

# Rapport de Soutenance

## QrCode Project

23 Mai 2023

---



Alexandre GIL - Matthias REY - Geoffrey ELBAZ - Clément LABBE

# Sommaire

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>État de l'art</b>	<b>6</b>
2.1	Qu'est ce qu'un QrCode ? . . . . .	6
2.2	Technologies Connues . . . . .	6
2.3	Notre Technologie . . . . .	7
2.3.1	Analyse du QrCode pour obtenu l'URL du site	8
2.3.2	Création d'un QrCode à partir d'un lien . . .	9
<b>3</b>	<b>Planning Prévisionnel</b>	<b>11</b>
<b>4</b>	<b>Objectif du projet</b>	<b>12</b>
<b>5</b>	<b>Etapas Réalisées</b>	<b>13</b>
5.1	Interface graphique . . . . .	13
5.2	Traitement de l'image . . . . .	17
5.2.1	Procédé . . . . .	17
5.2.2	Tranformation en matrice . . . . .	19
5.3	principe algorythmique de la lecture du Qrcode . . .	20
5.3.1	Détection des motifs d'alignement . . . . .	20
5.3.2	Extraction de la zone de format . . . . .	22
5.3.3	Extraction des données . . . . .	23
5.3.4	Correction des erreurs . . . . .	24
5.3.5	Interprétation des données . . . . .	25

5.4	Binarisation des données . . . . .	28
5.5	Creation du QrCode . . . . .	35
5.6	Transformation en image . . . . .	38
<b>6</b>	<b>Site Web</b>	<b>40</b>
<b>7</b>	<b>Planning Prévisionnel</b>	<b>43</b>
<b>8</b>	<b>Nos peines et joies</b>	<b>44</b>
8.1	Clément . . . . .	44
8.2	Alexandre . . . . .	45
8.3	Matthias . . . . .	47
8.4	Geoffrey . . . . .	48
<b>9</b>	<b>Conclusion</b>	<b>51</b>

# 1 Introduction

Le présent rapport décrit le développement et l'implémentation d'un projet innovant de création et de lecture de codes QR. Ce projet a été conçu dans le but de fournir une solution efficace et conviviale pour générer et interpréter les codes QR, qui sont devenus omniprésents dans notre société moderne.

Les codes QR, ou Quick Response codes, sont des codes-barres bidimensionnels qui peuvent être scannés à l'aide de smartphones, de tablettes ou d'autres dispositifs équipés de caméras. Ils sont utilisés dans une multitude de domaines, tels que la publicité, le marketing, la logistique, et même dans les processus de paiement. Leur popularité réside dans leur capacité à stocker de grandes quantités d'informations, qui peuvent être rapidement et facilement extraites par le biais d'une simple capture d'image.

Ce projet vise à créer une plateforme permettant aux utilisateurs de générer des codes QR personnalisés pour leurs besoins spécifiques, qu'il s'agisse de liens vers des sites web, de coordonnées de contact, de données d'identification ou d'autres types d'informations. De plus, notre plateforme offre une fonctionnalité de lecture de codes QR, permettant aux utilisateurs de décoder les informations contenues dans les codes QR scannés.

Pour réaliser ce projet, une approche basée sur les technologies

de pointe a été adoptée. Nous avons utilisé des langages de programmation modernes et des bibliothèques spécifiques pour la génération et la lecture de codes QR. Notre objectif était de créer une solution performante, fiable et facile à utiliser, tout en maintenant une flexibilité suffisante pour s'adapter aux besoins évolutifs des utilisateurs.

Ce rapport détaillera les différentes étapes du développement, les choix techniques effectués, ainsi que les tests et les résultats obtenus. Nous analyserons également les défis rencontrés tout au long du processus de développement, les leçons apprises et les améliorations potentielles pour l'avenir.

En conclusion, ce projet offre une solution complète et polyvalente pour la création et la lecture de codes QR. Nous croyons fermement que cette plateforme sera d'une grande valeur pour les utilisateurs, en leur offrant une méthode simple et efficace pour partager des informations tout en tirant pleinement parti du potentiel des codes QR.

## 2 État de l'art

### 2.1 Qu'est ce qu'un QrCode ?

Le QrCode, en forme longue quick response code (code à réponse rapide ), est un type de code-barres à deux dimensions constitué de modules-carrés noirs disposés dans un carré à fond blanc. Ces points définissent l'information que contient le code. Ce dessin, lisible par machine, peut être visualisé sur l'écran d'un appareil mobile ou imprimé sur papier.

Il existe d'autres formats de code-barres deux dimensions au fonctionnement similaire, comme le Flashcode ou encore le 2D-Doc, confondus avec le code QR.

### 2.2 Technologies Connues

Aujourd'hui la majorité des solutions se font sur téléphone, cependant il arrive parfois que nous ayons besoin de scanner un QrCode sur un ordinateur.

