

COMMUNICATION CLIENT SERVEUR

La communication client-**serveur** s'organise en deux **paquetages** distribués qui intègrent une **API** que nous avons réalisée permettant de se connecter, de lire (recevoir et traiter des informations envoyées par le client) et d'écrire (envoyer des informations au client) sur le réseau grâce à des méthodes simples.

Nous avons dû utiliser une architecture exploitant le parallélisme. En effet, afin de pouvoir accepter plusieurs connexions réseau simultanément, notre application doit être **multi threadée**. De plus, nos applications utilisant des interfaces graphiques, les opérations de lecture réseau ne doivent pas la bloquer, ce qui appelle l'utilisation de **threads**, c'est pourquoi nous avons été amenés à utiliser des **API** de parallélisme dans nos deux **paquetages**, **serveur** et **client**.

Le premier, sur le **serveur**, exploite l'**API** *javafx.concurrent* afin de créer des services réseau permettant de dialoguer sur le réseau avec les clients selon les différentes étapes du déroulement de l'application.

Le second, sur le client **Android**, utilise l'**API** *AsyncTask* d'**Android** afin de créer des services d'envoi et de réception sur le réseau, avec une gestion événementielle de la réception des messages grâce au **Patron de Conception** *Observer*, afin de pouvoir déclencher les actions graphiques au moment adéquat.

Arsène Lapostolet

PROGRAMMATION JAXAFX

L'interface **JavaFX** se décompose en deux **paquetages** : le package de gestion des vues et celui des managers.

Le premier **paquetage** est constitué de l'ensemble des **classes Java** servant d'interface à l'utilisateur. En effet, il est responsable de l'ensemble des éléments graphiques à afficher sur l'écran de l'ordinateur pour permettre aux utilisateurs d'utiliser l'application et de naviguer entre les différents écrans (appelés vues) de l'application. Ces éléments graphiques sont pour la majorité des éléments de base de la bibliothèque **JavaFX** et sont utilisés pour l'affichage des divers éléments reçus par la partie réseau du projet (cartes, pastilles ...) ou pour les contrôles utilisateurs (changement de vues, options ...). De plus, il contient aussi un composant essentiel à son bon fonctionnement qui est le **FX CSS**, un **CSS** propre à **JavaFX** qui permet de définir l'aspect, le positionnement et diverses améliorations graphiques de l'ensemble des contrôles FX.

Thomas Turbide

Ces contrôles utilisateurs font parfois appel au **paquetage** des managers qui contribuent au bon fonctionnement de l'application et qui sont l'ensemble des composants auxquels l'utilisateur ne doit pas accéder et ne doit pas utiliser lors du fonctionnement de l'application. Ceux-ci gèrent le fond vidéo, la musique ou encore la sauvegarde des options du jeu mais sont aussi les éléments fondateurs des vues (les objets "racine" y sont notamment définis) et les éléments qui permettent de les manipuler ou de les gérer.

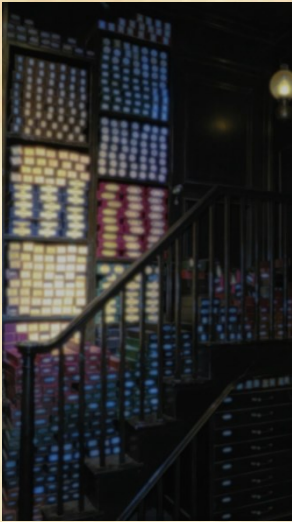


Enfin, la programmation **JavaFX** allait de pair avec l'aspect graphique du projet et il a plusieurs fois été nécessaire de demander aux responsables de cet aspect de réaliser des images utilisées dans nos menus.

Alexis Gomes Matias

ASPECT GRAPHIQUE

Nous avons voté la partie graphique au début du projet, et nous avons opté à l'unanimité pour inscrire le jeu dans l'univers d'Harry Potter. Nous devons donc modifier tous les graphismes pour qu'ils correspondent au monde créé par J.K Rowling. Nous avons choisi tous ensemble les personnages que pourront choisir les joueurs, ainsi que les meurtriers (mages noirs), les lieux et les armes. Une fois les choix fait, nous avons récupéré des photos de ces différents éléments sur internet grâce à différents sites (Pinterest, deviantArt, ...) et nous avons ensuite fait des montages, pour détourner les personnages, meurtriers et armes, ajouter un fond cohérent et rajouter les titres de chaque carte. Nous avons donc mis en place des montages où chaque type de carte (personnage, meurtrier et arme) possède une trame de fond.

Gabriel Guisset

		
Fond des cartes Arme	Fond des cartes Meurtrier	Fond des cartes Personnage

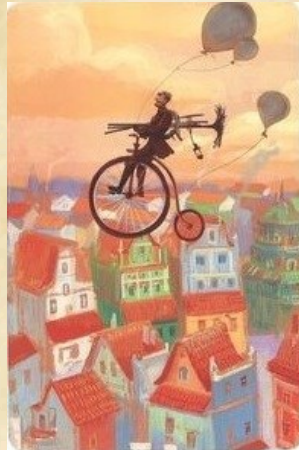
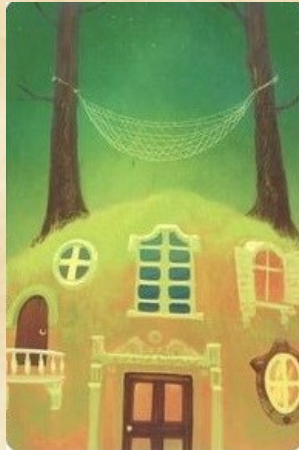
Fonds de cartes Enquête

Une fois les cartes enquêtes réalisées, nous avons dû partir à la recherche des cartes indices. Pour cela, nous avons cherché des fan art Harry Potter (et animaux fantastiques). Nous avons fait un grand tri, ce qui fait que nous n'avons pas pu trouver un nombre suffisant de cartes et avons donc dû rajouter aussi celle du Mysterium original. Il est donc possible de voir une différence de style entre les cartes indices que nous avons trouvé et les cartes indices provenant directement du Mysterium.

