

2048

version 4.0.0



DIVERSITY
BY EPITECH

I. Introduction

Une start-up de développement de jeux mobiles dirigée par Markus Sugarberg s'est faite hacker et son célèbre jeu, le 2048, si satisfaisant et addictif, a été cassé par les hackers ! L'entreprise a besoin des talents de la jeune Cerisia pour le réparer d'ici deux heures !

Le but du jeu est de fusionner des nombres d'une même valeur jusqu'à atteindre le nombre 2048. Voici un site que Markus Sugarberg donne à Cerisia pour se familiariser avec son fonctionnement : [Play2048](#).



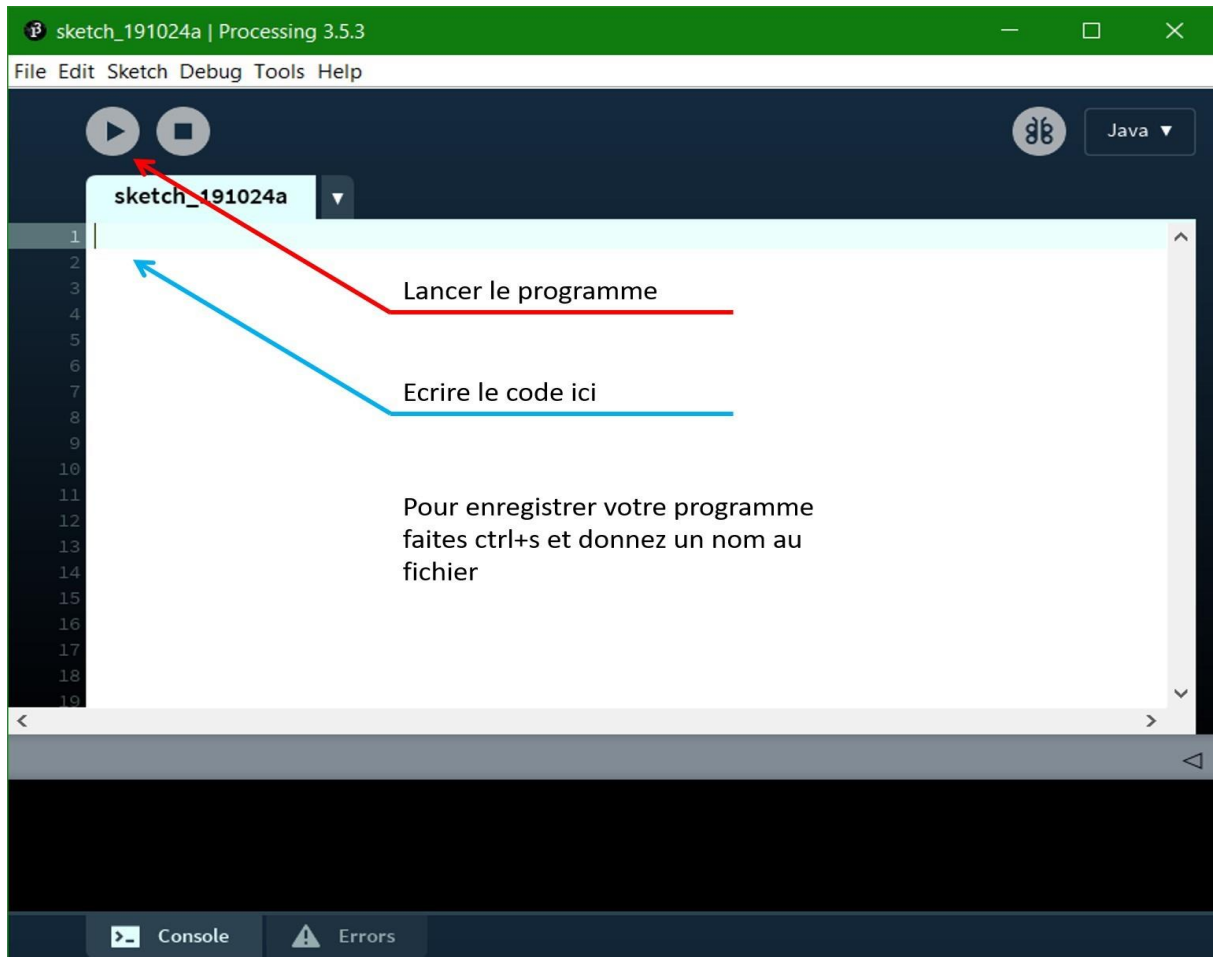
Logo du logiciel Processing

Consignes

- En cas de question, pensez à demander de l'aide à votre voisin de droite. Puis de gauche. Demandez enfin à un Cobra si vous êtes toujours bloqué(e).
- Vous avez tout à fait le droit d'utiliser internet pour trouver des réponses ou pour vous renseigner.
- N'hésitez pas à faire des bonus et à ajouter des fonctionnalités lorsque votre projet sera terminé.