En général, toutes les applications mobiles ou logiciels permettant la lecture puis l'analyse d'un QrCode fonctionnent de la même manière. En effet, le décodage et la lecture d'un QrCode suit de manière générale un algorithme universel puisque le QrCode est

très standardisé (à l'image du code barre par exemple). Ainsi, les différentes applications ou logiciels assurant la gestion de QrCodes se démarquent par leurs interfaces, leur ergonomie, leurs fonctionnalités...

Cependant, nous n'avons souvent pas la possibilité d'effectuer l'analyse et la génération d'un QRcode sur une seule application ou logiciel. En effet, nombre d'applications (surtout sur mobile) proposent la lecture de QrCode. D'ailleurs, de nombreux smartphones proposent cette fonctionnalité de base sur leurs systèmes. Seulement, rares sont les applications ou logiciels qui proposent aussi la création de codes à partir de liens. On doit souvent se diriger vers des solutions à part ce qui peut être fastidieux.

Notre but va donc être de rassembler les forces de ces applications et d'effacer leurs faiblesses afin de créer un logiciel tout-en-un permettant de faire tout cela sur son ordinateur de manière très ergonomique.

## **2.3 Notre Technologie**

Notre objectif est d'arriver à scanner un QrCode à partir d'une sélection de l'écran avec la souris de l'utilisateur. Une fois cette image sauvegardée, le but sera d'isoler le QrCode pour ensuite en extraire les données.

### 2.3.1 Analyse du QrCode pour obtenu l'URL du site

Une fois le QrCode détecté une sera analysé pour obtenir le lien contenu par le QrCode.

voici ci-dessous la suite d'étapes que nous allons suivre pour lire et afficher le resultat soit ici le site encodé par le QrCode

1. Dans un premier temps nous allons reconnaître les bits 1 ou 0 par la valeur RGB de chaque pixel du QrCode.
2. Puis nous identifierons le taux de code correcteur du QrCode pour ne pas comptabiliser plusieurs fois le même bit ou un bit de redondance.
3. Ensuite l'objectif sera de détecter la version du QrCode choisie. Elle fera varier le nombre de modules dont on a besoin pour décoder nos informations.
4. On ignorera les patterns de références trouvés qui nous auront servis de point de repère pour la détection du QrCode.
5. Il faudra ensuite détecter les erreurs et les corriger si certaines sont présentes.
6. Enfin vient la dernière étape, et pas des moindres, celle du décodage des données. Les données seront donc converties en binaire à partir des valeurs RGB des pixels du QrCode. Puis



en caractère à partir du code ASCII, pour en obtenir l'URL décrypté et afficher le résultat à l'utilisateur.

### **2.3.2 Création d'un QrCode à partir d'un lien**

Comme son nom l'indique cette partie portera sur la création d'un QrCode à partir d'un lien inséré par l'utilisateur.

1. Dans un premier temps, nous effectuerons une analyse des données à encoder et paramétrage du niveau de code correcteur. Le but est d'analyser le flux de données d'entrée pour identifier la variété des caractères différents pour être encodés. Si l'utilisateur n'a pas spécifié le niveau de code correcteur, la plus petite version de QrCode sera sélectionnée pour accueillir les données.
2. Il faudra ensuite convertir les caractères de données dans un flux de bytes. Par exemple en code ASCII puis en binaire pour obtenir une série de 1 et de 0.
3. Nous implémenterons la correction des erreurs. Le but est de séparer par blocs les bits de données et de générer leurs codes correcteurs. On placera donc un bit d'information et autour, on générera du code correcteur.
4. Nous insérerons les données avec le code correcteur dans la matrice.

5. Nous générerons la matrice et évaluerons le résultat qu'elle retourne. On optimisera ici la balance entre les modules noirs et les modules blancs.
6. Enfin la dernière étape sera de générer le QrCode au format image. C'est le résultat final qui pourra être lu par un lecteur de QrCode et sauvegarder.

### 3 Planning Prévisionnel

Tâche à réaliser	Alexandre	Matthias	Geoffrey	Clément
Sélection Écran	O			X
Détection QrCode		O	X	
Analyse du QrCode		X	O	
Création du QrCode	X			O

Table 1: Répartition des tâches (O : Suppléants)

Avancement Prévu	Soutenance 1	Soutenance 2	Soutenance 3
Sélection Écran	40%	80%	100%
Détection QrCode	15%	50%	100%
Analyse QrCode	30%	70%	100%
Création QrCode	50%	80%	100%

Table 2: Planning prévisionnel

## 4 Objectif du projet

Finalement, notre projet est assez ambitieux mais néanmoins réalisable. Nous le trouvons très intéressant tant d'un point de vue technique et pratique que d'un point de vue apprentissage.

Ce projet nous permettra d'en apprendre plus sur une technologie très présente dans notre quotidien, de renforcer nos capacités dans une pléthore de domaines informatiques allant de l'interface graphique à la gestion d'images, et de s'améliorer en travaux de groupes et en mise en place de solutions.

C'est ainsi une très belle opportunité d'essayer d'imaginer et de mettre en place notre propre version de la gestion de QrCodes sur ordinateur.

## 5 Etapes Réalisées

### 5.1 Interface graphique

Dans cette partie, nous allons expliquer comment nous avons créé et implémenté l'interface graphique de notre logiciel.