*Carte indice
de notre jeu*



*Exemples de
carte indice
du Myste-
rium*



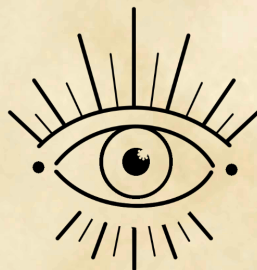
Exemples de cartes du jeu

Nous avons dû créer deux versions de toutes les cartes possibles, une version de base que nous présentons aux joueurs et sur le plateau, et une version validée qui apparait lorsqu'un joueur sélectionne une carte (soit lorsque le MJ choisit quelles cartes indices il enverra aux joueurs, soit lorsqu'un joueur sélectionne la carte qu'il pense être la sienne).

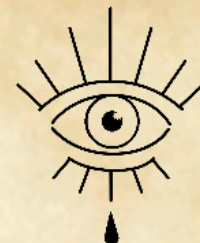


Cartes cochées

Nous avons aussi cherché et créé de nombreuses icônes et assets graphiques en tout genre pour créer un univers cohérent, nous avons aussi créé un logo pour l'application. Voici le logo :



Ce logo a été inspiré par le mouvement de baguette nécessaire pour lancer le sortilège Revelio (nom du jeu) dans l'un des premiers jeux Harry Potter. De plus, l'œil est un symbole de vérité, ce qui semble être un bon logo pour un jeu d'enquête. Avant de parvenir à ce logo nous avons pensé à de nombreuses versions dont voici quelques exemples :



Maud Gellée

VALIDATION ET TESTS

Afin de vérifier le fonctionnement de notre application, nous avons effectué différents types de vérifications que ce soit pour la logique applicative, l'application **Android**, l'interface **JavaFx** ou la communication client-serveur.

LOGIQUE APPLICATIVE

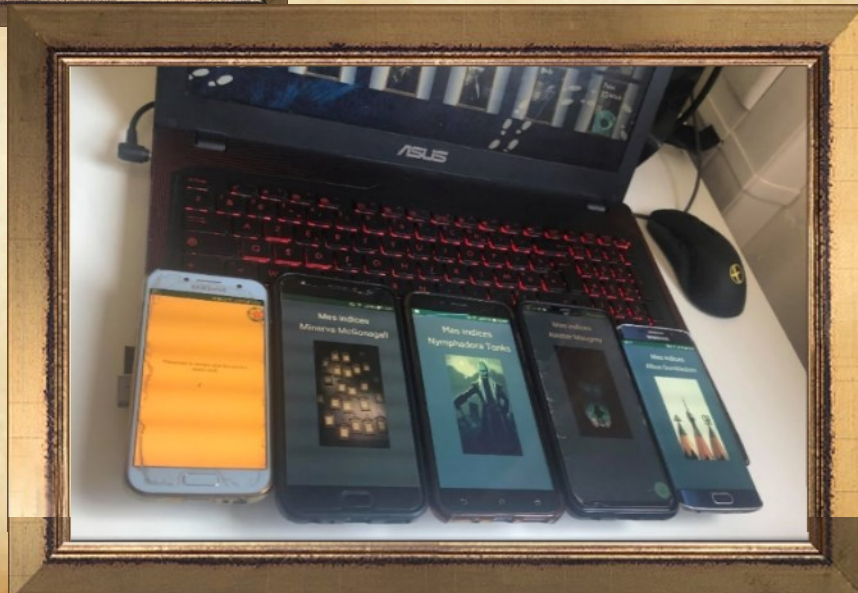
Puisque cette partie du programme est le fondement de l'organisation du jeu, nous devons obligatoirement tester le bon fonctionnement de celle-ci. Pour cela, grâce à **JUnit**, nous avons testé les différentes fonctionnalités du jeu grâce à des données factices fixes, qui nous ont de plus permis d'éviter l'effet aléatoire du tirage des cartes et de vérifier que les cartes étaient bien distribuées aux joueurs. Une fois ces tests **JUnit** réalisés et passés, nous avons de plus dû faire quelques modifications au code au fur et à mesure de l'avancement du projet pour mieux répondre aux problèmes trouvés ou pour faciliter le travail des développeurs **Android**.

APPLICATION ANDROID

Afin de réaliser l'ensemble des tests fonctionnels de notre application **Android**, nous n'avons pas pu faire autrement que de les réaliser en conditions réelles de jeu. En effet, la portion de code portant sur le côté mobile ne contient aucune logique applicative et n'est composée que de connexion réseau et d'aspects visuels liés à l'interface pour l'utilisateur, ce qui nous a empêché de réaliser des tests **unitaires**. Cependant, afin de vérifier le bon fonctionnement de l'application sur n'importe quel type de téléphone **Android**, nous avons réalisé des tests sur des appareils de différentes tailles et de différentes versions, ce qui nous a aidé à améliorer l'ergonomie de notre application.



*Photos du
Playtest Beta*



INTERFACE JAVAFX

L'interface **JavaFX** étant l'interface affichée sur l'ordinateur, les tests réalisés sur celle-ci ont consisté en des tests visuels afin de vérifier la bonne position et le bon fonctionnement des différents objets graphiques à afficher sur l'écran. Nous avons dû réaliser plusieurs utilitaires qui simulent la connexion au **serveur** de clients afin de pouvoir naviguer entre les différentes vues du jeu qui ne sont accessibles que lors du déroulement normal du jeu en réseau. De plus, la création d'un logger a facilité la phase de tests en affichant dans la fenêtre du projet les actions réalisées par le **serveur** à chaque étape.

COMMUNICATION CLIENT SERVEUR

Pour tester le fonctionnement de notre **API** de communication client-**serveur**, nous avons dressé une liste des cas d'utilisation possibles et nous les avons mis en œuvre, grâce à des **client BTTP** pour surveiller leur bon déroulement.

