Outils de développement logiciel

Projet - Jeu de Rôle

Encadrant de TP : SIBILLE Emmanuel

Contents

1 Introduction

Ce projet consiste en la réalisation d'un jeu de rôle en language C. Après avoir précisé certains choix qui ont été réalisés au niveau des règles du jeu qui ne couvraient pas toutes les configurations possibles, on abordera les différentes étapes de réalisation du projet, du choix des structures de données jusqu'à la réalisation des tests finaux.

Un trace d'exécution vidéo est disponible à l'adresse suivante http://bit.ly/TFnCUS

2 Remarques préliminaires

Certains choix on du être réalisés au niveau des règles du jeu.

(1) Lorsqu'un hero arrive sur une case où se trouvent déjà deux heros d'un même joueur, il combat le dernier arrivé sur la case uniquement.

(2) Lorsqu'un hero arrive sur une case question, il doit d'abord répondre à la question avant d'affronter le hero qui s'y trouve le cas échéant.

(3) L'invocation du magicien est proposée juste avant que ce dernier ne se déplace vers une autre case.

(4) Le Boss de fin est à expérience fixe et son énergie n'est pas réinitialisée à chaque combat.

3 Organisation du projet

3.1 Structures de données

On définit en premier lieu les structure de données qui constituent la base du travail. Elle permettent en outre d'avoir des algorithmes plus efficaces si elles sont judicieusement choisies.

Les strutures des données sont détaillée en annexe (2).

Pour plus de précisions, voir la documentation réalisée par doxygen et disponible à l'adresse suivante :

3.2 Algorithmes

Toutes les fonctions des différents modules de jeu qui correspondent aux déplacement et aux actions des différents heros, décris en énoncé, se font en temps constant et ne posent donc pas problème.

3.3 Modules

Ce projet est organisé en différents dossier suivant la forme classique d'une distribution logicielle libre. Dans le dossier source, les fichiers sources sont répartis suivant différents modules. Les dépendances entre ces modules sont précisée dans l'annexe (1).

Question Interprétateur de question. Renvoie un pointeur sur un élément de type Question en l'initialisant à partir d'un descripteur de fichier.

Memoire Gestion de la memoire. Alloue et libère les différentes structures.

Primitives Fonctions de base du jeu modifiant directement les structures.

Jeu Fonction évoluées de jeu, obtenues par composition des fonctions élémentaires du module `Primitives`.

Intfc Interface communiquant avec le jeu. Elle gère les entrées, qu'elle transmet au module de jeu et les sorties, qu'elle affiche. *Remarque :* Pour alléger l'affichage, on représente (exemple) `(T1,a)`pour une attaque d'un Troll et `(M2,-10)`la blessure du magicien 2 qui perd 10 points d'énergie.

Main Programme principal

4 Tests d'éxécution

Pour chaque module `'module'`, on utilise un fichier `module-test.c` qui contient un main et qui teste successivement les fonctions du module.

Le `makefile` se trouvant dans le dossier `src` permet de générer les différents fichier `.o` des modules (les fichiers `.o`des tests également).

Le makefile principal permet de faire l'édition des liens permettant de tester un seul module dont le nom est dans la variable `TEST`. Pour cela, il supprime tous les fichiers objets correspondant à des tests différents de `$(TEST).o` puis réalise l'édition des liens.

Par exemple, pour tester le module `'primitives'`qui dépend du module `'memoire'`qui lui même dépend du module `'questions'`on éxécutera les commandes suivantes :

```
make module MODULES='memoire questions primitives'
make link_module TEST='primitives'
```

Tous les modules ont été testés pour quelques exemples et fonctionnent correctement.

5 Compilation et exécution du programme

Pour compiler le programme, il suffit de lancer la commande suivante :

```
make all
```

Elle permet de créer tous les fichier objets (et supprime les fichier objets de test) puis de faire l'édition des liens pour créer l'éxecutable final.

Pour lancer le programme qui se situe dans le dossier `bin`,il faut l'exécuter avec pour paramètre le nom du fichier de questions.