Tout d'abord, nous devons choisir une librairie adaptée à notre travail. Etant donné que nous avons déjà travaillé dessus au s3, la librairie GTK a été rapidement choisie.

L'objectif de notre logiciel est d'être simple et pratique à utiliser. En effet, notre logiciel doit être un outil rapide à utiliser en cas de QR code à lire sur un ordinateur. Nous avons donc opté pour une interface simple avec peu de boutons et peu de possibilités pour empêcher l'utilisateur de se perdre.

Pour notre implémentation nous avons une fonction `create_general` qui s'occupe de créer la page d'accueil avec nos deux boutons permettant de décoder ou d'encoder un QR code, elle ajoute également le logo de notre projet.



Decoder un QR-Code

Generer un QR-Code

Image de notre page d'accueil

Lorsque l'un des deux boutons est pressé, une des deux fonctions permettant de créer les pages d'encodage ou de décodage (create decode, create encode) sont appelées.

La page d'encodage comporte 2 boutons et un champ permettant à l'utilisateur de rentrer du texte à encoder. Le premier bouton permet de retourner à l'accueil. Le deuxième bouton permet de générer un QR-code à partir des données entrées par l'utilisateur. Ici nous faisons aussi appel aux fonctions d'encodage que nous avons créés. Après l'encodage, le QR code résultat est construit et affiché à l'utilisateur.



Voici la page d'encodage avec un exemple de texte à encoder.

La page de décodage comporte 3 boutons. Le premier permet de revenir à l'accueil. Le deuxième de choisir un fichier. Ici l'utilisateur peut choisir une image de QR-code à travers son ordinateur. Le troisième bouton permet de décoder le QR-code choisi, ce bouton fait appel aux fonctions des autres parties du projet. Une fois le QR code décodé, le résultat est affiché a l'utilisateur.



Voici une image de la page de decodage avec le QRcode généré precedemment

En conclusion, cette partie du projet est parfaitement dans les temps. Pour la prochaine soutenance nous devront implémenter le système de capture d'écran du QR code à décoder directement dans notre logiciel. Nous allons aussi travailler sur l'esthétique du logiciel qui est assez terne pour l'instant.



## 5.2 Traitement de l'image

Pour cette partie une grosse partie du travail était le comment? Nous avons fait pas mal de recherche sur la bibliothèque à utiliser. Pour éviter de tout refaire si la bibliothèque ne convenait pas. Puis suite aux recherches et à nos connaissances nous avons décidé d'utiliser une bibliothèque déjà utilisée auparavant. Il s'agit de SDL car notre réflexion a été que le traitement d'image et l'isolation du QrCode était assez similaire au projet effectué en S3 sur le Sudoku.

### 5.2.1 Procédé

Nous avons effectué un parcours de pixel dont le but est de détecter le carré en haut à gauche du QrCode. une fois le carré détecté et le pattern enregistré. Il nous a fallu faire le même procédé pour le carré en haut à droite. Bien sûr nous vérifions que nos deux carrés ont la même taille.

Pour expliquer un peu l'algorithme une fois le carré gauche détecté on parcourt le QrCode et à chaque pixel l'algorithme teste si le pixel fait parti du carré droit alors l'algorithme va trouver le pixel le plus à droite possible

Voir ci-dessous avec l'image de QrCode avec en rouge la première détection du carré puis en vert sur le carré de droite on observe la détection des côtes.

Ensuite on calcul la largeur du QrCode et on fait le même procédé pour le carré du bas, et on rogne grâce au dimension et aux coordonnées des coins du QrCode.



### 5.2.2 Transformation en matrice

Une fois l'image enregistrer, il faut la transformé en matrice et pour cette étape ca n'a pas été très compliqué car nous n'avons que des pixels noir ou blanc.

Nous avons donc parcourut la liste de pixels de la nouvelle image nommé Pretraitement.PNG et deux conditions avec soit on met dans la matrice un 1 pour signifier un pixel noir soit un 0 pour le pixel blanc.

Un problème que nous avons rencontré est que le carré d'un QrCode ne fait pas un pixel. Nous travaillons donc pour convertir une matrice de pixel en matrice de module.

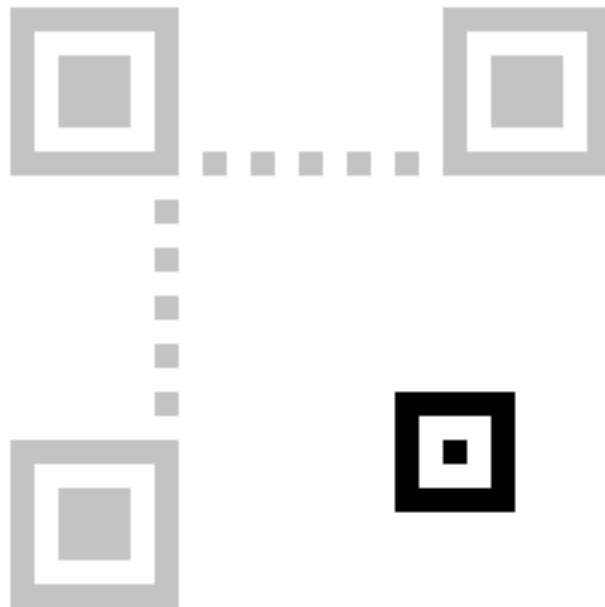
La fonction est partiellement faite mais il reste encore quelques cas à gérer. Mais elle sera complété a la deuxième soutenance.

