Projet présenté par :

BENLOULOU Sarra NEKKAA Yousra

-

Université Clermont Auvergne

Institut d'Informatique

Département de l'informatique

Réseau: Un protocole de communication graphique

Encadré par :

MBIADO SALEU Gertrude Raissa

Année Universitaire : 2018-2019

Plan:

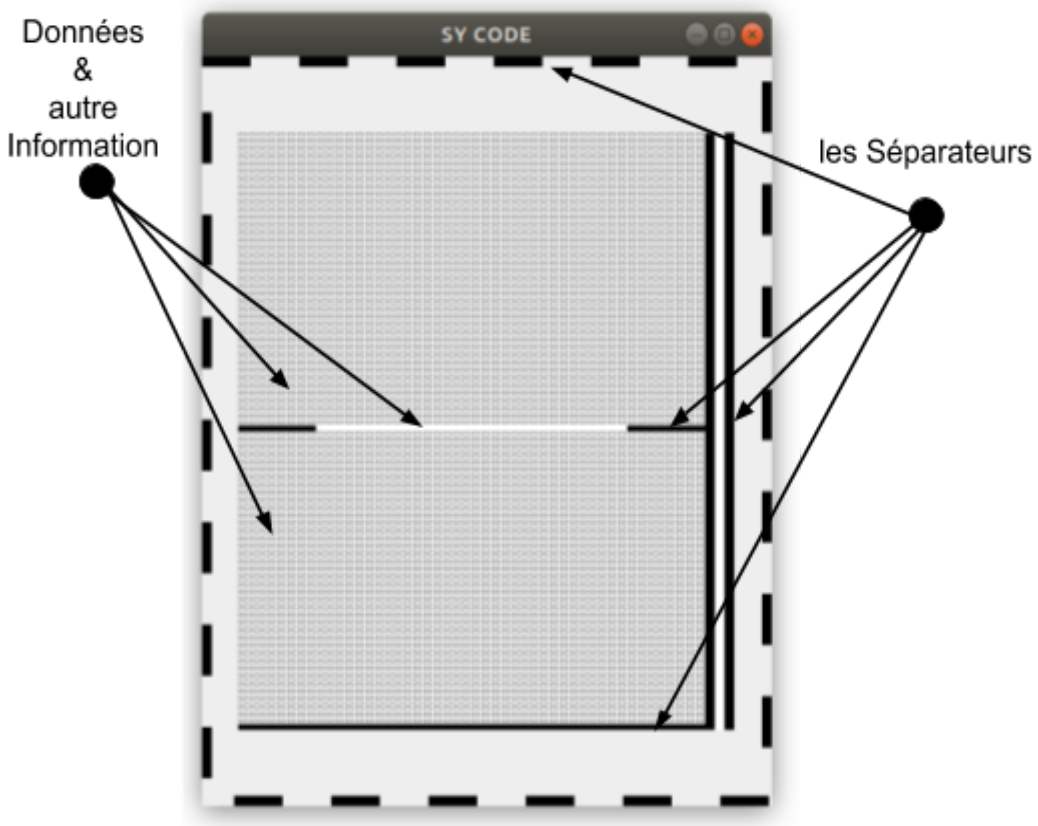
- **Définition**
- **Structure générale de SY-CODE**
- **Stockage des données**
  - **La capacité**
- **Fonctionnement**
- **Aspect technique**
- **Points à améliorer**
- **Retour d'information (FeedBack)**

- **Définition :**

Pour réaliser ce projet “ création d’un protocole de communication graphique “,nous avons défini notre propre protocole que nous avons appelé SY-CODE.
SY-CODE est basé sur une matrice 102x51 codée de 0 , 1 et -1.
Nous avons choisi de coder les données en binaire, où nous écrivons le code ascii de chaque caractère en binaire , chaque caractère est codé sur 8 bits. Une cellule de la matrices qui contient -1 ne porte aucune information et ne stocke pas de données, (toute la matrice est initialisée au départ à -1) .
Nous avons présenté notre matrice graphiquement avec un pixel blanc pour désigner une cellule qui contient 0, un pixel noir pour 1 et un pixel noir et blanc pour -1 .

