

# Université Lumière Lyon 2



## RAPPORT DE PROJET

### M1 INFORMATIQUE – PROGRAMMATION MOBILE

---

Projet : application “Living at Time”

---



Flavien Gonin

N°étudiant : 2235811

Guillaume Hostache

N°étudiant : 2236156

# I. Présentation générale

Notre application est un jeu narratif, du même genre que le jeu “Lapse : A Forgotten Future”. Nous avons repris le principe du jeu, avec notre histoire (une enquête au moyen-âge.) et quelques changements comme la galerie, les statistiques...

Voici donc une présentation de notre application mobile “Living at Time”, présentation qui pourrait être présente sur un éventuel Apple Store :

Mener l'enquête dans un royaume où le temps semble plongé dans le chaos total ! Vous êtes plongé au cœur du moyen-âge, afin de mener l'enquête sur de mystérieuses défaillances temporelles qui semblent provoquer des catastrophes dans le futur. Mais pour cela, vous allez devoir diriger et décidez de la destinée de tout un royaume, à l'aide d'un simple mouvement de doigt, vers la gauche ou la droite. Vous devrez prendre de lourdes décisions en ces temps, en proie à la maladie, à la révolte et la guerre. Arriverez-vous à éviter les révoltes, à repousser les invasions vikings ? Arriverez-vous à trouver les réponses à vos questions ? Découvrirez-vous la source de ce chaos temporel ?

Soyez vigilant, pesez bien le pour et le contre de chaque décision, car rien n'arrive par hasard. Chaque action pourra vous aider à atteindre la victoire finale, ou bien la défaite ! Qui sera votre ami, qui sera votre ennemi ? Ce sera à vous d'y répondre...

Ce qui est fait est fait, et vous ne pouvez que jouer vos cartes du mieux que vous le pouvez dans cette course inévitable contre le temps !

## II. Présentation du code

### A. Structure des données

Le jeu fonctionne avec un enchaînement d'événements, qui possède certaines caractéristiques communes et propres parfois. Une “carte” événement est donc composé : d'un personnage (Paysan, Chevalier, Reine, Adepté...), d'un dialogue, de deux choix de réponses associés au dialogue, d'un offset associé à la réponse (dans certains cas, pour que l'histoire principale puisse évoluer et que les choix influencent vraiment la suite des dialogues). Il y a aussi des conditions d'apparitions, liées au temps dans la majorité des cas, mais ce problème est traité avec des conditions dans le code. (Les événements sont juste séparés dans des fichiers différents). Dans tous les fichiers, on sépare les informations avec des ‘ ; ’ ou des ‘ \n ’

Voici un exemple avec le fichier GameEvent.txt, où les événements sont simples et où il n'y a pas d'offset.

Dans ce fichier (comme dans tous les fichiers d'ailleurs), la première ligne représente le nombre de personnages différents contenu dans le fichier. Ensuite, les

événements pour chacun des personnages sont écrits. D’abord, il y a le nombre d’événement associé au nom du personnage.

```
GameEvent.txt  NinjaDeath.txt
1 11
2 2;paysan
3 L'un de vos chevaliers à saccager notre moulin, je réclame une justice !
4 Je vais le sanctionner.;0;1;-1;0;10
5 Hors de ma vue, paysan !;0;-1;0;0;-10
6 L'un de vos conseillers est un voleur !
7 Comment osez-vous ! Calomnie !;0;-1;0;0;-10
8 Je vais enquêter...;0;1;0;0;10
9 Mon roi, je souhaiterais devenir chevalier ! M'accordez-vous votre autorisation ?
10 Oui;0;1;1;0;0
11 Non;0;-1;0;0;0
12 4;paysanne
13 Je cherche le chevalier Sir Andromède, il m'a offert cette rose... Et j'aimerais le revoir mon roi !
14 Je lui transmettrais...;0;-1;1;0;-10
15 Il doit être en ville...;0;1;1;0;10
16 Mes brebis se sont égarées dans le bois de Clementi, il me faut de l'aide !
17 Sir Pinpin, vous aiderez.;0;1;-1;0;10
18 Voici un dédomagement.;0;1;0;1;10
19 La tempête de cette nuit à détruit le toit de ma grange, mes bêtes vont mourir de froid !
20 Ce n'est pas grave...;0;-1;0;0;-10
21 Voici pour les réparations.;0;1;0;-1;10
22 Les routes sont dangereuses, mon roi, où sont les chevaliers qui doivent nous protéger !
23 Je m'en occupe !;0;1;-1;1;10
24 Ne vous en faites pas...;0;1;-1;-1;-10
```