En conclusion nous avons une matrice de 1 et de 0 qu'il nous reste a convertir en matrice de module !

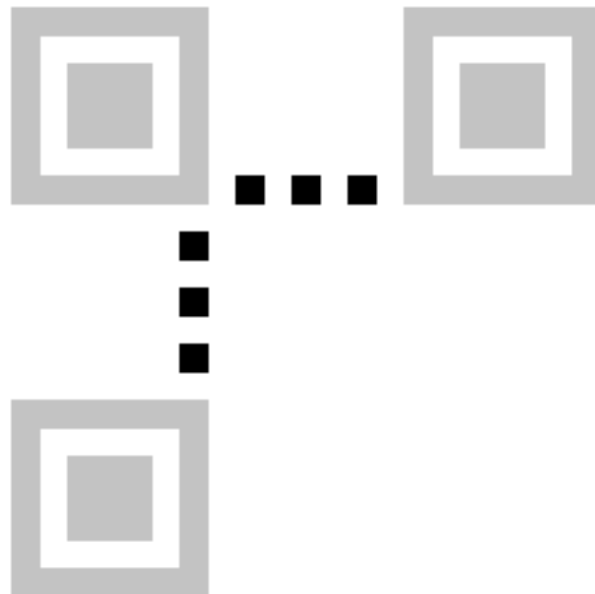
## 5.3 principe algorithme de la lecture du Qrcode

### 5.3.1 Détection des motifs d'alignement

La détection des motifs d'alignement dans un QR code est une étape importante du processus de décodage. Les motifs d'alignement sont des motifs carrés qui permettent d'aligner et de corriger les erreurs dans le QR code.



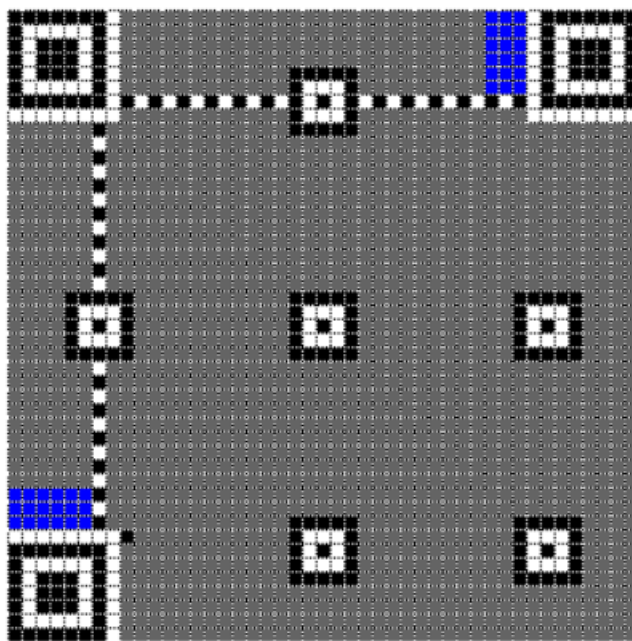
Cependant les motifs d'alignement sont très utilisés pour le passage de l'image d'un QRcode à la matrice de module. A ce niveau de la lecture cette étape a déjà été effectuée. Il nous faut néanmoins les marquer comme vides afin de ne pas les confondre pour des données lisibles. De même pour les zones de synchronisation visible sur cette image ci-dessous.



### 5.3.2 Extraction de la zone de format

Dans cette section le but est a partir de la matrice de module d'un QrCode extraire la zone de format du QR code, qui contient des informations telles que le masque utilisé, la version du QR code et les informations d'erreur de correction. C'est information se situe en haut a droit et en bas a gauche du QrCode.

Nous pouvons voir sur l'image ci-dessous en bleu les différent champ format disponible sur un Qrcode normalisé comme le décrit la norme ISO/IEC 18004 qui spécifie les caractéristiques et les règles pour les Qrcode.