- **Structure générale de SY-CODE :**

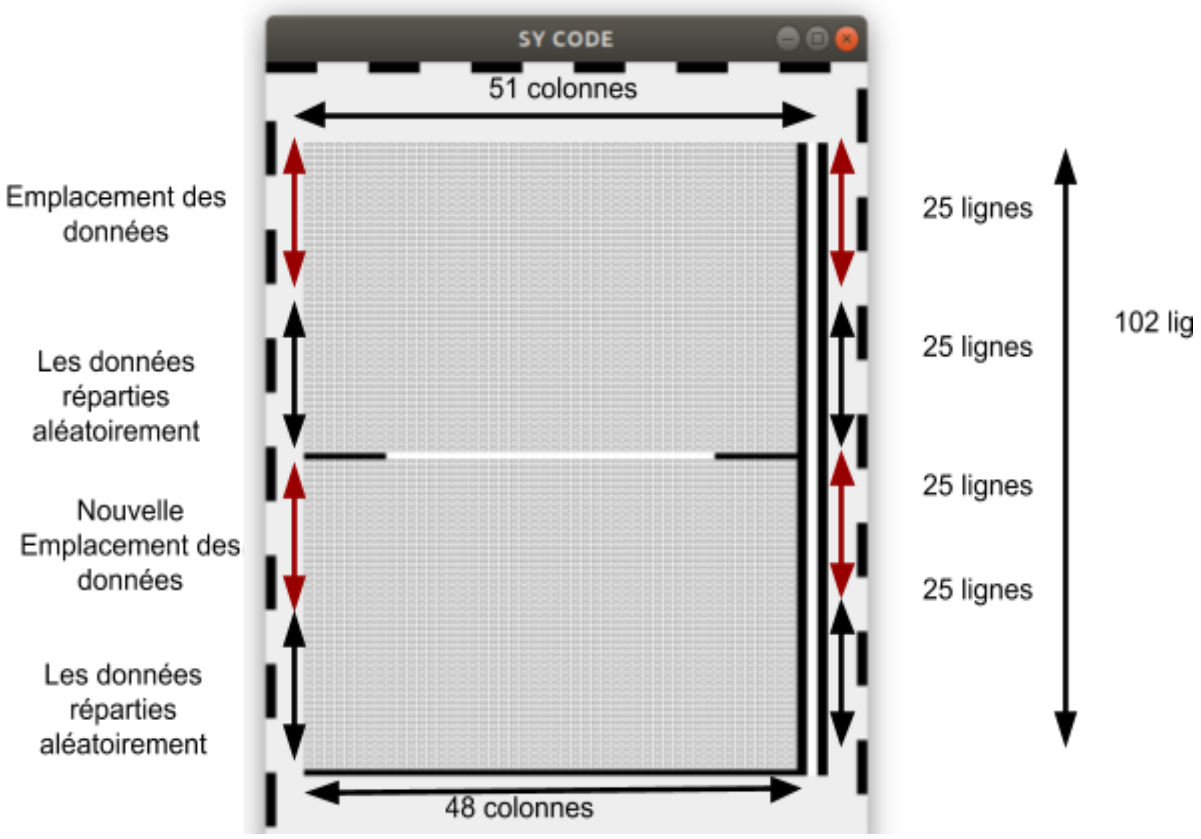
La première étape pour réaliser ce projet était de réfléchir à la structure générale de notre protocole. La question qui se posait était de comment faire pour différencier notre protocole de communication graphique d’un autre. pour cela nous avons mis en place certains séparateurs pour faciliter au lecteur la détection de SY-CODE qui peut se retrouver dans une feuille qui peut contenir d’autre informations.
Le premier séparateur est représenté comme étant des traits noir et blanc qui entourent les données, ce séparateur est suivi d’un espace blanc, puis nous retrouvons à droite du code trois colonnes : noir, blanc puis noir, ces trois colonnes permettent au lecteur de déterminer la fin de chaque ligne. Au centre du code nous retrouvons une ligne composée d’un trait noir, un trait blanc puis un autre noir, c’est l'équivalent de 8 bits composés de 1, ensuite 16 bits composés de 0 et enfin 8 bits de 1. Cette ligne permet de centrer le code au lecteur de scan, l’espace blanc ne contiendra pas toujours des 0, il contiendra également une information importante que nous détaillerons plus tard. Enfin, une ligne noir (48 bits de 1) qui détermine la fin de la lecture de chaque colonne.(Figure 1)



