



RAPPORT DE PJS4 – Groupe 19

Présenté par :

Paul Courbois  
Benjamin Di Santo  
Julien Leopardo  
Jules Plateau  
Velan Senguttuvan  
Simon Weber



UNIVERSITÉ  
**PARIS**  
**DESCARTES**

Encadré par :  
Anton Roy



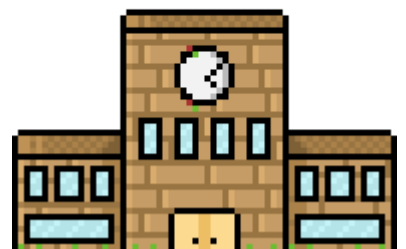
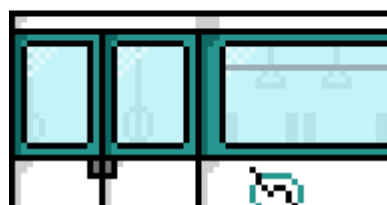
# REMERCIEMENTS

Nous tenons à remercier toutes les personnes qui ont contribué au bon développement de notre projet et qui nous ont aidés lors de la rédaction de ce rapport.

Tout d'abord, nous adressons nos remerciements à notre professeur encadrant, Monsieur Anton Roy, Développeur de Jeux Vidéo chez Amplitude Studios, pour son aide, le temps passé avec nous et le partage de ses connaissances. Grâce à son expertise et ses expériences, nous avons réussi à progresser rapidement et efficacement dans le développement de notre projet, tout en apprenant de nouvelles choses dans un domaine en constante évolution tel que le monde du jeu vidéo. Il fut d'une aide précieuse durant toute la durée de ce projet.

Nous tenons aussi à vivement remercier tous nos professeurs d'informatique au sein de l'IUT qui nous ont encadrés au cours de ces deux dernières années : Monsieur Jean-François Brette pour son expertise en techniques Java avancées, Messieurs Mikal Ziane et Mourad Ouziri pour leurs connaissances en maintenabilité et qualité de code, Monsieur Denis Poitrenaud pour nous avoir introduit au langage Java, Monsieur Julien Rossit pour sa culture en jeux vidéo et ses conseils sur l'aspect multijoueur de notre projet.

Enfin, nous tenons à remercier toutes les personnes qui nous ont conseillées et relues lors de la rédaction de ce rapport : nos parents, et notre professeur d'expression, Madame Ariane Mayer qui nous a fourni de précieux conseils pour la rédaction de ce rapport.





# SOMMAIRE

Auteur : Velan Senguttuvan

<b>INTRODUCTION</b>	<b>3</b>
<b>DEVELOPPEMENT</b>	<b>4</b>
- PHASE D'ANALYSE	5
- CAHIER DES CHARGES	7
- OUTILS & LANGAGES	8
- REPARTITION DES TACHES	9
- CONCEPTION DE L'APPLICATION	10
- DEVELOPPEMENT ALGORITHMIQUE	12
- DEVELOPPEMENT GRAPHIQUE & SONORE	14
- RESULTATS	17
<b>CONCLUSION</b>	<b>22</b>
<b>ANNEXES</b>	<b>23</b>
- DIAGRAMME DE GANTT	24
- MANUEL D'UTILISATION	25
- CHARTE EDITORIALE	29
- CHARTE GRAPHIQUE	33
- LEXIQUE	36
- BIBLIOGRAPHIE	41
<b>RESUMES INDIVIDUELS</b>	<b>43</b>
<b>POSTER</b>	<b>53</b>



# INTRODUCTION

Auteurs : Benjamin Di Santo & Simon Weber

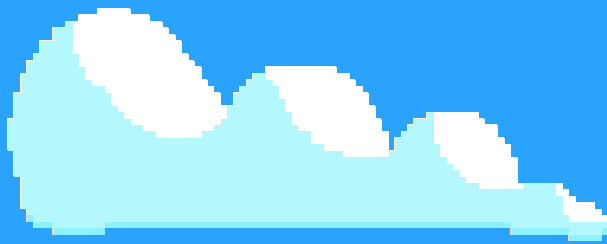
L'objectif de ce projet tuteuré de fin d'année est la réalisation d'une application permettant à chacun d'approfondir ses compétences en informatique, expression, communication et gestion de projet de manière autonome. Ce projet est à réaliser en groupe de 5 à 6 étudiants qui doivent choisir, de manière complètement libre, un sujet pertinent, innovant et réalisable.

Lorsque notre groupe s'est formé, nous avons eu plusieurs idées pour notre projet. Après un brainstorming sur les sujets possibles, il fallait choisir entre deux options : soit une application de réseau social pour gamers, soit une application similaire au jeu vidéo très populaire « Reigns ».

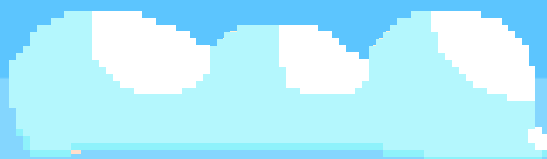
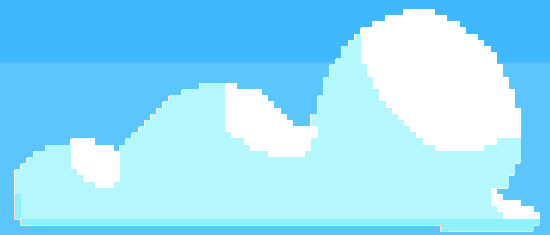
Ces deux propositions se sont formées de manière assez naturelle, car, en effet, certains membres du groupe avaient un penchant pour le développement de jeux vidéo, secteur dans lequel ils souhaitent travailler plus tard. De plus, la moitié du groupe avait déjà joué à « Reigns » et trouvait le concept du jeu innovant et surtout réalisable. En effet, le fait de faire des décisions stratégiques à partir de questions à réponses binaires était un aspect assez attrayant et intrigant du jeu.

À la suite d'un vote assez catégorique, nous avons opté pour faire un remake de « Reigns » et avons commencé nos recherches. C'est ainsi que nous avons décidé que notre jeu sera basé sur le gameplay de ce jeu adapté à des scénarios qui se présentent dans la vie d'un étudiant qui tente d'obtenir son DUT Informatique. Afin de réaliser ce projet efficacement, nous avons décidé d'implémenter la méthode SCRUM et avons mis en place un Carnet de Bord afin de répertorier le travail fait lors des séances.

Dans un premier temps, nous décrirons les phases de recherches et de conception pour notre application. Puis nous étudierons les phases de développement à la fois algorithmique et graphique avant de dresser un bilan de ce projet.



## DEVELOPPEMENT





## PHASE D'ANALYSE

Auteur : Simon Weber

Très rapidement, nous avons choisi de partir sur l'idée du jeu vidéo de fait à cause de plusieurs aspects :

- Le sujet s'avérait avant tout être original, en effet le sujet de réseau social avait déjà été abordé par d'anciennes promos et était déjà le sujet de certains de nos camarades, il était donc question de nous démarquer.
- Le jeu vidéo en lui-même se voulait réalisable : l'interface et le visuel n'était pas trop complexe et le fonctionnement du jeu en lui-même était simple de sorte à ce que nous puissions le réaliser entièrement de sorte à ce que le rendu final soit similaire à ce que nous avons imaginé au début.
- Paul et Jules voulant travailler plus tard dans l'industrie vidéoludique, ce sujet s'avérait être une bonne expérience pour eux.

Connaissant les fonctionnalités de base du jeu « Reigns », il nous fallait trouver d'autres fonctionnalités pour nous démarquer et ainsi rendre notre jeu original. Nous avons donc longuement pensé à ce qu'un étudiant en IUT vivait tous les jours pour penser différents objets, événements et cartes. L'UXDesign et la Direction Artistique étaient deux choses de très importantes. En effet, du fait que notre jeu s'avérait simple sur son fonctionnement algorithmique, il fallait que notre jeu soit ergonomique et ai sa propre identité visuelle.

Toutes les différentes fonctionnalités et idées que nous souhaitons implémenter ont toutes été notées dans un document intitulé Boîte à Idées que nous consultons à chaque séance pour voir l'avancement de l'implémentation.

Pour ce qui est de la concurrence sur le marché en générale, le secteur du jeu vidéo s'avère saturé. En effet, de nombreux jeux sortent chaque année et si ces derniers ne viennent pas de grands studios de développement, les jeux dit 'indépendants' ont du mal à sortir du lot. Les seuls titres à petits budgets qui se démarquent ont soit un concept et un design inédit, et ou l'appui d'un éditeur qui peut aider le jeu en budget et en influence. Mais étant donné nos maigres moyens nous cherchions juste à faire un jeu intéressant et facile à implémenter et qui soit agréable à jouer. Le but de ce projet était donc de nous faire plaisir dans le design et la programmation.



Au-delà de cet aspect, nous voulions que notre jeu plaise. Donc comme nous ne n'avons pas un gros budget pouvant séduire un large public, nous limitons notre public à un groupe très précis en étant sûre de lui faire aimer le jeu. Nous avons donc décidé de prendre un sujet que nous connaissons et qui parle aux gens : les études supérieures. Notre cible est donc les personnes ayant fini leurs études assez récemment qui peuvent revivre certains aspects de ces dernières à travers notre jeu et ressentir une certaine nostalgie. Nous avons alors cherché à lier la forme à ce sentiment, étant donné que notre public cible a entre 20 et 35 ans nous avons conclu qu'ils avaient connus dans leurs premières expériences de jeu sur 'GameBoy', 'GameBoy Advance' et autres. Des consoles ayant des jeux en pixel art, ce qui motive donc notre choix.

En plus de cela, notre jeu au principe simple possède une durée assez courte. En effet, de nos jours, nous disposons de moins en moins de temps pour nous, seuls les « vrais joueurs » investissent leur temps dans des jeux vidéo aux durées de vie longues. Ainsi, la durée de vie de notre jeu nous permet de toucher un plus large public que celui ciblé d'un minimum d'âge de 12 ans du fait de la mention de cigarettes dans notre jeu, le catégorisant ainsi PEGI 12.