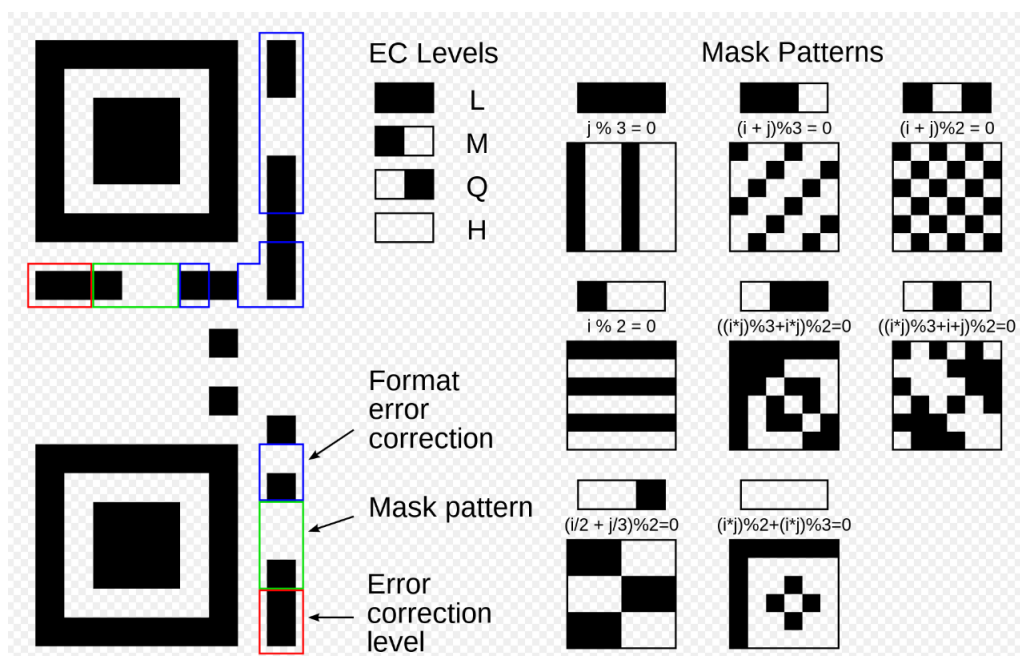


Une fois la position de la zone de format déterminée, il faut extraire les modules correspondants de la matrice de modules du QR code en respectant les dimensions spécifiées par la norme.

la dernière étape concernant la zone format étant la lecture des informations extraite, il nous faut savoir que la structure de cette dernière n'est pas décidé au hasard. Chaque bit sont organisé de la sorte. Les premiers bits de la zone de format indiquent le masque utilisé pour le QR code. La norme spécifie les bits spécifiques qui correspondent aux différentes options de masquage. Ensuite les bits suivants dans la zone de format représentent la version du QR code. La norme spécifie le nombre de bits réservés à la version en fonction de la version du QR code. Pour finir les derniers bits de la zone de format contiennent des informations sur le niveau de correction d'erreur appliqué au QR code, tel que défini par la norme.

### **5.3.3 Extraction des données**

Une fois les modules inutiles aux données retiré, puis, le masque analysé et la version détecté, il suffit de lire le QrCode comme décrit dans la norme ISO/IEC 18004. Cela implique de suivre les lignes verticale, tel un serpent qui remontrais le Qrcode, en partant du coin inferieur droit jusque le coin inferieur gauche dans le cas où la correction d'erreur existe et est prise en compte. comme décrit sur l'image si dessous. Dans le cas ou cette correction est estimées peu neces-



saire, du a la qualité des échantillons possible, nous n'aurions besoin de parcourir qu'une proportion du Qrcode. Cette proportion est bien évidemment déjà déterminé l'ors de la lecture du champ format.

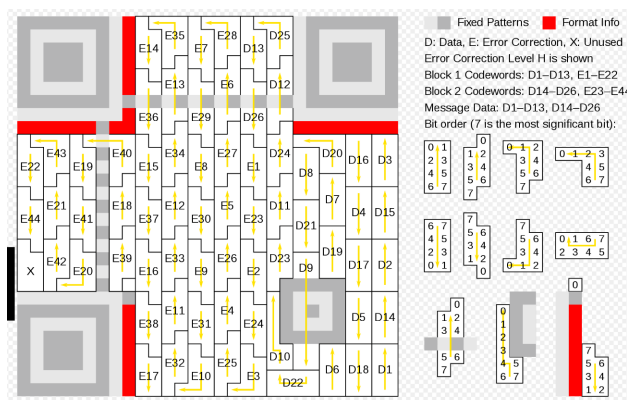
### 5.3.4 Correction des erreurs

Dans le cas où nous le sotions. Si le QR code utilise une correction d'erreurs, il est possible d'appliquer les algorithmes de correction d'erreurs spécifiés dans la norme pour corriger les bits de données endommagés ou manquants.



### 5.3.5 Interprétation des données

Après avoir effectué toutes c'est acrobassi algorithmique, nous devons annalyser afin de les rendre lisible par nous meme. La methode appliqué dependra du type de données encodé dans ce Qrcode. C'est methode se refère a la norme ISO/IEC 18004, qui expicite la manière dont nous devons décoder de l'ascii, du binaire, de l'hexadécimale, ou bien, du décimale.



voici les différentes étapes que nous avons réalisé pour lire notre QrCode pour vous expliquer de maniere simple et précise.

