



# Ingénierie Informatique

## TP n°7

### Bonus fin du projet

---

#### Étape 34 – Fonction ajouteNouveauxBonbons

---

La fonction `ajouteNouveauxBonbons` prend en paramètres le tableau « modèle » `tab_jeu` et le tableau `tab_match` qui contient l'état des bonbons (0 = supprimé, 2 = strié, -1 = pas de changement). Pour chaque cellule de la ligne 0 (la ligne la plus haute) de `tab_jeu` égale à -1 (la cellule ne contient donc pas de bonbon), la fonction affecte à la cellule un nouveau bonbon qui ne crée aucun `match3` avec ses voisins et elle affecte la valeur -1 à la cellule correspondante de `tab_match` (pour indiquer que ce bonbon ne doit être ni détruit ni strié). Réécrivez la fonction `ajouteNouveauxBonbons` et testez la en créant des `match3`. Vous devriez voir les bonbons descendre et de nouveaux bonbons apparaître tout en haut de l'espace de jeu.

#### Aide sur l'écriture de la fonction :

Pour créer de nouveaux bonbons qui n'induisent pas de nouveaux `match3`, vous pouvez avantageusement vous inspirer du code de la fonction `initialiseTableauJeu`, que vous avez écrite au TP n°2.

#### Utilité de la fonction :

Dans les étapes précédentes, vous avez réalisé des fonctions qui détruisent des bonbons et qui font descendre ceux qui sont au dessus. Par conséquent, certaines cellules de la ligne tout en haut de l'espace de jeu `tab_jeu` sont devenues vides (elles ont des valeurs égales à -1). Il faut donc remplir ces cellules avec de nouveaux bonbons. C'est l'objectif de la fonction `ajouteNouveauxBonbons`. Elle est appelée par la fonction `mouseReleaseEvent` de l'interface de jeu après avoir réalisé l'animation qui montre les bonbons en train de descendre d'une case vers le bas et après que les tableaux `tab_jeu` et `tab_match` aient été mis à jour via la fonction `descendTableau` pour tenir compte de ces déplacements vers le bas.

---

#### Étape 35 – Fonction calculeScore

---

Comme son nom l'indique, la fonction `calculeScore` calcule le score dû aux suppressions de bonbons des `match3`. Pour cela, on souhaite que plus le nombre de bonbons intervenant dans un `match3` est élevé, plus le score augmente fortement. Différentes méthodes sont possibles pour parvenir à ce résultat. Celle que l'on utilise ici est la suivante :

Dans un premier temps, on tient compte des match3 horizontaux. Pour cela, on parcourt le tableau `tab_match` ligne par ligne. Au début de l'inspection de chaque ligne `lig`, on crée une variable `tmp_score` qui vaut 0. On parcourt alors toutes les colonnes de cette ligne de la gauche vers la droite. Pour chaque colonne `col`, si sa valeur de `tab_jeu[col,lig]` est différente de -1 (il y a un bonbon) et si celle de `tab_match[col,lig]` est  $\geq 0$  (il y a une suppression ou une strie), on augmente `tmp_score` de 10, sinon on affecte la valeur à 0 à `tmp_score`. On augmente ensuite le score de la valeur de `tmp_score`. Enfin, si `tab_match[col,lig]` a une valeur de 2, on augmente à nouveau le score de 30.

Dans un deuxième temps, afin de tenir compte des match3 verticaux, on réalise les mêmes opérations mais en inversant le rôle des lignes et des colonnes. Autrement dit, on parcourt les tableaux `tab_jeu` et `tab_match` colonne par colonne. Au début du parcours de chaque colonne, on crée une variable `tmp_score` qui vaut 0. On parcourt alors toutes les lignes de cette colonne du haut vers le bas. Si on rencontre une case dont la valeur dans `tab_match` est  $\geq 0$  et la valeur de la case correspondante dans `tab_jeu` est différente de -1, on augmente `tmp_score` de 10, sinon on fait redescendre sa valeur à 0. On augmente ensuite le score de la valeur de `tmp_score`. Enfin, si la case de `tab_match` a une valeur de 2, on réaugmente le score de 30.

Enfin, la fonction renvoie la valeur du score qu'elle a ainsi calculé. Testez votre fonction en réalisant des match3 et en affichant via un `print` le score que vous renvoyez.

#### Compléments d'informations :

L'utilisation de `tmp_score` permet d'augmenter significativement le score quand on a des match3 de 4 bonbons ou plus : notez qu'un seul bonbon augmente le score de 10, deux bonbons consécutifs supprimés l'augmente de  $10+20=30$ , trois bonbons consécutifs de  $10+20+30=60$ , *etc.* Quand on fait repasser la valeur de `tmp_score` à 0, c'est que l'on est arrivé à une case qui ne fait pas partie d'un match3. Du coup, il faut arrêter la suite des nombres triangulaires, sinon le score augmente trop vite. C'est pourquoi `tmp_score` doit revenir à 0.