Le sujet choisi, il nous fallait maintenant discuter du langage que nous allions utiliser pour développer : Java et C# étaient deux langages sur lesquelles nous étions hésitants, mais bien que C# couplé avec le moteur de jeu Unity s'avérait plus optimisé pour réaliser un jeu vidéo, Java fut le langage que nous avons choisi. En effet, Unity nécessitait un temps d'adaptation du fait qu'aucun de nous ne le maîtrisait et il s'avérait aussi que Julien et Jules possèdent tous les deux de très bonnes bases en Java, il aurait été dommage de ne pas les exploiter. Facebook Messenger et Discord ont été choisis pour discuter du projet entre nous et avec notre professeur référent, Google Drive pour stocker et partager entre nous les différents documents et Framagit pour héberger notre projet.



# CAHIER DES CHARGES

Auteur : Benjamin Di Santo

## A – FICHE PRODUIT

- Genre : Jeu narratif à choix binaire
- Style : Stratégie / Réflexion / Rétro
- Plateforme : PC Windows / Mac
- Bibliothèque Graphique : JavaFx
- Nombre de Joueurs : 1 ou 2
- PEGI : 12+

## B – PITCH

« College War » est un jeu de cartes narratif qui met l'utilisateur dans la peau d'un étudiant Universitaire. Au cours d'une partie, une série de dilemmes s'imposent, et les décisions du joueur ont des conséquences et certaines sont parfois inattendues. Elles déterminent sa réussite scolaire ou ... son entrée chez Pole Emploi.

## C – PRINCIPE DE JEU

Le principe de jeu est assez direct. L'aspect le plus intrigant : chaque choix n'a que deux solutions. Le jeu se base sur ces cinq éléments fondateurs : « Savoir », « Popularité », « Santé », « Argent » et « Moral ». Afin de continuer à jouer et obtenir son diplôme, le joueur doit veiller à ne pas laisser la jauge d'un seul de ces éléments se remplir complètement, ni tomber à zéro et terminer l'histoire prévue.

## D – INSPIRATION

Pour la réalisation de notre application, nous nous sommes inspirés assez lourdement une du jeu mobile populaire « Reigns » ainsi que ses produits associés : « Reigns: Game of Thrones » & « Reigns: Her Majesty »

## E – POINTS-CLES

- Jouer à plusieurs dans un salon local
- Récupérer une partie en cours (Sauvegarde)
- Effet nostalgique des années d'étudiant
- Graphismes « Homemade »





# OUTILS & LANGAGES

Auteur : Velan Senguttuvan

Pour la réalisation de notre application, nous avons principalement utilisé deux outils de développement : Java comme langage de programmation et la bibliothèque JavaFx pour l'implémentation graphique. Ce choix nous permet notamment de travailler uniquement sur le logiciel Eclipse. Cette décision fut assez rapide et sans issues, car c'est un langage maîtrisé par tous les membres du groupe grâce aux différents cours que nous avons pu avoir lors de ces deux dernières années de DUT. Nous avons choisi la bibliothèque JavaFx, car elle nous a permis de créer une interface graphique relativement pratique et simple d'utilisation. Cela nous a aussi donné l'occasion d'élargir nos compétences informatiques.

Nous avons choisi cette combinaison d'outils, car cela nous a permis de construire notre application dans un environnement familier. Notre autre choix pour les outils et langages était le C#, un langage de programmation inconnu par la majorité du groupe, couplé avec le logiciel Unity qui nécessite une certaine période d'apprentissage.

Pour réaliser notre projet, nous avons utilisé plusieurs outils :

- Comme environnement de développement (IDE), nous avons choisi d'utiliser Eclipse car nous avons tous appris à l'utiliser lors de nos cours en programmation Java.
- Nous avons utilisé Google Drive pour stocker des ressources graphiques, partager des documents. (Comme la Boîte à Idées ou le Carnet de Bord)
- Nous avons découvert Git, un outil de gestion de versions, sur conseil de notre professeur référent, et car cet outil sera être indispensable dans nos projets futurs.
- Nous avons choisi Framagit comme hébergeur pour nos dépôts Git car certains des membres du groupe possédait déjà un compte sur cette plateforme.
- Pour mettre en forme le contenu des cartes utilisées et définir la progression du jeu, nous avons utilisé Microsoft Excel.
- Pour les messages courant nous avons utilisé Messenger, car tous les membres du groupe sont présents et disponibles sur cette plateforme.
- Pour régulièrement faire le point sur l'avancement du projet avec notre professeur référent, nous avons utilisé l'application Discord pour son chat vocal.
- Pour la partie graphique de notre projet, nous avons choisi d'utiliser Aseprite pour créer les ressources graphiques, car cet outil est particulièrement adapté à la création d'images suivant les principes du pixel art. Nous avons aussi utilisé Adobe Illustrator pour retoucher les images.



# REPARTITION DES TACHES

Auteur : Simon Weber

Étant donné qu'à la formation de notre groupe, nous nous connaissions déjà, nous avons vite su déterminer quelles étaient les forces et faiblesses de chacun du fait des précédents projets réalisés ensemble. Il a donc été simple pour nous de déterminer ce que chacun de nous apporterait à ce projet :

- Simon, de façon assez naturelle s'est vu être chef de projet en supervisant la répartition et l'avancement des tâches du projet tout en réalisant le tableur Excel où y figure les cartes, objets et statistiques du jeu ainsi que les différents menus de l'application et la page de chargement.
- Paul, du fait de ses 8 ans de passion pour le dessin, s'est occupé de la partie graphique du projet : réalisation de l'interface visuelle du jeu, des personnages, des arrière plans, des objets, de l'affiche et du manuel d'utilisation.
- Jules, du fait de son bon niveau en Application Serveur Java et de ses connaissances en jeux vidéo, s'est occupé de réaliser le mode de jeu multijoueur et d'une interface de jeu minime pour avoir un premier aperçu en étudiant la bibliothèque JavaFX.
- Benjamin, du fait d'un ensemble de compétences modulaires, s'est occupé d'implémenter l'interface graphique finale du jeu en étudiant la bibliothèque JavaFX, d'une partie de la page chargement, mais surtout de la compréhension, réalisation et maintenance du dépôt git de notre projet.
- Julien, du fait de ses compétences en optimisation et gestion d'architecture de projets Java, s'est occupé du développement de l'architecture du code, de l'algorithmique de fonctionnement du jeu et de l'optimisation du code des autres membres du groupe.
- Velan s'est occupé de réaliser des modes de sauvegarde et de scores du jeu en les implémentant dans le menu d'accueil, d'un exemple de déroulement d'une partie et de la création de quelques cartes.



# CONCEPTION DE L'APPLICATION

Auteurs : Julien Leopardo & Jules Plateau

Notre première séance a consisté à un « brainstorming » qui nous a permis de définir la démarche à suivre pour notre projet. Nous avons alors créé un dossier sur Google Drive et partagé des documents (boîte à idées, journal de bord, diagramme de Gantt, etc.). Nous avons discuté de la manière de distinguer notre jeu de « Reigns » dont nous nous inspirons, et des étapes à suivre pour le développement. Nous n'avons donc rien codé lors de cette séance. En effet, nous voulions d'abord bien définir nos besoins avant de commencer.

D'un point de vue algorithmique, le choix du langage Java implique la création d'objets (classes). Nous avons donc dû mettre en place une certaine organisation du code. Nous avons donc par la suite défini l'architecture de l'application. Les classes indispensables à son bon fonctionnement sont donc : les joueurs « Player.java », les cartes « Card.java », « SimpleCard.java » qui est un cas spécifique de « Card.java » (implémentation), « Excel.java » qui transforme le tableau Excel en objets Java, « CardManager.java » qui utilise « Excel.java » et crée les bonnes instances d'objets, et « Appli.java » qui teste si tout fonctionne correctement. Ces quelques classes, qui ont évolué progressivement, nous ont permis dès le début d'avoir une bonne vision de notre application. Dans le même temps, notre graphiste Paul, commençait déjà à dessiner les personnages et les décors pour l'interface graphique.

Pour résumer la suite des séances, avant de travailler chacun sur sa partie, nous faisons le point sur les choses terminées, celles à faire lors de la séance et leur répartition. Nous avons aussi essayé le plus possible de travailler par deux et non tout seul. En effet, cela permet d'être plus efficace et d'avancer plus vite. Pour tenir notre professeur référent, Anton Roy, au courant, nous avons utilisé Discord pour lui expliquer nos avancées. De son côté, ayant accès à notre code via Framagit, il nous donnait des conseils et des indications sur ce qu'on pouvait faire lors de la prochaine séance.

Après avoir créé le tableau Excel des cartes du jeu en plus de ce qui vient d'être décrit, nous avons pu nous lancer dans le développement de l'interface graphique, grâce à la bibliothèque JavaFx. C'est une bibliothèque externe, c'est-à-dire qu'on peut l'utiliser, mais pas la modifier. Il a donc d'abord fallu l'étudier pour comprendre son fonctionnement et identifier les composants qui nous intéressent. Cela nous a permis d'obtenir une première version du jeu dès la semaine du 26 février. Même si aucun élément graphique n'était présent, nous avons pu identifier les bugs dans notre algorithmique et les corriger. Tout en



continuant à développer l'application (ajout des modes online et backup), nous nous sommes par la suite concentré sur les aspects graphiques : tant la réalisation des éléments en eux-mêmes que leur implémentation dans notre code.

C'est après la semaine des DST que nous avons pu nous investir à fond dans le projet. L'aspect graphique du jeu a donc changé de plus en plus rapidement : notre application s'est mise à réellement ressembler à un jeu vidéo. À partir de ce moment, nous avons voulu améliorer le plus possible les fonctionnalités du projet (gestion des événements, des items, etc.). Nous avons aussi conceptualisé le menu du jeu. Cela nous a pris du temps, car la gestion des clics sur les boutons était un peu plus compliquée que prévue. Nous avons codé les différentes redirections, ajouté des musiques et images de fond. Cependant, le code relatif au menu a été séparé du code principal pour ne pas créer d'erreurs, car ils sont deux éléments bien distincts. Il a donc fallu par la suite les relier afin de pouvoir lancer une partie depuis le menu.