Installation de Processing

Si vous n'avez pas Processing sur votre ordinateur, il faudra le télécharger depuis [ce site](#) puis l'installer. Voici à quoi ressemble l'interface du logiciel, ainsi que son utilisation :



Un design sobre et intuitif, n'est-ce pas ?

Des cases et des chiffres

Markus Sugarberg attend de Cerisia que son jeu fonctionne de nouveau comme un charme. Cela implique que l'apparence soit identique à l'image présentée en-dessous, que les nombres fusionnent et que l'utilisateur puisse gagner ou perdre. Plus facile à dire qu'à faire, mais l'informaticienne est prête à relever le défi !



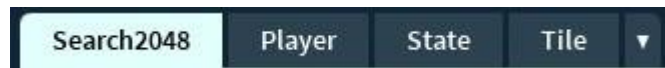
		2	8
	2	8	32
2	8	16	128
64	256	512	2048

Exemple de résultat que vous devriez avoir

A vos claviers !

Pour la mission, le grand chef de l'entreprise donne un protocole de réalisation ainsi qu'un dossier contenant des fichiers Processing à la jeune développeuse.

Avec le logiciel installé, double-cliquez sur les fichiers pour les ouvrir. Une fois qu'ils sont tous ouverts, un code incomplet du projet apparaît. Cerisia pourra maintenant naviguer à travers les différents onglets du projet correspondant chacun à un fichier :



Exemple d'onglets sur Processing

Il faut prendre le temps pour bien lire le code et essayer de comprendre ce qu'il fait dans un premier temps.

Les numéros ont été réinitialisés !

Un des hackers a cassé les numéros affichés dans les cases ! Les valeurs de celles-ci sont toutes 0 au lieu de 2 ou 4 quand on lance le jeu. Le fichier "Tile" doit avoir un problème...

Cerisia décide d'aller y faire un tour pour essayer de régler ce problème ! Elle se renseigne sur le fonctionnement des [variables](#). Monsieur Sugarberg lui donne un petit indice : "La réponse se retrouve dans les zéros". Un peu dubitative, la développeuse décide tout de même d'écouter son conseil.

Les cases ont perdu leur couleur !

Les hackers ont supprimé toute une partie du fichier "Tile", la fonction **setColour()** est maintenant vide. La fonction dans ce fichier et faites en sorte que la variable [color](#) représente une couleur différente selon sa [valeur](#). Il faut ajouter les couleurs typiques du jeu, c'est-à-dire le jaune, l'orange et le marron.

Pour cette étape, il y a plusieurs manières de réussir l'exercice. La jeune femme fait donc des recherches pour trouver la solution qui lui paraît la plus intéressante.

Les cases ne bougent plus !

Maintenant que le plateau de jeu ressemble au résultat attendu, il faut à présent réparer le mouvement des cases. Pour l'instant, le joueur ne peut même plus les faire bouger. Pour ça, il faut se servir de la fonction par défaut de Processing : **keyPressed()**. Rendez-vous dans le fichier "Search2048" et repérez-la pour la corriger.

Vous allez aussi utiliser la variable "p". Mais attention, cette variable est un peu particulière car elle contient des fonctions. Cherchez comment vous en servir et rendez ce jeu jouable !

Des bugs ? Mais où ?

Les deux heures sont bientôt écoulées. Cerisia teste son programme dans le doute. Après tout, un bug est si vite arrivé... C'est le moment de mettre à profit le temps que vous avez investi pour comprendre le code au début et vérifier que tout fonctionne correctement.

Conclusion

Cerisia a réussi sa mission ! Monsieur Sugarberg est extrêmement content et fier de s'être fier à la bonne personne. Sa start-up est de nouveau sur la route vers le succès avec son fameux jeu.

Cependant, pourquoi s'arrêter en si bon chemin alors qu'il y a tant à faire ? Voici quelques idées de bonus pour améliorer le 2048 :

- A la fin de chaque partie, le jeu se relance directement et les joueurs ne savent pas si c'est un bug ou si le jeu est fini. Ajoutez un écran avec un message pour indiquer si le joueur a gagné ou perdu.
- Les joueurs ont commencé à s'ennuyer à force de devoir recommencer autour des 2048 points. Donnez-leur donc du fil à retordre avec des parties plus longues en augmentant le nombre de cases dans l'écran de jeu !
- Les joueurs ne connaissent pas leur score et veulent savoir jusqu'où ils peuvent arriver. Affichez le score du joueur après chaque fusion de nombre.