

Compte rendu final

I. Le projet :

1. Démarrage :

Dans la version finale de notre application, le programme commence par demander à l'utilisateur quel map il veut charger et on crée la ville avec tous les types de cases et chaque bâtiment n'existe au maximum qu'une seule fois dans nos simulation. Ensuite, on demande à l'utilisateur avec quel type de personnage il veut jouer, standard, hippie ou homme pressé et s'il veut jouer cette partie avec ou sans permis, avec ou sans maillot de bain et avec ou sans vélo. Une fois qu'il a choisi, le personnage apparaît dans sa maison et enfin, on affiche à l'utilisateur la ville. La partie peut alors commencer.

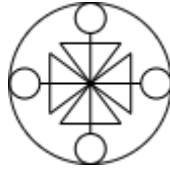
2. Affichage :

Après cela, on affiche la ville avec toutes ses cases. Chaque case est représentée par deux lettres :

- MA : maison
- BA : bar
- BI : bibliothèque
- FA : fast-food
- UN : université
- RO : route
- TR : trottoir
- FO : forêt
- EA : eau
- XX : case grisée

Le personnage se trouve dans la ville et commence sur la case maison. Il est représenté par des parenthèses () qui entourent la case. Par exemple, au départ, comme le personnage se trouve sur la case maison, alors il y aura une case "(MA)" qui est la case maison avec le personnage à l'intérieur.

Puis on affiche les caractéristiques de notre personnage.



3. Actions possibles du personnage :

L'utilisateur peut maintenant déplacer son personnage dans la ville à l'aide des lettres "z", "q", "s" et "d". A chaque fois que le personnage se déplace, on affiche à nouveau les caractéristiques du personnage car elles ont probablement changées, et la ville avec l'endroit où se situe le personnage.

Le personnage peut changer son mode de déplacement en écrivant "voiture", "velo" ou "pieds" et si, bien-sûr, cela est autorisé par la case actuelle.

Lorsqu'il se déplace, le personnage a un certain pourcentage de chances de tomber sur un piège et d'avoir un malus. Lorsque l'une des barres du personnage atteint 0, le personnage meurt et la simulation est finie.

Nous nous sommes permis de faire une modification, remplacer le nombre de points d'hydratation et de satiété perdus qui était à 10, à 3 pour les déplacements à pieds, et de 5 à 2 pour les déplacements en vélo, pour rendre la simulation plus longue et amusante, car le personnage mourrait trop vite sans ces changements.

4. Choix divers d'implémentation :

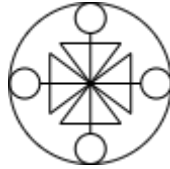
Dans nos Case, nous avons implémenté la méthode `suru()` de différentes façon. Au lieu de créer diverses méthodes pour effectuer des actions diverses liées au déplacement sur une Case, nous avons choisi cette méthode afin de ne garder qu'un seul point d'entrée sur une Case. Ainsi, lorsqu'un personnage se déplace sur une case, il suffit d'appeler `suru()` sur cette case.

`suru()` se charge alors par exemple d'afficher le mode de déplacement et d'affecter le personnage en fonction de la case. Elle appelle également la méthode `determinerPiege()` sur les cases susceptibles d'en présenter.

Nous avons décidé de ne pas générer de maps aléatoirement d'une part, par facilité et, d'autre part, afin d'avoir des maps relativement intéressantes.

5. Répartition des tâches :

- Nathanaël
 - Implémentation des `CaseDeDeplacement` (Route, Trottoir, Eau, Foret, `CaseGrisee`)
 - génération de la ville et boucle principale dans le main
- Sébastien



- Les classes Personnage, HommePresse, Hippie et Standard, ainsi que le questionnaire dans la classe main.
- Guillaume
 - Implémentation des bâtiments (Maison, Bar, Fast-Food, Bibliothèque, Université)

II. Avis Personnels :

Nathanaël :

Le sujet du projet était intéressant et stimulant. Le travail de groupe est une chose qui me plaît de plus en plus. Au vu de l'organisation de ce semestre, on comprend aisément que la ligne directrice de ce dernier était justement le travail de groupe. Nous avons ainsi pu tester différentes organisations avec un nombre de membres plus ou moins nombreux.

