

Projet : Flood

Étape 2 - La couleur se répand

La deuxième étape sera consacrée à l'algorithme central du jeu : la couleur doit maintenant se propager correctement sur le terrain de jeu. Pour cela, nous allons enrichir le modèle.

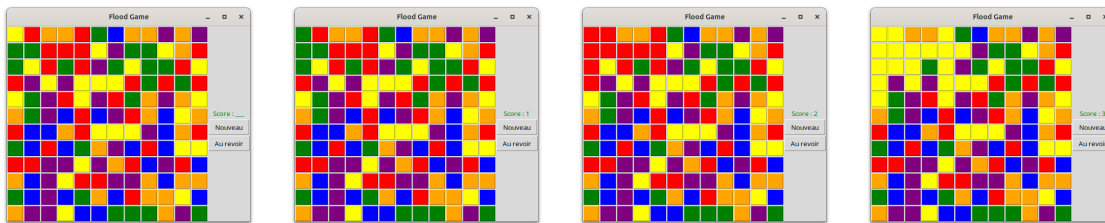


FIGURE 1 – Flood : Le joueur a choisi la couleur vert, puis rouge, puis jaune

Une case

Pour le moment, nous avons juste mémorisé (sous la forme d'un entier) la couleur de chaque carré. Cela va être insuffisant pour développer les nouveaux algorithmes. Pour pouvoir mémoriser plusieurs choses représentant un petit carré de la mosaïque, vous devez écrire une classe `Case` (dans le fichier `modele_flood.py`) qui modélise un carré :

- Les attributs conservés dans une case sont :
 - ses coordonnées
 - sa couleur (toujours sous la forme d'un entier)
 - un booléen qui indique l'état de la case : soit elle fait partie de la zone touchée par le carré en haut à gauche, soit elle n'en fait pas partie.
- La classe `Case` a aussi besoin de la dimension de la matrice du `Modele`.
- Les méthodes proposées par la classe `Case` sont :
 - des accesseurs en lecture sur les attributs
 - une méthode `voisines` qui retourne la liste des coordonnées des cases voisines (donc soit une liste de deux couples de coordonnées si la case est dans un coin, de trois couples si la case est sur un bord, ou de quatre couples dans les autres cas)
 - une méthode pour changer la couleur
 - une méthode pour changer l'état de la `Case`.

Le modèle - La couleur se répand

Dans un premier temps, vous devez adapter le `Modele` :

- au niveau du constructeur : la matrice n'est plus une matrice d'entiers mais une matrice de `Case`
- dans la méthode qui démarre une nouvelle partie, dans la méthode `__str__`, ...

- en ajoutant des méthodes comme `couleur`, `voisines`, ... qui, à partir d'un numéro de ligne et d'un numéro de colonne retourne la couleur à cette position, les coordonnées des cases voisines, ... À vous d'ajouter les méthodes au fur et à mesure de vos besoins.

Le travail le plus important maintenant est d'écrire la méthode `pose_couleur` qui prend un numéro de couleur en paramètre et diffuse cette couleur à partir du carré en haut à gauche.

La fin de partie

Et pour terminer, vous pouvez maintenant détecter la fin de partie.

Maintenant, le `Modele` peut calculer si la partie est finie (méthode `partie_finie` dans le `Modele`).

Quand cette méthode doit-elle être appelée ?

Lorsque la partie est finie, un message apparaît sur le côté droit et les boutons ne sont plus actifs.

Lorsqu'on re-démarre une partie, l'affichage doit changer et les boutons doivent redevenir actifs.

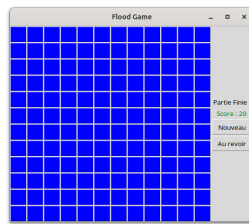


FIGURE 2 – Flood : La partie est finie !

L'application

Testez votre application. N'oubliez pas que vous avez écrit une méthode pour afficher le plateau de jeu en mode texte, cela sera très pratique pour débuser les erreurs.

Vous pouvez jouer avec votre application ? Bravo, c'est fini pour cette étape !