

Projet Programmation Concurrente – année 2017-18
Polytech’Nice Sophia – SI4
M. Riveill

Durée indicative : entre 15 et 30 heures dont uniquement 6 heures seront faites lors des séances de TD. [A faire par groupe de deux.](#)

Vision d’ensemble du projet : il s’agit de modéliser de manière simple le déplacement d’une foule. Nous sommes dans le cadre d’un **cours programmation concurrente** et donc **ce qui est important** ce n’est ni l’IHM, ni l’évolutivité du code **mais bien l’identification des contraintes liées au parallélisme et aux accès concurrent** afin d’y apporter une réponse adéquate.

Le langage de programmation est nécessairement Python.

Spécification de la mise en œuvre

Le terrain sur lequel se déplace la foule fait 512 x 128 et sera représenté par une matrice de taille identique ayant 512 colonnes (numérotées de 0 à 511) et de 128 lignes (numérotée de 0 à 127). Des obstacles de taille quelconque seront présent sur le terrain. En aucun cas ces obstacles touchent le bord du terrain. La sortie sera le coin supérieur gauche et fera 2 * 2. La figure 1 ci-dessous représente le terrain.

Afin de rendre le code modulable, vous pourrez utiliser les variables suivantes :

- Terrain = [512, 128] # Taille du terrain
- Sorties = [[0, 0], [0, 1], [1, 0]] # Cases permettant à une personne de sortir
- Obstacles = [[xmin₁, ymin₁, xmax₁, ymax₁], ..., [xmin_k, ymin_k, xmax_k, ymax_k]] # Taille des différents obstacles
- Personnes = [[x₁, y₁], ..., [x_p, y_p]] # Coordonnées des personnes

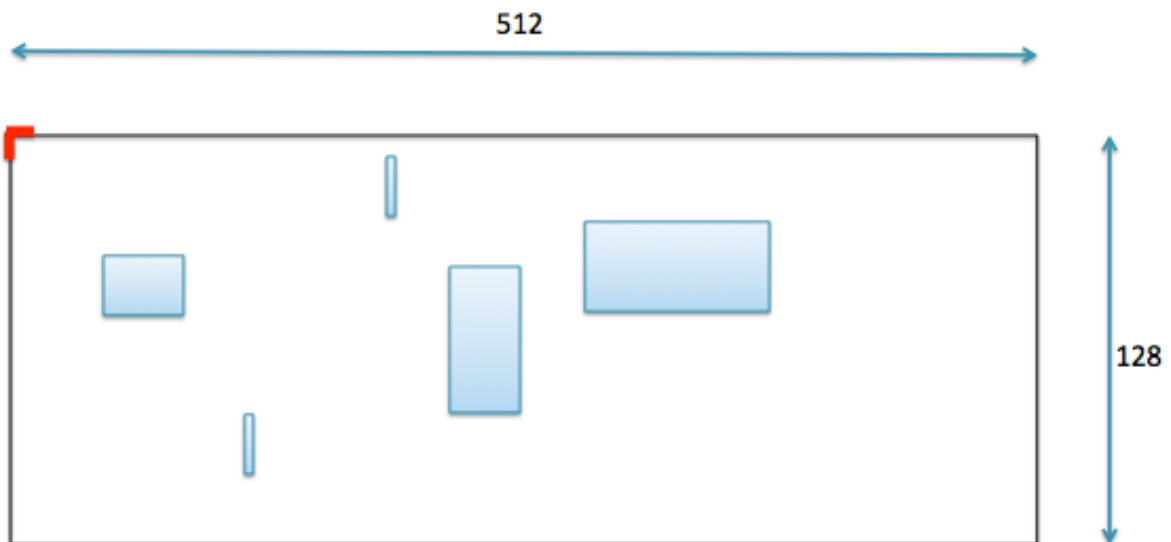


Fig. 1 – forme du terrain

Une personne occupe une cellule de la matrice et se déplace d’une seule case selon les directions Nord, Sud, Est et Ouest ainsi que les 4 diagonale Nord-Sud, Nord-Ouest, Sud-Est et Sud-Ouest.

Après une phase d’initialisation qui met en place le terrain et qui distribue aléatoirement P personnes, le simulateur fait avancer chacune des personnes vers une sortie. A chaque déplacement la personne doit se

rapprocher une sortie. Bien évidemment une personne ne peut pas occuper une place déjà occupée.

Afin de comparer les différentes mises en œuvre, il est souhaitable qu'à chaque exécution, les personnes partent de la même place.

Mises-en œuvre : vous allez programmer un simulateur ultra-simpliste du déplacement d'une foule. Toutes les versions seront intégrées dans le même code et des options du programme permettront de choisir la version à exécuter :

-p [0123456789] : nombre de personnes présentes sur le terrain

- p varie de 0 à 9 et le nombre de personnes vaut 2^p (i.e. le nombre de personnes varie de 1 à 512)

-t [012] : scénario de créations des threads

- -t0 : une seule thread
- -t1 : une thread est associée à chacune des personnes créées et chaque thread doit faire avancer la personne qu'elle gère ;
- -t2 : le terrain est partagé en 4 parties égales. Une thread est associée à chaque partie du terrain et chaque thread doit faire avancer successivement chacune des personnes présentes sur la portion de terrain que la thread gère.

-m : mesure du temps d'exécution

- mesure la consommation du CPU et le temps de réponse du programme
- lorsque des mesures sont effectuées : la phase d'initialisation du programme n'est pas prise en compte et l'affichage n'est pas actif
- pour effectuer les mesures, l'application est lancée 5 fois et la mesure est la moyenne des 3 valeurs intermédiaires

Le rendu du projet :

Le rapport doit contenir pour chacune des étapes :

- les algorithmes (pseudo code) mis en œuvre
 - insister en particulier sur le respect des conditions d'accès aux variables partagées. Pour cela on décrira ces variables et les stratégies mises en place pour contrôler les accès.
- présenter chaque fois que cela est nécessaire les principes de fonctionnement des bibliothèques utilisées et faire la comparaison avec les modèles vu en cours. On s'intéressera bien évidemment à la mise en œuvre des processus, thread et des primitives de synchronisation verrou, sémaphore, moniteur, etc.
- décrire les performances obtenues afin de comprendre comment évoluent le temps de réponse de l'application et le temps de consommation CPU en fonction du nombre de personnes. Il s'agit d'avoir un esprit critique sur la mise en œuvre et de trouver des justifications à ce que l'on observe.
- vous rendrez 2 rapports (avec le code associé)
 - le premier portera sur la mise en œuvre du scénario t0 (1 seule thread) dans un mois.
 - Ce rapport ne sera pas noté mais permettra de mieux affiner les attentes en termes de rendu.
 - le second, sera un rapport global et portera sur les trois scénarios avec une comparaison des différentes stratégies à la fin du cours.
 - Ce rapport sera noté.