

SAÉ Système - 2048

1- Principes du jeu

À partir d'une grille carrée de côté 4, les joueurs doivent fusionner des puissances de 2 jusqu'à faire apparaître la valeur 2048, en choisissant de déplacer l'ensemble des tuiles dans une direction.

Structure :

- Grille : 4×4 cases
- Tuiles : valeurs qui sont des puissances de 2 (2, 4, 8, 16, 32, 64, 128, 256, 512,...)

Déroulement d'un tour

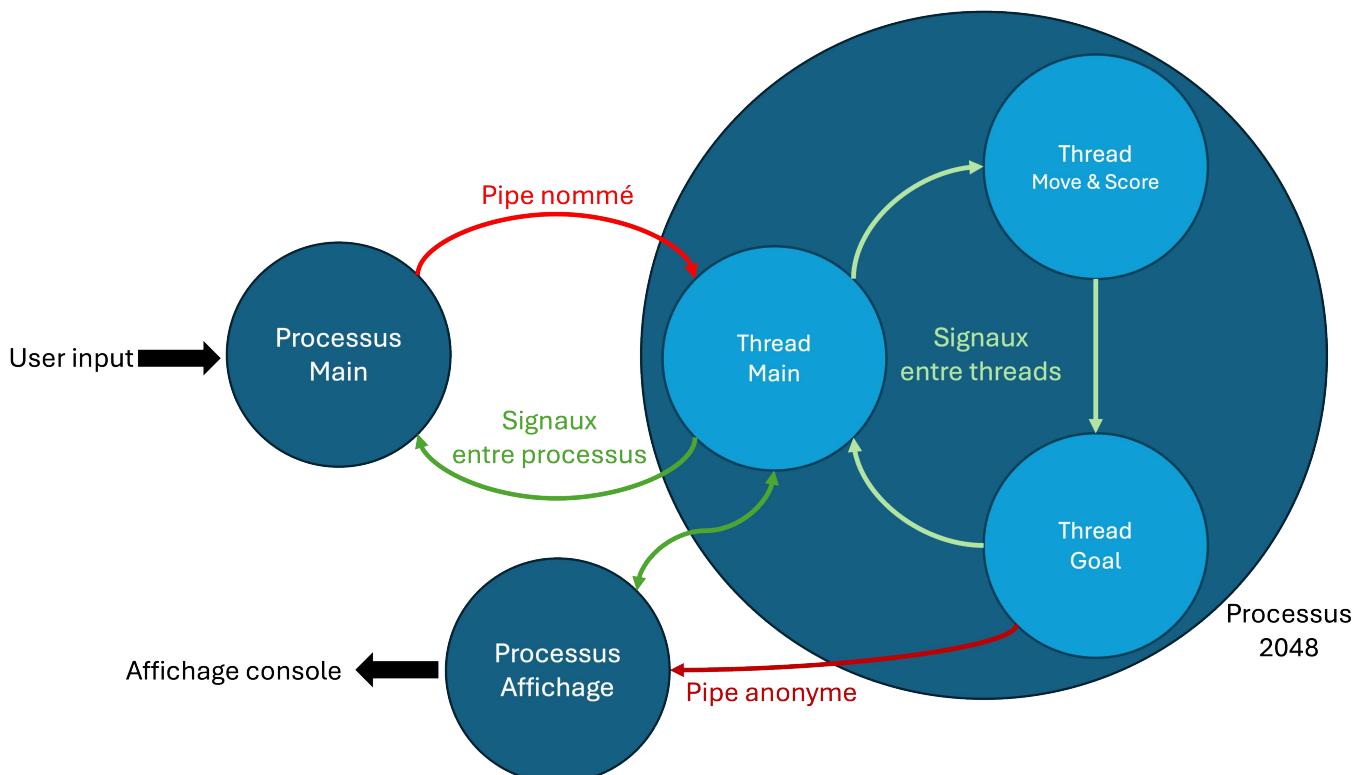
1. *Apparition d'une tuile* : au début et après chaque coup valide, une nouvelle tuile (valeur 2 ou 4) apparaît aléatoirement sur une case vide.
2. *Déplacement* : le joueur choisit une direction (\uparrow , \downarrow , \leftarrow , \rightarrow). Toutes les tuiles glissent dans cette direction jusqu'à rencontrer un bord ou une autre tuile. Un coup valide est un coup qui entraîne le déplacement d'au moins une tuile.
3. *Fusion* :
 - a. Deux tuiles adjacentes de même valeur fusionnent en une seule tuile de valeur double.
 - b. Chaque tuile ne peut fusionner qu'une seule fois par coup.
 - c. Le score augmente de la valeur de la tuile créée.

Conditions de fin

- *Victoire* : Une tuile 2048 est créée. La partie s'arrête.
- *Défaite* : Il n'est plus possible de réaliser un coup valide.

2- Ce que vous devez mettre en place

En trinôme, vous devez mettre en place d'ici le **19 janvier 2026** le jeu 2048 tel que décrit par l'illustration ci-dessous et les consignes suivantes.



Mise en place par étapes

1. Un premier processus, le processus « *main* », est à mettre en place. Son rôle est de récupérer les commandes utilisateurs permettant au jeu de se dérouler (déplacement des tuiles ou abandon). Ces commandes sont ensuite écrites dans un pipe nommé vers le second processus, le processus « *2048* », qui réalisera la logique attendue derrière ces commandes.
2. Le processus « *2048* » est composé de trois threads ayant chacun un rôle défini et ils devront être synchronisés par des signaux.
 - a. Le thread « *principal* » doit mettre en place les autres threads, gérer leur synchronisation et gérer les entrées utilisateurs qu'il doit lire sur le pipe nommé partagé avec le processus « *main* ». Les entrées utilisateurs permettant de déplacer les tuiles seront gérées par le thread « *Move & Score* » : elles doivent ainsi lui être transmises de la manière qui vous semble la plus judicieuse. L'entrée utilisateur associée à l'abandon est gérée par le thread « *principal* » qui devra alors proprement stopper l'exécution de tous les processus, à l'aide de signaux.
 - b. Le thread « *Move & Score* » se chargera de réaliser le déplacement des tuiles selon les règles du jeu et aussi, de calculer le nouveau score suite au déplacement. Il partagera ces informations avec le thread « *Goal* ».
 - c. Le thread « *Goal* » se chargera de vérifier si la partie en cours est terminée et de quelle manière (victoire ou défaite). Il a également pour responsabilité de transmettre toutes les informations à afficher au processus « *affichage* » par le biais d'un pipe anonyme.
3. Le processus « *affichage* » affiche l'état du jeu aux utilisateurs dans la console. Il communiquera et se synchronisera avec le processus « *2048* » par le biais de signaux.

Contact : ali.ayadi@unistra.fr ; rorhand@unistra.fr