

Un jeu de quête

UE 12 informatique - Apprentissage de la Programmation - année 2021-2022

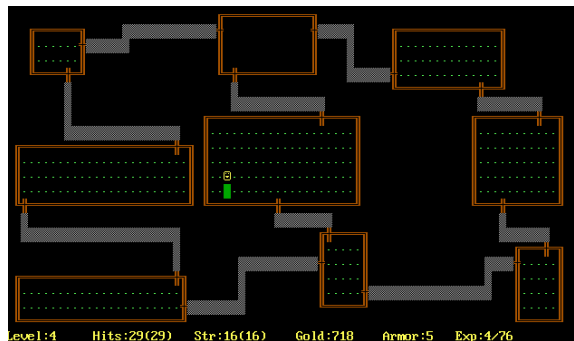
Hackthon du 28 janvier 2022

1 Le jeu

Le but de ce sujet est très simple : vous devez vous inspirer du très célèbre jeu de parcours de donjons Rogue pour réaliser votre propre jeu de donjon qui se joue sur un simple terminal.

Si vous ne le connaissez pas encore Rogue parcourez [https://en.wikipedia.org/wiki/Rogue_\(video_game\)](https://en.wikipedia.org/wiki/Rogue_(video_game)) ou essayer le jeu (pas trop longtemps) là https://archive.org/details/msdos_Rogue_1983.

Vous savez enfin à quoi jouaient les élèves au début des années 80...



rogue - wikipedia

1.1 La forme graphique du jeu

Inspirez-vous des techniques que vous avez utilisées pour le snake.

Faites les choses de **manière graduelle** et *commitez* à chaque fois que vous avez quelque chose qui marchouille.

La suite de ce document explique le jeu pour le C++, pour le Python, adaptez la présentation à la librairie choisie.

Votre terrain de jeux est les lignes et les colonnes de votre terminal où vous contrôlerez votre héros en tapant des caractères (par exemple les lettres 'i', 'k', 'j' et 'l' pour up, down, left et right).

1.2 À quoi ressemblait le premier jeu Rogue

Nous vous donnons ci-dessous très rapidement quelques idées sur la manière (approximative) dont fonctionnait Rogue. Vous êtes libres de vous en inspirer plus ou moins.

Votre *personnage* (le caractère '@') erre dans un donjon à la recherche d'un Graal, de pièces d'or ou pour toute autre bonne raison. Le donjon est constitué de '.' et de '-' pour les murs, de '.' pour le sol, de '+' pour les portes, de '#' pour les couloirs et éventuellement de '=' pour les escaliers mais commencer par un seul niveau est recommandé.

```
-----
|. @ . . . | ## + . . . | #####
|. . . . . | # | . . . | # #
|. . . . . + ### | . . . + ### #
|. . . . . | | . . . | #
|. . . . . | ### + . . . | #
|. . . . . | # - - - + - - #####
- - - + - - # # #
# # # - - + - -
# # # | . . . |
##### | . . = |
-----
```

figure 1: '@' dans la première pièce d'un niveau de 3 pièces

Notons qu'il est plus intéressant pour le jeu que le donjon se découvre petit à petit quand le personnage avance mais dans un premier temps, nous vous recommandons de le laisser visible.

Au cours de sa quête, votre personnage rencontre des ennemis (comme le terrible chevalier noir 'B') qu'il doit combattre vaillamment (par exemple en lui avançant dessus). L'issue du combat n'est jamais sûre, votre personnage s'affaiblit quand il recoit des coups de son ennemi (ses points de vie diminuent) il peut mourir (il n'a plus de points de vie), quand il en sort vainqueur il s'améliore (sa force augmente, il peut aussi gagner des points de vie).

```

-----
|. @ . . . | K # + . . . . | #####
|. . . . . | # | . . . . | # #
|. . . . . + ### | . . . . + ##### #
|. . . . . | | . . . . | #
|. . . . . | ### + . . . . | #
|. . . . . | # - - - + - - #####
- - - + - - # # #
# # # - - + - -
# # # | . . . |
##### | . . = |
-----

```

figure 2: '@' ne voit pas encore 'K' caché dans couloir

Votre personnage va trouver des trésors et des objets durant sa quête.

De l'or '*' qu'il pourra utiliser pour soudoyer un gardien, acheter des objets dans un magasin ou s'enrichir personnellement.

