

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22
téléphone : 04 79 75 94 22
email : Pierre.Hyvernat@univ-savoie.fr
www : <http://www.lama.univ-savoie.fr/~hyvernat/>

Info608, TP

Sujet du projet

But final : créer un jeu d'aventures de type *roguelike*.

I – Présentation générale du projet

Les “roguelikes” sont des jeux d'aventure de la famille de Rogue, écrit en 1980. C'était la préhistoire, et les cartes graphiques 3D n'existaient pas... Rogue est un jeu d'aventures tour par tour où le héros explore un donjon. L'interaction avec le joueur passait seulement par le clavier (les souris n'existaient pas non-plus) et l'affichage ASCII (les systèmes d'exploitations n'avaient pas de couche graphique).

Une situation typique pouvait ressembler à

```
-----  
|...|      #####                               Nom : Merlin  
|...|      #          #                         PV : 100  
|.$..+#####          #                       points : 4242 $  
|...|      #          ---+---  
-----   #          |.....|  
           #          |.!...|  
           #          |.....|  
           #          |..@..|  
-----   #          |.....|  
|..|      #####+.D..|  
|<.+###   #          |.....|  
----- #   #          |.?...|  
           #####          -----
```

Ici, le héros (le petit '@') se prépare à attaquer un dragon (le 'D').

Ou alors, il est en train de s'enfuir.

Ou alors, il est en train de discuter.

...

Seul le contexte de jeu pourrait nous donner ce genre de détails.

Les autres symboles traditionnellement utilisés sont :

- les lettres pour les monstres
- '.' pour une parcelle de sol éclairé (les pièces)
- '#' pour les couloirs (éclairés ou non)
- '>' et '<' pour les escaliers passants d'un niveau à l'autre
- '+' pour les portes
- '-' et '|' pour les murs
- '\$', '?', '!', ... pour les objets (pièces d'or, parchemins, potions magiques etc.)

Les commandes de base (Nethack) sont

- flèches : pour se déplacer (ou, si le pavé numérique n'existe pas, les lettres h, j, k et l pour les directions gauche, bas, haut et droite)
- 'i' pour obtenir le liste de ses objets ("inventory")
- ', ' pour ramasser un objet
- 'd' pour déposer un objet ("drop")
- 'q' boire une potion ("quaff")
- 'r' pour lire un parchemin ("read")
- ...

Vous allez programmer un jeu avec des spécificités similaires. Le thème ne doit pas forcément être celui du Rogue original... Tant que vous respectez les consignes (voir plus loin), vous pouvez modifier le monde, ou même le concept du jeu. Voici par exemple quelques possibilités :

- jeu d'aventure classique
- Némoto doit éviter les requins et retrouver son papa
- Marie a perdu son porte monnaie dans le grand centre commercial
- ...

Le côté minimaliste de ce genre de jeux laisse la part belle à l'imagination. Utilisez cet avantage...

II – Détails administratifs

Le but de ce cours est de "mimer" une situation simplifiée d'interaction client / programmeurs. Vous êtes les programmeurs et je suis le client.

Votre mission, si vous l'acceptez* se divise en deux parties.

Dans un premier temps, vous devrez fournir (dans environ un mois) un document en réponse au cahier des charges (voir plus bas) pour expliquer ce que vous proposez.

Une fois ce document validé par le client, le contrat est formalisé et vous devrez, en fin de semestre, fournir un produit fini (le programme) correspondant autant que possible au cahier des charges. Au long du semestre, il vous faudra également fournir plusieurs types de documents (spécifications de votre programme, documentation etc.) Les détails vous seront donnés en cours.

Pour faire tout ça, il faudra vous répartir en groupes de 4, 5 ou 6 personnes.

Votre évaluation finale prendra en compte

- la réponse au cahier des charges (description et spécification initiales du programme, planning des réalisations etc.),
- le rapport final (spécification finale, documentation, description du cycle de développement, différences entre le produit et les spécifications initiales etc.)
- la soutenance orale du projet devant le client et vos collègues, avec une démonstration
- le programme en lui même

Notez que le programme ne compte que pour une toute petite partie de la note. Mieux vaut un petit programme bien documenté qui suit le cahier des charges plutôt qu'un programme génial qui n'a rien à voir avec le produit annoncé !

Veillez aussi à essayer de faire une proposition réaliste : ni trop simple (le client ne l'acceptera pas) ni trop ambitieuse (vous risquez de ne pas pouvoir tout faire et le client refusera de payer).

Petits détails importants...

Pendant qu'on parle des documents à rendre, tous vos documents devront être, par ordre de préférence, soit au format pdf, soit au format texte soit au format Word/OpenOffice. Les autres

* et si vous ne l'acceptez pas, j'ai des moyens de pression !

formats ne seront pas accepté! Tous vos documents devront également comporter la liste des tous les membres du groupe ainsi que votre filière (TIC-TC, TIC-info ou MI-info). Tous les documents ne respectant pas ces consignes risquent d'être jetés sans être lus.

Enfin, si vous m'envoyez des emails, débrouillez-vous pour mettre "info608" dans le sujet. Si le sujet ne comporte pas "info608", le mail n'aura aucun caractère officiel, et je ne le lirais que si j'ai envie!