Un script permet de tout automatiser. Pour le lancer, taper :

```
sh ./projet.sh
```

6 Analyse d'exécution

6.1 Gprof

Gprof est un logiciel de profilage qui permet de générer (pour un éxecutable compilé avec l'option `-pg`) un profil d'utilisation des fonctions, où sont précisé le nombre d'appels de chaque fonction et leur temps d'éxécution. Il est très pratique pour savoir quelles sont les fonctions à optimiser en priorité.

Le profil plat généré par *Gprof*, pour une partie de 4 joueurs en mode automatique est disponible en annexe (3).

6.2 Valgrind

Un grand soin a été accordé à la gestion de la mémoire : le programme alloue exactement ce dont il a besoin et le libère en fin d'éxecution. Pour cela les structures sont allouées dynamiquement. Pour vérifier qu'il n'y a pas d'erreur d'écriture ou de lecture en mémoire, et surtout qu'il n'y a pas de fuite mémoire, on utilisera le logiciel *Valgrind*, qui permet de générer un profilage de la mémoire utilisée par le programme. Disponible en annexe, pour la même exécution que le profilage par *Gprof*.

7 Documentation

La documentation de ce projet a été générée automatiquement à partir des tags que l'on a remplit dans les fichiers d'en-tête. Elle est disponible à l'adresse suivante :

http://www.ecole.ensicaen.fr/ benjelloun/odl/projet/doc/index.html

8 Possibilités d'évolution / À faire

Un grand soin a été accordé à l'écriture du code et à la séparation des fonctions en différents modules pour faciliter éventuelle réutilisation du code à l'avenir.

(1) Par exemple, deux dossiers `moteur2d` et `intfc2d` figurent dans le dossier sources. Il correspondent à la version graphique de l'interface. Pour la créer, il suffit de reprendre le fichier `intfct.c` et de remplacer le corps de chaque fonction par son équivalent permettant un affichage graphique.

(2) De même, le dossier `ia` correspond à une intelligence artificielle communiquant avec l'interface pour répondre automatiquement aux questions. Une exécution d'un tour de jeu avec cette interface est disponible à l'adresse suivante : http://bit.ly/11n68TF. Toutefois, n'étant pas finalisée et comportement quelques bugs, elle n'apparaît pas dans ce projet.

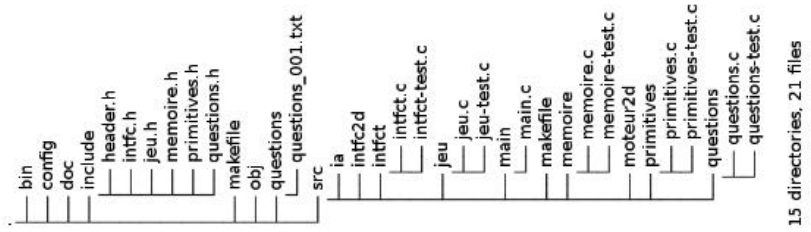
(3) Nombre de cellules variable, génération aléatoire automatique d'un plateau.

