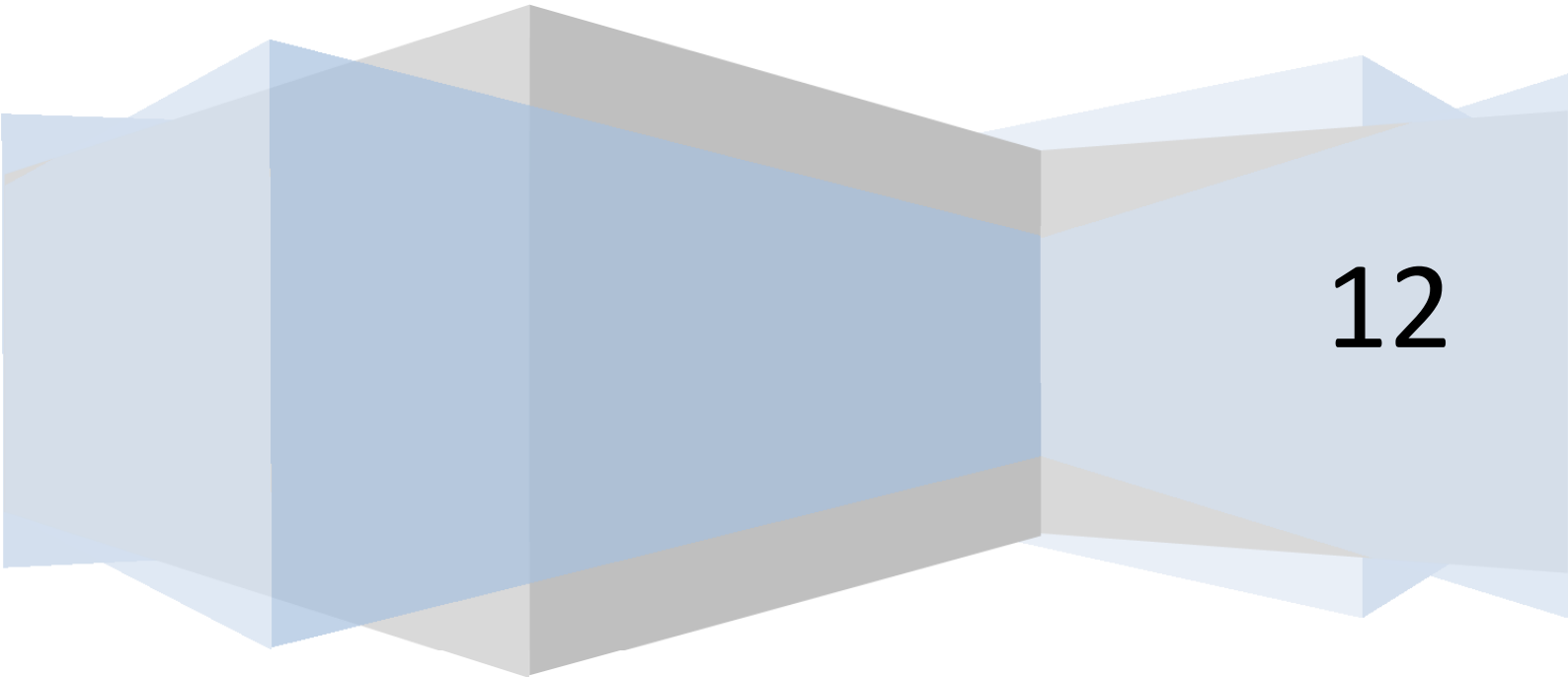


EFREI

Projet Unix L2

tpunix@efrei.fr



12

Table des matières

1	Descriptif du projet	3
1.1	Règles du jeu	3
1.2	Descriptif technique	4
1.2.1	Principe	4
1.2.2	Démarrage	4
1.2.3	Structure de données	4
1.2.4	IHM	5
1.2.5	Les pièces.....	5
1.2.6	Le contenu d'une pièce	6
1.2.7	Passages secrets	7
1.3	Optionnel.....	8
2	Rapport & script	9

1 Descriptif du projet

Le projet consiste à réaliser, à l'aide de scripts shell, un jeu de type jeu d'aventure.

Le joueur aura pour objectif de parcourir un labyrinthe afin d'en trouver la sortie. Le joueur sera amené à rencontrer des ennemis, répondre à des énigmes et à découvrir des objets.

L'interface sera entièrement textuelle, les actions et l'environnement seront décrits.

L'interaction avec le jeu se fera à l'aide des touches du clavier.

Le labyrinthe devra comporter au moins 20 pièces. La structure du labyrinthe est fixe, mais son contenu varie à chaque partie.