Des potions magiques 'j' qu'il peut ramasser, placer dans son sac pour les boire plus tard ou boire tout de suite au dépend de sa vie (les potions peuvent l'affaiblir, lui apporter des pouvoirs magiques, de la force...).

Des armes (poignards, dagues et épées '!', arcs et flèches '(' qu'il peut ramasser, placer dans son sac et prendre en main pour combattre des monstres.

Des armures ou casques '&' pour se protéger, des parchemins, des anneaux 'o' qui peuvent avoir des pouvoirs quand le personnage les enfle.

De la nourriture parce que votre personnage ne doit pas mourir de faim ni de soif.

Les objets peuvent être magiques alors attention aux mauvais sorts.

```

-----
|. @ . . . | ## + . . . ! . | #####
|. . . . . | # | . & . . . | # #
|. . . . . + ### | . . . . + ##### #
|. . . . . | | . . . o . | #
|. . j . . | ### + . . . . | #
| * . . . . | # - - - + - - #####
- - - + - - # # #
# # # - - + - -
# # # | . . . |
##### | . . = |
-----

```

figure 3: '@' avec une potion, de l'or, un poignard, un anneau, une armure.

Quand il marchera sur un objet, le jeu lui décrira l'objet et là soit le jeu considère qu'il le prend automatiquement soit lui propose de le prendre. Ainsi si " **votre personnage trouve 30 pièces d'or** ", celles-ci se rajoutent automatiquement à son porte-monnaie, si " **votre personnage trouve une potion magique bleue ? pour la prendre faire**) ".

On peut varier le jeu comme on le désire, par exemple, pour les objets autre que l'or, son sac peut ne pas avoir une capacité infinie. Votre personnage devra alors décider de jeter un objet pour en garder un autre ou trouver sur son chemin un sac plus gros (y'en a).

Le personnages peut faire des actions par exemple comme se déplacer, monter un escalier '<', le descendre '>', combattre, lire, manger, boire, s'armer, enfiler une armure, mettre un casque, allumer sa torche (des pièces peuvent être plongées dans l'obscurité) ... tout ce que vous voulez qu'il fasse.

Et votre personnage peut enfin trouver le Graal...

Le donjon peut avoir plusieurs niveaux (d'où les escaliers). Il peut avoir des passages secrets, des trappes; certaines pièces peuvent être plongées dans l'obscurité... tout ce que vous imaginez.

Nous vous recommandons d'y aller progressivement faire un seul niveau c'est déjà très bien...

2 Les consignes

1. découpage

Passer un bon moment en début de séance à réfléchir au jeu en groupe. Découpez le jeu en morceaux et décidez de comment les morceaux s'articulent (s'interfacent). Répartissez-vous les morceaux de manière intelligente. Chacun doit pouvoir proposer des versions de plus en plus évoluées de son "morceau". Partez d'une version minimaliste et complexifiez-la au fur et à mesure.

2. intégration

Faites bien attention à l'intégration des différents codes dans un même système c'est LA partie difficile. Ne codez pas 10.000 lignes de codes chacun en vous demandant à 15h25 comment vous allez bien pouvoir les réunir dans un même programme... Commencez à mettre vos morceaux ensemble le plus tôt possible de manière à identifier tout de suite des défauts qui pourraient compromettre l'intégrité du système. Attention même les groupes de "forts" ne sont pas à l'abri de déconvenues e.g. rien du tout à montrer à 16h45.

3. gestion de versions

Utilisez github pour échanger le code entre vous. Mais surtout utilisez github pour pouvoir revenir rapidement à une version qui "fonctionne" même si ce n'est pas la meilleure elle sera toujours mieux que rien.

4. finalité

Cet exercice est une expérience de programmation en groupe en temps limité avec sujet imposé. C'est très formateur. L'important est de vous donner les moyens, de vous impliquer, de jouer le jeu, d'être sérieux et de garder votre motivation tout au long de la journée.

En cas de démotivation: appelez l'enseignant de votre salle et racontez lui votre problème. Il est possible que le résultat ne soit pas à la hauteur de vos ambitions ou de votre niveau mais vous pourrez comprendre où et pourquoi ça a bloqué.

3 Parlons rendu

Nous sommes là pour répondre à vos questions et pour vous débloquer en cas de problème.

À partir de 15h30 dans chaque salle sera organisée un rendu en *lightning-talk*: vous avez un créneau de 3 minutes pour faire une démo de votre jeu, ou de ce qui fonctionne à ce moment là.