

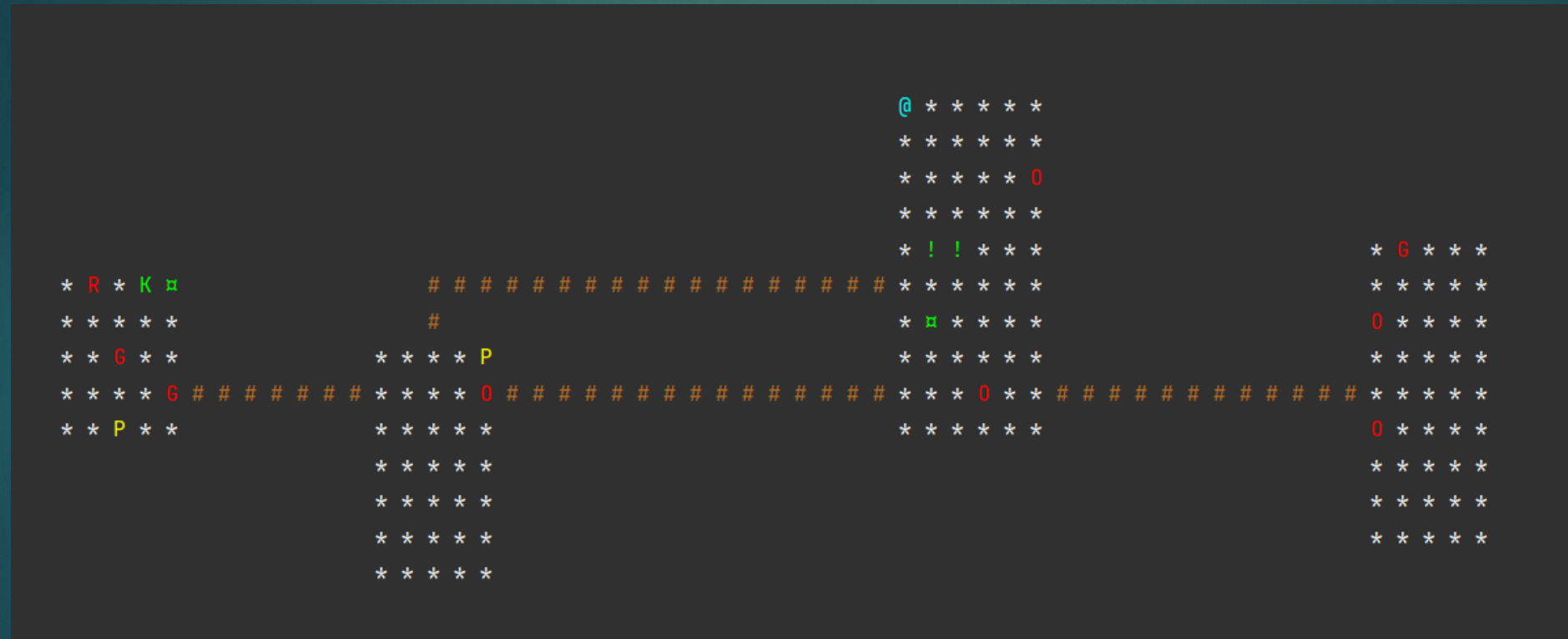


ROGUE-LIKE

Concepteurs

- ▶ PIERRE DAVID Adams
- ▶ NSHIMIYE Judicaël
- ▶ PRIGODIN David
- ▶ POISSON Rodrigue

Présentation Générale



FONCTIONNALITÉS

Le Code est subdivisé en 4 packages :

- . utils
- . map
- . entite
- . artefact

Utils

Le package Utils gère :

- tout ce qui est coloration des symboles utilisés pendant le jeu
- _ la recherche du chemin entre un Monstre et le Joueur
- _ la classe Position qui nous donne les coordonnées sur la map mais également des méthodes qui peuvent nous calculer la distance d'un point à un autre
- _ la classe Utils qui contient des méthodes qui nous renvoient des positions aléatoires dans une salle donnée ou encore une méthode qui nous dit si une position donnée est assez loin du Joueur ou pas

Map

- ▶ Le Package Map est l'un des plus remplis vu qu'il contient les classes des couloirs, des salles, des différentes informations affichées pendant le jeu et de la grille.
- ▶ Dans la Grille, on gère à peu près tout, que ça soit l'initialisation des monstres, des salles ou d'autres entités. On y réinitialise la map, on vérifie que telle entité est à telle position dans la map courante. Et c'est aussi dans cette classe qu'on appelle la méthode « searchPlayer » qui utilise la classe « DFS », dans le package Utils, pour avoir le chemin entre un monstre donnée et le Joueur. Et cette méthode « searchPlayer » sera à son tour appelée dans la classe move avec tous les monstres mais ça on y reviendra plus tard.
- ▶ Il contient également la classe « Draw » qui elle s'occupe de la représentation graphique dans le terminale de la grille.
- ▶ Le Jeu a une génération procédurale qui génère une Map aléatoirement à chaque lancement et à chaque niveau, cette génération est faite dans la classe Map où on génère les salles et tout ce qui a à l'intérieur.

Entite

- ▶ Ce package contient :
- ▶ - le code du Joueur, des différents monstres créés grâce à un pattern factory et un pattern singleton qui bloque la création de plusieurs factory.
- ▶ - la classe « Move » qui gère les déplacements du joueur et des monstres. Pour les déplacements des monstres, cela se fait notamment grâce à l'appel de « SearchPlayer » qui observe le chemin entre chaque monstre et le joueur.

Artefacts

- Ce package contient :
 - le code des entités abstraites comme les coffres, les potions, les clés, les portails etc.
 - la classe « Event » qui s'occupe de la génération de l'escalier et du passage au niveau suivant.

Tests

On a aussi des tests pertinents pour la classe grille, salle etc.

Symboles

- ▶ Le Joueur est représenté par « @ »
- ▶ Les Monstres : - Archer Goblin : « G »
- ▶ - Rogue : « R »
- ▶ - OrcWarrior : « O »
- ▶ Salle : « * »
- ▶ Couloirs : « # »
- ▶ Les clés : « K »
- ▶ Les Potions « ! »
- ▶ Les Coffres « ₪ »
- ▶ Les Portails « P »
- ▶ Les Escaliers « = »

Règles du Jeu

- ▶ On avance :
 - à droite avec « d »
 - à gauche avec « q »
 - en haut avec « z »
 - en bas avec « s »
- ▶ Les Potions augmentent les Points de vie du Joueur
- ▶ Les Coffres sont soit des pièges soit des Potions
- ▶ Les clés ne s'affichent que quand tous les monstres sont morts autrement dits pour ouvrir un escalier il faut trouver la clé (PS : la clé a la même couleur que la salle)
- ▶ Quand on tue tous les monstres on doit bouger n'importe où pour avoir une nouvelle map et donc avec la clé (même chose que quand on trouve la clé)

Quelques Précisions

- ▶ Le code a été fait sur le GitHub https://github.com/adamspd/terd_project
- ▶ Mais aussi sur <https://github.com/adamspd/rogue-like>
- ▶ On a fait un deuxième quand on a repris le code à zéro pour pouvoir garder le premier travail mais bien sûr le dernier travail se trouve sur le premier Github où le client est invité.