



2048

The game

Staff 42 bocal@staff.42.fr

Résumé: Ce document contient les instructions pour le rush 2048.

Table des matières

I	Préambule	2
II	Sujet	4
III	Bonus	7
IV	Consignes	8
V	Notation	9

Chapitre I

Préambule

Ce rush sera plus facile si vous visionnez au préalable cette [vidéo pédagogique](#)

Man: It takes a lot to make a stew
Woman: A pinch of salt and laughter, too
M: A scoop of kids to add the spice
W: A dash of love to make it nice, and you've got
Both: Too many Cooks
W: Too many Cooks
B: Too many Cooks
M: Too many Cooks
B: Too many Cooks
W: Too many Cooks
B: Too many Cooks
M: Too many
W: It takes a lot to make a stew
M: When it comes to me and you
W: And him and her and the baby, too
B: Too many Cooks, it's true
M: The saying goes, it'll spoil the broth
W: Honey, I think that's not true
M: Well, maybe too many Cooks will spoil the broth,
but they'll fill our hearts with
B: So much, so much lo-o-ove
Too many Cooks
M: A family is like a soup
W: Everyone adds an extra scoop
M: Mix an ounce of smile so sweet
W: A dash of cool to add the heat, and you've got
B: Too many Cooks
W: Too many Cooks
B: Too many Cooks
M: Too many Cooks
B: Too many Cooks
M: Too many Cooks
B: Too many Cooks
W: Too many Cooks

B: Too many Cooks

W: Too many Cooks

B: Too many Cooks

M: Too many Cooks

B: Too many Cooks

W: Too many Cooks

B: Too many Cooks

M: Too many Cooks

B: Too many Cooks

W: Too many Cooks

B: Too many Cooks

M: Too many Cooks

B: Too many Cooks

W: Too many Cooks

B: Too many Cooks

M: Too many

It takes a lot to make a stew

W: Especially when it's me and you

M: And him and Steve from corporate, too

...

Chapitre II

Sujet

Le but de ce rush est de recoder le jeu 2048 en C pour un affichage console. Ce n'est pas très compliqué, et pourtant, vous serez nombreux à échouer par abus de confiance et manque de vérifications... si si, même en vous prévenant ! Et pour être franc, 2048 est plus complexe qu'on ne se l'imagine.

Il vous faudra faire preuve de rigueur, veiller à ne rien oublier, bien lire ce cahier des charges, bien tester votre programme, et réussir à travailler à deux. Donc ne perdez pas de temps ! D'autant qu'il y a matière à faire des bonus facilement si votre programme est bien conçu dès le départ.

2048 is played on a simple gray 4 x 4 grid, with numbered tiles that slide smoothly when a player moves them using the four arrow keys. Every turn, a new tile will randomly appear in an empty spot on the board with a value of either 2 or 4. Tiles slide as far as possible in the chosen direction until they are stopped by either another tile or the edge of the grid. If two tiles of the same number collide while moving, they will merge into a tile with the total value of the two tiles that collided. The resulting tile cannot merge with another tile again in the same move. (source : Wikipedia)

Comme votre jeu s'affichera en console, ce ne sera pas "smoothly" mais "instantly".

Vous pouvez essayer 2048 ici : [clic](#)

Donc il vous faut :

- afficher un plateau de 16 cases en console (avec les caractères de votre choix) ;
- dans ces cases, des nombres (2 cases remplies à l'état initial) ;
- gérer les mouvements haut, bas, gauche, droite comme dans le jeu ;
- gérer l'apparition de nouveaux nombres random ('2' et '4') ;
- les '2' apparaissent plus souvent que les '4' ;
- indiquer au joueur quand il a gagné ou perdu ;
- laisser le joueur continuer sa partie après avoir atteint la condition de victoire ;
- ah, on me dit que je vous mâche trop le travail... bon vous avez compris, c'est un 2048.



EXCEPTIONNELLEMENT, vous êtes autorisés à jouer à 2048 dans le cadre de ce rush. Toute autre utilisation de jeu vidéo en cluster reste soumise aux règles habituelles...