1.1 Règles du jeu

Le but du jeu est de sortir du labyrinthe.

Les armes et potions peuvent aider le joueur à réaliser cet objectif.

Une pièce peut contenir une énigme à laquelle le joueur doit répondre pour pouvoir sortir de la pièce, et aller dans la pièce suivante.

Les ennemis tenteront de stopper la progression du joueur.

Un joueur est décrit par les éléments suivants (stockés dans un fichier) :

1. Son nom
2. Sa description
3. Son nombre de points de vie maximum
4. Son nombre de points de vie courant
5. L'arme qu'il transporte, ainsi que les points d'attaque correspondants
6. L'endroit où le héros se trouve (`pwd`)

Lors d'un début de partie le joueur devra saisir le nom de son héros et sa description, choisir une arme (choix multiple de quelques armes basiques), et le nombre de point de vie maximum sera déterminé aléatoirement (base plus tirage aléatoire).

En cours de partie si le héros ramasse une arme, elle viendra remplacer dans le fichier l'arme actuelle du héros. De même, les potions augmentent le nombre de point de vie courant, celui-ci ne peut bien entendu pas dépasser le nombre de point de vie maximum.

Le héros meurt lorsque son nombre de point de vie est égal à zéro (attention à la gestion des nombres négatifs)

Le joueur ne pourra franchir une pièce donnée s'il n'a pas triomphé des ennemis se trouvant dans la pièce.

Lors d'un combat :

1. Le joueur a l'initiative du combat (ce qui lui permet de ramasser les potions et armes de la pièce avant le début du combat)
2. Le joueur frappe toujours en premier
3. Si la pièce contient plusieurs ennemis, les combats sont séquentiels (un ennemi puis l'autre)
4. Le combat s'arrête quand tous les ennemis sont défaits ou que le héros est mort
5. Les deux adversaires tentent de frapper chacun leur tour
6. Le joueur ne peut pas prendre de potion pendant le combat
7. Le joueur ne peut pas changer d'arme pendant le combat
8. Les points de dégâts infligés – c'est-à-dire les points de vie perdus à l'adversaire – correspondent aux points d'attaque caractéristiques de l'arme.
9. Un tirage aléatoire est réalisé pour savoir si le coup porte ou non : 50% de chance pour le héros, 33% pour les adversaires.

Le joueur ne pourra franchir une pièce donnée s'il n'a pas répondu aux énigmes se trouvant dans la pièce.

1.2 Descriptif technique

1.2.1 Principe

La progression du héros est modélisée par le parcours des répertoires représentant le labyrinthe. Ses rencontres et trouvailles sont modélisées par les fichiers contenus dans ces répertoires. Par exemple si le joueur se trouve dans la pièce numéro 5, le script sera positionné dans le répertoire `piece5` de l'arborescence modélisant le labyrinthe.

1.2.2 Démarrage

Le script du jeu devra se nommer `labyrinthe.sh`.
Ce script devra fonctionner sur les machines de l'EFREI, et sera testé sur cet environnement.

1.2.3 Structure de données

Le labyrinthe sera modélisé sous forme de fichiers plats et de répertoires.

La structure des répertoires représente le labyrinthe.

Le labyrinthe est créé par les étudiants et sera la base du projet

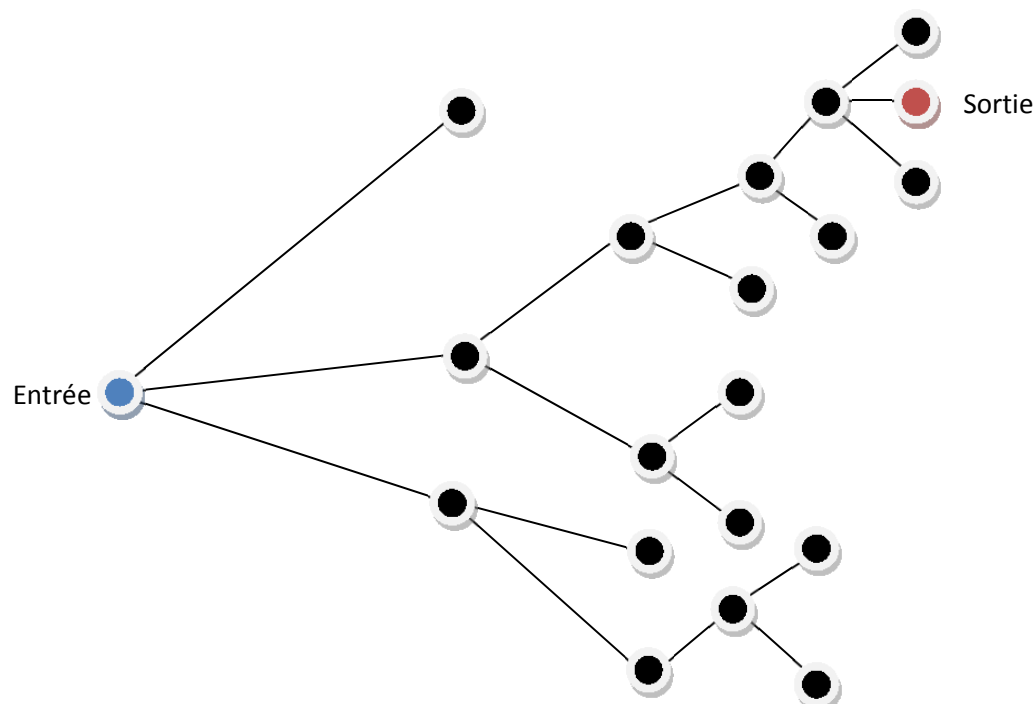
La structure du labyrinthe a les caractéristiques suivantes :

