06/04/2020

PHELIP Maxence GOMBAULT Mallory TROUILLET Nicolas SYLARD Augustin

EPF Life

Projet programmation

Table des matières

[I-Introduction 2](#_Toc71124284)

[II- Cahier des charges 3](#_Toc71124285)

[A) Présentation 3](#_Toc71124286)

[B) Les jauges, la difficulté et la victoire 3](#_Toc71124287)

[C) Les évènements 4](#_Toc71124288)

[a) Les évènements « Daily » 4](#_Toc71124289)

[b) Les évènements « Random » 5](#_Toc71124290)

[c) Diagramme d’activité 6](#_Toc71124291)

[d) Les choix et leurs conséquences 7](#_Toc71124292)

[D) L’interface graphique 9](#_Toc71124293)

[III) Rapport technique 11](#_Toc71124294)

[A) Les classes 11](#_Toc71124295)

[B) Le déroulement du jeu 14](#_Toc71124296)

[a) Le diagramme « Use-Case » 14](#_Toc71124297)

[b) Le logigramme 15](#_Toc71124298)

[IV) Gestion du groupe 16](#_Toc71124299)

[A) Organisation 16](#_Toc71124300)

[B) Difficultés rencontrées 17](#_Toc71124301)

[C) Outils utilisés 18](#_Toc71124302)

[V) Conclusion 19](#_Toc71124303)

[VI) Table des illustrations 20](#_Toc71124304)

[Annexes 21](#_Toc71124305)

# I-Introduction

Notre projet est un jeu du nom de « EPF Life ». C’est un jeu de gestion où l’utilisateur incarne un élève de l’EPF qui est confronté à des choix qui détermineront sa vie sociale, sa réussite scolaire et sa santé.

L’école nous proposant des cours de CPO (Conception Programmation Objet), nous devions réaliser un projet en groupe de 4 afin de mettre en œuvre nos connaissances de programmation et notre esprit d’équipe. Nous avons donc créé notre groupe composé de Mallory GOMBAULT, Maxence PHELIP, Augustin SYLARD et Nicolas TROUILLET.

Après s’être réunis en vue de trouver un projet qui nous plairait, nous avons trouvé une idée qui paraissait intéressante, tout en étant ludique. Nous avons ensuite passé du temps à réfléchir en détail au déroulement du jeu, et nous sommes arrivés à développer un concept que nous allons vous décrire dans ce rapport.

Nous avons évidemment réalisé ce projet avec les restrictions dues au COVID-19. Il a donc été nécessaire de se réunir régulièrement sur Teams, pendant et hors des cours de CPO, afin de garder une vue d’ensemble du projet. En effet, chaque membre du groupe était attitré à des tâches particulières et répertoriées dans un planning.

# II- Cahier des charges

## Présentation

Notre projet est un jeu de gestion crée sur Java, NetBeans. Le joueur incarnera un élève de l’EPF, régulièrement confronté à des choix de tous types, et qui influenceront trois jauges représentant respectivement la santé, l’éducation et la sociabilité. La vie de l’élève sera divisée en quatre parties : le matin, le midi, l’après-midi et le soir.

Le jeu comportera également trois niveaux de difficulté : facile, moyen, difficile. Le niveau facile permet de découvrir le jeu dans sa totalité, car il est presque inconcevable de perdre au jeu. Le niveau difficile est plus challenging que les autres, et permet de réellement chercher à gagner.

## Les jauges, la difficulté et la victoire

Chaque caractéristique (santé, éducation, sociabilité) sera représentée par une jauge de couleur différente : la santé sera représentée en rouge, l’éducation en vert et la sociabilité en bleu.

Chaque jauge aura une valeur maximum de 100 points, avec pour valeur de départ 70/100 en mode facile, 60/100 en mode moyen et 50/100 en mode difficile.

Le mode de difficulté influencera la perte de points : En mode facile, chaque perte de points est à coefficient 1. En mode moyen, chaque perte de point est multipliée par 1,25 et en mode difficile, par 1,5. Le gain de point n’est pas modifié, quel que soit le mode de difficulté.