Pour le reste du temps qu'il nous restait, nous avons essayé de bien assembler tous les éléments du projet (menu, code principal, musique, images, fichiers de sauvegarde, etc.) tout en terminant les derniers éléments graphiques, dont le poster. En effet, nous avons voulu gardé la même esthétique entre le jeu et le poster. Nos avatars sont donc dans le même style de pixel art, la police et les décors sont les mêmes. Cette volonté d'harmonisation se reflète également dans le dossier.

Pour finir, nous nous sommes rendu compte que nous manquions peut-être de cartes. Au tout début, nous pensions que le jeu symboliserait le temps passé à l'IUT, soit deux années. De plus, les statistiques associées aux cartes n'étaient pas toujours vraiment pertinentes. Nous avons donc dû globalement relire le tableau Excel et corriger les détails, en plus d'ajouter de nouvelles cartes.



# DEVELOPPEMENT ALGORITHMIQUE

Auteurs : Julien Leopardo & Jules Plateau

Comme dit dans la partie précédente, l'algorithme de base a été codé assez rapidement avec comme packages principaux « appli » (Appli.java, Player.java et Card.java), « cards » (SimpleCard.java et CardManager.java, puis plus tard Event.java et Item.java) et « xcel » avec Excel.java et ExelControlPoint.java. Ces dernières classes récupèrent les tableaux Excel et les transmettent à CardManager.java, qui crée les instances de Card, Event et Item. Appli.java quant à elle, vérifie le fonctionnement global du jeu ainsi que les point de contrôle pour le mode multijoueur.

Pour utiliser la bibliothèque JavaFx, nous avons créé un nouveau package « AppliFx », qui contient des classes principales pour former notre interface graphique. Il s'agit de Appli.java et Window.java (elles ont été dupliquées et adaptées pour les modes online et backup). Appli.java (du package « AppliFx ») symbolise une partie, et contient donc plusieurs traitements relatifs au jeu en lui-même. Elle s'occupe notamment de choisir quelle carte afficher, celle à afficher après et vérifier si la partie est terminée. Quant à Window.java, elle représente l'interface graphique en elle-même, avec tous les éléments à afficher. Ces éléments graphiques ont été rassemblés dans le package « javaFxComponents ». Les classes du package implémentent celles de la bibliothèque JavaFx et utilisent directement son code. Notre travail ici consistait donc juste à adapter le code à notre besoin. Il ne reste plus que deux packages : « menu » qui contient les classes relatives au menu et lance les différents modes du jeu, et « network » qui détaille le fonctionnement du mode online.

Rentrons maintenant un peu plus dans le détail. Pour le mode classique, le joueur (Player.java) possède un tableau de statistiques qui lui est propre et qui évolue au cours de la partie. Il a aussi une liste d'items, il peut donc simplement déléguer les traitements qui leur sont relatifs à la classe Item.java. L'interface Card.java (et son implémentation SimpleCard.java) symbolise les cartes. Une carte possède plusieurs attributs (statistiques, événement, image, description, etc.). Les méthodes se résument donc à obtenir ces attributs (getters) et les modifier (setters). La séparation entre Card.java et SimpleCard.java a été faite dans la volonté de rendre le code maintenable. Si plus tard on veut ajouter un autre type de cartes, notre classe devra juste implémenter Card.java. Le projet n'aura donc pas à être modifié. Pour les classes Event.java et Item.java, elles détaillent le fonctionnement de nos événements et items. Quant à CardManager.java, elle utilise Excel.java pour lire le tableau Excel des cartes, et crée les objets Java associés.



Pour le mode backup (avec sauvegarde), Player, Appli et Window ont été dupliquées et adaptées. Nous aurions préféré garder les mêmes classes, mais ce mode ne gère pas les items. De plus, PlayerBackup.java possède de nouvelles méthodes pour lire et écrire dans le fichier de sauvegarde.

Network est le package contenant toutes les classes liées à la gestion des connexions entre les joueurs (mode online). Ce mode est prévu pour deux joueurs uniquement : le premier démarre un serveur et attend une connexion, tandis que le second s'y connecte. À savoir que la classe Server.java ne gère qu'un seul socket. Pour le client et le serveur, l'utilisation de la méthode runLater de la bibliothèque permet d'effectuer des tâches sur le thread principal ainsi que des actions graphiques. Le socket est passé aux classes qui gèrent les entrées et sorties de données (ManagerConnexionIn.java et ManagerConnexionOut.java) auxquelles la classe AppliOnline.java fera appel. Il y a deux données différentes à transmettre : l'atteinte d'un point de Contrôle, et les statistiques du joueur (qui ne sont utilisés que pour savoir si l'autre joueur a perdu et ainsi finir la partie, mais peut servir pour de futures améliorations). La classe NextControlPoint.java gère un GroupView affichant si le passage aux points de contrôle s'est activé.

Le package menu contient toutes les classes concernant le menu ainsi que la gestion de la musique dans la classe BackgroundMusic.java. La classe StartGame.java est utilisée comme classe de lancement du jeu. Toutes les constantes y sont stockées ainsi que le « stage » (nom de la fenêtre en JavaFx). Elle contient également l'écran de chargement. Les différents écrans du menu sont : accueil (HomePage.java), multijoueur (Online.java), continuer (Continue.java), score (Score.java) et crédits (Credits.java). Les objets graphiques particuliers selon les écrans sont stockés et traités dans les classes Title.java, MenuBox.java, MenuItem.java, MenuBoxOnline.java et MenuItemOnline.java (Les textes sont passés en blanc pour l'écran multijoueur.)



# DEVELOPPEMENT GRAPHIQUE & SONORE

Auteurs : Paul Courbois & Benjamin Di Santo

## Justification Graphique et UX design

Notre projet possède une patte graphique affirmée. Nous avons choisi le pixel art et un aspect rétro comme style graphique, comme dit précédemment, nous voulions évoquer une certaine mélancholie chez notre public cible en rappelant les jeux de jeunesse. De plus, le style rétro et pixel art est extrêmement présent dans les jeux indépendants de nos jours, car ce sont des dessins faciles à créer, à animer et avec un bon rendu pour un minimum de travail. Cela colle donc avec le peu de temps et de budget de notre projet, de plus c'est un style qui plaît à un public très large, les plus jeunes qui sont attirés par toutes ces couleurs et les plus vieux qui ont connus les jeux pixelisés. C'est donc pour cela que c'est un style qui est à la mode de nos jours et qui plaît beaucoup sur des jeux comme « Red String Club », « Owlboy » et « Hyper Light Drifter »

Concernant l'UX design qui est tout ce qui touche à la structure et au design de l'application en général, c'est à dire l'interface utilisateur. Pour ce qui est des menus nous voulions rester simple et fonctionnel pour facilement naviguer entre les menus et ne pas perdre le joueur avec un menu stylisé inutilement. En ce qui concerne la phase de jeu on a voulu mettre en avant le personnage avec lequel on interagit pour donner une idée des conséquences de nos choix sur notre entourage. Notre interface de question est donc inspiré des jeux de rôle japonais de la « GameBoy Advance » notamment de « Final Fantasy Tactics Advance » pour en plus de cela ajouter à l'aspect nostalgie. On a aussi ajouté des phases de journées pour ajouter de la vie au jeu avec des nuages qui passent et aussi un lieu qui est indiqué pour montrer un aspect quotidien au joueur. Enfin, nous avons situé les jauges de vie au-dessus de tout et assez grandes pour être visibles et surtout lisibles et marquer leur importance.



## Intégration Graphique

Après avoir fait nos choix esthétiques, pendant la création des images nous avons créé un prototype d'interface de jeu avec JavaFX pour avoir une application fonctionnelle nous permettant de déboguer le projet et son fonctionnement sans l'intégration des images. Une fois le prototype de jeu fini avec des boutons et des images tests, nous y avons ajouté les images créées par nos soins.

À ce moment-là nous avons donc eu quelques petites difficultés, d'une part nous devions camoufler les boutons et asset de JavaFX pour y ajouter la texture voulue, d'autre part les images devaient être positionnées au pixel près ce qui a pris beaucoup de temps à faire et surtout certaines images n'étaient pas à la bonne échelle, ce qui faisait un rendu incohérent, et les jauges avaient un fond transparent ce qui rendait très peu lisible leur niveau sur un fond de nuit. Nous avons donc retouché quasiment toutes les images pour avoir le rendu le plus propre possible.

Nous avons ensuite répété l'opération pour toutes les autres parties de l'application (les scores, le menu principal...) Enfin après avoir réglé tout cela, nous avons fait des animations comme les boutons qui changent de couleur au survol de la souris, des nuages qui défilent dans le fond, et le fond, les couleurs des nuages et les lieux qui changent au fur et à mesure du jeu.





## Sound Design

Ce n'est qu'après l'intégration des éléments graphique que nous nous sommes penché sur le Sound Design. En effet, le jeu correspondait à nos attentes visuellement parlant, mais ce dernier manquait de sonorités. C'est ainsi que nous avons commencé à décorer notre jeu de sons et musique.

Le premier élément que nous avons intégré était le son des boutons qui survenait lorsque le curseur passait au-dessus de ces derniers. Il fallait que ce dernier ai une sonorité qui colle à notre jeu, nous avons cherché à travers les différentes banques de sons gratuites jusqu'à en trouver un qui ai un style « retro / pixel ».

En parallèle de celui-ci, nous avons cherché 4 musiques pour habiller notre jeu :

- Une musique calme et accueillante qui serait présent dans le menu d'accueil et dans les autres menus ainsi que dans la page de chargement.
- Une musique de jeu un peu plus rapide faisant ressortir un sentiment de réflexions, stratégie compte tenu de notre style de jeu,
- Une musique de victoire aux sonorités joyeuses,
- Une musique de défaite aux sonorités tristes,

Ces 4 sons ont été rapidement trouvés, collants toujours à notre style retro. En effet, ces dernières sont des musiques 8-bit correspondant au musiques des jeux des premiers consoles, dite de nos jours « rétro ».

Un petit détail que nous avons rectifié était la transition entre la musique d'accueil et de jeu qui s'avérait trop brutale, nette. Nous avons donc implanté une fonction diminuant progressivement la musique pour derrière augmenter progressivement la suivante pour que cela soit plus agréable à l'oreille.



