

Rapport commun UE Projet groupe ajmn

Répartition des rôles et organisation du travail

Notre groupe est composé de 4 personnes : - ARAB Mounir : **Modèle mémoire**
- BENNOUAR Adel : **Coordinateur** - DOUZI Jibril : **Interface graphique** -
NAGARADJE Nandan : **Fichier de niveau et interface textuelle**

Mounir a réalisé le modèle mémoire, en parallèle avec le travail de Jibril. Jibril a entièrement fait l'interface graphique, et il a bien aidé et contribué au travail de Mounir. Nandan a réalisé l'interface textuelle. Tandis qu'Adel a aidé à été l'interlocuteur principal lorsqu'il y avait un soucis sur le code ou le git, et a aussi fait en partie le chargement de niveau dans l'interface graphique.

Nous n'avions pas eu de meeting hebdomadaire ou mensuel, on avançait selon le rythme de chacun.

Présentation des classes

- **Direction** : Classe enum pour les directions
- **Box, Player Vide, Wall** : Classes permettant de créer respectivement une boîte, un joueur, un bloc de vide, un mur.
- **Element** : Classe commune aux boîtes, joueur, vide, murs, mondes(matrice)
- **Img** : Classe permettant de "générer" une image de boîte, joueur, mur
- **Map, Niveau** : Classes utilisées pour initialiser un niveau dans l'interface textuelle
- **Reader** : Interface textuelle
- **Matrice** : Classe contenant un monde, elle est notamment composée d'un `Element[] []`
- **DrawLevel, FirstApp** : Classes relatives à l'interface graphique. FirstApp s'occupe de toutes les fenetres à l'écran, tandis que DrawLevel s'occupe des actions en fonction de l'actionnement des boutons de l'interface graphique.

Points forts/faibles

Forts

- Communication
- Interface graphique et modèle mémoire réalisés rapidement

Faibles

- Un chargement de niveau incomplet

Difficultés principales

- Faire entrer des sous mondes dans des sous mondes.
- Corriger un bug en lien avec 'ctrl+z'.

- Afficher le contenu des boites avec une profondeur de 4 et le contenu extérieur de la boite courante de 1/4 de cases de chaque côté.
- Lire un fichier qui contient un niveau récursif. Proposition de solution : Utiliser un tableau de **Matrice** pour stocker les sous-mondes et appeler récursivement la fonction de chargement de niveau **loadLvl** (d'où l'argument **Matrice[]** qui n'est finalement pas utilisé !)