Si une des jauges passe à 0, le joueur perd la partie. Un compteur sera présent pour compter le nombre de jours passés sans perdre, et si ce compteur atteint 152 jours (soit l’équivalent de 5 mois ou un semestre), alors le joueur gagne. Cela est cependant très long à atteindre, et le jeu deviendrait répétitif car nous n’avons créé qu’une vingtaine d’évènements. Ils se répéteraient donc sans cesse. Nous avons simplement mis cette fonctionnalité car il faut qu’il y ait un moyen de gagner, tout en restant réaliste. Le but étant de surtout s’amuser en découvrant les graphismes et les évènements.

## Les évènements

Les évènements proposés au joueur varieront en fonction du moment de la journée et seront composés de plusieurs choix possibles (quatre ou moins) : le joueur devra choisir l’une de ces propositions. De plus, ces évènements seront répartis en deux catégories : les évènements « daily » et les évènements « random ». Le jeu se déroulera en progression linéaire, chaque évènement ne dépendant pas du choix précédent.

### Les évènements « Daily »

Chaque évènement daily sera attribué à un moment de la journée et il se reproduira chaque jour au moment de la journée concerné. Par exemple, chaque matin, le joueur sera confronté à l’évènement montré ci-dessous

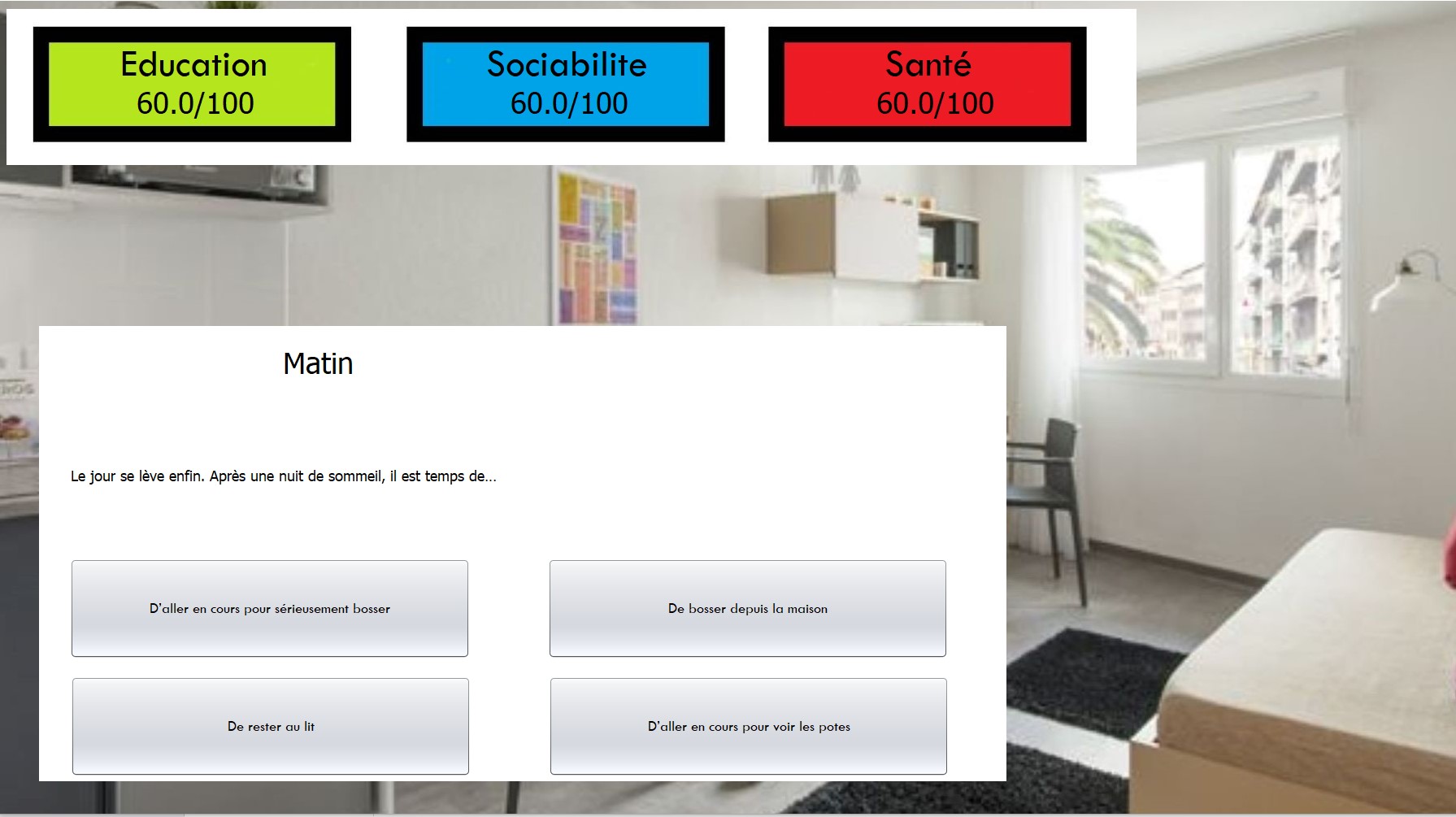
Il pourra alors choisir l’une des quatre propositions, chacune influençant différemment ses jauges. Par exemple, la première proposition augmentera sa jauge d’éducation mais baissera les jauges de santé et de sociabilité, la dernière aura en revanche pour effet de diminuer l’éducation, augmenter la sociabilité, et n’aura pas d’incidence sur la jauge de santé.