Afin de déboguer plus efficacement, nous avons développé une API Logging pour avoir tous les événements se produisant sur l'application sur la console de l'application en temps réel. Ainsi, nous pouvions surveiller les messages envoyés et reçu par le serveur.

Ensuite pour tester l'intégration de la communication aux applications, nous avons effectué des tests de communication en condition réelles en déployant l'application coté **serveur** et coté client.

Gabriel Guisset

CONCLUSION

RESULTAT ABOUT

Nous avons finalement réussi à réaliser l'itération 2, c'est à dire qu'il est possible de jouer une partie sans prendre en compte les points de clairvoyances (voir évolution possible). Pour se connecter au PC les téléphones peuvent inscrire directement l'adresse **IP** ou bien scanner le **QR** code. Nous avons réussi à appliquer une charte graphique esthétique sur toutes les vues **Android**, et cohérente avec celle de la vue du plateau.

Les associations de chaque joueur sont données aléatoirement par le **serveur**.

Au moment de la sélection des personnages et au fur et à mesure du choix des autres joueurs, la liste des personnages disponible change pour n'afficher que les personnages qui n'ont pas encore été pris.

Gabriel Guisset

DIFFICULTES RENCONTREES

Ce projet a été le projet le plus complet que nous avons pu faire. En effet, nous avons déjà eu l'occasion de travailler en groupe lors de notre DUT mais jamais pour un projet de cette envergure. Travailler sur un projet aussi long et complexe en équipe de 6 fut une expérience très enrichissante. Nous nous sommes réparti le travail en fonction des compétences et des envies de chacun pour que tous les membres du groupe prennent plaisir à travailler et que nous puissions avancer vite et dans la bonne humeur. Bien entendu nous avons rencontré quelques difficultés au sein du groupe qui ont pu nous faire perdre un peu de temps ou créer des tensions, mais nous avons toujours réussi à faire des compromis et à finalement nous entendre entre nous. Pour cela, nous nous réunissions et nous posions afin de régler les problèmes et désaccords que nous rencontrions afin de trouver des solutions en équipe pour que chacun puisse exposer son point de vue et ainsi prendre une décision en toute connaissance de cause.

Une des principales difficultés que nous avons rencontrées est de trouver des salles informatiques libres pour pouvoir travailler au calme et sans être dérangés. Les salles informatiques réservées pour le PJS4 n'étaient pas affichées et la cellule informatique ne semblait pas au courant des réservations de salles pour le projet.

Pour la partie graphique nous avons dû faire de nombreuses recherches sur internet et trouver un grand nombre de cartes de bonnes qualités et correspondant à ce qu'on veut fut parfois très compliqué, en particulier pour les cartes indices. Nous devons faire beaucoup de tri pour garder une cohérence graphique et conserver des fan art de qualité. Nous n'avons

donc pas pu trouver un nombre suffisant de cartes pour permettre de jouer au jeu, nous avons donc dû ajouter les cartes indices du jeu original.

D'un point de vue techniques, la plus grosse difficulté a été de mettre sur le même plan le développement d'interfaces graphiques événementielles et celui d'interactions réseau et du parallélisme. Nous avons également rencontré quelques problèmes avec le packaging de logique d'application, dans lequel nous avons découvert quelques bugs au moment de son intégration dans le code client. Android Studio nous a également donné un peu de fil à retordre avec son débogueur qui provoquait des arrêts inexplicables de notre application.

Arsène Lapostolet

EVOLUTIONS POSSIBLES

Il existe deux principales améliorations pour notre jeu. La première serait de permettre de jouer de 2 à 7 personnes, comme le *Mysterium* de base, et non pas être nécessairement 5, ainsi il serait moins contraignant de jouer à notre jeu. Nous avons préféré nous concentrer sur l'aspect logique du jeu, pour avoir plus facilement une version utilisable, mais nous n'avons pas pu revenir sur cela après avoir fini la deuxième itération. Cependant, nous avons pensé l'application de telle manière qu'il ne serait pas très dur de la faire évoluer dans ce sens, car nous savions dès le début de la création que nous voudrions implémenter cette utilisation.

La deuxième amélioration majeure serait d'ajouter les points de clairvoyance. Les points de clairvoyance font partie du jeu de base mais ne sont pas nécessaires au bon déroulement du jeu. Le principe est que tous les joueurs possèdent des jetons de clairvoyances (chaque joueur en possède 4 ou 6 en fonction du nombre de participant), une fois que tous les joueurs ont sélectionné la carte enquête qu'ils pensent être la leur (meurtrier, lieu ou arme), tous les joueurs peuvent indiquer s'ils soutiennent ou non la décision de chaque joueur, par rapport aux cartes indices distribuées par le MJ.

Les joueurs ayant eu la bonne intuition sur les autres (c'est à dire les joueurs ayant bien prédit la validité ou l'invalidité du choix d'un joueur) gagnent des points de clairvoyance et ces derniers serviront à la fin du jeu. Une fois le moment où tous les joueurs doivent choisir la bonne association, ils n'auront pas accès au même nombre de cartes indices en fonction de leurs points de clairvoyances obtenus. Plus un joueur a eu de points, plus il aura d'indices pour trouver la vérité.

Cette option, quoi que très intéressante et permettant de rajouter un peu de challenge, n'est pas indispensable à la partie et nous l'avons donc laissée pour la dernière itération, que nous n'avons pas eu le temps de réaliser. Il serait cependant très intéressant de la rajouter, cette implémentation ne serait pas très dure à réaliser mais prendrait un certain temps et nous demanderait de modifier beaucoup de code.

Une autre amélioration serait de permettre au MJ de choisir lui-même les associations des joueurs, et non pas de faire la répartition uniquement aléatoirement. En effet, nous supposons que les joueurs se connaîtront entre eux et seront un minimum à l'aise avec l'univers d'Harry Potter et voudront peut-être faire des associations cohérentes. Nous voudrions aussi ajouter la possibilité aux joueurs d'ajouter des cartes personnalisées, rajouter des personnages, des meurtriers ou d'autres aspects graphiques pour ainsi offrir de plus en plus d'associations

et de possibilités d'histoire aux joueurs (nous fournissons une histoire de base, mais les joueurs sont libres de la modifier ou d'en inventer comme ils le souhaitent). Ainsi le jeu serait beaucoup plus accessible et laisserait une grande liberté aux joueurs.