# RESULTATS

Auteur : Velan Senguttuvan

Voici le résultat de notre travail. Vous pouvez y retrouver les différentes fenêtres du jeu.

## Le Menu :



Le menu est le point d'entrée aux fonctionnalités de notre jeu. Il est composé de notre logo placé au centre de la fenêtre et d'une liste de boutons permettant d'accéder aux fonctionnalités.

Le joueur se retrouve sur le menu lorsqu'il lance le jeu ou lorsqu'il clique sur le bouton retour dans une des fonctionnalités.

## La Barre de Chargement :



Lorsque l'utilisateur lance une partie, il se retrouve sur une page de chargement.



## L'Interface de Jeu :



Le jeu est composé de quatre blocs qui sont :

- Les statistiques du joueur
- La fenêtre de dialogue
- Le lieu et / ou Item
- La fenêtre de réponse
- L'arrière-plan de la fenêtre change aléatoirement

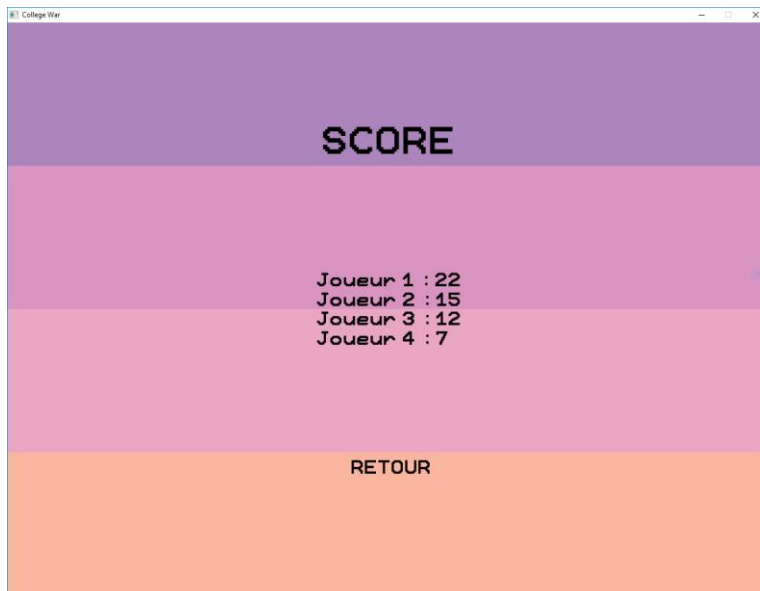
## La Fenêtre de Sauvegarde :



La fenêtre « CONTINUER » permet de lancer le jeu depuis une sauvegarde existante.



### Le Tableau des Scores :



La fenêtre « SCORE » affiche le score des joueurs enregistré dans l'ordre décroissant.

### La Fenêtre Multijoueur :



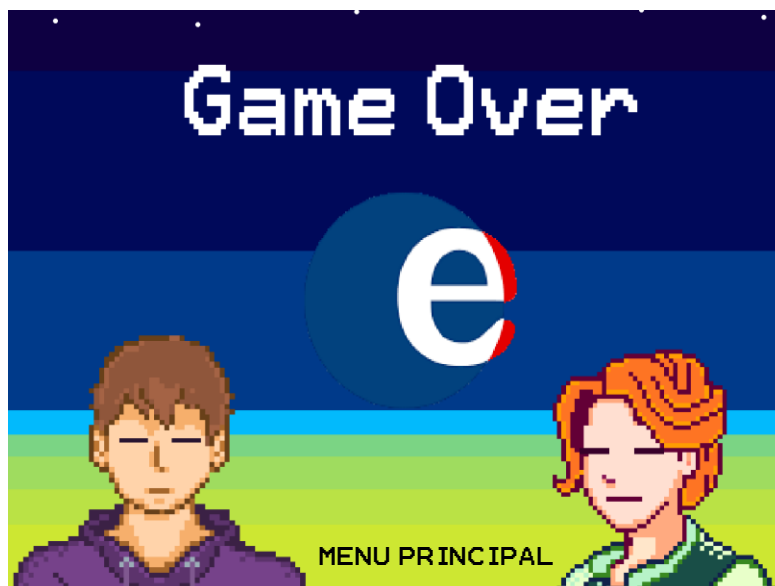
Le mode multijoueur permet à 2 joueurs de rentrer en compétition sur une partie pour savoir qui survivra le plus longtemps.



Ecran de Fin, Partie Gagnée :

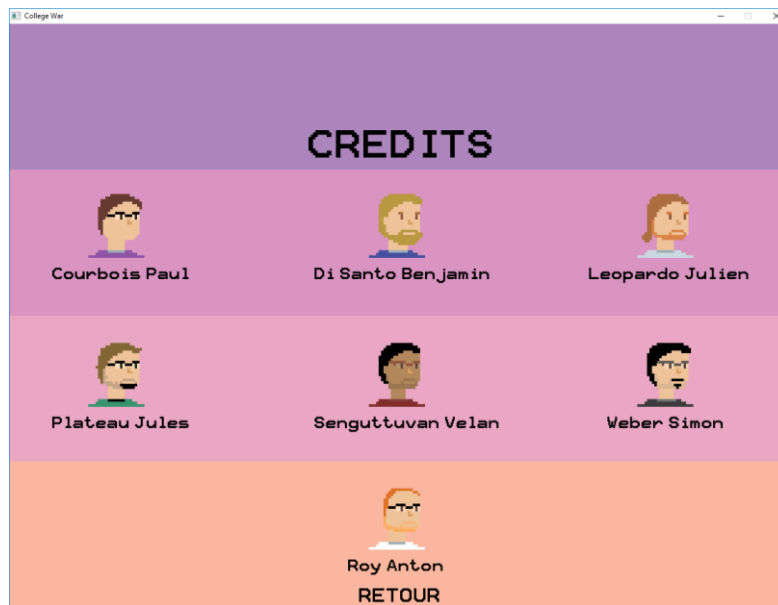


Ecran de Fin, Partie Perdue :





## Les Crédits



La fenêtre des crédits affiche les noms des personnes ayant participé au projet avec une image en « pixel art » les représentant.



# CONCLUSION

Auteur : Benjamin Di Santo

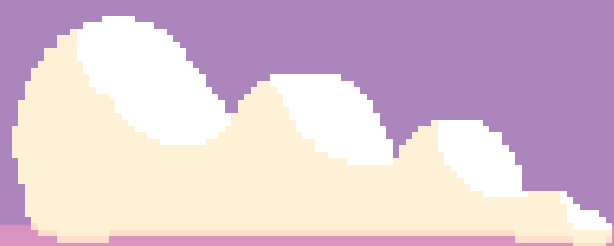
Arrivé à la fin de la période de développement du PJS4, nous avons réussi à développer une version avancée et fonctionnelle de notre application. En effet, la majorité des fonctionnalités que nous avons imaginées dans la Boite à Idées ont été implémentées et fonctionnent correctement, mais quelques améliorations sont nécessaires pour aboutir véritablement à une version complète de notre application qui pourrait être publiée en ligne.

Tout au long du développement de l'application, les principaux problèmes techniques que nous avons rencontrés concernaient surtout la bibliothèque JavaFX et la manipulation des données à partir d'Excel. En effet, face à ces difficultés, nous avons appris en autodidacte à manipuler ces nouvelles technologies pour arriver à avancer dans notre projet. Une autre difficulté technologique et organisationnelle rencontrée était la gestion du dépôt Git sur Framagit, un outil qu'aucun membre du groupe ne savait utiliser.

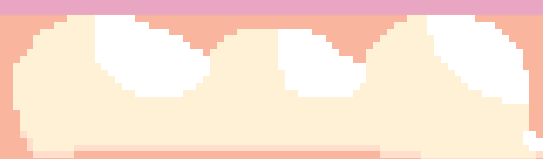
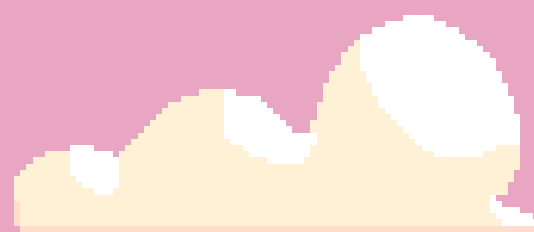
Pour les améliorations possibles, il serait intéressant :

- D'ajouter un mode multijoueur en ligne où deux joueurs sur 2 réseaux différents pourraient se retrouver dans des salons.
- De réaliser un College War sur Android du fait que ce dernier a été développé sous Java faisant que la portabilité ne sera pas complexe à réaliser ;
- D'ajouter un système de cycle de périodes comme en IUT pour connaître l'avancement du joueur sur les 2 ans d'étude.
- De fusionner le mode Backup et normal de sorte à ce que le mode sauvegarde soit fonctionnel avec les objets.
- De voir les jauges qui seront impactées par les réponses des questions avant de choisir la réponse.
- D'améliorer les objets où les statistiques s'avèrent trop brutales et d'ajuster les statistiques des cartes en général.
- De créer d'autres cartes avec des personnages différents et d'événements déblocables sous des conditions plus complexes.

Pour conclure, ce projet s'est révélé très enrichissant dans la mesure où il a consisté en une approche concrète à la réalisation d'un projet informatique complet. En effet, la prise d'initiative, le respect des délais, la répartition des tâches, surmonter les difficultés et le travail en équipe seront des aspects essentiels de nos futurs métiers que nous avons appris tout le long de ce projet.