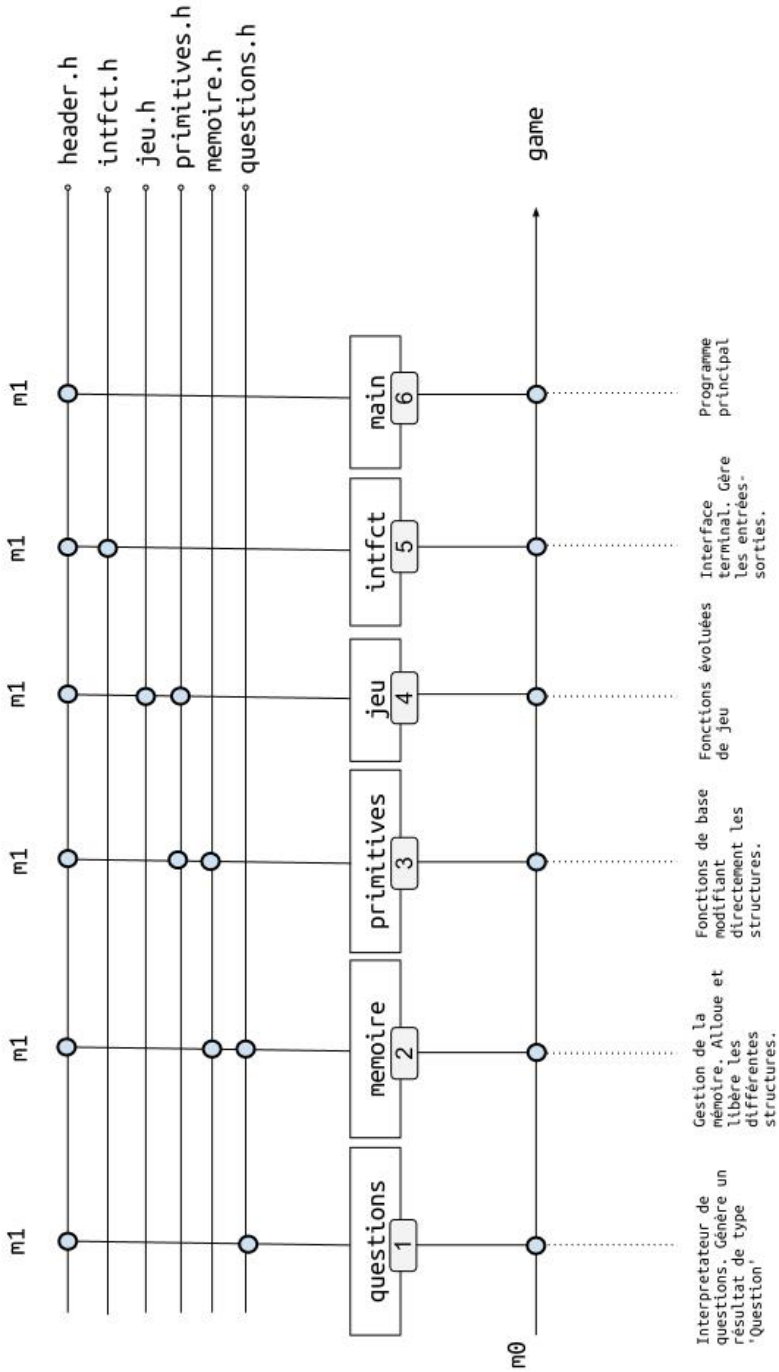
(4) Gestion des erreurs de formatage du fichier de questions. Si le fichier n'est pas bien formaté, le programme rencontre un erreur de segmentation. À corriger.