Bien entendu, nous pourrions aussi améliorer la partie graphique en créant nous-même des cartes par exemple, ou bien en faisant plus attention au design, en étudiant par exemple l'esthétisme des jeux de sociétés en général.

Thomas Turbide

ANNEXES

GLOSSAIRE

Android : Android est un système d'exploitation mobile développé par Google. Il utilise une version modifiée du kernel linux. Android est open source et est utilisé aussi bien sur des smartphones et des tablettes que sur des Smart TV ou des objets connectés.

API : Application Programming Interface, un ensemble de définition, protocoles de communication et outil pour créer des logiciels.

API REST : Representational State Transfer est un style d'architecture logicielle qui définit un ensemble de contraintes utilisées pour créer des services web.

AWT : Abstract Window Toolkit est une bibliothèque graphique pour Java, faisant partie de la bibliothèque standard de Java.

Bluetooth : Technologie de réseau sans fil local pour échange des données entre appareils sur une courte distance.

Backlog : Historique des propriétés ajoutées à un produit au cours du développement agile avec la méthode Scrum.

CSS : Cascading Style Sheets est un langage de feuilles de style utilisé pour décrire la présentation d'un document écrit dans un langage de balise comme par exemple HTML.

Classe java : En programmation orientée objet, une classe correspond à la spécification d'un ensemble de propriétés (les attributs) et de traitement (les méthodes) propres aux objets qui seront les instances de cette classe.

Client BTTP : Le client Brette Text Transfert Protocol est une spécification de client TCP/IP synchrone en Java définie par Jean François Brette.

FramaGit : Framagit est un hébergement gratuit et libre par l'association Framasoft du logiciel open source GitLab. GitLab est un environnement web de gestion de dépôt git proposant des fonctionnalités de développement agile, de DevOps et d'intégration continue.

Framework : Code abstrait qui permet à l'utilisateur d'utiliser des fonctionnalités générales

Git : Git est un système distribué de contrôle de version pour suivre les changements du code source durant le développement logiciel.**HTTP** : Hyper Text Transfer Protocol, est le protocole d'application d'échange réseau principale du World Wide Web. Il permet de faciliter l'échange de document HTML.

Git Fork : Git Fork est un client graphique git qui permet d'effectuer les tâches de commit, push, pull et merge, etc ... à l'aide d'une interface graphique.

IntelliJ : IntelliJ IDEA est un environnement de développement intégré Java. Il est développé par JetBrains sous une licence commerciale.

IP : Internet protocol est le protocole principal de la couche réseau. Il a pour charge d'encapsuler les données dans des paquets réseau et à les étiqueter en fonction de leur source et de leur destination.

Java : Langage de programmation orienté-objet et réflexif, compilé pour la machine virtuelle Java (JVM).

JavaFX : JavaFX est un framework graphique Java, qui permet de créer des interfaces graphiques client riche pour des applications de bureau.

JSON : JavaScript Object Notation est un standard open source de représentation des données de façon intelligible par l'homme. Il est utilisé notamment pour la communication client- serveur sur le web.

JUnit : Framework de tests unitaires pour le langage de programmation Java.

Kernel : Noyau de système d'exploitation, parties fondamentales des systèmes d'exploitation qui gère les ressources de l'ordinateur.

Paquetage Java : Un paquetage Java organise des classes Java en un espace de nom, fournissant un espace de nom unique à chaque classe du même package.

Patron de Conception : Solution générale et réutilisation à un problème récurrent dans un contexte donnée en conception logicielle.

QR Code : (Quick Response Code) code barre en deux dimensions, lisible par une machine, qui contient des informations sur l'objet sur lequel il est attaché.

Multi threadée : un application dite multi threadée est une application faisant appel à plusieurs threads, elle utilise donc le parallélisme.

Parrallélisme : Caractéristique d'un programme qui lui permet de d'exécuter des instructions sur plusieurs processeurs en même temps.

Refactoring : processus de restructuration de code existant, c'est à dire en changer la conception interne sans en changer le comportement externe.

Serveur : Programme informatique qui fournit des fonctionnalités pour d'autres programmes ou machines (les clients).

SliderAdapter : Composant d'application qui fait le lien entre une structure de donnée et un contrôle d'interface graphique.

Sockets : Tube de communication entre deux processus sur deux machines différentes à travers le réseau.

Swing : Swing est une bibliothèque graphique pour le langage de programmation Java, faisant partie de la bibliothèque standard de Java. Swing offre la possibilité de créer des interfaces graphiques identiques quel que soit le système d'exploitation sous-jacent.

SWT : Standard Widget Toolkit est une bibliothèque graphique libre pour Java, initiée par IBM.

TCP : Transmission Contrôle Protocole est un des principaux protocoles de la couche transport du modèle OSI. Il assure la livraison fiable et avec la gestion d'erreur d'un flux de données entre des applications.

Tests unitaires : Protocole de test logiciel qui expérimente le bon fonctionnement de plusieurs modules de code (fonctions) avec des données de contrôle.

Thread : Processus léger du système d'exploitation.

Websockets : Protocole de communication fournissant une communication full duplex sur une seule connexion TCP.

XML : eXtensible Markup Language est un langage de balisage qui définit un ensemble de règles d'encodage pour encoder des données dans un format lisible à la fois par les humains et les machines.