SY-CODE vide

- **Stockage des données :**
    - **La capacité :**

La matrice sans séparateurs est de taille 48x100, en effet elle contient deux partie, une partie haute et une autre basse.
La partie haute de la matrice est de taille 50x48, elle aussi composée de deux sous parties “emplacement” et “données” de taille 25x48 chaque une.
Notre matrice peut stocker jusqu'à 150 caractères, dans la partie donnée nous avons 25*48 = 1200 bits , 1200 / 8 bits (la taille d’un caractère) = 150 caracteres .
Nous avons besoin du même nombre de bits pour stocker l’emplacement des caractères dans la partie “Emplacement” d’où la taille est de 25x48.



- **Fonctionnement :**

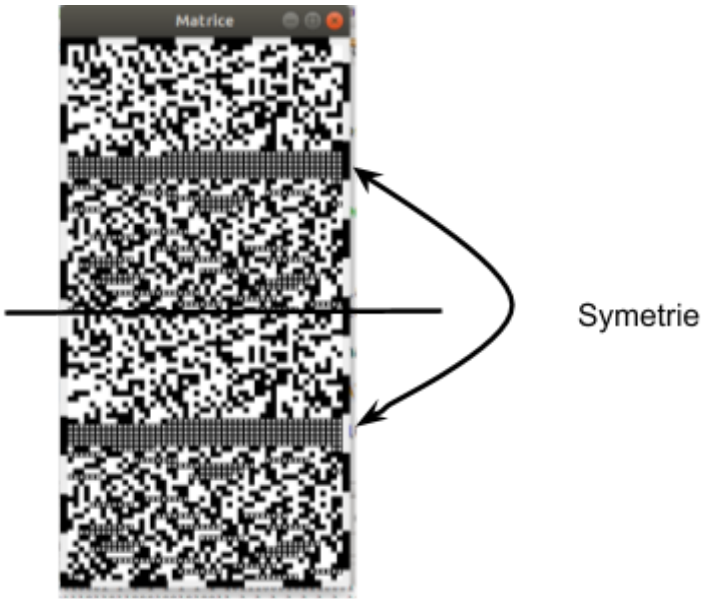
Si nous voulons stocker le mot “abc”, le protocole va tirer au hasard un nombre aléatoire entre 0 et 150 pour chaque caractère, ce nombre assignera l’emplacement du caractère dans la matrice. Exemple : 52 pour le caractère ‘a’, 2 pour ‘b’, et 100 pour ‘c’.

Dans la partie “Emplacement” nous plaçons ces nombres en respectant l’ordre : 52,2,100. (stocker en binaire sur 8 bits).

Dans la partie “Données” nous plaçons dans le 52 ème octets le caractère ‘a’ (codé en binaire sur 8 bits), dans le 2ème octets le caractère ‘b’ et enfin le caractère ‘c’ dans le 100 ème octet .

Cette stratégie nous garantie de récupérer le maximum de données au cas où une partie de notre SY-CODE est cachée ou abimée par exemple.