Nombre de pièces minimum : 20

Profondeur d'arbre minimum : 5

Nombre de monstre minimum : 8

Nombre d'énigme minimum : 8



1.2.4 IHM

Le jeu étant en mode texte, il faudra créer une interface permettant de gérer les déplacements du héros, ses combats, sa recherche de passage secret, etc. cette interface sera à base de question réponse et de commande simple en mode texte.

Un menu contenant les options du jeu permettra de :

- Lancer une nouvelle partie
 - Cette option déclenche la construction des nouveaux objets, énigmes et monstres ainsi que la pièce de sortie.
 - La création du personnage sera également faite lors de la sélection de cette option
- Continuer une partie sauvegardée
 - Cette option permet de rappeler une ancienne partie (décompression du tar.gz et remise en place de la structure et du lieu où se trouve le héros)
- Sauvegarder la partie en cours
 - Le contexte du jeu devra être sauvegardé sous forme d'un tar.gz du répertoire du labyrinthe en cours. Il faudra penser à sauver le lieu où se trouve le héros (`pwd`)
 - Lors de la sauvegarde, le labyrinthe courant est supprimé
- Voir la feuille de personnage
 - Donne les informations sur le personnage, sont niveau de vie actuel, son arme, etc.
- Quitter
 - Quittera le jeu

Durant la partie une touche spécifique permettra d'afficher le menu décrit ci-dessus.

1.2.5 Les pièces

Chaque pièce du labyrinthe est un répertoire.

Le répertoire symbolisant la première pièce se nommera « entree », la pièce symbolisant la sortie se nommera « sortie ».

Voici un exemple :

```
[skyce@weatherwax projet2012]$ tree
```

```
├── entree
│   ├── piece1
│   ├── piece2
│   │   ├── piece3
│   │   │   ├── piece5
│   │   └── sortie
```

```
6 directories, 0 files
```

Dans cet exemple, 6 pièces ont été représentées : entrée, pièce 1, pièce 2, pièce 3, pièce 4, et enfin pièce 5.

Pour se rendre de la pièce 1 à la pièce 5, le joueur devra obligatoirement passer par : l'entrée, puis la pièce 2, puis la pièce 3, et enfin la pièce 5.

La pièce « entree » est une pièce comme les autres. La pièce « sortie » change de place à chaque nouvelle partie.

1.2.6 Le contenu d'une pièce

Tout le contenu d'une pièce (objets, ennemis, énigmes, description...) sera représenté par des fichiers plats contenus dans le répertoire symbolisant la pièce.

1.2.6.1 *Ordre des actions réalisables dans une pièce*

- Gestion des armes
- Gestion des potions
- Gestion des combats
- Gestion des énigmes
- Gestion des déplacements (Passage secret et porte)

1.2.6.2 *Source de données*

Les objets, les descriptions des pièces, etc. sont contenus dans une bibliothèque. Cette bibliothèque sera utilisée pour créer aléatoirement les 5 fichiers en début de partie. Il faudra donc prévoir des fichiers plats pour chaque type de contenu avec les définitions dans lequel il faudra piocher lors de la création du labyrinthe en début de partie.

Chaque pièce devra contenir 5 fichiers représentant son contenu (les pièces dans lesquelles il n'y a pas d'ennemis contiendront un fichier « ennemis » vide, cela vaut aussi pour les fichiers « armes », les fichiers « potions » et les fichiers « énigmes »).