Figure 1 - Interface Évènement Daily

### Les évènements « Random »

Les évènement random (aléatoires), quant à eux, auront un certain pourcentage de chance de se produire, toujours suivant le moment de la journée. Ils pourront concerner une évaluation, une soirée avec des amis, … lesquels influenceront, encore une fois, différemment les jauges mais avec, pour la plupart, un impact plus important que les évènements « daily ».

Ci-dessous un exemple d’évènement random :

Figure 2 - Interface Évènement Random



Ici la première proposition augmentera la jauge d’éducation, mais fera diminuer la jauge de sociabilité et celle de santé, à l’inverse de la 2e et de la 3e proposition qui augmenteront les jauges de sociabilité et de santé, tout en diminuant celle d’éducation.

*(Voir en annexe plus d’images du jeu tels que les panels de présentation du jeu, des règles, du choix de difficulté, etc)*

### Diagramme d’activité

Un diagramme d'activité permet de modéliser le comportement du système, dont la séquence des actions et leurs conditions d'exécution.

Le déroulement des choix dans le jeu peut être décrit grâce au diagramme d’activité ci-dessous :

Une image contenant texte

Description générée automatiquement

Figure 3 - Diagramme d'activité

En l’occurrence, le joueur va d’abord lancer le jeu, puis un choix propre au moment de la journée va lui être proposé. Ensuite ses jauges en seront affectées et un nouveau moment de la journée commencera.