CHIFFRES AMUSANTS

Durant la réalisation de notre projet, nous avons réalisé un nombre varié et divers d'actions non liées à la programmation ou à la conception de l'application en elle-même. Nous avons répertorié ces actions et retenus certains chiffres que nous jugeons extrêmement importants à communiquer :

- Plus de 50 euros dépensés à la cafétaria de l'IUT avec tous les petits déjeunés, cafés, muffins et autres douceurs consommés sans lesquels nous n'aurions pas pu travailler.
- Près de 200 sushis avalés par l'équipe dans un restaurant asiatique. Comment un projet peut-il se dérouler sans ingurgitation de sushis ?
- Un nombre ahurissant de câbles et d'outils informatiques en tout genre : plus de 27 câbles comportant de l'HDMI, de l'Ethernet, de l'USB mais encore des switch internet, des hub USB ou de caméras.
- 138 chansons écoutées en groupe depuis Deezer ou Spotify pour booster notre productivité.
- 68 allers-retours entre la SIM, le secrétariat et les salles informatiques pour ouvrir et accéder à nos salles et travailler.
- 9569 lignes de code écrites pour avec le plus grand amusement, 344 commits sur GIT, qui nous ont permis d'avoir un projet complet d'environ 560Mo.
- Plus de 120h de travail en groupe, ça fait beaucoup de temps à se supporter les uns les autres ...



SOURCES

CARTES

Warner Bros, Saga « Harry Potter »

Warner Bros, Saga « Les Animaux Fantastiques »

Auteur exact inconnu, Pinterest, <https://www.pinterest.fr/pin/375839531388498919/>

« Ottowl », HP-Lexicon, https://www.hp-lexicon.org/?attachment_id=33422

Levi Pinfold, *couverture du livre « Harry Potter and the Chamber of Secrets, 20th anniversary edition »*, UK , https://harrypotter.fandom.com/wiki/Harry_Potter_and_the_Chamber_of_Secrets

VladislavPANtic, *DeviantArt*, <https://www.deviantart.com/vladislavpantic/art/Harry-Potter-and-the-Philosopher-s-Stone-FanArt-04-767713972>

Auteur exact inconnu, *Hitek*, https://hitek.fr/42/fond-decran-harry-potter-pc-smartphone_4636

Auteur exact inconnu, *Pinterest*, <http://www.Pinterest.com>

Auteur exact inconnu, *Pinterest*, <http://www.Pinterest.com>

Auteur exact inconnu, *RedShift3d*, https://www.redshift3d.com/?ACT=61&fid=18&aid=25401_2lCzn2OMNcJU7eG3zNI&board_id=1

P. Bessa, *Herbier général de l'amateur, vol. 8, 1817-1827*, <https://www.flickr.com/photos/swallowtailgardenseeds/15590183021/>

Giorgia, 2015, <https://www.webstagram.one/media/BhlnhcdnUdu>

Auteur exact inconnu, *Reddit*, https://www.reddit.com/r/wallpapers/comments/2x8noe/diagon_alley/

“Elle Artiste peintre”, *Elle Art*, <https://www.elle-art.com/>

Shergoku, *Wattpad*, <https://www.wattpad.com/story/88433454-1%27ann%C3%A9e-des-maraudeurs-1976>

Auteur exact inconnu, *Society6*, https://society6.com/product/the-opossum-and-peonies_duvet-cover?sku=s6-8659618p38a46v343

Blooming Concepts, <http://blooming-concepts.tumblr.com/post/72894347500/harry-potter-and-the-goblet-of-fire-concept-art-by>

The wizarding blog, <http://the-wizarding-blog.over-blog.com/2016/06/les-tableaux-de-poudlard.html>

Salavat Fidai, *Facebook*, <https://fr-fr.facebook.com/salavatfidaiart/photos/deathly-hallows-elder-wand-and-9%C2%BEhey-fans-what-signs-make-next-%EF%B8%8F/1357330731050789/>

« Le bestiaire d'Harry Potter », *Monsternovieitalia*, <https://monsternovieitalia.com/2017/09/28/il-bestiario-di-harry-potter/>

Lucy Callan, *Pinterest*, <https://www.pinterest.fr/lucycallan/harry-potter/>

Donald Alvarado, *Pulauubinstories*, <http://www.pulauubinstories.com/fond-d039ecran-harry-potter-pour-telephone/fond-damp039ecran-harry-potter-pour-telephone-beautiful-fonds-d-ecran-harry-potter-pour-telephone-mobile-kazageek-style/>

VAYALBA, *Gallery Wacom*, <http://gallery.wacom.com/gallery/74237493/Harry-Potter-Horcrux-Scenes-Characters>

Richard Wiswesser, *Eyes Love Tosee*, https://eyeslovetosee.de/harry-potter-studios/img_3966

Adam Brockbank, « Harry Potter and the Prisoner of Azkaban », *Adam Brockbank*, <http://adambrockbank.com/harry-potter-and-the-prisoner-of-azkaban#/id/i5306175>

Andreas Rocha, 2018, *The Art of Animation*, <http://theartofanimation.tumblr.com/post/27645293515/andreas-rocha>

Domitor Invictus, 2014, <https://sakrogoat.tumblr.com/post/85897409163>

darkdreams, 2011, <http://darkdreams.centerblog.net/rub-paysages-fantastique--4.html>

Auteur exact inconnu, <http://fillingthesoulwithbeauty.tumblr.com/post/33313298617>

Wizards, 2015, https://www.passionmagic.com/illustrations-exclusives-officielles-fournies-par-wizard/?epik=0SAIkE_IWH310

Waldemar Bartkowiak, « Master of the books », <http://emporioefikz.tumblr.com/post/111221767090/steampunktendencies-master-of-the-books-by>

Pierre Raveneau, « Lucius Deatheater », *Art Station*, <https://www.artstation.com/artwork/n8eV1>

Simon Bull, « The Journey Begins », <http://dadalux.tumblr.com/post/177533091402/simon-bull-the-journey-begins>

VladislavPANtic, « Guardian of The Forest », <http://harry-potter-aesthetic.tumblr.com/image/152298448677>

FionaHsieh, « Traveler », *DeviantArt*, <https://www.deviantart.com/fionahsieh/art/Traveler-479526297>

FionaHsieh, « Near the Spirits », *DeviantArt*, <https://www.deviantart.com/fionahsieh/art/Near-the-Spirits-517706250>