## ANNEXES

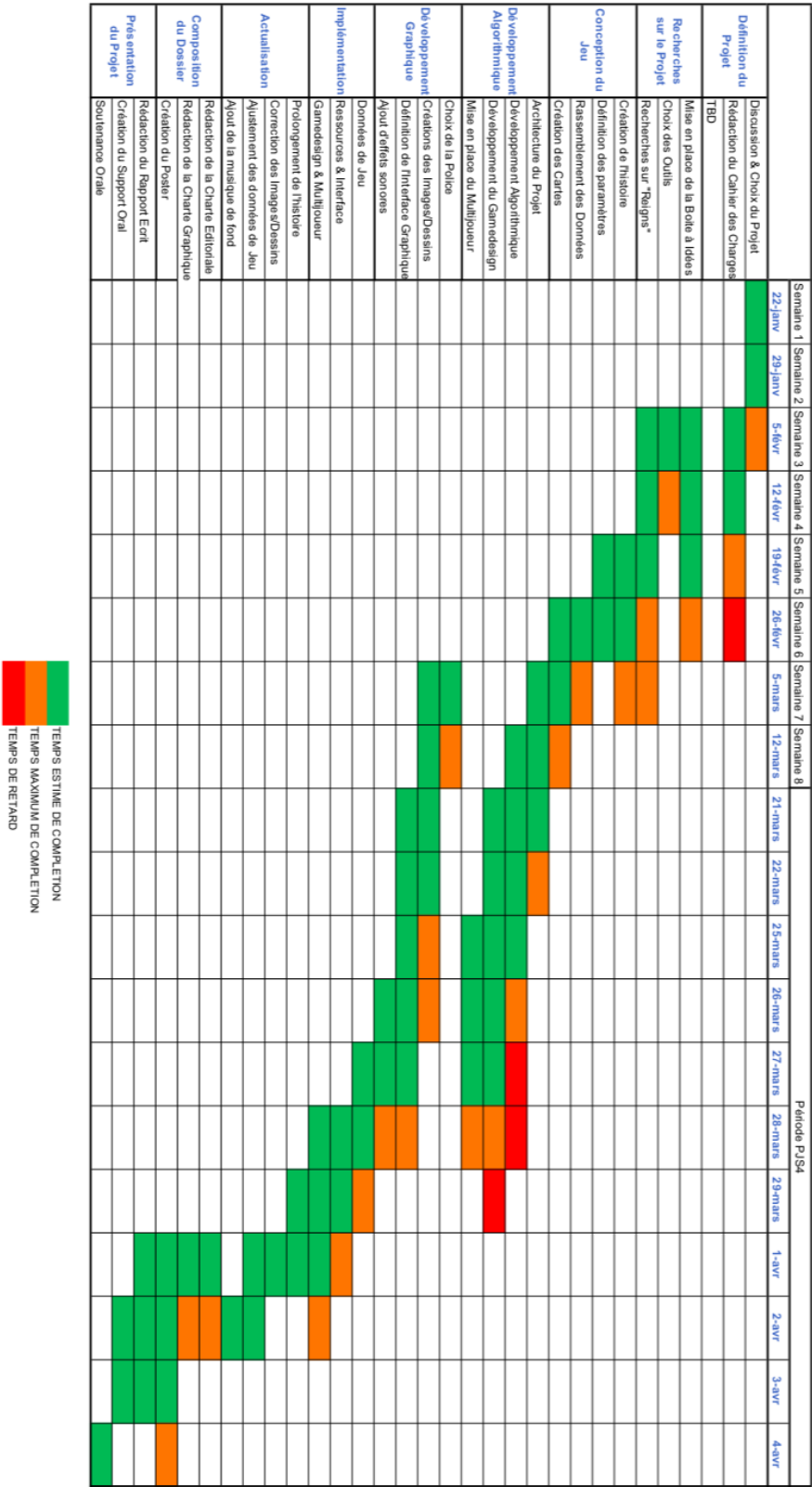






# DIAGRAMME DE GANTT

Auteur : Benjamin Di Santo

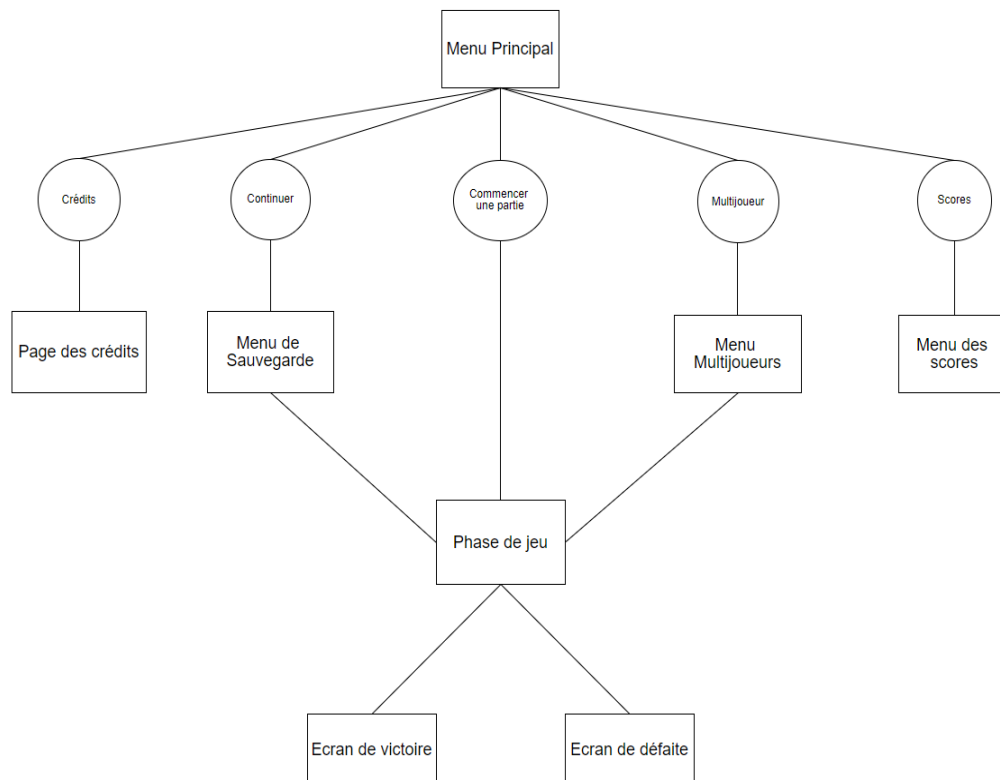




# MANUEL D'UTILISATION

Auteur : Paul Courbois

## Organigramme de l'Application :



## Menu Principal :

But : permet d'accéder à une fonctionnalité de l'application



Commencer une partie : Lance une phase de jeu depuis le début

Continuer : accède au menu de sauvegarde

Score : accède à la page des scores

Multijoueur : accède au menu multijoueur

Crédits : accède à la page des crédits



### Menu Choix de Sauvegarde :

But : permet de choisir la partie que le joueur veut continuer



Joueur 1 : permet d'accéder à la partie sauvegardée de l'utilisateur « joueur 1 », le menu s'incrémente en fonction des sauvegardes et des noms des utilisateurs.

Retour : permet de retourner au menu principal

### Menu des Scores :

But : permet de voir les scores des différents joueurs



La Zone de Texte sert à rentrer l'adresse IP de l'hébergeur d'une partie.

Host : prend l'adresse IP du champ, et attend une connexion d'un autre utilisateur

Connexion : prend l'adresse IP du champ, et attend se connecte à un autre utilisateur

Retour : permet de retourner au menu principal

**Ecran de Jeu :**

But : permet de jouer au jeu



Jauges : indique votre niveau en certaines catégories, elles varient en fonction de vos réponses et de vos objets, et si l'une est vide le jeu est perdu. (De gauche à droite : savoir, popularité, santé, moral, argent)

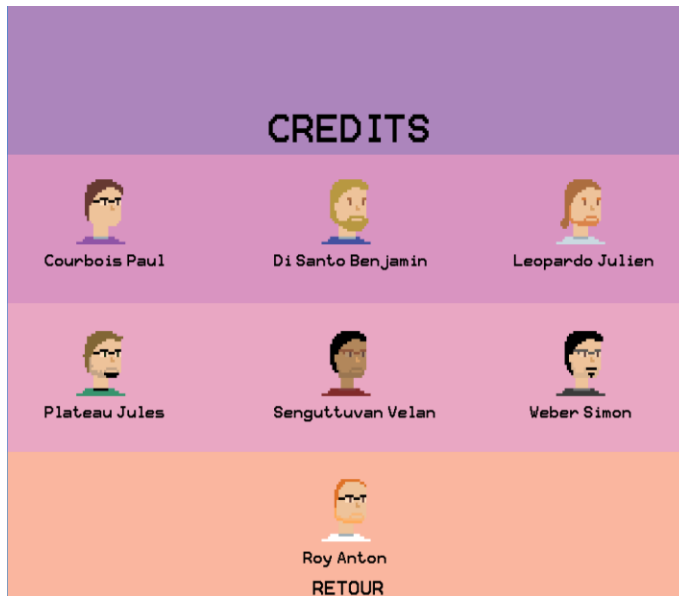
Objets : influent sur les effets des réponses

Lieu : indique le lieu où le joueur se situe

Boutons Réponses : deux réponses qui influent sur les jauges

**Ecran des Crédits :**

But : permet de voir les créateurs du jeu



Retour : permet de retourner au menu principal



# CHARTRE EDITORIALE

Auteur : Benjamin Di Santo

## Titre de Page

<b>Police</b>	Pixel12x10
<b>Taille de Police</b>	18
<b>Style</b>	Aucun
<b>Alignement</b>	Centré
<b>Espacement (Après)</b>	9 pt
<b>Couleur du Texte</b>	Noir
<b>Interligne</b>	1.3

## Nom de l'Auteur

<b>Police</b>	Pixel12x10
<b>Taille de Police</b>	12
<b>Style</b>	Souligné
<b>Alignement</b>	Gauche
<b>Espacement (Après)</b>	10 pt
<b>Couleur du Texte</b>	Bleu
<b>Interligne</b>	1.3

**Sous-Titre de Page**

<b>Police</b>	Calibri (Body)
<b>Taille de Police</b>	12
<b>Style</b>	Gras
<b>Alignement</b>	Gauche
<b>Espacement (Après)</b>	0 pt
<b>Couleur du Texte</b>	Noir
<b>Interligne</b>	1.3

**Texte de Page**

<b>Police</b>	Calibri (Body)
<b>Taille de Police</b>	12
<b>Style</b>	Aucun
<b>Alignement</b>	Justifié
<b>Espacement (Après)</b>	0 pt
<b>Couleur du Texte</b>	Noir
<b>Interligne</b>	1.3