### Les choix et leurs conséquences

Ci-dessous la liste des différents évènements, les choix daily et random associés, et leurs conséquences sur les différentes jauges (un « + » équivaut à +10, deux « + » équivalent à +20, etc) :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **N°** | **Nom** | **Personnage** | **Description** | **Choix** | **Santé** | **Education** | **Sociabilité** |
| 1 | Matin | X | Le jour se lève enfin. Après une nuit de sommeil, il est temps de… | 1. Aller en cours pour sérieusement bosser 2. Rester au lit 3. Bosser depuis la maison 4. Aller en cours pour voir les potes | .  ++  .  . | +  -  ++  - | .  -  -  ++ |
| 2 | Midi | X | Le midi arrive ! Qu’ai-je le temps de faire en 45 min ? | 1. Manger en travaillant 2. Manger avec des potes 3. Travailler | +  +  - | +  .  ++ | -  +  - |
| 3 | Après-midi | X | Les cours reprennent… Est-ce que j’ai vraiment envie d’y aller ? | 1. Oui. Je dois m’y mettre sérieusement ! 2. Oui, mais que pour voir les potes. 3. Non, je bosserai mieux à la maison 4. Non. J’ai bien envie de me faire une petite séance de sport ! | .  .  .  ++ | +  -  ++  - | .  ++  -  - |
| 4 | Soir | X | Le soleil commence à se coucher… Après une longue journée bien occupée, il est temps de… | 1. Travailler tard, je suis en retard sur le programme ! 2. Aller dormir, ça me fera du bien 3. Passer une soirée chill sur mon ordinateur | -  ++  - | ++  .  . | .  .  + |
| 5 | Week end | X | Enfin le Week-end ! Idéal pour rattraper le temps perdu ! Sur quoi est-ce que je devrai me concentrer ? | 1. Bosser ! 2. Voir ses potes 3. Prendre soin de moi 4. Faire un peu de tout | -  -  +++  . | +++  --  -  . | --  +++  -  . |
| 6 | Soirée | Pote ? | Une méga soirée vient d’être organisée par tous tes potes ce soir. Un programme bien chargé t’attends. Alors, tu fonces t’éclater ? | 1. Non, j’ai besoin de dormir 2. Non, j’ai besoin de bosser 3. Oui, mais j’y vais en mode chill 4. Oui, et j’y vais à fond ! | ++  .  -  -- | .  ++  .  -- | --  --  ++  ++++ |
| 7 | Maladie | X | Et moi qui pensais qu’une bonne journée t’attendait… Tu t’es réveillé malade ! Que faire ? | 1. Pas le choix, direction le médecin ! 2. Je peux pas rater les cours, tant pis si je me sens mal… 3. J’essaye de bosser à la maison | .  --  - | --  ++  + | .  .  . |
| 8 | ExamenMA | Winston | Winston t’a préparé un partiel de maths abs sur la moitié du programme ! Pas question de rater ça quand même, si ? | 1. Je travaille à fond 2. Flemme de travailler, tant pis 3. Je préfère voir des potes, j’en ai besoin 4. Je fais le minimum de révision | -  +  +  . | +++  --  ---  - | --  +  ++  . |
| 9 | ExamenME | Jousset | Jousset t’indique que Laugerette a préparé un examen de techno méca pour demain. Tu avais complètement oublié ! | 1. Je travaille à fond 2. Flemme de travailler, tant pis 3. Je préfère voir des potes, j’en ai besoin 4. Je fais le minimum de révision | -  +  +  . | +++  --  ---  - | --  +  ++  . |
| 10 | ExamenPH | Barandon | Un examen de méca flotte pour la route ! Barandon ne rigole pas cette fois, révise bien le barrage ! | 1. Je travaille à fond 2. Flemme de travailler, tant pis 3. Je préfère voir des potes, j’en ai besoin 4. Je fais le minimum de révision | -  +  +  . | +++  --  ---  - | --  +  ++  . |
| 11 | Ta copine te trompe | Copine | Tu viens d’apprendre que ta copine t’a trompé pendant une soirée auquel tu n’étais pas invité… Il est temps de prendre une réelle décision. | 1. J’ai pas de copine ! 2. Je me venge direct 3. Je laisse couler, mais que cette fois… 4. Je la quitte et je tourne la page | -  +  -  -- | ++  --  -  - | --  +  .  ++ |
| 12 | Agressé dans la rue | Racaille | Un malfrat t’approche avec un couteau et te menace ! Il veut ton portefeuille et ton téléphone ! Réagis vite ! | 1. Je lui donne tout ce que j’ai sur moi 2. Je m’enfuis en tapant mon meilleur sprint 3. Je réplique. Ça va chauffer. | .  -  --- | .  .  . | --  -  +++ |
| 13 | Tram en panne | X | Et mince, une panne générale sur le réseau de Tram. Tu vas rater les cours si tu trouves pas une solution rapidement ! | 1. Je tape le sprint de ma vie. 2. J’y vais en marchant, tant pis si je suis en retard 3. Flemme, je rentre chez moi | --  .  + | ++  .  - | +  +  - |

Figure 4 - Tableau récapitulatif des évènements et choix

## L’interface graphique

La mise en situation du joueur se fera grâce à des photos en fond d’écran qui représenteront les différents lieux. Certains lieux seront récurrents comme l’appartement de l’élève le matin et le soir, le campus l’après-midi, et d’autres lieux ne seront qu’occasionnels comme les soirées, etc …



Figure 5 - Lieu "Soirée"

Une image contenant intérieur, mur, plancher, pièce

Description générée automatiquement

Figure 6 - Lieu "Appartement étudiant"

Certains évènements seront amenés par des personnages. Plusieurs professeurs seront présents dans le jeu et proposeront occasionnellement des évaluations. Des élèves pourront proposer une soirée, etc…



Figure 7 - Un professeur apparaissant dans le jeu

Une image contenant graphiques vectoriels

Description générée automatiquement

Figure 8 - La copine de l'élève incarné par le joueur

# III) Rapport technique

## Les classes

1. La classe Jauge

La classe Jauge a été développée afin de pouvoir créer trois jauges distinctes. Celles-ci contiennent une valeur max à ne pas dépasser, en l’occurrence 100. Leur valeur sera déterminante pour connaître la victoire ou la défaite du joueur. Une jauge est associée à un multiplicateur. En effet, lors d’un ajout d’une valeur négative uniquement, en fonction de la difficulté du jeu, un multiplicateur viendra intensifié par 1, 1.25 ou 1.5 (en fonction de la difficulté choisie) la valeur perdue lors d’un choix, pour rendre le jeu plus « challenging ». Enfin, une jauge porte un nom, car c’est comme cela que nous les différencions : santé, sociabilité et éducation.

