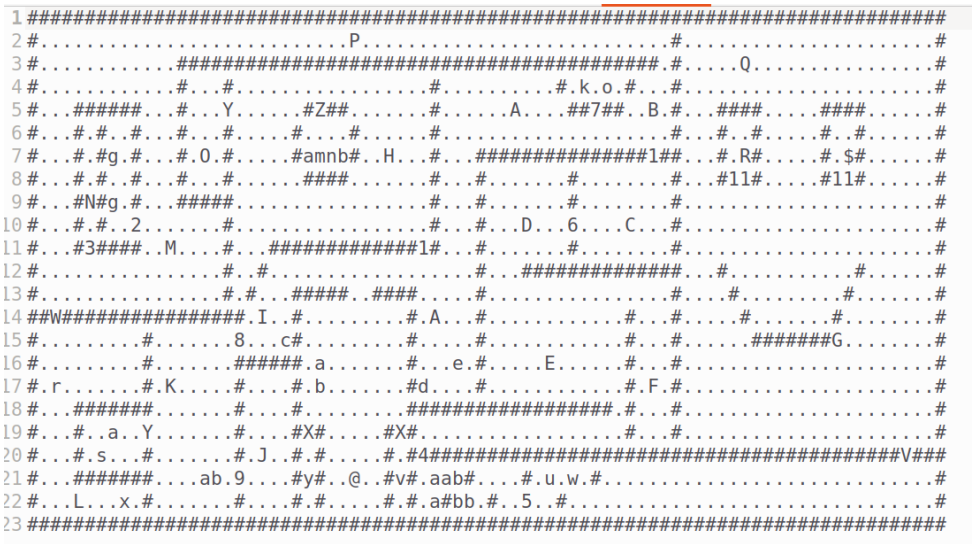
Licence 1
XLG1IE010 - Informatique

Nantes Université
2025/2026

Projet de fin de Semestre
Rogue-Like dans le terminal

1 Contexte

Dans ce projet, nous créons un jeu s'exécutant en mode texte dans le terminal. Il consiste à déplacer un personnage depuis un point départ jusqu'à un point d'arrivée (**$**) sur une grille définie dans un fichier texte. D'autres caractéristiques du jeu seront également définies dans des fichiers.



Le personnage (**@**) se déplace dans les quatre directions. Quelques contraintes :
- il ne peut pas traverser les murs (**#**)
- il ne peut pas traverser les monstres (**majuscules**) sans les affronter [1]
- il ne peut pas traverser les portes (**chiffres**) sans en avoir la capacité[1]
- il ramasse les équipements (**minuscules**)
- il possède six caractéristiques entières :
  - la première nommée *vision* représente la distance à laquelle il peut voir
  - le nom et le sens des autres sont à déterminer

Le jeu est encadré par deux type de contraintes :
- un ou plusieurs objectifs à atteindre (en terme d'équipements ou de caractéristiques) pour gagner
- une ou plusieurs conditions de défaite (par exemple, voir sa caractéristique *vie* tomber en dessous de 1)

1. ce qui peut nécessiter des objets ou des caractéristiques minimales

2 Éléments de conception

Les éléments seront définis en s'inspirant du sujet de TD7 (conception) :
— les équipements sont assimilables à des *articles*
— les portes et les conditions de victoire ou de défaite sont assimilables à des *contraintes*

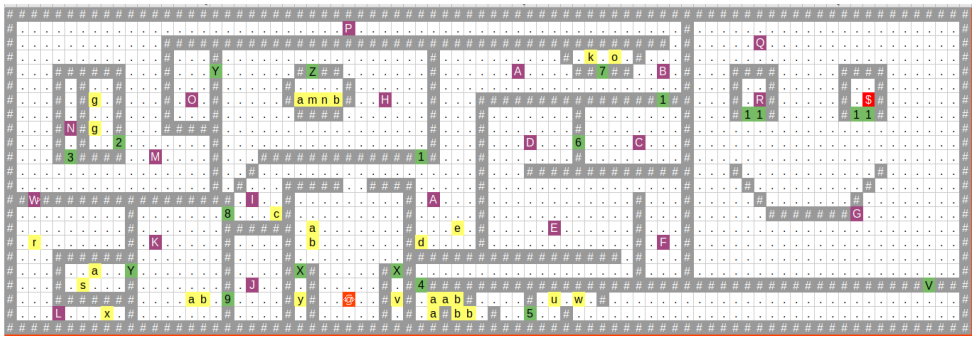
Les monstres sont définis comme un personnage, avec en plus les éléments suivants :
— son caractère actif ou non (ce qui conditionne son déplacement)
— le type d'IA
— optionnellement des contraintes de défaite et de succès
— la portée (qui détermine à quelle distance il peut détecter un personnage)

3 Fonctions attendues

Les fonctions minimales attendues sont :
— lecture et écriture de l'état du jeu (carte, personnage, équipements, monstres et portes)
— affichage de l'état du jeu via la bibliothèque fournie sur Madoc [2]
— déplacement du personnage en tenant compte des murs et des équipements

Les fonctions additionnelles demandées sont :
— gestion des conditions de succès ou de défaite
— gestion des portes
— gestion du combat contre les monstres
— une IA pour les monstres (au minimum, déplacement aléatoire et mécanisme d'activation/désactivation)
— dévoilement progressif de la grille en fonction des déplacements (possiblement sans permettre de voir à travers les murs)

Les fonctions optionnelles demandées sont :
— personnalisation de l'affichage (couleur, par exemple)
— des IA plus variées pour les monstres



2. Aucune autre bibliothèque ne devra être utilisée.