FionaHsieh, « Below the Canyon », *DeviantArt*, <https://www.deviantart.com/fionahsieh/art/Below-the-Canyon-508758468>

Dan Elijah Fajardo, *Pixalry*, <http://pixalry.io>Dan Elijah Fajardo, *Pixalry*, <http://pixalry.io>

APPLICATIONS

Dhut, 2017, Background animé du menu : https://www.desktophut.com/harry-potter-diagon-alley-animated-wallpaper/?fbclid=IwAR0JWhhB55zBWuZtE5osVsgXlKPJkdwHRPAzfAz0IB9Tzmcjx5_2FPSiTfU

Videvo, 2013, Background animé de l'écran de game over : <https://www.videvo.net/video/falling-stars-on-deep-blue-background/1632/>

Detracteur écran game over sur telephone

Phoenix écran victoire

GENERAL

Graphe sur les langages de programmation : <https://pypl.github.io/PYPL.html>

Graphe des OS mobiles : <http://gs.statcounter.com/os-market-share/mobile/worldwide>

Droit d'auteur : <http://desdroitsdesauteurs.fr/2012/01/la-protection-des-jeux-de-societe/>

EXEMPLES DE CARTES

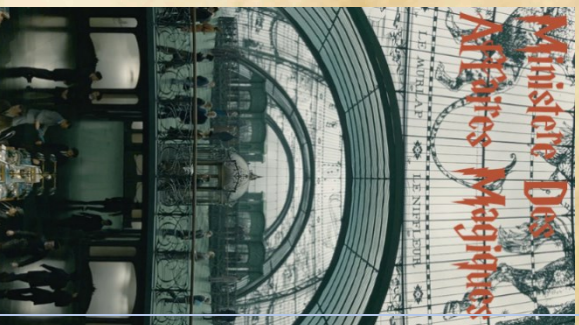
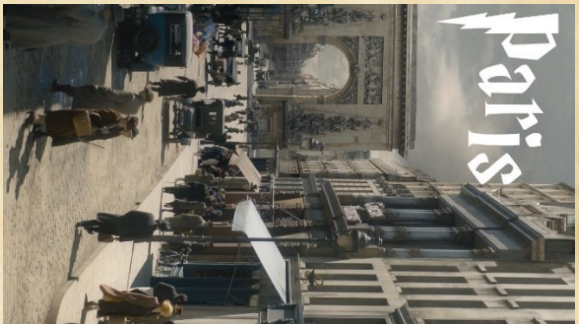
Exemples de cartes Personnage



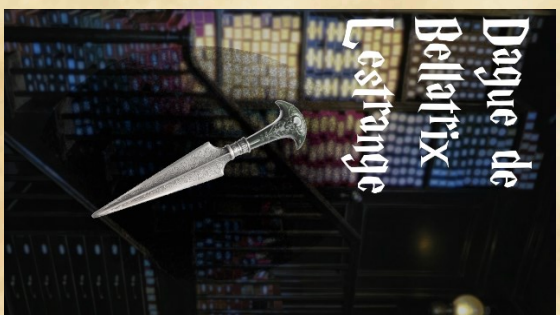
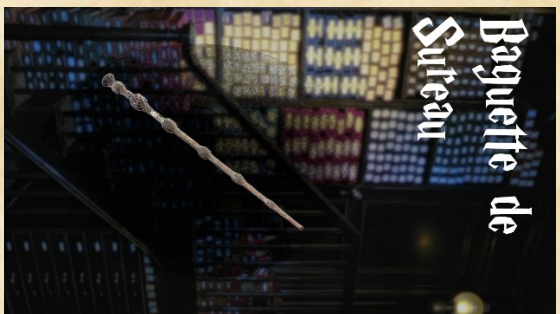
Exemples de cartes Meurtrier



Exemples de cartes Lieu



Exemples de cartes Arme





REVELIO

LE JEU DE SOCIÉTÉ NUMÉRIQUE



INTERAGISSEZ AVEC LE PLATEAU



Gellert Grindelwald

93 HOGWARTS EXPRESS

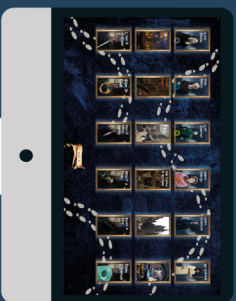


Le Ministère De La Magie

93 HOGWARTS EXPRESS



L'épée de Gryffondor

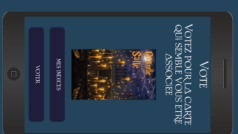
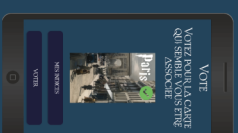
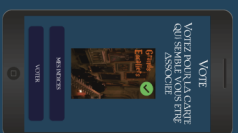
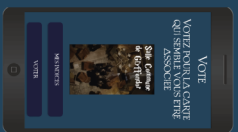
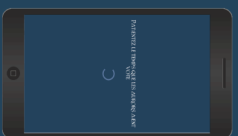


COOPÉREZ POUR RESOUDRE L'ENIGME

INTERPRÉTEZ LES INDICES

PLONGEZ DANS LE MONDE DES SORCIERS

INTERAGISSEZ AVEC LE PLATEAU



THOMAS TURBIDE

Our 4th semester project is a game based on the *Mystérium*, a board game, and the *Wizarding World* by J.K. Rowling. If I had to give a name to the type of game our "homemade" game *Revelio* is, I think I would choose "interactive board game". Interactive because of the digital part of *Revelio* and the remote interactions, board game for the several key elements our game share with classical board game.

This one of a kind project was really exciting because of the free subject choice. It was a good opportunity to do something we were really interested in. I came before the beginning of the project with this idea to focus on an application displayed on a wide screen and controlled by the players' phones, inspired by the game "Use your words". With this key concept in mind, we have been searching for ideas to fit with it. And it appears that board games were perfectly fitting! As we browsed for games, the *Mystérium* concept was the most interesting idea we found because of the different player roles.