9 Bugs

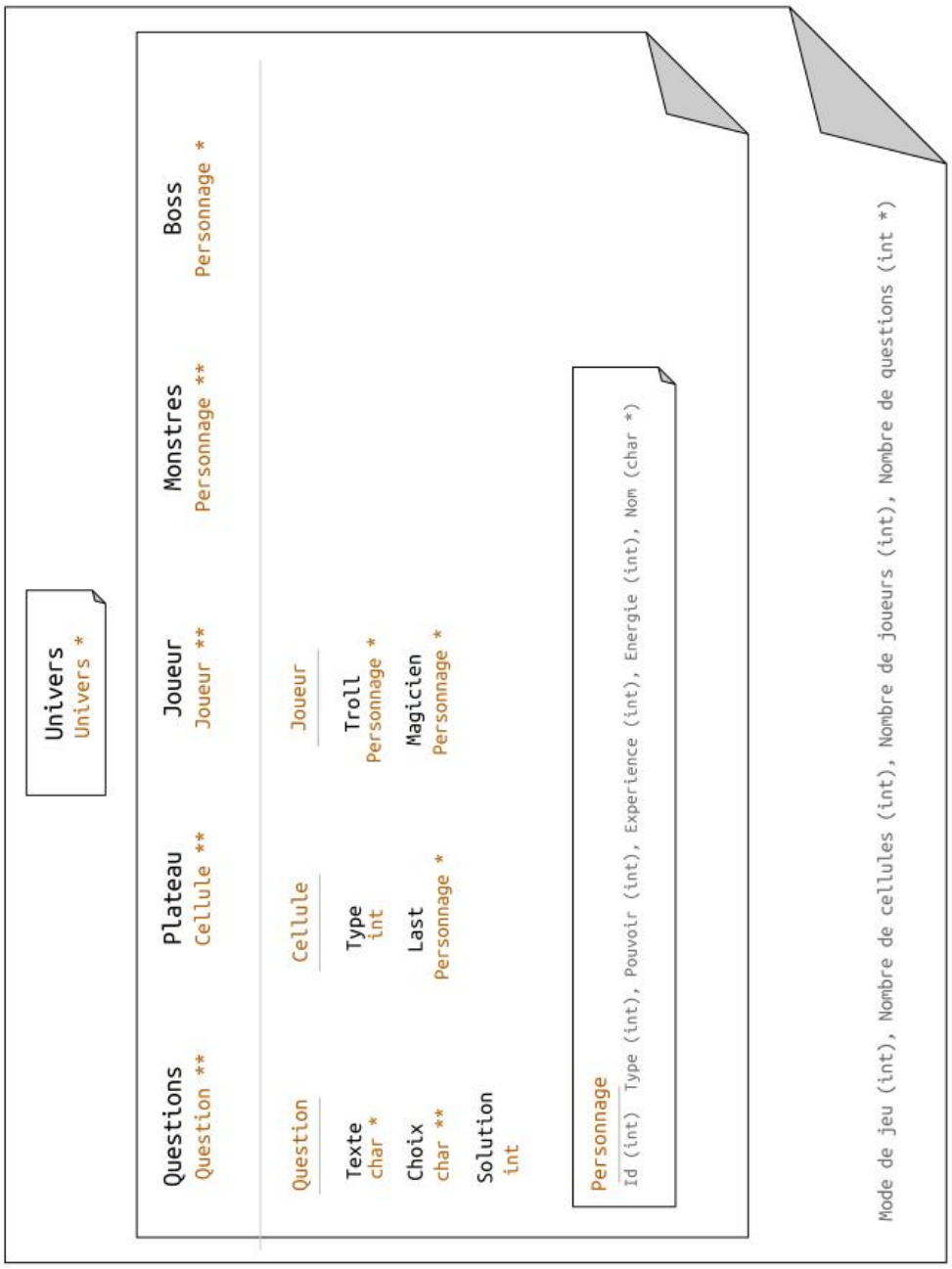
Pas de bug détecté dans des conditions de test normales. ☺

