Rapport du projet de Réseau II

Mohammed Damene
Valentin Loirette

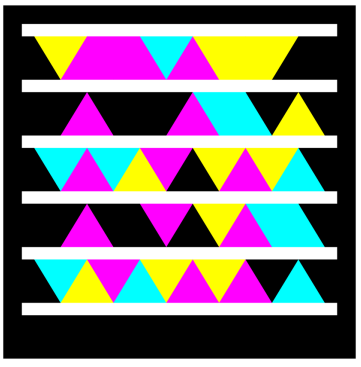
Pour notre projet de réseau nous avions comme consigne de créer un protocole de communication graphique, similaire au QR Code.

I Premier Prototypes

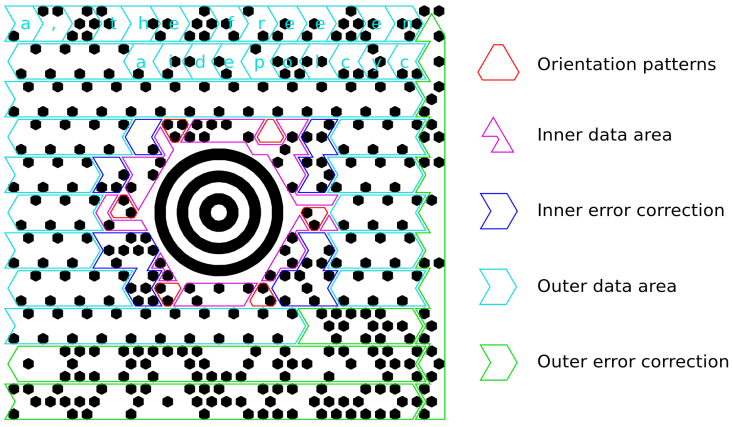
Dans un premier temps, nous avons fait des recherches sur les différents types de code de communication graphique existant. Nous avons tout de suite été intéressé par le Maxicode ainsi que par le HCCB. Nous avons donc pris la décision de nous inspirer de ces deux codes là pour créer le nôtre.

Pour la structure de notre code, c'est le HCCB qui nous a le plus inspiré. Tout comme lui, nous somme partie sur une base de ligne de triangles de couleur.

Pour la partie codage de l'information nous nous sommes inspirés du Maxicode. Nous avons donc repris le codage par bloc de point en mettant les points dans des triangles.