1. Les classes DailyEvent et RandomEvent

Ces deux classes sont ensembles car elles possèdent exactement les mêmes attributs et méthodes. Cependant, il est important de les différencier car elles ne contiendront pas les mêmes types d’évènements. Ces classes permettent de créer un évènement, régit par plusieurs attributs : un titre, une description de l’évènement, un personnage associé à l’évènement, un fond d’écran associé à l’évènement, la description des choix associés à l’évènement ainsi que le nombre de points perdus/gagnés par choix.

Chaque méthode permet de récupérer ces informations primordiales qui seront utilisées dans l’affichage des évènements sur l’interface graphique. Les méthodes sont donc simplement des constructeurs pour les choix, et des récupérations d’attributs car ils sont tous « private ». La seule différence entre évènement « Daily » et « Random » est que les « Daily » sont les mêmes évènements tous les jours (un par moment de la journée), alors que les « Random » apparaissent de manière aléatoire et ont un impact beaucoup plus important sur les jauges.

1. La classe ListeEvent

Cette classe est très importante. C’est celle qui permet de mettre un commun les évènements en fonction des moments de la journée. En effet, c’est dans cette classe que nous avons créé les évènements en leur attribuant leur titre, description, choix, personnage et fond par moment de la journée. Nous avons donc au total 9 listes d’évènements. 2 le matin (une pour l’évènement daily du matin, une pour les évènements random qui peuvent se produire le matin), 2 le midi, 2 l’après-midi, 2 le soir et un le week end. Séparer ces évènements était essentiel car certains évènements n’ont pas lieu d’être le soir par exemple, et donc cela permet de coder plus facilement, en fonction du moment de la journée et du type d’évènement, notre jeu. Les méthodes permettent simplement de récupérer un évènement précis dans une Liste.

1. La classe FenetreDeJeu

C’est la dernière classe de notre jeu. Elle permet de faire l’interface graphique. Elle réunit toutes les autres classes, créant les trois jauges, tous les évènements et déroulant le jeu. La base repose sur l’affichage de la description d’un évènement, puis l’attente du choix d’un joueur, puis l’attribution des valeurs du choix aux jauges, puis l’affichage du prochain évènement et ainsi de suite.   
De nombreux attributs nous permettent de garder en mémoire l’évènement courant afin de ne pas se tromper sur la Liste d’évènement à prendre en compte pour l’incrémentation des valeurs. La classe contient, notamment, des premiers panels pour les règles du jeu et le choix de difficulté ainsi qu’un panel pour la fin du jeu (défaite et victoire).

Toutes ces classes sont représentées dans l’UML ci-dessous qui récapitule notre code sur Netbeans. Il montre les conditions d’appartenance de nos 5 classes (Jauge, RandomEvent, DailyEvent, FenetreDeJeu et ListeEvent).

Nous avons exposé nos attributs, les méthodes et les constructeurs.