1. Nous analysons la version le mask et la correction d'erreur dans la zone définie du QrCode. Pour connaitre toute ses information avant de le traiter sinon il nous serait impossible de lire le QrCode et nous aurions des résultats tous autant incohérent les un que les autres.
2. Une fois le format enregistré on appliquera le filtre pour lire les données du QrCode qui je le rappelle a été récupéré précédement.
3. Puis Nous lisons les données en partant en bas a gauche puis en remontant. l'objectif lire le QrCode en serpentant de haut en bas. Une des difficultés que nous avons rencontré a été d'ignorer les patrons d'alignement et les patrons pour lire le QrCode. Ce sont de petits carrés au milieu du QrCode qui permettent de justement bien lire celui-ci.
4. Une fois toute la data extraite en ne lisant que la moitié du QrCode, car le reste est surtout de la répétition pour eviter une erreur de lecture. Nous obtenons une série de 0 et de 1.
5. Derniere étape et pas des moindre celle de lire cette série de donnée binarisée. Pour cela il fallait avant tout séparer en block les informations. Nous avons donc appliqué une algorithme propore au QrCode. Le but de clui ci est de faire des calculs

sur les blocs de bits en alphanumérique pour les transformer en caractere.

## 5.4 Binarisation des données

Pour ce qui est de la binarisation des données et donc de l'encodage de données en QrCode, nous avons dû effectuer de nombreuses recherches. En effet, si nous pensions qu'il était facile de trouver des renseignements sur cette partie, ce n'était pas vraiment le cas. L'encodage d'un QrCode est relativement complexe et passe par de nombreuses étapes toutes plus techniques les unes que les autres. Nous avons donc dû retrousser nos manches et chercher les informations au plus profond d'internet.

Après cette longue période de recherche, nous en sommes venus à élaborer une liste d'étapes à faire pour réussir à encoder correctement une URL par exemple. (la taille d'une URL étant parfaitement adaptée à la version de QrCode que nous avons choisi : la version 3).

Voici la liste d'étapes que nous avons et allons suivre :

1. Préparation des données : Il faut d'abord préparer les données à encoder, en l'occurrence l'URL que nous souhaitons encoder.
2. Détermination de la taille du QrCode : La taille du QrCode dépend de la quantité de données à encoder. Il est possible de déterminer la taille en utilisant des algorithmes de calcul de la taille du QrCode. En l'occurrence, nous avons décidé de nous concentrer autour de la version 3 des QrCode, soit une taille de  $29 \times 29 = 841$  modules.

3. Initialisation du tableau de bits : Il faut créer un tableau de bits pour stocker les données encodées. La taille du tableau dépendra de la taille du QrCode déterminée à l'étape précédente. Ici 841 donc.
4. Encodage des données : Il faut encoder les données en utilisant un algorithme d'encodage pour convertir l'URL en séquence de bits. Il existe plusieurs algorithmes d'encodage pour le QrCode, tels que l'encodage alphanumérique, l'encodage numérique et l'encodage binaire. C'est à cette étape que nous nous sommes arrêtés pour la première soutenance. Nous avons choisi l'encodage alphanumérique pour l'URL.
5. Ajout des informations d'en-tête : Il faut ajouter les informations d'en-tête au début du tableau de bits, telles que le type de QrCode, la version, la correction d'erreur, etc.
6. Division en blocs : Il faut diviser les données encodées en blocs pour les organiser de manière à ce que le QrCode puisse être lu correctement.
7. Ajout de correction d'erreur : Il faut ajouter des informations de correction d'erreur aux blocs pour permettre la correction des erreurs qui peuvent survenir lors de la lecture du QrCode.
8. Dessin du QrCode : Finalement, il faut dessiner le QrCode en utilisant les informations stockées dans le tableau de bits.

L'utilisation de fonctions graphiques ou le passage par de l'ASCII sont possibles.

Comme vous avez pu le lire, nous sommes à présent à l'atape 4 sur 8. En résumé, nous sommes capables de créer le tableau de bits représentant le QrCode. Nous pouvons encoder une URL quelconque respectant la taille de la version 3 des QrCodes c'est-à-dire une URL allant de 35 à 77 caractères maximum. Cette URL a été encodée avec un algorithme de codage alphanumérique dont voici le détail :

On récupère une URL sous forme de string qu'on considère valide. On récupère sa taille et on vérifie si elle est paire ou impaire car cela a un impact sur le traitement. Une fois ceci fait, on parcourt l'URL en s'arrêtant tous les deux caractères. On récupère ces caractères et on prend leurs valeurs alphanumériques c'est-à-dire leur nombre dans le tableau alphanumérique de la norme ISO que voilà :

Code	Character	Code	Character	Code	Character	Code	Character	Code	Character
00	0	09	9	18	I	27	R	36	Space
01	1	10	A	19	J	28	S	37	\$
02	2	11	B	20	K	29	T	38	%
03	3	12	C	21	L	30	U	39	*
04	4	13	D	22	M	31	V	40	+
05	5	14	E	23	N	32	W	41	–
06	6	15	F	24	O	33	X	42	.
07	7	16	G	25	P	34	Y	43	/
08	8	17	H	26	Q	35	Z	44	:

Pour obtenir la valeur alphanumérique d'un caractère, nous avons procédé mathématiquement en vérifiant si le caractère actuel était un chiffre ou bien une lettre de l'alphabet. Avec cette information, on peut réussir à déduire la valeur alphanumérique du caractère mathématiquement étant donné que: si c'est un chiffre alors sa valeur alphanumérique est égale à lui-même. si c'est une lettre alors sa valeur alphanumérique est égale à la valeur ASCII du caractère actuel moins la valeur ASCII de 'A' (65) + 10.