Annexe 1 - Diagramme d'interdépendance des modules





Annexe 2 - Structures de données



Annexe 3 - Profil plat - *Gprof*

```
Flat profile:

Each sample counts as 0.01 seconds.
 no time accumulated

  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ts/call  Ts/call  name
  0.00      0.00     0.00     1554     0.00     0.00  lancerDeAttaque
  0.00      0.00     0.00     1398     0.00     0.00  INTFC_blesser
  0.00      0.00     0.00     1398     0.00     0.00  blesser
  0.00      0.00     0.00     1394     0.00     0.00  INTFC_attaque
  0.00      0.00     0.00     1394     0.00     0.00  attaque
  0.00      0.00     0.00      205     0.00     0.00  deplacer
  0.00      0.00     0.00      151     0.00     0.00  INTFC_deplacer
  0.00      0.00     0.00      151     0.00     0.00  action
  0.00      0.00     0.00      151     0.00     0.00  deplacement
  0.00      0.00     0.00      151     0.00     0.00  lancerDeDeplacement
  0.00      0.00     0.00      115     0.00     0.00  INTFC_deconcentration
  0.00      0.00     0.00       72     0.00     0.00  soigner
  0.00      0.00     0.00       43     0.00     0.00  INTFC_gainExp
  0.00      0.00     0.00       43     0.00     0.00  gainExp
  0.00      0.00     0.00       42     0.00     0.00  estAllie
  0.00      0.00     0.00       38     0.00     0.00  reunion
  0.00      0.00     0.00       35     0.00     0.00  INTFC_poserQuestion
  0.00      0.00     0.00       35     0.00     0.00  INTFC_resultatQuestion
  0.00      0.00     0.00       35     0.00     0.00  poserQuestion
  0.00      0.00     0.00       33     0.00     0.00  INTFC_combat
  0.00      0.00     0.00       33     0.00     0.00  INTFC_victoire
  0.00      0.00     0.00       33     0.00     0.00  combat
  0.00      0.00     0.00       33     0.00     0.00  victoire
  0.00      0.00     0.00       31     0.00     0.00  allouerPersonnage
  0.00      0.00     0.00       31     0.00     0.00  libererPersonnage
  0.00      0.00     0.00       19     0.00     0.00  INTFC_afficherEtats
  0.00      0.00     0.00       19     0.00     0.00  INTFC_afficherPlateau
  0.00      0.00     0.00       19     0.00     0.00  INTFC_tourDeJeu
  0.00      0.00     0.00       19     0.00     0.00  tourDeJeu
  0.00      0.00     0.00       14     0.00     0.00  INTFC_demanderInvocation
  0.00      0.00     0.00        8     0.00     0.00  invocation
  0.00      0.00     0.00        7     0.00     0.00  INTFC_demanderMassue
  0.00      0.00     0.00        4     0.00     0.00  INTFC_combatFinal
  0.00      0.00     0.00        4     0.00     0.00  INTFC_invocation
  0.00      0.00     0.00        4     0.00     0.00  INTFC_issueCombatFinal
  0.00      0.00     0.00        4     0.00     0.00  INTFC_massue
  0.00      0.00     0.00        4     0.00     0.00  combatFinal
  0.00      0.00     0.00        4     0.00     0.00  interpreterLigne
  0.00      0.00     0.00        4     0.00     0.00  libererQuestion
  0.00      0.00     0.00        4     0.00     0.00  massue
  0.00      0.00     0.00        3     0.00     0.00  defaiteJoueur
  0.00      0.00     0.00        1     0.00     0.00  INTFC_ChoixModeDeJeu
  0.00      0.00     0.00        1     0.00     0.00  INTFC_ChoixNbJoueurs
  0.00      0.00     0.00        1     0.00     0.00  allouerJoueurs
  0.00      0.00     0.00        1     0.00     0.00  allouerMonstres
  0.00      0.00     0.00        1     0.00     0.00  allouerPlateau
  0.00      0.00     0.00        1     0.00     0.00  allouerUnivers
  0.00      0.00     0.00        1     0.00     0.00  chargerQuestions
  0.00      0.00     0.00        1     0.00     0.00  libererJoueurs
  0.00      0.00     0.00        1     0.00     0.00  libererMonstres
  0.00      0.00     0.00        1     0.00     0.00  libererPlateau
  0.00      0.00     0.00        1     0.00     0.00  libererQuestions
```

```
  0.00      0.00     0.00        1     0.00     0.00  libererUnivers

 %         the percentage of the total running time of the
time       program used by this function.

cumulative a running sum of the number of seconds accounted
 seconds   for by this function and those listed above it.

 self      the number of seconds accounted for by this
seconds    function alone.  This is the major sort for this
           listing.

calls      the number of times this function was invoked, if
           this function is profiled, else blank.

 self      the average number of milliseconds spent in this
ms/call    function per call, if this function is profiled,
   else blank.

 total     the average number of milliseconds spent in this
ms/call    function and its descendents per call, if this
   function is profiled, else blank.

name       the name of the function.  This is the minor sort
           for this listing. The index shows the location of
   the function in the gprof listing. If the index is
   in parenthesis it shows where it would appear in
   the gprof listing if it were to be printed.
```

Annexe 4 - Rapport *Valgrind* avec l'option `--leak-check=full`

```
==32403==
==32403== HEAP SUMMARY:
==32403==     in use at exit: 15,492 bytes in 2 blocks
==32403==   total heap usage: 124 allocs, 122 frees, 18,799 bytes allocated
==32403==
==32403== LEAK SUMMARY:
==32403==    definitely lost: 0 bytes in 0 blocks
==32403==    indirectly lost: 0 bytes in 0 blocks
==32403==      possibly lost: 0 bytes in 0 blocks
==32403==    still reachable: 15,492 bytes in 2 blocks
==32403==         suppressed: 0 bytes in 0 blocks
==32403== Rerun with --leak-check=full to see details of leaked memory
==32403==
==32403== For counts of detected and suppressed errors, rerun with: -v
==32403== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 2 from 2)
Profiling timer expired
```