# CHARTE GRAPHIQUE

Auteur : Paul Courbois

Les Arrière-Plans :



*Fond Matin*



*Fond Jour*

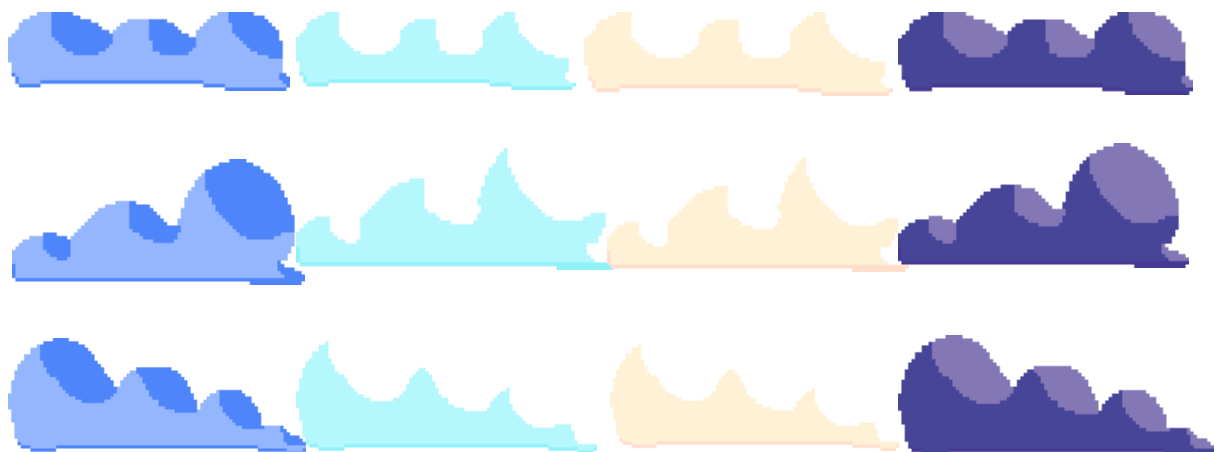


*Fond Crepuscule*



*Fond Nuit*

*Trois nuages déclinés pour chaque fond*







**Les Fonds de Texte :**



*Fond des questions et nom du personnage*



*Fond des réponses*

**Les Bonus :**



*Travail*



*Triche*



*Fumeur*



*Amour*



*Billet de  
Retard*



*Buveur*



**Les Personnages :**



*Léa*



*Ryan*



*Ordinateur*



*Norton*



*Anonyme*



*Marvin*



*Marc*

**Les HUDs (Heds Up Display / Information Interface):***Jauge d'Argent**Jauge de Moral**Jauge de Popularité**Jauge de Santé**Jauge de Savoir*

## Les Ecrans de Fin de parties :

*Partie Gagnée**Partie Perdue*

## Les Avatars de l'Equipe :



Paul



Benjamin



Julien



Jules

Velan

Simon



Anton



# LEXIQUE

Auteurs : Julien Leopardo & Jules Plateau

**Algorithme :**

« Processus à effectuer pour répondre à un problème. Ce processus est toujours le même pour un même problème. »

InfoWebMaster, « Glossaire », *InfoWebMaster*,

<http://glossaire.infowebmaster.fr/algorithme/>

**Architecture :**

« Structure générale inhérente à un système informatique, organisation des différents éléments du système et des relations entre eux. »

Collectif, « Architecture (informatique) », *Wikipédia*,

[https://fr.wikipedia.org/wiki/Architecture\\_\(informatique\)](https://fr.wikipedia.org/wiki/Architecture_(informatique))

**Bibliothèque externe (external library) :**

« Ensemble de routines (portion de code) de classes compilée et prête à être utilisée par des programmes. »

Collectif, « Bibliothèque logiciel », *Wikipédia*,

[https://fr.wikipedia.org/wiki/Biblioth%C3%A8que\\_logicielle](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle)

**Brainstorming :**

« Réunion de groupe ou chacun est invité à émettre librement et spontanément des idées ou suggestions en relation avec le sujet de l'étude. »

Définitions marketing, « Définition : Brainstorming », *Définitions marketing*,

<https://www.definitions-marketing.com/definition/brainstorming/>

**Cahier des charges :**

« Le cahier des charges décrit précisément les besoins auxquels le prestataire doit répondre, et organise la relation entre les différents acteurs tout au long du projet. »

Anonyme, « Cahier des charges : définition simple et traduction », *JDN*,

<https://www.journaldunet.fr/business/dictionnaire-economique-et-financier/1199141-cahier-des-charges-definition-traduction/>

**Conflit :**

« Problème détecté par un système d'exploitation indiquant que deux composants ou plus utilisent une même ressource. » Un conflit dans le code arrive lorsqu'au moins deux versions différentes d'un même fichier entrent en concurrence.

Collectif, « Conflit matériel », *Wikipédia*,

[https://fr.wikipedia.org/wiki/Conflit\\_mat%C3%A9riel](https://fr.wikipedia.org/wiki/Conflit_mat%C3%A9riel)

**Diagramme de Gantt :**

« Outil efficace pour représenter visuellement l'état d'avancement des différentes tâches qui constitue un projet. »

Gantt.com, « Qu'est-ce qu'un diagramme de Gantt ? », *Gantt.com*,

<https://www.gantt.com/fr/>

**Direction artistique :**

« Le directeur artistique de jeux vidéo est responsable de l'avancement visuel du jeu, c'est un expert en design qui gère les graphismes du jeu. »

Anonyme, « Directeur artistique de jeux vidéo : salaire, formation, description du métier... », *JDN*,

<https://www.journaldunet.fr/management/guide-du-management/1199605-directeur-artistique-de-jeux-video-salaire-formation-description-du-metier/>

**Discord :**

« Logiciel gratuit conçu à la base pour les communautés de joueurs, qui permet de créer des salons vocaux ou textuels. »

Collectif, « Discord (logiciel) », *Wikipédia*,

[https://fr.wikipedia.org/wiki/Discord\\_\(logiciel\)](https://fr.wikipedia.org/wiki/Discord_(logiciel))

**Eclipse :**

« Logiciel visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java. »

Collectif, « Eclipse (projet) », *Wikipédia*,

[https://fr.wikipedia.org/wiki/Eclipse\\_\(projet\)](https://fr.wikipedia.org/wiki/Eclipse_(projet))

**Environnement de développement (IDE) :**

« Ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels »

Collectif, « Environnement de développement », *Wikipédia*,  
[https://fr.wikipedia.org/wiki/Eclipse\\_\(projet\)](https://fr.wikipedia.org/wiki/Eclipse_(projet))

**Framagit :**

« Forge logicielle (système de gestion de développement collaboratif de logiciel) reposant sur le logiciel Gitlab. »

Framasoft, « Framagit / Gitlab . GitBook », *Framasoft*, <https://docs.framasoft.org/fr/gitlab/>

**Git :**

« Logiciel de gestion de versions décentralisé »

Collectif, « Git », *Wikipédia*, <https://fr.wikipedia.org/wiki/Git>

**Google Drive :**

« Service de stockage et de partage de fichiers dans le cloud lancé par Google. »

Collectif, « Google Drive », *Wikipédia*, [https://fr.wikipedia.org/wiki/Google\\_Drive](https://fr.wikipedia.org/wiki/Google_Drive)

**GUI / interface graphique (Graphical User Interface) :**

« Dispositif de dialogue homme / machine, où les objets à manipuler (le plus souvent avec une souris) sont dessinés sous forme de pictogrammes. »

Collectif, « Interface graphique », *Wikipédia*,  
[https://fr.wikipedia.org/wiki/Interface\\_graphique](https://fr.wikipedia.org/wiki/Interface_graphique)

**Implémenter :**

« Effectuer l'ensemble des opérations permettant de définir un projet et de le réaliser. » Nous utilisons aussi ce mot dans le sens suivant : coder l'algorithme relatif à une fonctionnalité.

Académie française, « Implémenter », *Académie française*, 07/02/13,  
<http://www.academie-francaise.fr/implémenter>

**Industrie vidéoludique :**

« Secteur d'activité économique de la conception, la production et la commercialisation du jeu vidéo. »

Collectif, « Industrie vidéoludique », *Wikipédia*,  
[https://fr.wikipedia.org/wiki/Industrie\\_vid%C3%A9oludique](https://fr.wikipedia.org/wiki/Industrie_vid%C3%A9oludique)

**Interface :**

« Ensemble de méthodes abstraites et constantes, doit être implémenté par une classe, et est utilisée comme telle sans pour autant implémenter son comportement. »

UPMF Grenoble, « Cours java : Interface », *UPMF Grenoble*,

<http://imss-www.upmf-grenoble.fr/prevert/Prog/Java/CoursJava/interface.html>

**Java :**

« Langage de programmation orienté objet, permettant la création de logiciels facilement portable mais un peu lourds à exécuter. »

Collectif, « Java (langage) », *Wikipédia*, [https://fr.wikipedia.org/wiki/Java\\_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage))

**Maintenance :**

« Mise à jour ou adaptation ou amélioration d'un programme (corrections de bugs, répondre aux nouveaux besoins, etc). » (traduit depuis l'anglais)

Anonyme, « maintenance Definition from PC », *PCmag*,

<https://www.pcmag.com/encyclopedia/term/46542/maintenance>

**Messenger :**

« Système de messagerie instantanée créé par Facebook et incorporé au réseau social. »

Collectif, « Facebook Messenger », *Wikipédia*,

[https://fr.wikipedia.org/wiki/Facebook\\_Messenger](https://fr.wikipedia.org/wiki/Facebook_Messenger)

**Méthode Scrum :**

« Méthode de gestion de projet, qui est développé sous forme de sprint (en cycle) qui se concentre sur quelques fonctionnalités voire une seule. »

Définitions marketing, « Définition : Méthode Scrum », *Définitions marketing*,

<https://www.definitions-marketing.com/definition/methode-scrum/>

**Package :**

« Unité / fichier regroupant des classes Java liées entre elles. »

Anonyme, « Java - les packages », *CCM*,

<https://www.commentcamarche.net/contents/562-java-les-packages>



**PEGI (Pan European Game Information) :**

« Classification d'un jeu qui évalue la classe d'âge auquel est destiné un jeu vidéo. »

Service public, « Jeux vidéo : qu'est-ce que le classement PEGI ? », *Service public*,

<https://www.service-public.fr/particuliers/vosdroits/F33922>

**Pixel art :**

« Technique de dessin où l'on dessine des objets ou des personnages pixel par pixel »

Anonyme, « Pixel art », *Wikimini*, [https://fr.wikimini.org/wiki/Pixel art](https://fr.wikimini.org/wiki/Pixel_art)

**Portabilité :**

« Pour un programme informatique, capacité à être porté pour fonctionner plus ou moins facilement dans différents environnements d'exécution. »

Anonyme, « Portabilité (informatique) : définition », *Le Parisien*,

[http://dictionnaire.sensagent.leparisien.fr/Portabilit%C3%A9%20\(informatique\)/fr-fr/](http://dictionnaire.sensagent.leparisien.fr/Portabilit%C3%A9%20(informatique)/fr-fr/)

**PJS4 / projet tuteuré (Projet Semestre 4) :**

Projet pluridisciplinaire en groupe de 5 ou 6 qui dure tout le semestre 4, en autonomie avec un professeur référent, et dont le choix du sujet est libre.