During this project, I was able to work on the computer's interfaces with the use of JavaFX and its basic graphical controllers. It was the really first time I used this library, and so it was difficult to be efficient at the beginning of the development phase. Thanks to the advices and experiences of Arsène and some tutorials he shared with me, my progression hasn't been slowed too much. This helpful hand let me enjoy the fun and satisfaction to create interfaces. For a developer, seeing our work taking form by having a more instinctive HMI (human machine interface) than white characters on a black console is a real pleasure. This accomplishment feeling is really encouraging and brings us to be more efficient as we go deeper into the project. I was joined later by Alexis and Maud who gave me an appreciated and useful helping hand. This important task led me to work with the graphist team and the network communication team.

One of the biggest issues we had to deal with was to master JavaFX, and to be able to use it with its full potential. Searching for tutorials and explanations on this library is not easy as searching for help about the Java language. Moreover, the different point of view and interpretation our team could have on a feature slowed us in case of misunderstanding. This had led to interface's adjustments on almost every element we worked on, to make it match with the global vision of our team.

To conclude, this project is a part of the best project I ever had to realize. If I may say so, the stars were aligned to make this project shine. The possibility to choose freely the subject, our group, and so our co-workers, our technologies, our organization, our supervising professor, our tasks, and so on was the perfect opportunity to improve our technical skills, our teamwork skills and our project management skills within an exciting project. However, I'm disappointed we couldn't take this project as far as we wanted. The main steps were completed, and I'm proud of what we made, but the game could be better. If I could express one last regret, I haven't been able to follow entirely the work of the network and android team, on which I'm a bit curious.

ARSENE LAPOSTOLET

Our project is a distributed software that allows its users to digitally play a board game. This board game is a cooperative investigation game in which a witness has to make investigators guess the circumstances of a murder. The game's lore is the Wizarding Worlds, created by J.K. Rowling in her books *Harry Potter* and *The Fantastic Beasts*. The game's board is shown on a PC which is also the server. The client (used by the players to interact with the game) is an Android app which interacts with the server over TCP sockets.

The purpose of this project was on one hand to discover and learn new technologies and on the other hand to develop our skills on technologies we already know. We also wanted to create something innovative that we would be happy to use. Furthermore, as a group of board game fans we wanted to develop a software that would allow us to play more fluently, by getting rid of managing the cards or having to check on the appliance of the rules.

I far as I am concerned, I worked on the network communication API and on the server graphical user interface. It was a very dense, motivating and interesting commitment because my work was the backbone of the whole project. It made everyone interact with each other in the work. My code was indeed using the applicative logic while being used by the different graphical user interfaces. I learnt a lot because this work involved a deep knowledge of TCP sockets and threads management. For instance, in order to create the round loop, I designed a recursive thread to achieve better optimization and code readability. This experience developed my mastery of distributed and concurrent programming a lot. I also designed the underlying logic and algorithms that are used to create the server's views. I really enjoyed this part because on the one hand I created complex algorithms that improved my logic and abstraction ability and on the other hand it was very delighting to see the graphical result of my code.

We came through this project with little hassle. Our major difficulty was to put graphical user interface on the same ground as multi-threaded and networking logic. We overcame that obstacle by learning from programming books and online tutorials. Another challenge was to integrate the application logic to the networking API. This integration indeed showed bugs for which we had to provide hotfixes on the go. This experience enlightens the importance of exhaustive and well-designed unit tests. We also struggled a little with the integration of our Android networking API with the android graphical user interface because the debugger provided by Android Studio tricked us a little by crashing our app for no reason. Once we went aware of this, we just stopped using this debugger.

As a conclusion, we were able to bring forth a very usable and enjoyable piece of software which we can be proud of. We learnt a lot from this project, both from the technical and the relational point of view. We discovered and learnt to master a large set of modern and useful technologies but we also learnt to work together, overcome the disagreement and be a cohesive team. This project let us put our strength together to achieve something greater.

SIXTINE TRAVERSE

Revelio is an hybrid board game inspired by an existing board game called *Mystérium* which has received the Golden Ace of the year 2016. We have decided to associate our project with the *Harry Potter* theme, because many of ours love this fantasy world, and it is famous among the world. To play, you have to be five people. Your goal is to uncloak the true story of the Leta Lestrange murder: the murderer, the weapon used, and the place it happened, in fact, like a *Cluedo*, except that our game is cooperative.

This project has been exciting and informative for me, on many aspects. First of all, about my knowledges on computer science, specifically with the development of Android application. Indeed, it was my main mission in this project. I was really in a hurry to learn more about this technology before the project, and it was an opportunity to take to improve my skills in this domain. Thus, I developed the android side program, with the help of the graphic team and network-communication team. At the beginning, I was alone working on this part of the project. I had to find the solution of the problems I met, by myself. The main difficulty I had was to adapt the application view to different sizes of screen. I found out to do that on many websites like StackOverFlow and forums. I had on many views a "scroll view", so that all users can scroll if some elements exceed the screen. I also had some difficulties with the creation of list views, more precisely in the selection of an element of the list. I managed to solve many issues in the best way I could, by doing many tests. Moreover, I had to make many researches to know how to implement the specificities of the application, like the sliders and their adapter.

When my part was well advanced, we were able to share with the network team to create communication between the server and the phones. After this, the whole suite of the android project went very smoothly, despite some minor problems with too large image sizes.

Finally, the project allowed me to develop my skills in Android, but also develop my autonomy and my ability to solve problems, and my team spirit. Indeed, this was the first time I bring forth such a big project with six teammates. It was really instructive and pleasant to work with a team as passionate as ours. We worked with regularity, seriousness and fun. We were all motivated and we met each day from 9 am to 17 pm. However, I was used to stay until 19 pm in order to advance as best as I can and as quickly as possible, and because I was really appreciating what I was doing. This help me in being more conscientious and assiduous. To finish, this project brought me new interests and confirmed me even more my desire to continue in the field of computing, especially in programming within a team. I hope I can relive such an experience again and again.