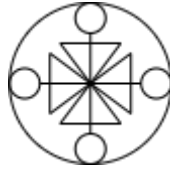
Pour ce qui est du management, il apparaît qu'un groupe se dirige ou "s'auto-dirige" plus facilement lorsque le nombre de participants est limité. Dans ce projet, nous avons par exemple parfaitement réussi à nous diviser les tâches en 3. C'était plus difficile en génie logiciel, lorsque nous étions 5 à travailler sur plus ou moins les mêmes tâches.

Cependant, à l'instar de mes collègues, je trouve que le temps nous a manqué en raison du nombre de projets que nous avons eu à faire en parallèle. Par ailleurs, j'ai l'intime conviction que si nous avions terminé le cours de Design Pattern avant de commencer le projet, cela aurait pu nous aider.

Finalement, j'ai trouvé cette expérience enrichissante. Je garde certains regrets dû au manque de temps comme le fait de ne pas avoir implémenté d'interface graphique. Par ailleurs, j'ai eu parfois quelques réserves sur la justesse de notre modélisation mais nous allons nous améliorer avec le temps.

Sébastien :

Malgré le manque de temps évident dû à la période de fin de semestre et l'ambiance pesante de la situation sanitaire, j'ai trouvé ce projet assez long mais intéressant et facile à réaliser et surtout, nous avons pu utiliser diverses connaissances acquises les semestres précédents.



Je partage l'avis de Nathanaël concernant le travail d'équipe qui était facile à répartir et à collaborer en autogestion, contrairement à Génie Logiciel où c'était un peu plus complexe et désordonné.

Il serait plus judicieux d'avancer le début de la conception et de la réalisation plus tôt dans le semestre afin de donner le temps aux élèves, pour leur permettre de bien finaliser leur travail (comme ajouter une interface dans notre cas) et éviter la surcharge de travail.

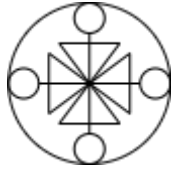
Guillaume :

J'ai personnellement aimé ce projet, et ce, pour plusieurs raisons :

- Tout d'abord, le choix du Sujet. Faire une simulation d'une ville en projet était assez sympathique à faire, cela ressemble un peu à un mini jeu-vidéo, et cela a rendu le projet amusant. Les actions que peut réaliser notre personnage peuvent aussi être drôles, comme par exemple les pièges avec la peau de banane, la poussette et les autres, ce qui a permis d'aborder le sujet sur de bonnes bases.
- Nous avons été mis dans de vraies conditions pour le développement de ce projet, nous devions le faire de A à Z, en passant par les différents diagrammes, à l'implémentation du code, et enfin aux tests.
- Nous avons eu des libertés pour le développement de l'application, et nous avons pu également faire des modifications qui nous semblaient nécessaires, pour rendre l'application plus agréables.
- Les contraintes n'étaient pas très nombreuses (voir inexistantes à l'exception du sujet), nous avons le choix du langage de programmation, et ça fait du bien de ne pas se sentir "limiter", d'être forcé à respecter des règles.

Pour les difficultés du projet, on peut retrouver :

- Respecter la date de rendu, si nous n'avions que ce travail, cela aurait été facile à respecter, mais à cause du reste, cela a rendu le temps de travail limité et challengeant, en restant possible.
- Respecter les dates que nous nous étions fixées sur le diagramme de Gantt (En lien avec le point ci-dessus).
- L'utilisation de Git n'est toujours pas encore totalement maîtrisée, il est arrivé d'avoir des conflits qui auraient pu être évités.
- Faire des diagrammes que l'on peut suivre tout au long de l'implémentation du code. Nous avons réussi à faire un diagramme de classe que nous avons plutôt bien respecté, mais des modifications étaient à faire au fur et à mesure



que nous implémentions le projet. Le diagramme n'était donc pas parfait (même si je pense que cela est normal).

- Respecter toutes les fonctionnalités demandées dans le sujet, notamment celle qui touche à la sauvegarde de la partie. Nous n'avons pas eu le temps d'implémenter un moyen de sauvegarde, c'était ce qui semblait être le plus difficile (ou en tout cas quelque chose que l'on ne connaissait pas).

Lefèvre Nathanaël

Rocca Sébastien

Bonenfant Guillaume