Après, il y a tous les événements écrits avec les réponses en dessous de chaque événement. À côté de toutes les réponses, il y a les différents impacts que la réponse pourra avoir sur les statistiques. L’ordre est le suivant : [Religion, Population, Militaire, Richesse, Popularité/Élection]. Ces chiffres sont séparés par des ‘;’.

Sinon, voici un autre exemple avec le fichier NinjaDeath.txt, qui fait appel aux offsets, permettant de faire des sauts de lecture dans le fichier.

```
GameEvent.txt  NinjaDeath.txt
1 4
2 9;ninja
3 Mon seigneur, mon empereur me rappelle pour une affaire urgente, je venais vous remercier pour votre accueil...
4 Le plaisir est pour moi !;0
5 Attendez...;1
6 En remerciement, acceptez ce présent, mon roi !
7 Merci !;10
8 Oui !;10
9 Oui, une dernière chose majesté ?
10 Vous me rappelez quelqu'un...;0
11 Vous êtes une anomalie ?;2
12 C'est fort probable, je suis avant tout une anomalie égarée dans une drôle d'époque... Je ne suis que l'imaginaire de quelqu'un d'autre...
13 Hein !;0
14 Alors, il faut...;0
15 Oui, vous devez être la personne qui à chassez les autres anomalies... Faites ce que vous avez à faire.
16 Bien !;0
17 Je suis désolé...;0
18 Effectivement. Même si je n'aime guère cette appellation, je me qualifierais de maître intemporel...
19 C'est vous l'origine ?;0
20 Euh...;2
21 Non, je ne suis que l'imaginaire d'un autre...
22 Dans ce cas, je vais...;0
23 Bien;0
24 Faites donc, majesté...
25 Au revoir;1
26 ...;1
27 Je me considère comme la plus maline des anomalies, bien que je ne sois pas la plus grande des menaces pour votre royaume...
28 Mais faites ce que vous devez faire. Ce fut un honneur pour moi !
29 Tout l'honneur est pour moi !;0
30 Oui;0
31 1;inconnu
32 ;0
33 ;0
34 1;ninja
```

Le début du fichier ne change pas, sauf que pour tous les événements, il n’y a pas d’impact sur les statistiques, mais un offset, à la fin de chaque réponse (Ici, le but est d’avoir un enchaînement d’événement lié à l’histoire principale, il est compliqué de mettre un impact sur les différentes statistiques. Cette offset indique si un saut d’événement est nécessaire ou non. Si l’offset n est à 0, pas de saut, lecture “classique”, sinon, on fait un saut de n événement (même s’il y a un changement de personnage)

## B. Quelques explications du code

Tout d’abord, le code est très lourd (plus de 1000 lignes) mais cela est principalement dû aux nombreuses conditions pour gérer l’ordre des événements.

## 1. Lecture des événements

La première fonction vraiment importante est la fonction `lectureEvent`. Celle-ci prend en paramètres divers arguments tels que le nom du fichier, son extension et deux booléens, un pour indiquer si le fichier possède un offset et l'autre pour indiquer si on doit lire des valeurs.