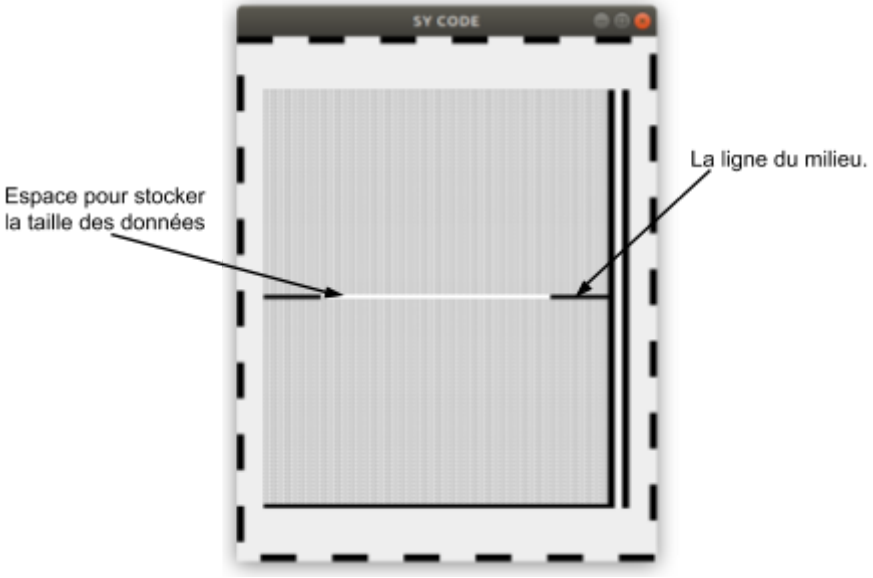
En effet, si les caractères sont stockées successivement, ‘a’ ‘b’ ‘c’ le risque de leur perte est beaucoup plus grand que celui de l’emplacement aléatoire de notre protocole .
Et si jamais la partie “emplacement” est perdue de notre protocole, comment peut-on récupérer les données ?
C’est pour cette raison que nous avons ajouté la partie basse de notre matrice, où nous faisons de la redondance. Au départ, nous avons copié exactement la partie haute dans la partie basse. Nous nous sommes retrouvées avec une symétrie par rapport à la ligne centrale horizontale .



Premiere version du SY-CODE

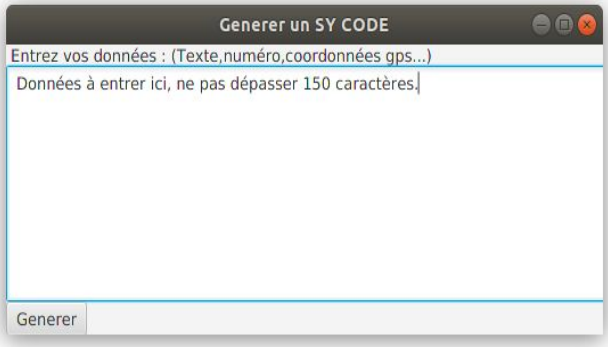
Cette symétrie ne joue pas à notre faveur, car si nous perdons une ligne verticale de notre SY-CODE, les même caractères seront perdus de la partie haute et basse, nous ne pourrons donc pas les retrouver même si nous aurions fait de la redondance.
Voilà pourquoi nous avons mis un nouveau emplacement dans cette partie. C’est à dire, notre exemple ‘abc’, nous donnera par exemple : 16 pour ‘a’, 30 pour ‘b’ et 120 pour ‘c’, les données seront stockées donc différemment dans la partie haute et il n’y aura plus de symétrie.