Ensuite, il reste à encoder les caractères spéciaux ne répondant à aucun logique en terme de valeur comme '!', '/' ou '+' par exemple. Pour ceux là, pas d'autre choix que de parser l'URL en utilisant un

switch case contenu dans une autre fonction. Ainsi, si le caractère actuelle n'est ni un chiffre, ni une lettre, il est envoyé dans la fonction qui teste tous les caractères possibles et qui renvoie la valeur correspondante.

Une fois la valeur alphanumérique du caractère récupérée, on la multiplie par 45 si le caractère est le premier des deux. Ensuite on additionne le deuxième.

Par exemple, si nous voulons encoder le mot "ca", il faut récupérer la valeur alphanumérique de 'C' (12) puis de 'A' (10). Ensuite, nous effectuons le calcul suivant :  $12 * 45 + 10 = 550$ .

Une fois le calcul effectué, nous passons le résultat dans une fonction qui converti ce nombre en binaire sur 11 bits. Puis nous effectuons cette opération tant que nous n'avons pas fini de parcourir l'URL.

Comme précisé plus haut, si la taille de l'URL est impaire, le traitement est légèrement modifié puisque nous prenons les caractères deux à deux. Dans ce cas, la valeur du dernier caractère de l'URL est encodé sur 11 bits sans être multiplié par 45.

Toutes les valeurs sont ensuite concaténées et on se retrouve avec une chaîne binaire. Cependant il manque quelques informations en début de chaîne. En effet, nous devons rajouter le nombre de caractères à encoder (ici la taille de l'URL) sur 9 bits puis le code du



format utilisé (ici l'alphanumérique) qui est 0010 dans notre cas.

Résumons le tout par un exemple. Si nous voulons encoder le mot "krisboul!", nous allons procéder ainsi : On parcourt le mot en s'arrêtant tous les deux caractères et en encodant comme expliqué plus haut.

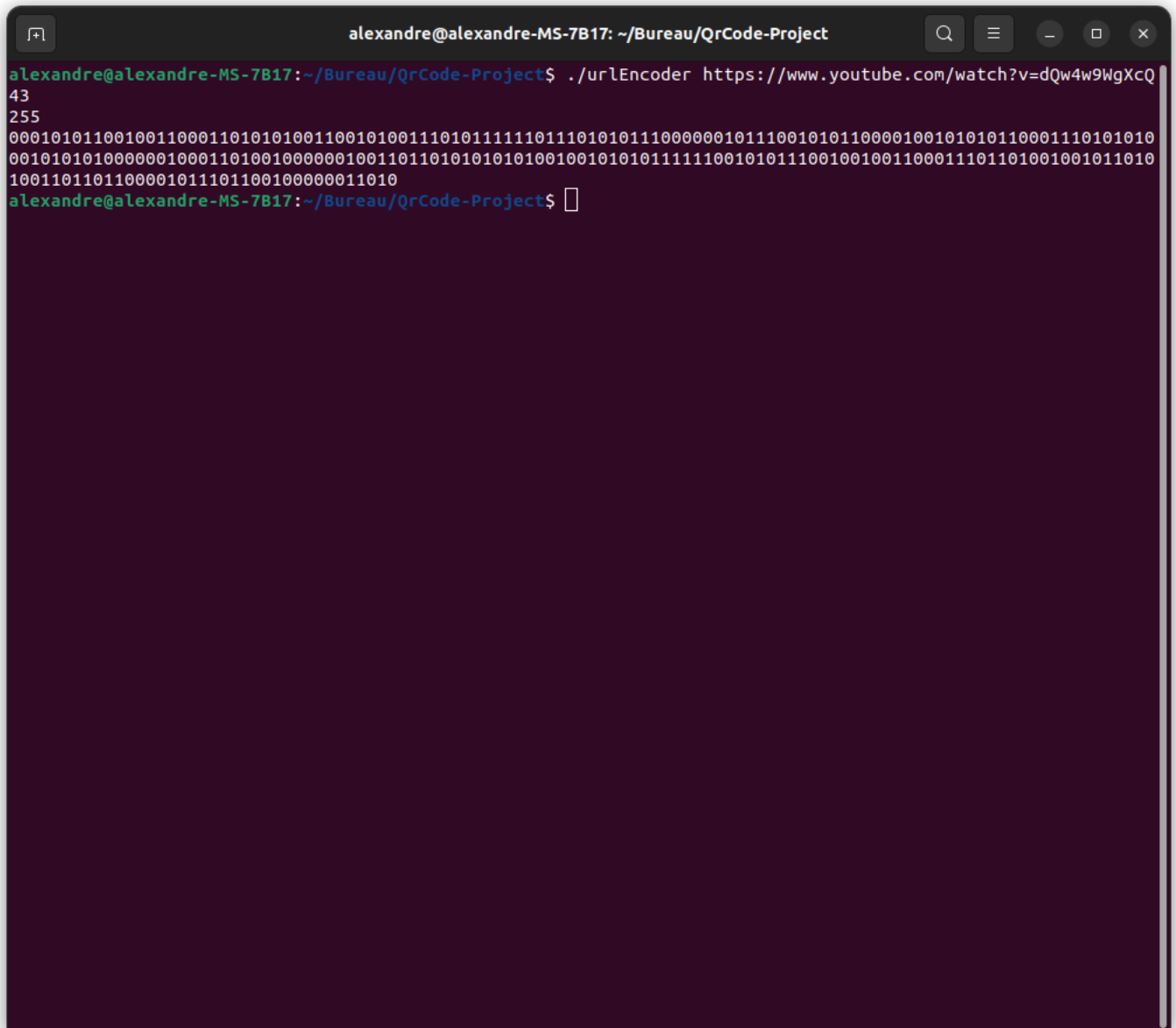
Le résultat final sera donc :

```
000001001 0010 0111001111101101000110010000001111010101101100000000000
```