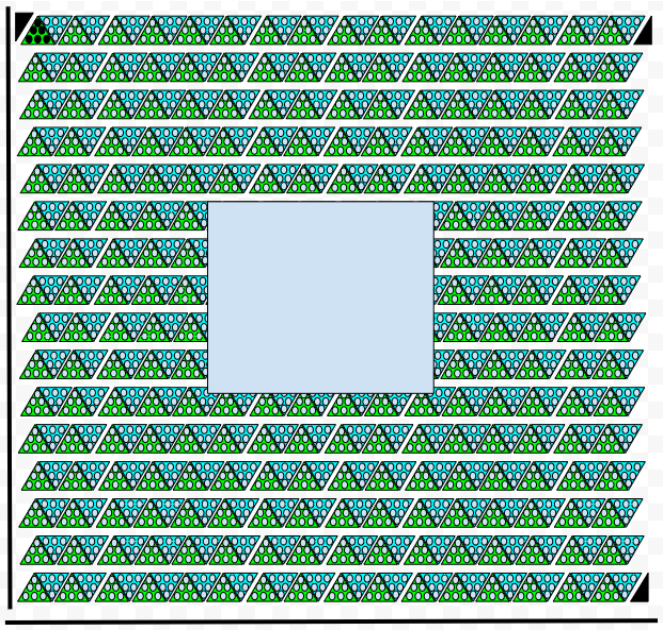


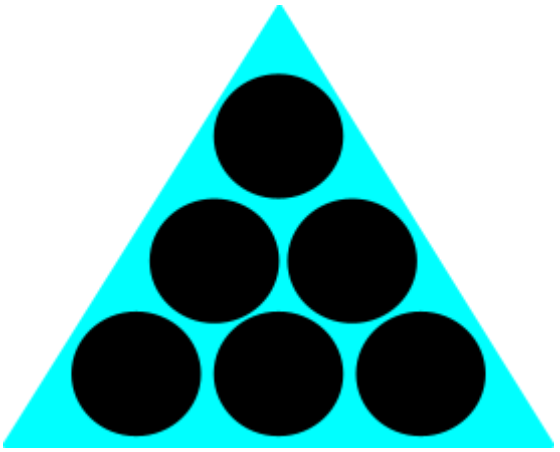
HCCB



Maxicode

Voici donc le premier prototype de code que nous avons fait simplement avec la fonction dessin de Google Doc :





Nous avons donc fait des lignes de triangles colorés, composé chacun de 6 points noir. Ces triangles sont encadrés par 3 bornes qui déterminent l'orientation de l'image. Dans ce prototype, le rectangle bleu au milieu représente un potentiel logo.

II Réalisation

Pour coder notre protocole de communication graphique, nous avons choisi d'utiliser le python, avec la librairie PIL (Python Imaging Library), ce qui nous simplifie grandement la partie programmation graphique.

III Fonctionnement

Étant donné la forme de notre protocole graphique, nous avons choisi de coder un caractère par triangle. Dans chaque triangle, nous avons 6 points, qui chacun codent 1 bit. Chaque triangle est d'une couleur, nous avons 4 couleurs de triangle disponible ce qui nous ajoute 2 bits de plus. Chaque triangle constitué de points code donc 8 bit d'information.

Nous avons choisi d'utiliser la table ascii pour coder les caractères, ce qui fait que chaque caractère est codé sur 7 bit.

Notre protocole se devant d'avoir une certaine sécurité, nous avons décidé d'utiliser le code correcteur Reed Solomon, notamment utilisé par les

QRCode, ainsi qu'un système de redondance qui nous permet de lire le code même si une partie de l'image est corrompue.

Aux quatres coins de notre protocole se trouvent des polygones composés de 3 carrés colorés qui permettent à un appareil de connaître l'orientation du protocole. Entre ces polygones se trouvent une succession de carrés blanc et noir, qui donnent une série d'information à l'appareil qui lit le protocole. Ces informations commencent à droite du polygones cyan et contiennent, dans cet ordre :

- la version du protocole sur 3 bit
- le nombre de caractère du message sur 10 bit
- la redondance sur 4 bit
- la taille sur 7 bit
- la présence du logo sur 1 bit

Ces informations sont écrites avec une redondance pour garantir leur lecture. Le nombre de redondances se calcule automatiquement en fonction de la taille des carrés et de l'espace disponible pour écrire le message.

Lors de la création d'une image, l'utilisateur choisit le nombre de redondances qu'il souhaite en fonction du nombre de caractères du message qu'il veut entrer. Grâce à ça, il peut se permettre de rentrer un message allant jusqu'à 728 caractères.

IV Version sans Interface

La façon de base pour créer une image est de passer directement par le terminal et par le code source.

Pour créer une image, il faut :

- lancer le programme python
- entrer le nombre de redondances voulu en s'aidant du tableau au dessus
- Entrer le message, (la limite du nombre de caractère se réaffiche au dessus)
- Entrer Yes ou No (casse non importante) pour indiquer si on veut mettre un logo où non.