**Statistiques :**

Les différentes jauges qualifiant un personnage (dans le cas de notre jeu, Intelligence, popularité, moral, argent, santé).

**UXDesign (User eXperience) :**

« Penser et concevoir un site web ou une application de manière à ce que l'expérience utilisateur soit la meilleure possible. »

Gabriel Dabi-Schwebel, « UX Design : qu'est-ce que l'expérience utilisateur ? », *1 min 30*,

<https://www.1min30.com/dictionnaire-du-web/ux-design-user-experience>

**Unity :**

« Moteur de jeu multiplateforme permettant de sortir un jeu sur de nombreuses plateformes différentes. »

Collectif, « Unity (moteur de jeu », *Wikipédia*,

[https://fr.wikipedia.org/wiki/Unity\\_\(moteur\\_de\\_jeu\)](https://fr.wikipedia.org/wiki/Unity_(moteur_de_jeu))



# BIBLIOGRAPHIE

Auteurs : Jules Plateau & Simon Weber

## **Musique de Jeu (1) :**

HeatleyBros, « Block Party! Chill Fun Hip Hop Game Music by HeatleyBros »

*Youtube.com*, 06/05/2017 - (Libre de Droits)

<https://www.youtube.com/watch?v=89H4jhuRKpo>

## **Musique de Jeu (2) :**

HeatleyBros, « 8 Bit Street! Funky City Game Music by HeatleyBros »

*Youtube.com*, 13/01/2018 - (Libre de Droits)

<https://www.youtube.com/watch?v=kGemKgdFYWg>

## **Musique de Jeu (3) :**

HeatleyBros, « 8 Bit Space Groove! Chill Pretty Game Music by HeatleyBros »

*Youtube.com*, 09/06/2015 - (Libre de Droits)

<https://www.youtube.com/watch?v=Bok8nLviThg>

## **Musique de Jeu (4) :**

8 bit, « Village Theme - Royalty free 8bit Music »

*Youtube.com*, 03/03/2018 - (Libre de Droits)

<https://www.youtube.com/watch?v=e2foisOQ0Bw>

## **Musique de Fin de Partie (Victoire) :**

carlitoshacker, « 8-Bits Music (No Copyright!) »

*Youtube.com*, 02/10/2016 - (Libre de Droits)

<https://www.youtube.com/watch?v=XEMQ-kj2nlo>

## **Musique de Fin de Partie (Défaite) :**

redjakOfficial, « Redjak : bareknuckle »

*Youtube.com*, 29/08/2012 - (Libre de Droits)

<https://www.youtube.com/watch?v=rTJVjkptFlc>

**Musique de Menu :**

HeatleyBros, « 8 Bit Beginning ! Slow Chill Game Music by HeatleyBros »

*Youtube.com*, 02/07/2016 - (Libre de Droits)

[https://www.youtube.com/watch?v=ptlYSrZU\\_ZU](https://www.youtube.com/watch?v=ptlYSrZU_ZU)

**Police :**

Corne2Plum3, « Pixel 12x10 »

*Dafont.com*, 05/03/2019 - (Libre de Droits)

<https://www.dafont.com/pixel12x10.font>

**Inspiration Ressources Graphiques - Stardew Valley :**

ConcernedApe, « Stardew Valley »

*stardewvalley.net*, 26/02/2016

<https://www.stardewvalley.net/>

**Inspiration Ressources Graphiques - Final Fantasy Tactics Advance:**

Square Product Development Division 4, « Final Fantasy Tactics Advance »

*Nintendo.com*, 08/09/2003

<https://www.nintendo.com/games/detail/final-fantasy-tactics-advance-wii-u>

**Inspiration Ressources Graphiques - Hyper Light Drifter :**

Heart Machine, « Hyper Light Drifter »

*SteamPowered.com*, 31/03/2016

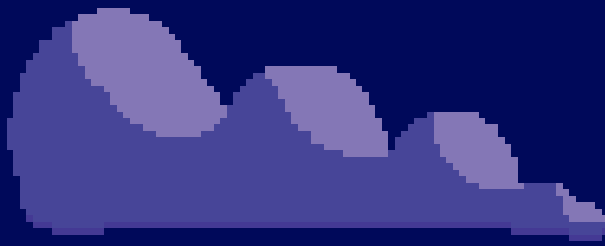
[https://store.steampowered.com/app/257850/Hyper\\_Light\\_Drifter/](https://store.steampowered.com/app/257850/Hyper_Light_Drifter/)

**Inspiration Ressources Graphiques - The Red Strings Club :**

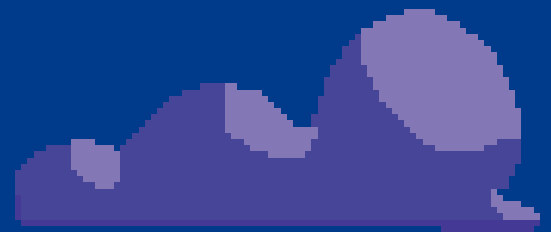
Deconstructeam, « The Red Strings Club »

*SteamPowered.com*, 22/01/2018

[https://store.steampowered.com/app/589780/The\\_Red\\_Strings\\_Club/](https://store.steampowered.com/app/589780/The_Red_Strings_Club/)



## RESUMES INDIVIDUELS





# RESUME INDIVIDUEL EN ANGLAIS

Auteur : Paul Courbois

We had some difficulties when choosing our topic, but Jules and I want to work in the video game industry, so after some debating, our group decided to work on a video game. We chose to work on a « Reigns like » game because it is an attainable and realistic project, with many interesting facets such as the development and the design. Once our subject was defined, we analyzed everyone's skills and we concluded that my 8 years of experience in graphic design would be a useful tool and my love for video games would help designs our game's user experience. Another that I was assigned the data exploitation in « Eclipse » (our Java programming platform) from an Excel sheet.

Our application is an interactive narrative game with binary choices. Then the player is asked a question and he has only two possible answers. Each response has multiple effects on his different gauges. These gauges are the player's life. If one of them hits 0: its game over. Our game is about the life of a DUT student, so we decided to use 5 gauges: « Knowledge », « Popularity », « Health », « Morale » and « Money ». We also implemented some objects which the player can unlock with specific choices and each of them will have various effects. Also, in our game, there are some events which the player can have access through specific choices.

During this project, I've worked a lot on digital drawing and defined our game's artistic direction. I decided to use pixel art to create a retro and melancholy feeling. Moreover, with « Eclipse » and the « Apache POI » library, I created a prototype of the data exploitation process so we could access the data created by Julien from our app. In this project, I designed the game with Jules, but all the game design choices were always approved by all of them before implementation. After all of this Benjamin and I worked on the animation and incorporating my drawings in the app. It was a difficult part: some images weren't the right size and I had to rework a lot of them.

Overall, we didn't have a lot of problems: our project looks very complete and professional. Given more time, we would have expanded on different areas such as drawing, cards, and testing. To conclude, this project helped me and my future. It's my first video game with a team. It will be a very useful and intriguing experience that will help me get accepted in a video game school. I've learned so much about the team around a video game project, all the tools, a lot of new aspects of the job. This project allowed me to confirm my passion for video games and the fact that I want to make video game design into my career.



# RESUME INDIVIDUEL EN ANGLAIS

Auteur : Benjamin Di Santo

This end of the year tutored project (also name PJS4) consisted of the development of an application of our choosing. Our team decided to go for an original video game inspired by the wildly popular app « Reigns ». Its game concept is quite simple: the player becomes a medieval king and faces a series of tough choices with only two possible options that will have a tradeoff effect on four different key components (« Army », « Money », « Religion » and « Populace »). The goal of the game is to last as long as possible without having one of these components become too high or too low. So strategy is key in this game.

As a group, we all played the game multiple times and really enjoyed the story as well as its creative gameplay, so we decided to recreate it and adapt it to our own storyline. We chose to make the player a University student who tries to get his diploma while dealing with tough choices along the way. However, instead of dealing with a kingdom's army, wealth or populace, the player now has to worry about his « Knowledge », « Popularity », « Health », « Morale » and « Money ». Since our group has a general likeness for old video games, we decided to completely change the aesthetics of our app from « Reigns' » and proposed a very retro-like GUI (Graphic User Interface), using pixel-art graphics and a very simple color scheme.

My role in this project was to develop and implement the GUI into the application. During the first couple of weeks, I followed several tutorials online to try and master this new challenge: the JavaFx Library. This library allows its users to implement a functional, dynamic and aesthetically pleasing interface. Once all the structural code was put in place by Jules & Julien and the pixel-art graphics were created by Paul, it was time for me to contribute to the app. I was able to mix my skills with the JavaFx Library and my creative design skills to realize the interface our group had agreed on. To achieve this, I implemented new Java methods to create the graphics components that we needed for our app and worked closely with Jules to allow a fluid rendition of the gameplay.