Considérations techniques :

- La partie se termine quand le joueur ne peut plus bouger ou quand une case atteint la valeur 2048. Cette valeur fera l'objet d'une enum dans un header et pourra être modifiée en soutenance (notamment pour vérifier ce qu'il se passe lorsque la condition de victoire est atteinte, parce qu'avec 2048 ça risque d'être long).
- Vous devez donc rendre au minimum un fichier *.h contenant par exemple :

```
enum    e_const
{
    WIN_VALUE = 2048
};
```

- Vous veillerez à ce que cette valeur ne soit prise en compte que s'il s'agit d'une puissance de 2.
- La touche ECHAP doit permettre de quitter le jeu proprement.
- Vous avez droit à votre libft, aux fonctions `rand(3)`, `srand(3)`, `time(3)`, `signal(3)`, et à toutes les fonctions de la lib ncurses. Vous pouvez utiliser d'autres fonctions si elles sont nécessaires à vos bonus.
- Interdiction d'utiliser des variables globales, hormis celles qui pourraient déjà être définies pour vous. On tolérera une exception à cette règle pour gérer les signaux.
- Vous n'utiliserez pas de variable statique.
- La taille du plateau doit s'adapter à la taille de la fenêtre lorsque vous lancez le programme, et dynamiquement lorsque vous redimensionnez le terminal (voir vidéo).

Chapitre III

Bonus

Les bonus ne seront évalués que si la partie obligatoire est parfaite. Si vous faites un bonus perso qui entre en conflit avec le cahier des charges (par exemple faire popper des '8' random), ce doit être désactivable. Voici quelques idées :

- des couleurs ;
- le joueur peut choisir de jouer sur une grille 4x4 ou 5x5 ;
- le lancement de la partie est précédé d'un menu (de simples lignes de texte numérotées suffisent), proposant au minimum deux choix dont un pour lancer la partie ;
- les chiffres sont dessinés en ascii art plutôt qu'en chiffres... ;
- les meilleurs scores sont enregistrés avec un pseudo dans un fichier scores.txt, créé dans le dossier courant s'il n'existe pas, puis affichés dans le jeu.

Ces bonus sont déjà un peu avancés et il y a sûrement des choses plus évidentes à faire avant. D'ailleurs, un certain nombre de bonus évidents et relativement simples vous permettront de gagner facilement des points si vous les trouvez.

Chapitre IV

Consignes

- Vous êtes libres d'organiser et nommer vos fichiers comme vous le désirez.
- L'exécutable doit s'appeler `game_2048`.
- Vous devez rendre un Makefile.
- Votre Makefile doit compiler le projet, et doit contenir les règles habituelles. Il ne doit recompiler le programme qu'en cas de nécessité.
- Votre Makefile doit compiler avec les flags `-Wall`, `-Wextra` et `-Werror`.
- Si vous êtes malin et que vous utilisez votre libft, vous devez en copier les sources et le Makefile associé dans un dossier nommé libft qui devra être à la racine de votre dépôt de rendu. Votre Makefile devra compiler la librairie en appelant son Makefile avant de compiler votre projet.
- Votre projet doit être à la Norme.
- En aucun cas votre programme ne doit quitter de façon inattendue (segmentation fault, bus error, double free, etc).
- Toute fonction en sus de la première dans un fichier `.c` doit être définie `static`.
- Toute mémoire allouée sur le tas doit être libérée proprement.
- Vous ne devez jamais rendre de code que vous n'avez pas écrit vous-même. En cas de doute, vous serez invités à une séance de recode au bocal pour juger de votre bonne foi.
- Vous devez rendre à la racine de votre dépôt de rendu un fichier nommé `auteur` contenant les logins des deux membres du groupe suivis d'un `'\n'`, tel que :

```
superprompt> cat -e ./auteur
login1$
login2$
superprompt>
```

Chapitre V

Notation

- S'il vous manque une des fonctionnalités principales du jeu, vous risquez de ne pas valider le rush.
- Les deux membres du binôme devront être capables d'expliquer un point de détail du code ou le fonctionnement du programme dans sa globalité. Les correcteurs sont encouragés à poser des questions à chacun des deux équipiers.
- Pour vous mettre un peu la pression, on vous rappelle que les correcteurs devront appliquer scrupuleusement ces règles :
 - Fichier auteur invalide : 0 ;
 - Makefile incomplet ou qui relink : 0 ;
 - Faute de norme, sur n'importe quelle partie du code rendu : 0 ;
 - Segfault, même si causé par l'ajout d'un bonus : 0 ;
 - Consigne du chapitre Consignes non respectée : 0.
- En contrepartie, les correcteurs seront plus conciliants sur certains points :
 - si votre programme a du mal à gérer trop de requêtes à la suite ;
 - si votre programme ne gère pas quand on maintient une touche appuyée ;
 - si votre programme est lancé dans un "environnement vide" ;
 - si votre jeu est mauvais (du moment qu'il marche) ;
 - si votre jeu est moche (du moment qu'on voit les cases et les chiffres).

Bon courage à tous pour ce rush !