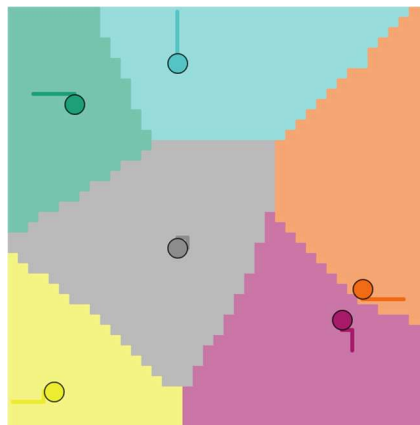


LIFAMI – TP : Simulation économique des marchands de glaces

Objectifs : Info/Math, des outils pour l'économie ?
Manipulation des structures, des boucles, etc.

La **loi de Hotelling** affirme que sur la plupart des marchés, la concurrence conduit les producteurs à réduire la différence entre leurs produits. Cette loi est aussi appelée **principe de différenciation minimale**. Elle a été formulée par le statisticien et économiste américain [Harold Hotelling](https://fr.wikipedia.org/wiki/Harold_Hotelling) (1895-1973) dans un article intitulé *Stability in Competition*. (Source Wikipedia : https://fr.wikipedia.org/wiki/Loi_de_Hotelling).



Le **problème des marchands de glace ambulant** est un exemple célèbre de la [théorie des jeux](#), et une approche simplifiée du modèle de Hotelling.

Soit une zone géographique où viennent s'installer N marchands de glaces. Pour simplifier, tous les marchands vendent la même glace, exactement identique mais pas au même prix. Un client choisit son marchand en minimisant la fonction « distance + prix ». A chaque tour un marchand se pose deux questions :

- bouger son stand avec 4 choix possibles : $x+1$ ou $x-1$ ou $y+1$ ou $y-1$;
- monter ou baisser ses prix de 10%.

Chaque marchand teste les 4 positions, les compare à la position actuelle et change s'il augmente ses rentrées. Il fait de même pour les prix.

1. Un marchand est représenté par une position, un prix de vente, une couleur (ceci pour l'affichage) et le nombre de client. Pour la position, vous pouvez récupérer la structure Point des TP précédents. On retrouve la rentrée d'argent d'un marchand en multipliant le nombre de clients par le prix de vente.

LesMarchands sera une structure comportant un grand tableau de *Marchand* et la taille réellement utilisée dans ce tableau. En algo :

```
structure LesMarchands
    mar : Tableau[100] de Marchand
    nm : Entier
FinStructure
```

Ecrivez les deux structures *Marchand* et *LesMarchands* comportant tous les marchands de la région.

2. Ecrivez la procédure *init* qui initialise aléatoirement tous les marchands. Les marchands auront une position dans la fenêtre *Gravic* de taille DIMW x DIMW. Les prix seront compris entre 2 et 6 euros.
3. Ecrivez la procédure *drawMarchands* qui affiche un cercle rouge de rayon 2 pixels à la position de chaque marchand.
4. Ecrivez la procédure *TraiteVente* qui traite une zone géographique. Avec *Gravic*, cette zone sera toute la fenêtre. En initialisation, vous mettrez à 0 le nombre de client de chaque marchand. Chaque pixel de la fenêtre sera un client qui prendra comme couleur celle du marchand qu'il choisit. Rappel : un client choisit le marchand qui minimise la somme distance + prix. Chaque fois qu'un client choisit un marchand, il faut ajouter le prix de la glace à la rentrée d'argent du marchand.

Pour chaque pixel, vous appellerez la procédure *put_pixel* pour colorier le pixel avec la couleur du marchand choisi :

```
Proc put_pixel(x,y : Entier, r,g,b : Entier)
```

Remarque : n'appellez pas la fonction *Gravic* d'affichage *winDisplay* dans cette procédure car cette procédure pourra être utilisée pour faire des calculs intermédiaires lors de l'évolution des positions/prix des marchands à la question suivante.

5. Ecrivez la procédure *MiseAJourMarchands* qui va faire évoluer chaque marchand. Pour cette procédure vous devez avoir deux structures *LesMarchands*, celle pour avant la mise à jour et celle qui sera mise à jour. En effet, tous les marchands doivent faire leur évolution de manière simultanée et non séquentiellement.
6. Ajoutez le code qui permet d'afficher sous forme de courbes les prix de tous les marchands. Faites des observations avec différentes initialisations ! Quand les prix sont-ils les plus bas ?