Comme expliqué précédemment, les fichiers possèdent une forme qui est toujours, ou presque, la même. Cela permet donc de pouvoir lire n'importe quel fichier en ayant seulement quelques informations qui sont représentées par les deux booléens.

Lors de la lecture d'un fichier, on commence par récupérer tout le texte puis, il est séparé entre chaque ligne. On récupère sur la première ligne le nombre de personnages qu'il y aura dans le fichier. Ensuite, pour chaque personnage, on a sur la première ligne, le nombre d'événements liés à ce personnage, mais aussi son nom séparé par un “;”. Chaque événement est représenté sur trois lignes. La première est la question posée lors de l'événement, la deuxième est la première réponse possible avec en plus des valeurs et des offsets dans certains cas et la troisième est une deuxième réponse similaire à la première.

Après lecture des trois lignes cités plus haut, on crée un nouveau événement comprenant toutes ces informations. On l'ajoute ensuite à un tableau temporaire qui sera renvoyé à la fin de la fonction.

## 2. Sauvegarde et chargement

La deuxième fonctionnalité importante de ce jeu est la sauvegarde. Dans les fichiers du jeu, il existe un fichier nommé “`save.txt`”. Ce fichier est, de base, vide. Lors de la chaque partie, on charge le jeu afin de savoir si le jeu a déjà commencé ou si on est sur la première partie.

Lors du chargement, on vérifie la taille du fichier de sauvegarde en nombre de ligne. Si le fichier contient moins de trois lignes, cela signifie que le fichier est soit vide soit qu'il n'y a pas assez d'informations. Si le fichier est dans cette situation, on le remplit en sauvegardant toutes les informations importantes et on affirme que le jeu ne doit pas être chargé. Dans le cas où le fichier serait rempli, on récupère toutes les informations permettant de savoir où le joueur se situe dans la partie.

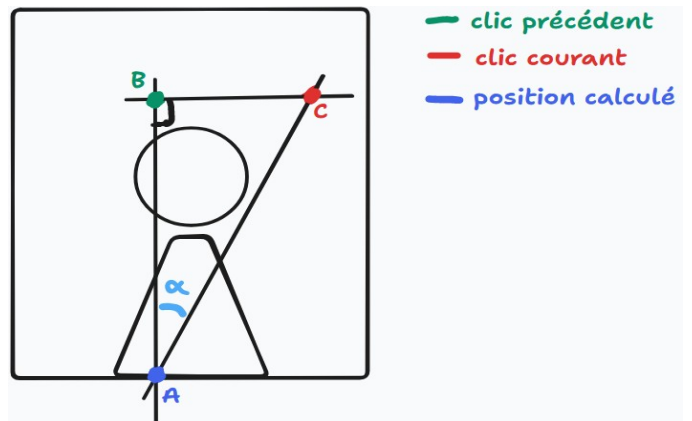
Comme pendant les chargements, lors d'une sauvegarde, on récupère le fichier et ensuite on crée une chaîne de caractères contenant sur chaque ligne les informations que l'on cherche à garder pour une prochaine partie. Dans ces données, il y a, par exemple, le dictionnaire des succès, le dictionnaire des personnages, différentes valeurs (argent, foi, armée, population, popularité), l'année du jeu et différents booléens permettant de gérer les événements du jeu.

### 3. Rotation et mouvement de la carte

Lors d'une partie type, le joueur interagit avec les images à l'écran en les déplaçant vers la droite ou vers la gauche. Pour permettre ces déplacements, on utilise les fonctions de base `touchesBegan`, `touchesMoved` et `touchesEnded`. Lors du début du « clic », on vérifie que la position est dans le carré de l'image. S'il est dans ce carré, on en profite pour sauvegarder les coordonnées du dernier point et autoriser le déplacement.

Si le « clic » se déplace, on calcule la distance entre les deux points avant de l'appliquer à la carte et aussi aux labels de réponses. Ensuite, on commence les calculs pour la rotation de la carte. Pour cela, on utilise la trigonométrie.