III – Sujet (cahier des charges)

Nous voudrions un petit jeu simple avec les spécificités suivantes :

- style aventure / réflexion
- un seul joueur (pas de synchronisation)
- tour par tour (pas de temps réel)
- interface utilisateur réduite (pour pouvoir y jouer sur un PDA, ou pourquoi pas sur un téléphone portable)
 - . l'utilisateur ne dispose que d'un clavier
 - . le hardware ne peut pas afficher beaucoup plus que des caractères alphanumériques
- une grande part d'aléatoire (même si on arrive à la fin, on peut y rejouer sans s'embêter)
 - . la création des niveaux
 - . la création et le placement des PNJ (personnages non joueurs) et des objets
- lorsque le joueur quitte un niveau, ce niveau est supprimé : si le joueur revient sur ce niveau, il est recréé (comme dans Moria / Angband)
- une petite famille d'objets plus ou moins utiles pour aider la tâche du joueur. Ces objets doivent avoir un impact sur le déroulement du jeu. (Le héros devient plus rapide / moins rapide etc.)
- un système de description en langue naturelle de ce qui se passe dans le donjon. ("Vous rentrez dans un nouveau niveau." ; "Le dragon vous attaque sans succès." ; ...)
- on peut arrêter une partie et la reprendre plus tard (dans la plupart des roguelike traditionnels, on ne peut pas sauver sa partie : la mort est irrémédiable...)
- un classement final de tous les joueurs où le score d'un joueur dépend de sa partie (nombre de pièces d'or ramassées, d'objets trouvés, ...)
- une documentation (online et offline) des différents aspects du jeu
- le jeu doit pouvoir être classé dans la famille des "roguelike".

Pour ce projet, vous devrez implanter le programme sur un ordinateur de type PC avec un système d'exploitation Unix ou Windows.

Pour les ambitieux, vous pourrez rajouter (une fois que tout ça est fait) les choses suivantes

- petite couche graphique pour pouvoir, si l'utilisateur le veut, remplacer l'affichage ASCII par des dessins simples (voir Nethack pour un exemple)
- moteur d'intelligence artificielle simple pour les PNJ (les monstres ne se déplacent pas aléatoirement)
- un système de niveaux permanents. (Lorsque le joueur quitte un niveau, celui-ci est sauvegardé : le joueur peut revenir et retrouver les trucs qu'il a laissés là...)
- un système d'attributs (intelligence, force, ...) influant sur le jeu ; un système de compétences (manuel, sportif, ...) ; ...
- supprimer les bugs
- rendre le jeu intéressant pour le public. (Les autres gens doivent avoir envie de jouer à votre jeu parcequ'il est bien!)
- ...

IV – Détails techniques

Vous allez donc écrire un programme. Un des premiers choix à faire sera celui du langage de programmation.

Pour les MI-info, nous vous facilitons la tâche en imposant le langage : ADA.

Pour les TIC-info et TIC-TC, le langage est “libre”. Vous devrez utiliser un langage impératif séquentiel connu. (C, C++, C#, objective C, Java, ADA, Pascal, ...) Le choix du langage Intercal est possible et de part sa difficulté multipliera votre note par deux ! (Mais on soustraira ensuite 10 points à votre note finale à cause de votre esprit pervers.)

Si vous voulez utiliser un langage interprété (Perl, Ruby, Python, ...), demandez-nous avant de commencer, et préparez des arguments pour justifier le choix du langage.

Les langage fonctionnels (Haskell, LISP, Caml, Scheme) et logiques (Prolog) ne sont pas autorisés.

Si vous avez un doute, demandez-nous avant de commencer...

Pour la réalisation du programme, il vous faudra probablement utiliser des bibliothèques. C’est possible à condition que celles-ci soient un minimum standard. (Par exemple, vous ne pouvez pas utiliser la bibliothèque “faire_un_roguelike_en_deux_heures” que vous avez trouvé sur Internet...)

Encore une fois, si vous avez des doutes lors de la réalisation, demandez-nous.

Conseils

Le style de programmation doit être le meilleur possible. Je ne relirais pas tout votre code, mais je regarderais un peu comment vous avez fait. Si c’est illisible, vous risquez de me mettre de mauvaise humeur. (Ce qui a tendance à se répercuter sur la notation...)

Donnez-vous des consignes de programmation pour que le code écrit par plusieurs personnes soit cohérent.

Faites des sauvegardes à plusieurs endroits (“mon disque dur est cassé” n’est pas une excuse pour ne pas rendre le projet !) et utilisez un système de gestion de versions (Darcs, svn, cvs, ...) pour pouvoir revenir en arrière facilement.

V – Références

Voici quelques liens qui pourraient vous être utiles :

- article wikipedia français : <http://fr.wikipedia.org/wiki/Roguelike>
- FAQ : <http://www.roguelikedevelopment.org/development/FAQ.php>
- usenet : rec.games.roguelike.development
- un roguelike pour téléphones portables :
<http://www.roguelikedevelopment.org/dweller/index.php>
- un roguelike minimaliste (en C) : <http://www.adom.de/misc/qhack.php3>
- source de plusieurs vieux roguelike :
<http://roguelikedevelopment.org/archive/#SOURCECODE>

Pour d’autres références, Google is your friend...