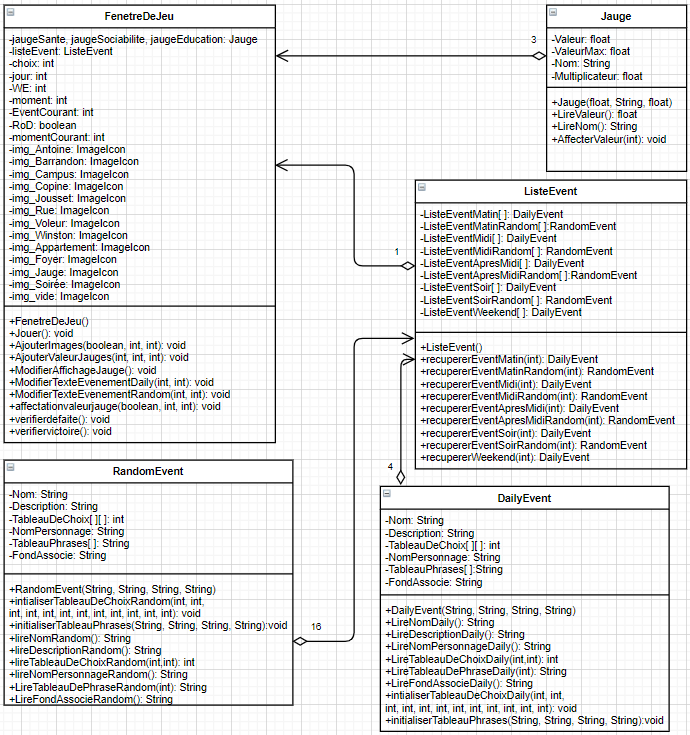


Figure 9 - UML

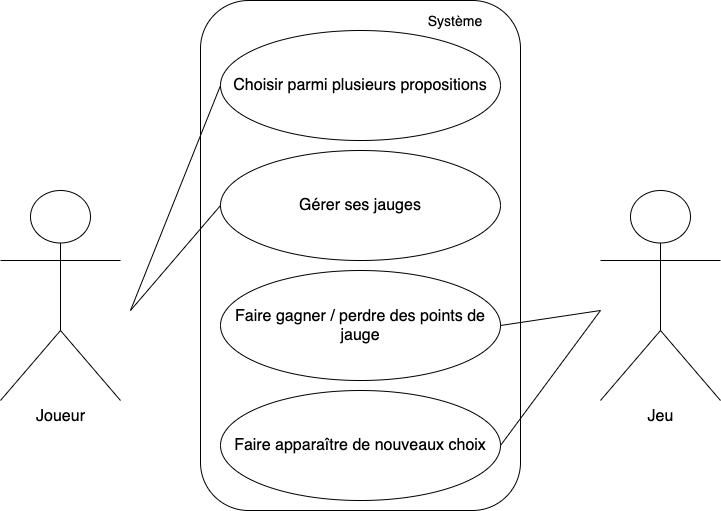
## Le déroulement du jeu

Pour coder le jeu, nous nous sommes basés sur deux diagrammes qui représentent le séquencement du jeu. Ces diagrammes étaient essentiels pour avoir une vision claire et commune à tous de l’objectif final.

### Le diagramme « Use-Case »

Le diagramme de cas d'utilisation permet de visualiser la séquence d'actions exécutées par le système, en interaction avec le joueur. Ici, le joueur interagit sur deux actions différentes (« choisir » et « gérer »). Le jeu va, en conséquence, interagir sur les deux autres, auxquelles le joueur n’a pas accès (« faire gagner/perdre des points », et « faire apparaître des choix »)

Figure 10 - Diagramme Use-Case



### Le logigramme

Le logigramme permet de modéliser plus en détail le déroulement du jeu, et la conséquence de chaque action sur la suivante.

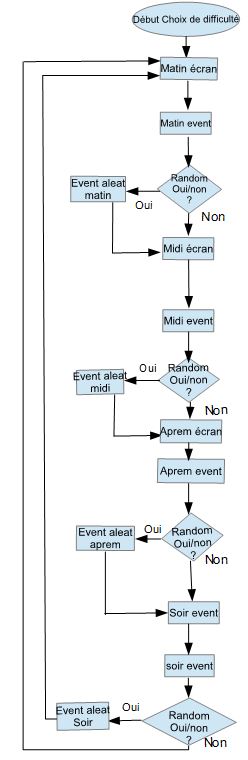


Figure 11 - Logigramme

# IV) Gestion du groupe

## Organisation

Le groupe est composé de 4 membres. Nous avons tous un rôle important à jouer. Le chef de groupe, Mallory Gombault, a organisé un planning afin que nous le respections. Nous avions une deadline à respecter toutes les deux semaines. Nous avons réussi à respecter chaque deadline. Ci-dessous le planning.



Figure 12 - Planning

Augustin s’est occupé de plusieurs classes, et de récupérer des captures d’écran du jeu afin de les intégrer au rapport, Nicolas a fait plusieurs classes et s’est intéressé aux ajouts potentiels dans le jeu, Maxence a majoritairement fait le rapport technique, les diagrammes et le diaporama de soutenance, et Mallory a fait plusieurs classes mais surtout l’interface graphique et a aidé sur le rapport technique. Augustin, Nicolas et Mallory se sont également tous penchés sur l’interface graphique qui a posé certains problèmes évoqués plus bas.

Figure 13 - Diagramme de Gantt

Ici, le diagramme de Gantt représente l’avancée de notre projet en fonction des tâches à effectuer. Le plus gros du travail aura été la rédaction du rapport technique ainsi que le codage de l’interface graphique.

