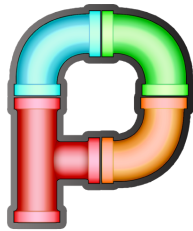


Projet de programmation PREMIER SPRINT

February 1, 2024

Membres du groupe

Khadir Abdessamed
Semani Issac
Tachou Andrei
Zhou Ludovic



Contents

1	Introduction	2
2	Compréhension du projet	2
2.1	Classe case	2
2.2	Condition de victoire	2
2.3	Génération d'une partie	3
3	Repartition des taches	3
4	Futures améliorations	3
5	Conclusion	4

1 Introduction

Ce document vise à vous présenter comment nous nous sommes organisés pour cette première séance de projet de programmation, notre façon de décomposer le travail et la répartition des tâches. Notre groupe s'occupe du projet ED3 (CrazyPlumber)

2 Compréhension du projet

Nous avons commencé par lire le PDF disponible sur moodle et jouer à la version flash du jeu en ligne afin de mieux comprendre son fonctionnement.

2.1 Classe case

On pense à implémenter une classe case qui représentera les cases du jeu sur lesquelles il y a un tuyau. Chaque case aura deux attributs : Un ***int typeDeTuyau*** qui représentera s'il y a un tuyau et quel est son type (Vertical, courbé, triple....) ainsi qu'un ***int orientation*** qui déterminera l'orientation du tuyau (Haut, bas, droite, gauche).

2.2 Condition de victoire

Tout d'abord en réfléchissant nous sommes partis vers une implémentations assez complexe : la classe case devrait permettre à chaque case de vérifier si elle est reliée à une autre et si elles le sont toutes de la base jusqu'à la fin alors le joueur a gagné.

Mais on a trouvé un moyen *BEAUCOUP* plus simple de vérifier la condition de victoire. Il suffit de vérifier que l'orientation des cases (tuyaux) actuelle corresponde à la bonne orientation. Par exemple si on a 3 tuyaux et que la bonne combinaison d'orientation pour mener de la base à la fin est droite, droite, haut, alors il y aura une fonction qui vérifiera constamment l'état actuel des cases(tuyaux) et la comparera avec la bonne combinaison. Le joueur devra les faire tourner jusqu'à l'obtenir.

Concrètement l'attribut ***int typeDeTuyau*** de la classe case n'entre pas en compte avec la condition de victoire et n'est utile que pour afficher le bon type de tuyau (enfin, pour l'instant...)

2.3 Génération d'une partie

Nous pensons écrire notre carte de cases dans un fichier texte *map.txt* et la solution de chaque carte dans *solution.txt*, ensuite un scanner les transformeras en deux listes de cases, la première ***case[] mapDeDepart*** qui sera la liste de cases de départ du niveau avec l'orientation des tuyaux incorrecte et puis ***case[] mapDarrivee*** qui sera la liste de cases avec les bonnes orientations qu'il faut atteindre (on comparera constamment les deux pour savoir si le joueur a gagné) Lorsqu'on lance le jeu les positions initiales des tuyaux seront générées aléatoirement par la formule

```
mapDeDepart[i].orientation = mapDarrivee[i].orientation + random.nextInt(2);
```

Ce qui nous permet d'avoir une orientation des tuyaux toujours différente tout en étant sûr qu'elle ne forme jamais la solution

3 Repartition des taches

Nous avons décidé de nous répartir les taches comme suit et ce jusqu'à avoir fini notre 1er objectif :

- Issac : Coder la classe case et la logique des niveaux ainsi que la création de menu
- Andrei : Créer des sprites pour les tuyaux et s'occuper de leur implémentations graphique
- Ludovic : Afficher graphiquement la map générée par un fichier texte
- Abdessamad : Coder un scanner qui transforme les fichiers .txt en liste de cases

4 Futures améliorations

Il nous paraît impératif d'ajouter notre touche personnel en basant notre jeu sur un thème avec par exemple des tuyaux d'apparence différente (bam-bou, futuriste etc) et nous voulons créer un mode de jeu différent du type contre la montre avec des fuites d'eau graphique qu'on voit à l'écran.

5 Conclusion

En conclusion, cette première séance de projet de programmation nous a permis de mettre en place une organisation efficace pour le développement de notre jeu, CrazyPlumber. Nous avons travaillé de manière collaborative pour comprendre les exigences du projet, décomposer le travail en tâches spécifiques, et répartir ces tâches entre les membres du groupe.

Cette première étape a jeté les bases solides du développement de CrazyPlumber, et nous sommes impatients de poursuivre notre collaboration pour atteindre les objectifs futurs du projet.