

2048

version 3.0



DIVERSITY
by EPITECH

I. Introduction

Processing est un outil permettant d'apprendre à coder par la création graphique. Il est utilisé par des étudiants, artistes, chercheurs et curieux pour apprendre ou créer rapidement des visuels. Par exemple, la chaîne YouTube [Coding Train](#), qui fait de la découverte de l'algorithmie, se sert souvent de Processing dans ses vidéos. Dans une de ses vidéos, il crée un [Star field](#) (affichage d'étoiles) !

Aujourd'hui, vous allez apprendre à coder avec Processing. Tout au long du sujet, vous allez être confrontés à de nouvelles notions de programmation. Ce faisant, n'hésitez pas à vous arrêter sur un exercice et à vous amuser avec !



Logo de Processing

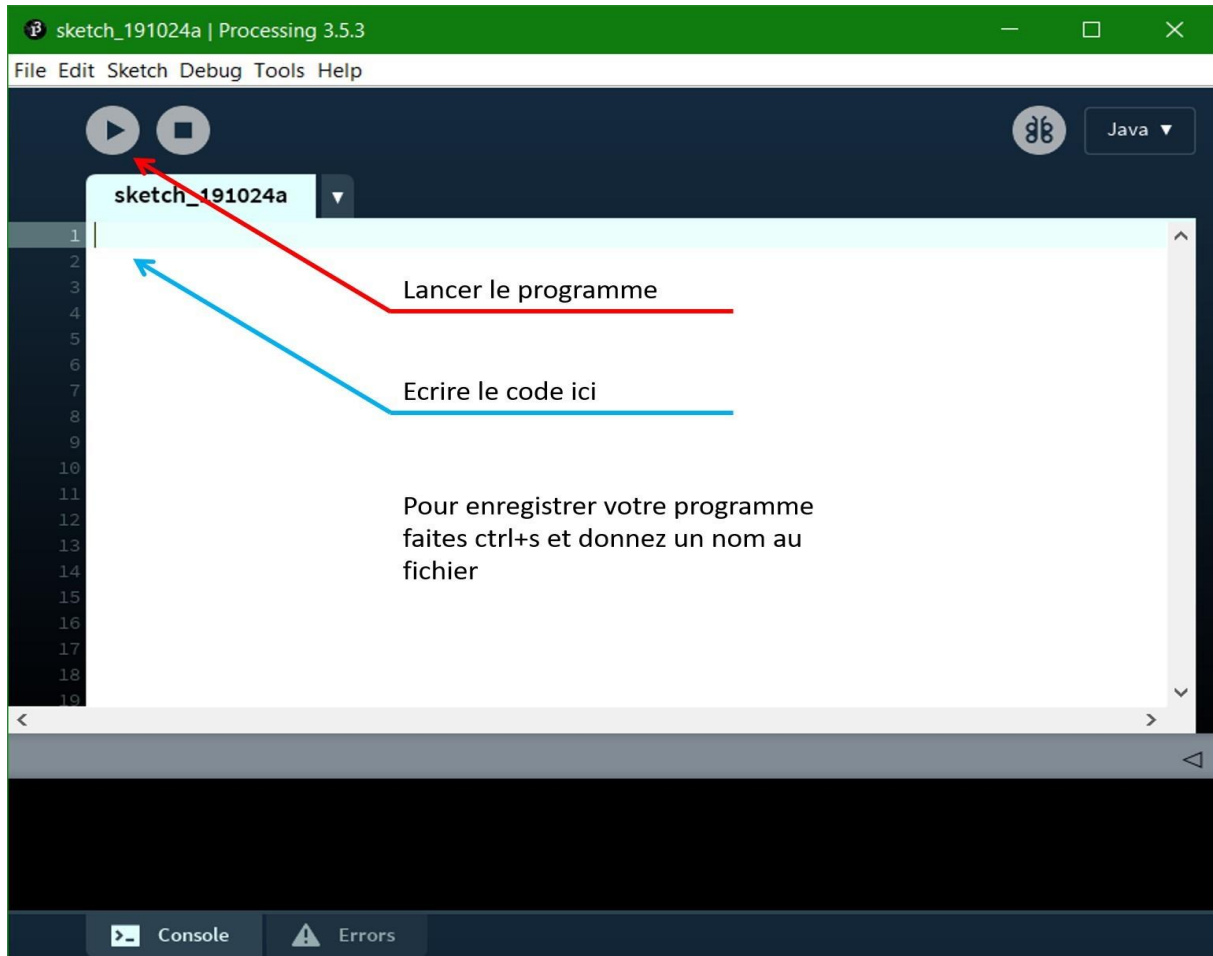
II. Consignes

- * Sur ce projet, il vous sera demandé de choisir comme nom de repository : cc_2048.
- * N'oubliez pas de push régulièrement.
- * En cas de question, pensez à demander de l'aide à votre voisin de droite. Puis de gauche. Demandez enfin à un Cobra (ceux-là ne mordent pas) 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 et validé.

III. Installation de processing

Si processing n'est pas installé sur votre ordinateur, il faudra le télécharger depuis [ce site](#) puis l'installer.

Voici à quoi ressemble l'interface de processing, ainsi que son utilisation :



Interface de processing

IV. Le jeu

Une startup de développement de jeux mobiles dirigée par Markus Sugarberg s'est fait 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 2048 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 le jeu : [Play2048](https://play2048.com/)



Exemple de screen que vous devriez avoir

V. Le code

Pour la mission, le grand chef de l'entreprise donne un sujet dans un dossier séparé des fichiers Processing : "nomdufichier.pde" à la jeune développeuse.

Avec Processing installé, double-clique sur les fichiers pour les ouvrir. Une fois 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.

a. Les numéros ont été réinitialisés !

Un des hackers a cassé les numéros affichés sur les cases ! Les valeurs des cases sont toutes 0 au lieu de 2 ou 4. Le fichier "Tile" doit avoir un problème. Cerisia décide d'aller faire un tour et 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".

b. Les cases ont perdu leur couleur !

Les hackers ont supprimé toute une partie du fichier "Tile". La fonction setColour est 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 des recherches sur internet pour trouver la solution.

c. 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 ce servir de la fonction par défaut de Processing : "keyPressed". Rendez-vous dans le fichier "Search2048" et repérez la fonction keyPressed.

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

d. Des bugs ? Mais où ?

Cerisia teste son programme dans le doute, un bug est si vite arrivé. Il faut trouver le bug et le régler ! Si bug il y a bien sûr.

VI. Conclusion

Cerisia a réussi sa mission ! Monsieur Sugarberg est extrêmement content et fière. Sa start-up est de nouveau sur la route vers le succès avec son fameux nouveau jeu.

Voici quelques idées de bonus :

- * Les joueurs ont commencé à s'ennuyer à force de devoir recommencer autour des 2048 points. Pour cette mission supplémentaire, donnez-leur des parties plus longues en augmentant le nombre de cases dans l'écran de jeu !
- * Ajoutez un temps pour mettre à l'épreuve les joueurs et tester leur rapidité !
- * 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.
- * 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 marquer la fin de la partie.
- * A chaque fin de partie le joueur ne se rappelle plus le score qu'il avait réalisé. Sauvegardez le meilleur score du joueur et affichez-le sur l'écran.
- * Les joueurs ne retiennent pas leurs scores et on en a même vu l'écrire sur une feuille avec leur nom.
- * Ajoutez un board avec les noms d'utilisateurs et leurs scores.