## Difficultés rencontrées

Durant notre projet, les principales difficultés rencontrées tournaient autour de l’interface graphique. Notre premier problème était l’implémentation des images. En effet, nos images changent à chaque évènement, et nous avons eu des problèmes pour réussir à les faire changer dans des JLabel. Nous avons fini par comprendre notre erreur après avoir étudié les anciens projets vus au cours de l’année. Nous n’avons finalement pas eu à créer une classe spéciale pour les images. Nous avons pu tout écrire dans l’interface graphique en créant une méthode « setIcon ».

Le deuxième problème rencontré était basé sur la forme algorithmique de notre jeu. Nous n’arrivions pas à intégrer les boutons de choix à notre jeu comme nous le souhaitions. Nous sommes passés par des boucles while, ce qui faisait tourner le jeu dans une boucle infinie. A l’aide de notre professeur, nous avons découvert une nouvelle manière de procéder, cependant, même si nous n’étions plus dans une boucle infinie, tous nos évènements étaient en décalage. Pour illustrer cela, l’évènement du midi qui s’affichait proposait en réalité les choix de l’évènement du matin précédent, et les jauges n’étaient pas correctement incrémenté. Nous avons mis du temps pour comprendre ce problème de décalage. Nous avons donc réécrit sur le papier le déroulement souhaité du jeu, ce qui nous a permis de modifier tout l’ordre et la manière de procéder du code, tout en gardant l’idée que notre professeur nous avait fourni.

## Outils utilisés

Afin de réaliser ce projet, nous avons utilisé principalement 2 outils. Le principal est Netbeans, sur lequel nous avons réalisé le code, mais nous avons également énormément utilisé « Github ». C’est un outil essentiel à notre projet qui nous a permis de travailler ensemble, à distance. Chacun pouvait interagir sur le projet et donc chacun était à jour concernant l’avancement des autres membres.

Pour les différentes réunions, ne pouvant être faites à l’école, nous avons beaucoup utilisé Teams où nous pouvions également déposer au fur et à mesure tous les fichiers concernant le projet, et les modifier pour conserver l’avancement de chacun.

# V) Conclusion

Ce projet nous a permis de travailler à plusieurs sur un jeu que nous avons choisi et développé, et qui, en conséquence, nous a énormément intéressé.

Nous avons pu, grâce aux différentes deadlines, apprendre à gérer notre temps afin de compléter les objectifs voulus, mais également développer une entraide et un esprit d’équipe en faisant face aux multiples difficultés rencontrées …

Nous avons également beaucoup appris concernant la programmation en Java, mais aussi concernant l’utilisation, pour des travaux de groupes, de différents logiciels de partage de fichiers comme Github ou Teams. Nous découvrons de plus en plus les multiples possibilités de la programmation, et cela nous fait croire que tout est possible.

En effet, de la création des classes à l’interface graphique, toutes nos compétences de codage ont été mises en œuvre, et nous en avons développé de nouvelles.  
De plus, malgré les heures prévues à travailler en groupe, nous avions parfois besoin d’avancer seul, et nous devions sauvegarder notre avancement afin que chacun puisse le réutiliser et à son tour avancer sur ses objectifs. Il était donc très important de partager régulièrement sur les différentes plateformes les fichiers utiles.

# VI) Table des illustrations

[Figure 1 - Interface Évènement Daily 4](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360186)

[Figure 2 - Interface Évènement Random 5](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360187)

[Figure 3 - Diagramme d'activité 6](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360188)

[Figure 4 - Tableau récapitulatif des évènements et choix 8](#_Toc71360189)

[Figure 5 - Lieu "Soirée" 9](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360190)

[Figure 6 - Lieu "Appartement étudiant" 9](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360191)

[Figure 7 - Un professeur apparaissant dans le jeu 10](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360192)

[Figure 8 - La copine de l'élève incarné par le joueur 10](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360193)

[Figure 9 - UML 13](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360194)

[Figure 10 - Diagramme Use-Case 14](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360195)

[Figure 11 - Logigramme 15](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360196)

[Figure 12 - Planning 16](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360197)

[Figure 13 - Diagramme de Gantt 17](file:///C:\Users\Utilisateur\Desktop\Cours%20fiches%20et%20aides\2e%20année\projet%20programmation\EPF%20LIFE\Rapport%20EPF%20Life.docx#_Toc71360198)