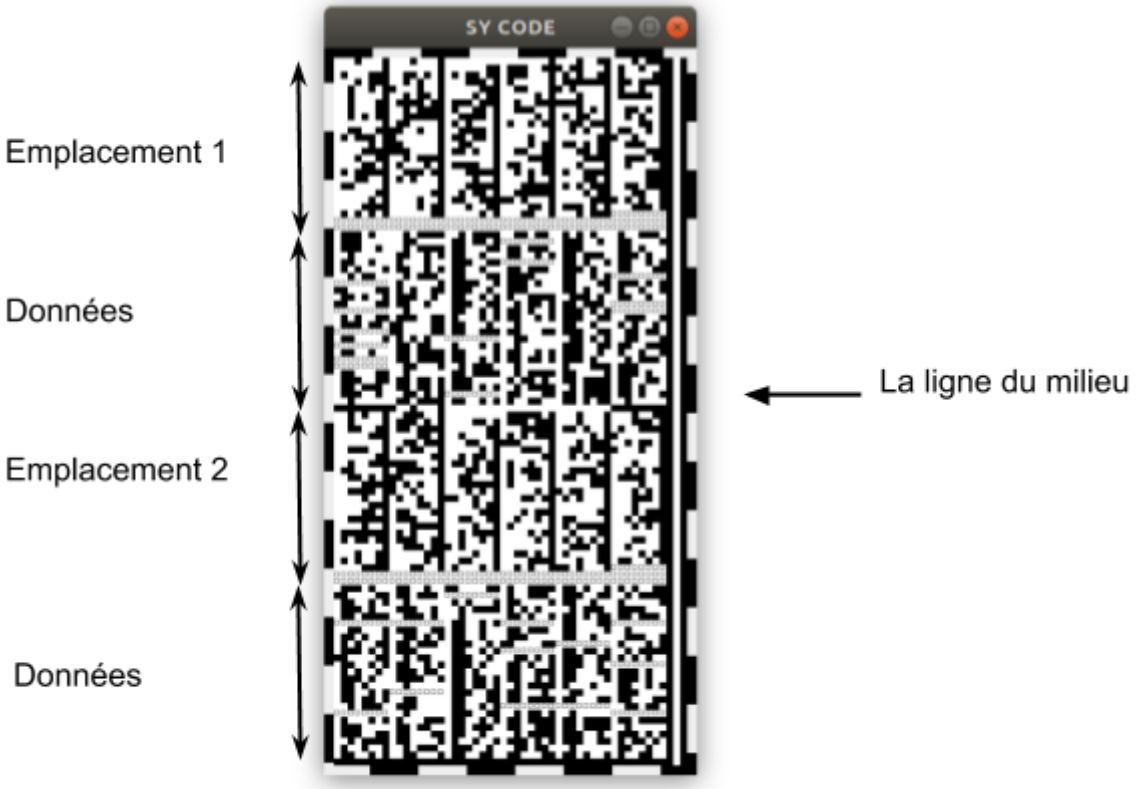
La ligne au milieu représente une séparation entre la partie haute et basse de notre matrice, cela aidera donc le lecteur de scan à bien visualiser les deux parties.
Dans l'espace blanc que nous avons évoqué précédemment nous stockerons la taille des données entrées par l'utilisateur, cette information aidera le lecteur du scan à vérifier si les données retrouvées sont complètes.



- **Aspect technique :**

Nous avons réalisé ce projet en langage objet java où nous avons un choix en terme de bibliothèques graphiques, nous avons utilisé la bibliothèque javafx pour réaliser tout ce qui concerne la partie graphique du projet .




SY CODE
Emplacement 1
Données
Emplacement 2
Données
La ligne du milieu

- **Points à améliorer :**

Si jamais nous continuons à travailler sur ce projet, on pourra améliorer les points suivants:

  - Optimiser tout l'espace libre de la matrice et ne pas laisser des cellules qui contiennent -1, ces cellules qui ne livrent aucune information.
  - Gérer de manière efficace l'emplacement aléatoire des caractères, nous pourrions faire en sorte d'éloigner les caractères les uns aux autres dans la matrice. Avec cette stratégie nous pourrions récupérer un maximum de données au cas où une partie de code n'est plus visible.

- **Retour d'information (FeedBack):**

Ce projet assez créatif nous a permis de mieux connaître les protocoles de communications graphiques, il nous a également permis de mettre en place plusieurs stratégies et de choisir la meilleure et aussi d'améliorer nos connaissances dans le langage orienté objet java, et surtout d'utiliser pour la première fois une bibliothèque graphique java.