Compte Rendu :
Groupe :

Tests de couverture, analyse statique de code et profilage

2 gcov et cmake :

les parties du code des fonctions qui ne fonctionnent pas

le fichier game.c
void add_mirror_ext(game game, content mirror_type, int col, int line){

```
#####:  304:   if (game==NULL){
#####:  305:     fprintf(stderr, "Invalid game pointer sent to add_mirror_ext!\n");
#####:  306:     exit(EXIT_FAILURE);
   -:  307:   }
#####:  308:   if (col>=game_width(game) ||col<0){
#####:  309:     fprintf(stderr, "Invalid integer col sent to add_mirror_ext!\n");
#####:  310:     exit(EXIT_FAILURE);
   -:  311:   }
#####:  312:   if (line>=game_height(game) || line<0){
#####:  313:     fprintf(stderr, "Invalid integer line sent to add_mirror_ext!\n");
#####:  314:     exit(EXIT_FAILURE);
   -:  315:   }
#####:  316:   game->board[game_width(game)*line+col] = mirror_type;
#####:  317: }
```

dans set_requered_nb_monster on a ça aussi :
  game→required_nb_spirits=value;

la fonction copy_game en entier du fait aussi qu'on ne l'a jamais utilisé :

game copy_game (cgame g_src){

```
#####:  388:   if (g_src==NULL) {
#####:  389:     exit(EXIT_FAILURE);
   -:  390:   }
#####:  391:   game g= new_game_ext(g_src->width,g_src->height);
#####:  392:   g->required_nb_zombies= g_src->required_nb_zombies;
#####:  393:   g->required_nb_ghosts= g_src->required_nb_ghosts;
#####:  394:   g->required_nb_vampires=g_src->required_nb_vampires;
#####:  395:   g->required_nb_spirits=g_src->required_nb_spirits;
#####:  396:   g->height=game_height(g_src);
#####:  397:   g->width=game_width(g_src);
#####:  398:   for (int i=0; i<g->width*g->height; i++){
#####:  399:     g->board[i] = g_src->board[i];
   -:  400:   }
#####:  401:   for (int i=0; i<NB_DIR; i++){
#####:  402:     if (i<2){
```

```
#####:  403:        for (int j=0;j<game_height(g); j++)
#####:  404:        g->required_nb_seen[i][j]=g_src->required_nb_seen[i][j];
    -:  405:    }
    -:  406:    else{
#####:  407:        for (int j=0;j<game_height(g); j++)
#####:  408:        g->required_nb_seen[i][j]=g_src->required_nb_seen[i][j];
    -:  409:    }
    -:  410:  }
#####:  411:   return g;
    -:  412: }
```

la fonction restart_game :

```
void restart_game(game g){
#####:  592: if(g==NULL){
#####:  593:   fprintf(stderr, "Invalid argument");
#####:  594:   exit(EXIT_FAILURE);
    -:  595: }
#####:  596: for (int i=0; i<game_width(g)*game_height(g); i++){
#####:  597:   if (g->board[i]==GHOST || g->board[i]==VAMPIRE || g-
>board[i]==ZOMBIE || g->board[i]==SPIRIT){
#####:  598:     g->board[i]=EMPTY;
    -:  599:   }
    -:  600: }
#####:  601:}
```

le fichier game_io.c

<results version="2"><cppcheck version="1.76.1"/><errors><error id="invalidPrintfArgType_sint" severity="warning" msg="%d in format string (no. 1) requires 'int' but the argument type is 'unsigned int'." verbose="%d in format string (no. 1) requires 'int' but the argument type is 'unsigned int'." cwe="686"><location file="game.c" line="161"/></error><error id="invalidPrintfArgType_sint" severity="warning" msg="%d in format string (no. 1) requires 'int' but the argument type is 'unsigned int'." verbose="%d in format string (no. 1) requires 'int' but the argument type is 'unsigned int'." cwe="686"><location file="game.c" line="171"/></error><error id="invalidPrintfArgType_sint" severity="warning" msg="%d in format string (no. 1) requires 'int' but the argument type is 'unsigned int'." verbose="%d in format string (no. 1) requires 'int' but the argument type is 'unsigned int'." cwe="686"><location file="game.c" line="245"/></error><error id="invalidPrintfArgType_sint" severity="warning" msg="%d in format string (no. 1) requires 'int' but the argument type is 'unsigned int'." verbose="%d in format string (no. 1) requires 'int' but the argument type is 'unsigned int'." cwe="686"><location file="game.c" line="255"/></error><error id="variableScope" severity="style" msg="The scope of the variable 'len' can be reduced." verbose="The scope

of the variable 'len' can be reduced. Warning: Be careful when fixing this message, especially when there are inner loops. Here is an example where cppcheck will write that the scope for 'i' can be reduced:\012void f(int x)\012{\012   int i = 0;\012   if (x) {\012   // it's safe to move 'int i = 0;' here\012   for (int n = 0; n < 10; ++n) {\012   // it is possible but not safe to move 'int i = 0;' here\012   do_something(&i);\012   }\012   }\012}\012When you see this message it is always safe to reduce the variable scope 1 level." cwe="398"><location file="game_io.c" line="24"/></error><error id="invalidPrintfArgType_sint" severity="warning" msg="%ld in format string (no. 2) requires 'long' but the argument type is 'unsigned long'." verbose="%ld in format string (no. 2) requires 'long' but the argument …

4 Optimisation et profilage