1.2.6.3 *Fichier « description »*

Contient la description de la pièce. Le fichier devra comporter une ou plusieurs lignes de description.

1.2.6.4 *Fichier « potions »*

Contient la liste des potions. Chaque ligne décrit une potion selon le formalisme suivant :

- <nom de la potion>:<description de la potion>:<nombre de point de vie rendue par la potion>

Les potions ne peuvent pas soustraire des points de vie au joueur, et ne peuvent pas être transportées.

1.2.6.5 *Fichier « armes »*

Contient la liste des armes. Chaque ligne décrit une arme selon le formalisme suivant :

- <nom unique de l'arme>:<description de l'arme>:<nombre de point d'attaque maximum>

1.2.6.6 *Fichier « ennemis »*

Contient la liste des ennemis. Chaque ligne décrit un ennemi selon le formalisme suivant :

- <nom de l'ennemi>:<description de l'ennemi>:<nombre de point de vie>:<nombre de point de dégâts maximum>

1.2.6.7 *Fichier « énigmes »*

Contient une énigme à laquelle le joueur doit répondre pour pouvoir passer à la pièce suivante. La réponse doit être sous forme de choix multiple. La bonne réponse devra changer les droits du répertoire (symbolisant la pièce suivante), permettant au héros de s'y rendre.

- <nom unique de l'énigme>:<question>:<choix 1>:<choix 2>:<réponse>

Les énigmes ne sont pas mortelles, une mauvaise réponse n'arrête pas le jeu.

Voici un exemple :

```
[skyce@weatherwax piece1]$ ls -l
```

```
total 20
-rw-r--r-- 1 skyce users 45 Dec 18 15:21 armes
-rw-r--r-- 1 skyce users 53 Dec 18 15:08 description
-rw-r--r-- 1 skyce users 69 Dec 18 16:11 enigme
-rw-r--r-- 1 skyce users 71 Dec 18 15:10 ennemis
-rw-r--r-- 1 skyce users 26 Dec 18 15:21 potions
```

Et voici le contenu de chacun des fichiers:

```
[skyce@weatherwax piece1]$ for i in *; do ls -l $i; cat $i; done

-rw-r--r-- 1 skyce users 45 Dec 18 15:21 armes
Epée du chevalier:Imposante et tranchante:2

-rw-r--r-- 1 skyce users 53 Dec 18 15:08 description
La pièce est sombre, les murs sont en pierres nues.

-rw-r--r-- 1 skyce users 69 Dec 18 16:11 enigme
rick:qu'est ce qu'il y connaît aux femmes Rick Hunter:Rien:Tout:Rien

-rw-r--r-- 1 skyce users 71 Dec 18 15:10 ennemis
Minotaure:Grand et cornu:10:2
Gobelin:Petit, vert et braillard:2:1

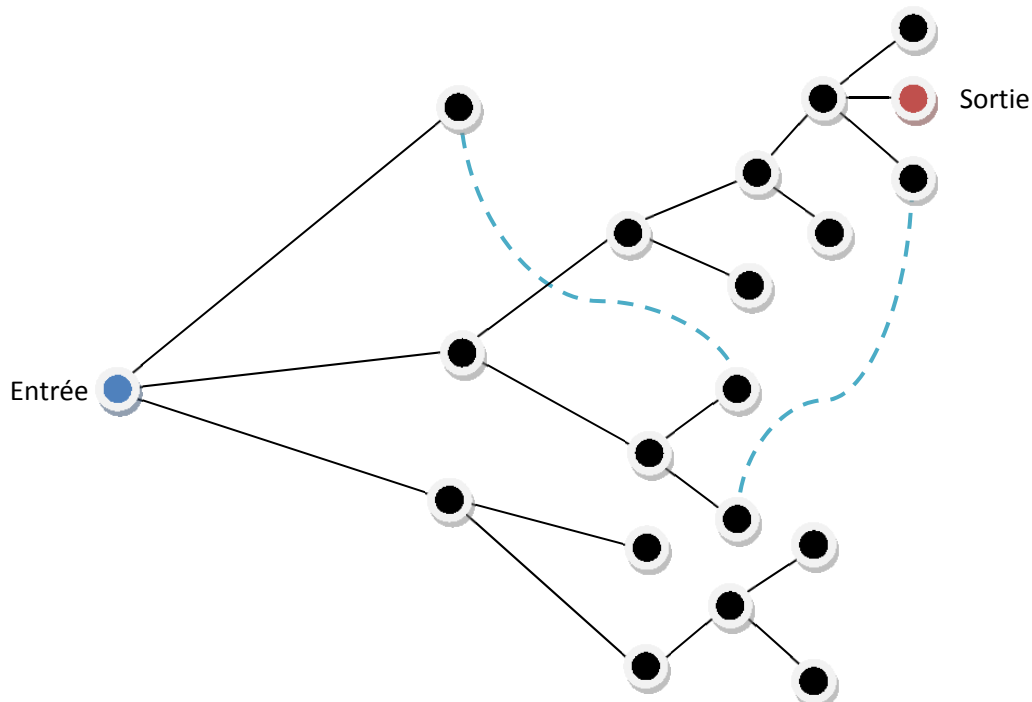
-rw-r--r-- 1 skyce users 26 Dec 18 15:21 potions
Potion:verte et gluante:2
```