Comme montré sur le schéma ci-contre, on crée trois points dans l'espace comme suit : Point vert (précédent.x, courant.y), Point rouge (courant) et Point bleu (précédent.x, bas de la carte.y). Avoir ces trois points permet de créer un triangle rectangle dans l'espace afin de calculer l'angle  $\alpha$ . Pour avoir cet angle, on calcule la distance entre les points A et C et aussi entre les points B et C. Avec ces distances, on peut calculer  $\alpha$  avec la formule suivante :  $\alpha = \arcsin\left(\frac{\text{distBC}}{\text{distAC}}\right)$



Grâce à cette formule, on peut appliquer le résultat à la carte pour la rotation, mais aussi connaître l'angle afin de changer d'événement si l'angle est supérieur à une certaine valeur.

### 4. Changement d'événement

Lorsque l'événement se finit par une réponse ou une autre, on change d'événements selon certaines conditions. La plupart du temps, le jeu choisira un événement aléatoire appartenant à une liste d'événements. Cependant, comme notre jeu possède un scénario, à intervalle régulier, le jeu va vérifier l'année afin de jouer d'autres événements comme l'arrivée de Robin des bois ou alors la mort d'un autre personnage. Lors de ces événements, ceux-ci peuvent posséder des offsets. Ces offsets permettent au joueur de répondre différemment et d'avoir quelques changements de dialogue.

En plus, lors du changement d'événement, on applique les potentiels changements de valeur indiqués par l'événement. Ceux-ci doivent être compris entre 0 et 8 ce qui implique beaucoup de vérification pour chaque valeur afin de ne pas casser le jeu.

### III. Difficultés rencontrées

Durant ce projet, nous avons rencontré de nombreuses difficultés. De nombreux problèmes de compatibilité entre l'environnement MacOS et celui de Windows. Au départ, nous avons écrits nos fichiers texte sous Windows, mais le formatage n'étant pas le même entre les deux environnements, nous avons dû traiter beaucoup d'erreurs de formatage qui posait problème lors de la lecture des données.

Ensuite, la difficulté fut de gérer le fil du temps. En effet certains événements ont des conditions d'apparitions qu'il a fallu définir et mettre en place. Nous avons aussi dû réaliser entièrement le scénario et gérer l'équilibrage pour que le jeu reste jouable, cela n'était pas dur mais représentait beaucoup de travail.

Globalement, les difficultés rencontrés sont principalement dû à un léger manque de temps et à une inexpérience sur l'utilisation de MacOS.

### IV. Pistes d'évolutions du projet

Tout d'abord, il faudrait peut-être faire de l'optimisation, pour simplifier le code qui est assez indigeste (nous avons manqué de temps pour cela.). Cela permettrait de pouvoir plus facilement le faire évoluer. Après cela, on pourrait ajouter de nouveaux événements, améliorer le scénario, ainsi que les assets qui ont été réalisé à la main (nos talents de graphisme sont très limités). L'idée serait aussi d'ajouter de nouveaux événements liés à la trame principale, de sorte à avoir une histoire encore plus incroyable.

Ensuite, on pourrait ajouter de nouvelles fonctionnalités sur le jeu, comme le choix entre plusieurs musiques différentes, un menu de sauvegarde pour avoir plusieurs joueurs ou l'ajout de "Joker" que l'on obtiendrait en prenant certaines décisions dans l'aventure, cela permettrait que lorsque le joueur atteigne le game over dans une statistique, le "Joker" se déclenche et évite au joueur la mort.

Un débat que l'on a eu était aussi sur la difficulté et l'ajout des triangles vert ou rouge, qui indique au joueur l'impact qu'aura sa décision sur les différentes statistiques (à l'exception de la popularité / chance de réélection). Dans le jeu de base, il y a seulement une icône blanche, ce qui indique juste au joueur que la statistique va être impacté. Nous avons longuement hésité, mais on a décidé de garder cette aide.