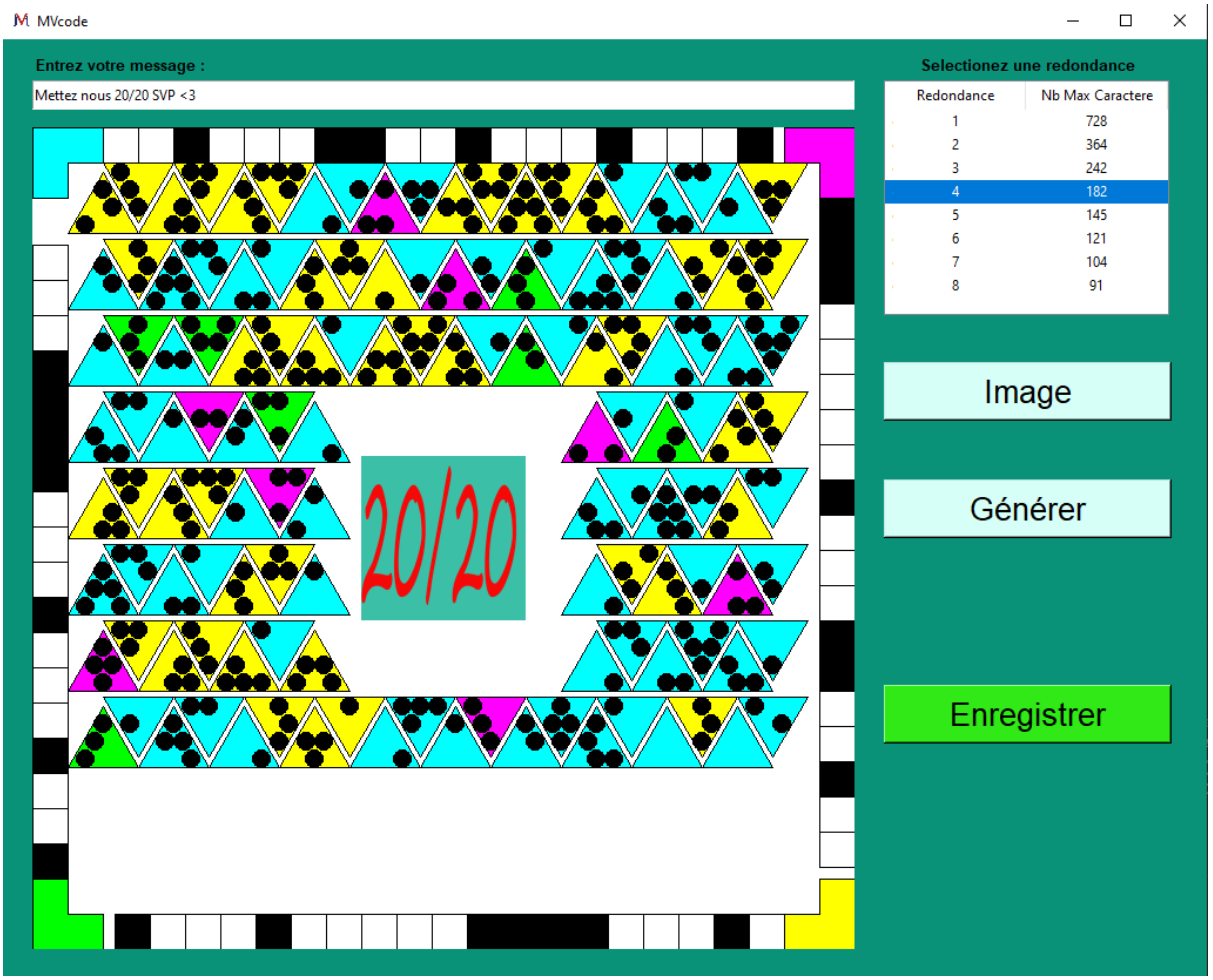
Dans le cas où on veut mettre un logo, il faut penser à donner le lien de l'image dans la variable LogoLink (environ ligne 219).

Après avoir fait tout ça, l'image s'affiche et s'enregistre dans le dossier source sous le nom MVcode.png.

Ensuite, les informations de code s'affichent dans le terminal.

V Version avec Interface

Pour faciliter la création d'une image à l'aide du protocole, nous avons choisi de créer une interface graphique. Cette interface graphique, créée grâce à tkinter, permet à l'utilisateur de rentrer son message, choisir le nombre de redondances, ajouter un logo/une image, générer l'image et d'enregistrer



Nous avons pris la décision d'appeler ce protocole MVcode, à partir de nos initiales (Mohammed et Valentin).