1.2.7 Passages secrets

Le labyrinthe devra contenir des passages secrets, ceux-ci seront modélisés par des liens symboliques.

Ainsi sur le schéma ci-dessous, sont représentés en ligne pointillée bleue deux passages secrets.

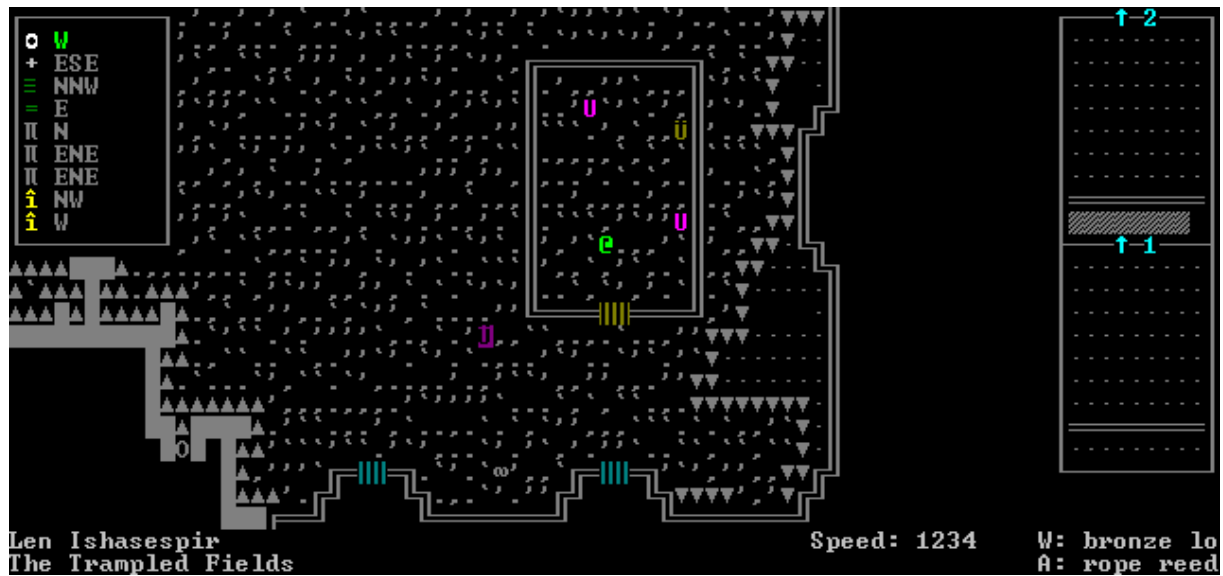
Le héros devra décider s'il recherche un passage secret dans une pièce ou non (option dans le menu d'exploration). La détection de la présence d'un passage secret devra être faite en détectant le lien symbolique.



1.3 Optionnel

Il pourra être mis en place une représentation graphique de la pièce (en ascii), afin d'agrémenter le jeu.

Pour inspiration un screenshot du jeu Dwarf Fortress (<http://www.bay12games.com/dwarves/>).



2 Rapport & script

Le rapport sera à rendre au format PDF, la page de garde devra contenir le nom des binômes et le groupe.

Après une rapide introduction, vous donnerez les informations relatives à l'interprétation du sujet et les choix que vous avez effectués.

Vous effectuerez ensuite une analyse de votre travail, vous fondant sur des exemples et des screenshots.

Les scripts devront être inclus en annexe.

Le rapport devra contenir une vingtaine de page.

Les scripts devront être commentés, vous les transmettez par mail à tpunix@efrei.fr archivés, le nom du fichier devra respecter la nomenclature suivant : *nombinome1_nombinome2_groupe.tar.gz*.

Le mail devra être expédié de votre messagerie EFREI.

Le script `labyrinthe.sh` doit être prêt à être exécuté sur les machines de l'EFREI après décompression de l'archive.