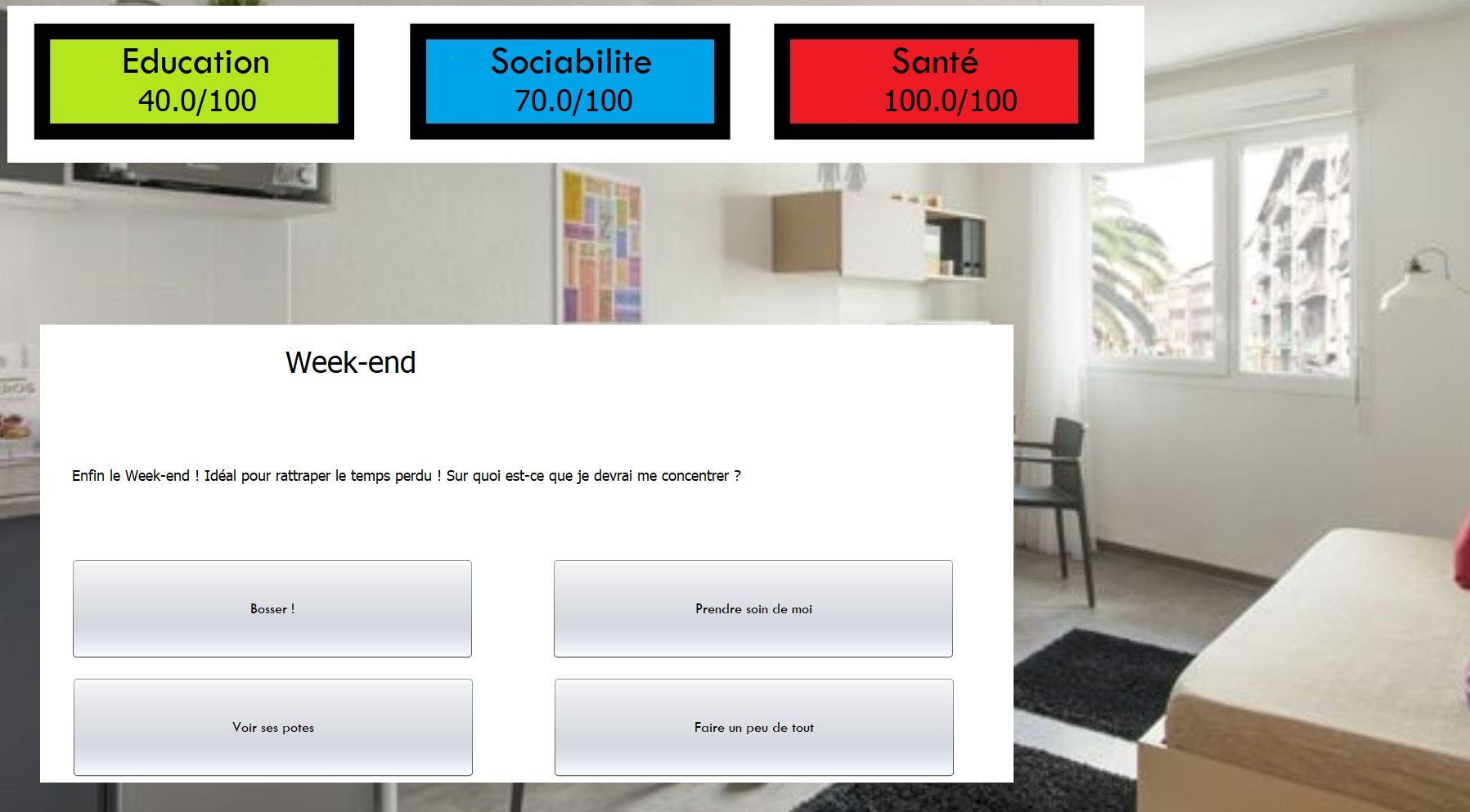
# Annexes

Une image contenant texte, arbre, extérieur

Description générée automatiquement**Annexe 1 :** Fenêtre de lancement du Jeu

Une image contenant texte, arbre, capture d’écran, extérieur

Description générée automatiquement**Annexe 2 :** Choix de la difficulté

**Annexe 3 :** Évènement du week-end

Une image contenant texte, plancher, intérieur

Description générée automatiquement**Annexe 4 :** Fenêtre en cas de défaite