I also had a secondary role within this group, as project maintainer, I learned how to use Git technology and since it has various websites that exploit it, we chose Framagit because our team members already had accounts there. This technology allows users to collaborate on a project by sharing and editing code on multiple devices. As the maintainer, I hosted the repository and was in charge of ensuring our project kept a logical structure as well as eliminating redundancies and keeping it up to date. I also taught my team how to use this technology so they could share their work from home or the IUT without damaging the project's integrity.

Overall, I am really proud of my team's work on this end of the year project. It was an important part of our curriculum and our first major group project. The fact that we were able to work well together on a subject that interested us all, I believe, motivated us even more to perfect our project. In the end, we are able to present a complex, functional and complete application that we are all very proud of.



# RESUME INDIVIDUEL EN ANGLAIS

Auteur : Julien Leopardo

Our project consists of the development of a video game. We took inspiration an existing mobile game « Reigns ». To summarize, the player experiences various situations where he or she has to select one option between two choices. The player has « statistics » that evolve (increase or decrease) based on the choices. The goal is to survive until the end, and you lose if one of the « statistics » becomes too high or too low. So you need to use your head and make the relevant choices.

In the original game « Reigns », the player embodies a king from the Middle Ages. To distinguish ourselves from it, we chose to adapt our game to our student lives. The player goes to our University, works, spends time with friends, etc. We also changed the player's statistics, that are now « Knowledge », « Popularity », « Health », « Money » and « Morale ». Finally, we decided to completely transform the game's aesthetics. The original game's choices are symbolized by some cards that pop up and then slide out of the screen. We wanted our game to have a strong retro atmosphere, because of our love for old games. We all played some video games on old Gameboys when we were younger. Thus, we created a graphic interface that reminded us of these games. We designed the game's characters and our avatars on the poster using pixel art for the same reason.

Regarding my role in the project, I coded the basic architecture: the classes and objects that are necessary for the game to function properly. I used the knowledge that I acquired during the last two years in order to make the code as optimal as possible. The goal is to allow easier maintenance: if we want to work on the project again, it will take less time and effort. I also worked a lot on the implementation of all the game's features. We decided to use some of « Reigns » and we invented some others. Therefore, I focused my efforts on coding the algorithms of these features (handling of events, the creation of the Java objects from the Excel table, etc.). I also helped Velan, who developed the backup mode for the game, in the implementation of his work. He made the code, but I did the fusion between his code and the game's code. I had to make sure to avoid any conflict or error. Finally, I reviewed all of the other members' work, to optimize the code. I did my best to make it more efficient with fewer lines.





I am really proud of my work because this was the first time I took part in such a major project. The fact that we chose our group and our subject motivated me even more. In the beginning, we had many different ideas, and I thought that we would never have enough time. But we managed to deliver a product that is as complete as possible. We could not add all the features we wanted and there are many things to improve, but we did our best and I really enjoyed the time we worked together.



# RESUME INDIVIDUEL EN ANGLAIS

Auteur : Jules Plateau

The project our group made is inspired by the game « Reigns », a game with a mixture of strategy and reflection with only a binary choice answer. The concept was innovative for me and the amount of work required to complete this application fit with the allotted time and means at our disposal. The game design we had originally thought of turned out to be a bit more complex than we thought, the gameplay was simplified to match the code we were able to develop.

My job during this project was to develop the online aspect of our game and built the code. This work implied a fair amount of use from socket (an online flow of data, a new Java concept we were taught this year) with two clients who need the same data to send and to receive. The concept of the online game is very similar to a solo game, but to add competition between the two player, there are « Control Points » that separate the cards of each player. These « Control Points » are cards with beneficial attributes to the first player to hit them. When that happens, the player who is falling behind is brought up to the same level and the game continues. You win if the other player loses or leaves, or if you reach the end of the game, and you lose if one of your stats goes to zero. At the end of the game, if both players haven't lost yet and reach the end, the app compares their scores and the one with the highest score is declared the winner.

I also worked closely with Julien to unite the codes, resolve some of the implementation errors. With my general knowledge of all the parts of the code, I was able to quickly implement new functionalities and / or resolve some quick changes that were needed. At the beginning of the project when we really began to code, Simon and I assigned tasks to each member to get a starting point for everyone so could quickly define what was the following steps were as the project advanced. On my part I would have liked to resolve some bugs that might happen and to improve on the "fluidity" of our game so that the player may not feel frustrated about the game's simplicity.

I would say I'm very happy and proud to have worked in this very effective and creative team. I think we are all proud of what we achieved even if the result isn't perfect. The game works very well and has a pleasing aesthetic thanks to Paul and Benjamin. Overall, there are no big regrets on any aspect of the game and no major issues were encountered during the development of the game.



# RESUME INDIVIDUEL EN ANGLAIS

Auteur : Velan Senguttuvan

As a final project for the DUT Informatique (named PJS4), my team and I did a video-game inspired by « Reigns », which is a popular video game where the player plays as a king that has to make decisions that impact 4 aspects of his kingdom. If a statistic is too low or too high, the player loses. Our team liked the concept of the game and we decided to develop a game that mostly followed the same rules of « Reigns », but also has his own rules adapted to our game's storyline and environment. If we chose this type of game, it is because it seemed doable in the time that we had and also because it had a good « skeleton » (basic and essential structure) that we could personalize as much as we want.

In this project, my main role was to implement the save function into the game so the player could quit the game and continue where he left when he came back. I also did a function to implement a leader board system into the game so we can see who got the best score. I created the score and continue tab on the menu screen. All of my work was realized with the help by Julien, Simon and Jules to finish it and implement it in the final file.

At the beginning of the project, before we had a full week dedicated to PJS4, I helped create the game design and define the storyline. This meant creating cards, thinking about how a player could win and how a game would go.

I think our final result is close to a Beta version of a game because it is mostly functional but lacks certain functionalities that could not be added because we didn't have the time to implement them. We had some difficulties with merging our work in the beginning because we were modifying the same files and were not accustomed to Git management. However, after some time, Benjamin was able to resolve this issue and updating our code was made very simple. He also taught us how to update our code by ourselves from our homes.

To conclude, this project was the first where we really got to do something that was our choice and not imposed by someone else. We got to develop an app from the ground up. The way we were organized was like we were working in a start-up company. It was a first step into the life within a professional environment because we learned and evolved with our project. For example, we needed to use Git so Benjamin taught us how to use it. I am really proud of the work we have done in a such small frame of time and now I know how to build a project from A to Z.



# RESUME INDIVIDUEL EN ANGLAIS

Auteur : Simon Weber

Our PJS4 project's goal was to develop an application of our choice. So it was quite natural for us to develop a video game, although the question of developing a social network for gamers arose during the first week of our project. Our app was developed on the mechanics of the popular game « Reigns » but we adapted it in our own way. Indeed the principle of our game is simple: obtain the DUT, but to achieve this, the player will need to answer to a lot of questions of binary choice, which depending on the response will affect 5 gauges: « Knowledge », « Popularity », « Health », « Money » and / or « Morale » reaches 0, the game is over and you have to start the game over.

The idea of the game came from me. I had from the first sessions a fairly accurate idea of the progress of a typical game as well as some features to implement. By sharing this, the idea was accepted well and so I naturally became project manager. As it turns out, I will be starting the L3 MAGE program next year, I could thus have a glimpse of this job within this project, which proved to be very interesting and resourceful. Indeed, although I do not think I have done all the work, a project manager should realize. I did take action on some aspects such as : supervising my team, taking precedence on some differences of opinion and giving advice on some tasks. However, that was not enough for me to be fully useful within the group.

So I also worked on technical aspects of our game. I devised the Excel spreadsheet which stored all the cards information and the objects that I have for the majority written, a work that took me no more than 2 days because it was entirely editorial and imaginative. In addition to that, I was interested in JavaFx, a UI library that can be used in Java. I used it to develop the home page of the game where we find the main menu and the logo of our game and one of the backgrounds drawn by Paul. This page turned out to be a crucial element of our game because it is here where the user will make his first opinion on the game. So, I decided to add sound effects so our game had more character. I also helped develop other menu items such as the credits page, the loading page as well as the design of the score page and multiplayer screen.



The last thing that was close to my heart was the Powerpoint, as in all the projects I was able to do. Indeed, this essential facet must be perfect to best reflect our work and to ensure that our message comes across as clearly as possible. So, I took the initiative to realize it to be sure that nothing is forgotten and that everything is presented as I imagined then to have it validated by my friends and to take the slides back to my teammates.

To conclude, I found this project to be very enriching both humanly and professionally. Indeed, I was able to discover more about my team: their qualities, their faults and their habits. In parallel with this, as previously said, professionally this project gave me a taste of what I could realize later and prepared me more seriously in Java and in group work. I am so proud of what we have been able to develop and present together as part of this PJS4.



## POSTER



Lors de la création de l’affiche, notre motivation première était de mettre en avant l’interface de notre jeu qui permet d’en comprendre son but très facilement et d’en apprécier le style. C’est pour cela que nous avons commencé par mettre les questions et les réponses en avant avec le fond et les nuages en arrière-plan et un curseur de souris pour signifier que le joueur doit choisir parmi les deux réponses.

Par la suite nous avons mis en avant notre logo en le faisant ressortir un maximum, par soucis d’équilibre dans la composition de l’image et pour que le joueur comprenne un maximum notre jeu, nous avons mis d’un côté les jauges à gérer et de l’autre les objets. Ensuite nous avons rajouté des textes et un sous-titre. Le sous-titre justifie le nom du jeu et exprime son principe de base, les études, les choix, et la ‘survie’ : ‘Pour survivre aux études les choix sont cruciaux’. Pour les autres textes nous avons ajouté des textes de significations des objets et des jauges, nous avons aussi ajouté deux phrases qui finissent de définir notre jeu et équilibrent l’image. Nous avons mis tous ces textes en gris pour ne pas surcharger l’affiche, sauf le sous-titre pour s’accorder au logo, et les deux phrases de cotés sont en majuscules pour ressortir un minimum et équilibrer l’image.

Enfin, nous avons ajouté les éléments dit ‘contractuel’ des affiches de pjs4 en dessous dans une partie séparée par un élément du jeu et s’accorde à la couleur de la question. Enfin, nous avons mis les avatars de notre équipe et enfin les différents outils utilisés.