B2-CPE

Responsable : Younes SERRAJ
Contact : younes2.serraj@epitech.eu

Bootstrap Allum1

Dernière mise à jour : 10/02/2016 11h52

- ➔ Si ce n'est pas déjà fait, nous vous invitons à lire le sujet de l'Allum1 !
- ➔ Si vous n'avez pas assisté au Kick-off Allum1... vous auriez du ;)
- ➔ Chaque exercice est à rendre dans un fichier "nom-de-la-fonction.c" à la racine du dépôt, et chaque fichier sera compilé séparément avec notre fonction main().
  - ◆ Exemple pour l'exercice Etape 0 : print_game_board.c
- ➔ Nom du dépôt : CPE_year_allum1_bootstrap
  - ◆ Exemple pour l'année 2015/2016 : CPE_2015_allum1_bootstrap

Overview

L'Allum1 sera le premier jeu vidéo que vous allez coder. L'objectif est de vous familiariser avec les composants de base d'un jeu afin que vous puissiez, par la suite, inventer des jeux plus complexes et plus aboutis. Les composants de base d'un jeu sont :

- Un plateau
- Des règles
- Un ou plusieurs joueurs
- (une gestion d'erreurs béton, personne n'aime les bugs dans les jeux)

Votre programme consistera donc en une boucle de jeu qui permettra aux joueurs d'évoluer sur ou faire évoluer le plateau, en suivant les règles, afin de gagner (ou perdre..).

Partie 1 : Le plateau

Etape 0

Ecrivez une fonction qui affiche un plateau de 4 lignes d'allumettes de forme pyramidale. Les allumettes seront représentées par des '|'.

- ❏ void print_game_board()

Etape 1

Ecrivez une fonction qui affiche le plateau précendent dans un joli cadre carré. Vous utiliserez le caractère '*' pour dessiner ce joli cadre.

- ❏ voic print_board_game_in_a_pretty_frame()

Faites bien attention à ne pas avoir d'espaces vides inutils, cad que la dernière ligne d'allumettes ne devrait pas avoir d'espaces entre les allumettes et le cadre.

Etape 2

Quand un joueur (IA ou human) retire un certain nombre d'allumettes, le plateau mis à jour est affiché. Ecrivez une fonction qui, partant d'un plateau de 4 lignes (comme généré à l'étape précédente),, prend la ligne et le nombre d'allumettes à retirer, et affiche le plateau mis à jour.

- ❏ void print_updated_board_game(int line, int nb_matches)

Etape 3

- ➔ Nous compilerons cet exercice avec notre main, le fichier .c de l'exercice ainsi que vos fichiers get_next_line.{c,h} qui devront se trouver à la racine du dépôt.

Il est temps de créer une vraie intéraction avec le joueur. Votre get_next_line() est prêt ? Votre my_getnbr() aussi ? Alors ça devrait être assez facile d'écrire la fonction suivante :

- ❏ void read_player_move_and_print_updated_board_game()

Cette fonction devra lire le nombre de lignes et le nombre d'allumettes sur l'entrée standard, et afficher le plateau mis à jour. Vous veillerez à implémenter la gestion d'erreur telle que demandée dans le sujet du projet. Exemple :

```
Line: 999
Error: this line is out of range
Line: -1
Error: this line is out of range
Line: 2
Matches: 789
Error: not enough matches on this line
Line: 2
Matches: 0
Error: you have to remove at least one match
Line: 2
Matches: -1
Error: you have to remove at least one match
Line: 2
Matches: 2
Player removed 2 match(es) from line 2
*********
*   |   *
*  |    *
* ||||| *
*|||||||*
*********
```

Partie 2 : IA

Etape 0

➔ Nous compilerons cet exercice avec notre main, le fichier .c de l'exercice ainsi que vos fichiers get_next_line.{c,h} qui devront se trouver à la racine du dépôt.

Reprenez la fonction de l'étape précédente. On permet au joueur de jouer en premier, et vous devez maintenant ajouter la partie IA, cad que vous allez coder une fonction qui génère un échange similaire à l'exemple ci-dessous :

❏ void make_human_and_ai_play_a_round()

```
*********
*   |   *
*  |||  *
* ||||| *
*|||||||*
*********

Your turn:
Line: 3
Matches: -1
Error: you have to remove at least one match
Line: 3
Matches: 99
Error: not enough matches on this line
Line: 15
Error: this line is out of range
Line: 3
```

```
Matches: 2
Player removed 2 match(es) from line 3
*********
*   |   *
*  |||  *
* |||   *
*|||||||*
*********

AI's turn...
AI removed 3 match(es) from line 2
*********
*   |   *
*       *
* |||   *
*|||||||*
*********
```

The end.