MAUD GELLEE

We came up with the idea of Revelio when we discovered some new videogames based on board games like the Monopoly or the trivial pursuit. We wanted the computer to be the board and use smartphones to interact with it. In order to make the application entertaining we had to choose the right game. After a brainstorming we picked out the Mysterium, an investigation and cooperative game comparable to the Cluedo. The rules were quite simple to understand, and we imagined well how we could digitize it.

It was very stimulating, not only for the technological discovery but also for the teamwork. Indeed, we never had the opportunity to work on a project as complex and as exhaustive as this one, and on top of that, we needed to learn how to work and communicate with each other.

When we distributed the tasks, I choose to work on the communication between the client (the phones) and the server (the computer) and on the graphics. Nevertheless, I worked a bit on every field of the project, which was very interesting, since I could touch and learn many different things. While I was working on the network communication, I could review some of the things I had learn during the year. We built a server accessible by clients thanks to its IP address, which could receive, process and send data. Afterwards, I've worked on the board and android interface, for we had to think again the ergonomic principles of the game for it to be adapted to a digital environment.

At last, I have worked on the graphics. We unanimously decided to include our game in the Harry Potter universe. Thus, we had to look for image in order to design new cards for the game, corresponding to the world of J.K. Rowling. It was quite time-consuming as we wanted high-quality picture, and we tried our best to find artwork that had the same design. Furthermore, we searched various icons and symbol in connection with the wizarding world, additionally I created the game logo. I love art and design, therefore handling the graphics was enjoyable, however I would have liked to spend a bit less time on the interface and more on the code itself, especially JavaFX, that I tried a bit by doing the game over and the win screen, but I had not the time to fully use it.

I really loved working on this project, I had the chance to develop my technological skills while we were creating a beautiful application. Moreover, the atmosphere of the team was great, obviously we had some tension from time to time, but we always managed to work things out and make compromised for the sake of the project. I really appreciated what we achieved together, thanks to this project, I have improved my hard and soft skills, and I am very grateful to have had the opportunity to work on this project with my teammates, since I will have the use of this knowledge in my next experiences.

ALEXIS GOMES MATIAS

Our project, Revelio, is a video game which offers to its users to play on a digital board game inspired by Mysterium. We applied the Wizarding World's historical background created by J.K Rowling in Harry Potter. In this game, the players must co-operate to investigate about a murder and they must guess the murderer, the weapon and the place of the crime. They must use a computer for the displaying of the board and the different game options (like cutting or modifying the music volume). All of this actions may be do on the computer or on the player's smartphone.

There were a lot of different purposes to this project that allowed us to learn and discover a bunch of new technologies and to increase our skills in a few programming domains. Furthermore, we wanted to create a game in which we can quickly and easily play, without the rid of managing the large number of little pieces usually present in a classic board game.

My tasks were essentially about the server graphical user interface (server GUI) and applicative logic of our game. First, I worked on the applicative logic which is the main Java code used by every other part of the game and consists of atomical operations represented by Java functions, used to store data and play to the game at the smaller level of the application. With this development, I acquired a lot of knowledges in object-oriented programming, in logical modelling and in breaking "dangerous" dependencies. Moreover, the server graphical interface taught me a lot more competences because I never used JavaFX before this project and I had to learn and understand the fundamentals of this technology quickly so it could be useful on the project. Therefore, a lot of difficulties came forward, especially on understanding issues on basics of panel's behavior and FX-CSS errors. However, with Oracle's JFX books and the official documentation, I overcame the issues which occurred and learnt about my errors. With these books, I also found how to manage the video background and the sound which is played through the game. I added a bunch of music randomly played for a better immersion of the player in our application.

In regard with my non-IT skills, I also acquired a lot of competences in project management and project working style because it is one of the most complete project we ever did in our studies and we had to think about the end of the project even if we were only at the first week of working. I also understood how to manage team disagreements and how to bypass them to continue to work and end the project.

Finally, this project allowed me to acquire a lot of new knowledges and competences that I will surely use in my future projects and professional life. I think I can consider it as one of my best experiences since I started programming as there was a good atmosphere between me and my teammates.

GABRIEL GUISSET

Our project, called Revelio, is a board game recreated and adapted into a software. 5 friends can play from their Android phone by cooperating to solve a crime in Harry Potter's World. One of them will assume the role of game master and will help the other players to find 4 cards associations and after that, discover the real murder by sending vision cards.

This project was very ambitious at first, we proceeded to an analysis to check all languages, tools and components we'll need and organize ourselves in teams to work on different parts of the programming. We worked by iteration to regularly have a functional version of our game. This method is called working in agile.

On my side, I worked on several parts of the project. In a first part, I programmed, with Alexis Gomes Matias, the applicative logic which is the basis of the project by organizing data and facilitate every treatment for phones and the server. One of the difficulties we had in this work was the respect of SOLID's principles to make sure the code is customizable and keep it accessible to modify. In a second time, I worked on the different graphical assets of the game (different cards, logo etc) with Maud GELLÉE. To continue, I worked on small parts and bits of server interface and Android interface. Last, I worked on different pictures that were necessary in many domains: poster, oral presentation and this sum up text.

I really liked this project and the overall ambiance in this working group. We were working tight together and good intended with one another.

While working on the project, I realized that I was spending too much time on the graphical assets. To bear this problem, I proposed to the group that every pole should present what has been made so that everyone can understand every part of the project, code or graphics. This suggestion was a good idea because after we got to present our parts to the other members of the project team, everybody was able to work on the different poles and therefor discover each technology used. I also would've preferred to work more on the JavaFX, the server interface because it seemed like an interesting part while still being complex and a primary element for the application to work. For the applicative logic, I was proud of my work because I could use, in a more personal domain, the knowledge I acquired during my studies.

To prepare the presentation, I also could use one of my passion which is photography. Filming the totality of a game to have a good demonstration video about how the game plays through, was also a nice way to apply this passion of mine while interlacing it with this study related project.

To me, this project is helpful on improving team cohesion and team work. I really liked to work on all the different poles of the project because on first hand, it gave me a first contact with components we never thought we would use.