



ft_minirogue

Non, on ne tape pas un slime avec une hache

42 staff staff@42.fr

Résumé: Ce mini-projet va vous faire réaliser un petit jeu similaire, dans son aspect général, au célèbre jeu vidéo Rogue.

Table des matières

I	Préambule	2
II	Le projet	3
II.1	Rogue	3
II.2	Partie Obligatoire aka Keskejfé ?	4
II.3	Partie Bonus aka Keskejfémintnankjéfiniiii ?	4

Chapitre I

Préambule

Selon Wikipédia :

La méthode du canard en plastique consiste à expliquer méticuleusement le code source que l'on a écrit à un collègue, à un simple passant, ou même à un objet inanimé comme un canard en plastique. Le simple fait d'exprimer ses pensées à voix haute est censé aider à trouver les erreurs de programmation. Comme les réactions de l'interlocuteur ou son niveau de compréhension du problème n'ont aucune importance dans ce processus, on peut le remplacer par un canard en plastique.

Lorsque le programmeur ou l'analyste explique un bout de code ou un algorithme qui est incorrect, il peut en effet se rendre compte de son erreur. La méthode du canard en caoutchouc exploite la dissonance cognitive en confrontant le concepteur au fait que le code qu'il a écrit ne correspond pas au code qu'il avait l'intention d'écrire.

L'avantage du canard en plastique sur un interlocuteur humain est que sa capacité d'écoute et sa patience sont sans limites. Son temps est aussi moins précieux que celui d'un consultant ou d'un collègue. Le fait qu'il soit petit permet de le placer discrètement à côté d'un écran d'ordinateur.

N'oubliez pas votre canard en plastique pour ce projet.

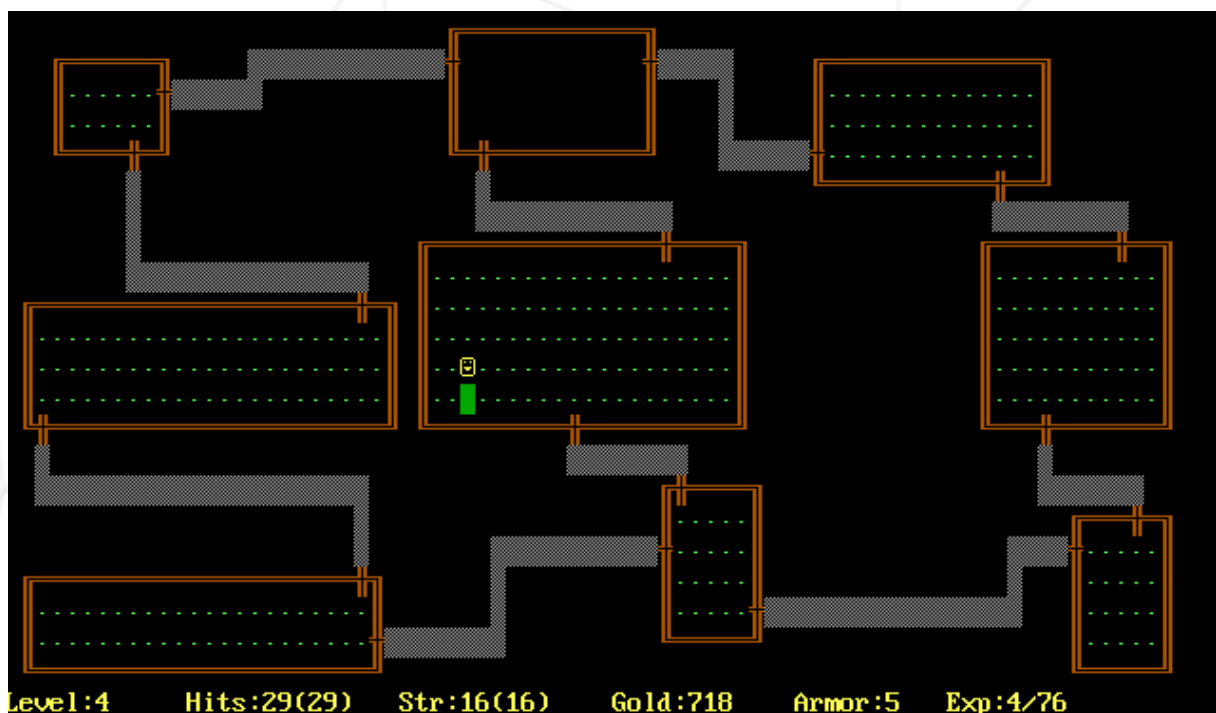
Chapitre II

Le projet

II.1 Rogue

[Rogue](#) est un jeu de parcours de donjon créés procéduralement, qui se joue sur un terminal. Si vous êtes inculte et ne connaissez pas encore, lisez la page Wikipédia.

Si vous voulez essayer rapidement le jeu (et vous devriez!), faites un tour [ici](#).



II.2 Partie Obligatoire aka Keskejfé ?

Le but de ce projet est très simple : Vous devez réaliser un mini-clone de **Rogue**.

Comme on n'est pas des bêtes, et que vous n'avez que très peu de temps, on ne vous en demande pas TROP. Voici la liste des choses obligatoires pour que l'on considère que votre rendu n'est pas boiteux :

- Votre jeu doit avoir une interface graphique basée sur le terminal uniquement. Libre à vous d'utiliser la librairie de votre choix (curses, ncurses, termcaps, ...)
- Votre rendu doit être en Python (2 ou 3, à votre convenance).
- Vous devez avoir au moins un niveau complet. Vous n'avez pas besoin de le générer aléatoirement à chaque lancement. Par niveau complet, on entend quelques rooms avec des passages cohérents.
- Il doit y avoir un personnage joueur (vous!), des monstres, et des trésors à ramasser.
- Votre joueur a des points de vie, qu'il perd quand un monstre le touche. À 0 points de vie, il meurt.
- Il doit être possible de déplacer le joueur, ramasser des trésors (qui vous donneront un bonus cohérent au choix), et frapper un monstre.
- Chaque action entreprise par le joueur (Déplacement, changement d'arme, attaque, nourriture, etc) fait avancer le temps d'un tour.
- Il va sans dire que votre jeu doit ressembler un minimum à **Rogue** ...

II.3 Partie Bonus aka Keskejfémintnankjéfiniiii ?

Si vous avez fait tout ça, et que votre projet est stable, pourquoi ne pas essayer d'en faire un vrai **Rogue-like** ?

- Découverte du niveau au fur et à mesure de l'avancement.
- Un inventaire : Armes, armures, objets de soin, etc ...
- Génération aléatoire de niveaux
- Plusieurs niveaux, avec des monstres plus durs, et des trésors plus cool!
- De la magie!
- Des classes de personnage!
- Et tout ce que vous aurez envie d'ajouter. Si vous avez le temps...