La première partie correspond à la taille du mot en binaire sur 9 bits ici 9. La seconde correspond au code du format alphanumérique c'est-à-dire 0010. La dernière correspond au mot encodé. Sa taille est cohérente car on sait que 2 caractères forment 11 bits dans le résultat et que le dernier aussi dans le cas d'un mot impair. La taille de cette partie devrait donc être de  $4 * 11 + 11 = 55$  et c'est bel et bien le cas.

Voilà notre avancement sur la partie binarisation des données pour cette première soutenance. Il reste évidemment beaucoup de choses à faire mais nous sommes sur la bonne voie.

Petit exemple du fonctionnement de la fonction d'encodage actuelle:



```
alexandre@alexandre-MS-7B17: ~/Bureau/QrCode-Project
alexandre@alexandre-MS-7B17:~/Bureau/QrCode-Project$ ./urlEncoder https://www.youtube.com/watch?v=dQw4w9WgXcQ
43
255
000101011001001100011010101001100101001110101111101110101011100000010111001010110000100101010110001110101010
00101010100000010001101001000000100110110101010100100101011111001010111001001001100011101101001001011010
1001101101100001011101100100000011010
alexandre@alexandre-MS-7B17:~/Bureau/QrCode-Project$
```

Elle nous donne la taille du mot, la taille du mot encodé, et enfin le mot encodé évidemment.

## 5.5 Creation du QrCode

Une fois la binarisation effectuée, on se retrouve avec notre mot encodé et donc nos données qui sont prêtes. Cependant, depuis la première soutenance, nous avons légèrement modifié le programme afin de lui permettre d'adapter la version du Qr Code en fonction de la taille du mot. Ainsi, notre programme s'adapte à différents types de Qr Code et est donc versatile et pratique. Il faut préciser qu'il ne marche pas sur tous les formats. Il s'arrête aux Qr Codes de type 4.

Passons maintenant à l'évolution depuis la dernière soutenance. Si la dernière fois Alexandre s'était arrêté à la binarisation des données, à présent il a bien avancé. En effet, le programme est maintenant capable de prendre un mot en entrée. Le mot peut être de tout type : URL, mot de passe, phrase quelconque... Cependant, le programme est fait pour fonctionner avec des URL et ne fonctionne donc pas avec les caractères en dehors du tableau alphanumérique. Aussi, quand une URL présente un point d'interrogation ou certains autres caractères, le décodage se passe mal. Nous sommes évidemment sur le coup pour corriger ces problèmes.

Revenons à la création. Une fois la binarisation terminée, il a fallu ajouter les corrections d'erreur. Effectivement, tous les Qr Codes possèdent des parties stockant les informations pour pouvoir les récupérer en cas de problèmes de lecture. Par exemple, si le Qr Code est endommagé pour x ou y raison, nous pourrions récupérer

les informations manquantes dans certains cas. En effet, selon la taille du Qr Code, nous devons choisir un pourcentage de correction d'erreur. Plus ce pourcentage est élevé, plus la récupération des données est efficace. Pour ajouter ces corrections d'erreur, Alexandre a implémenté l'algorithme de Reed-Solomon. Le principe de base de Reed-Solomon repose sur la division de la donnée en blocs de données appelés symboles. Ces symboles sont ensuite transformés mathématiquement pour ajouter des informations de redondance, également appelées codes de correction d'erreur, qui permettent donc de détecter et de corriger les erreurs de transmission. La transformation de Reed-Solomon est effectuée en utilisant des polynômes mathématiques, où chaque symbole de données est représenté par un coefficient de polynôme. Les coefficients de polynôme sont ensuite multipliés par une série de polynômes générateurs pour créer les codes de correction d'erreur. Les coefficients de ces codes sont ensuite ajoutés aux coefficients originaux du polynôme pour former un nouveau polynôme appelé le polynôme de codage.

Une fois les polynômes calculés, il suffit de les ajouter à la fin de la séquence de bits générée précédemment. Evidemment, il faut les ajouter dans un certain ordre afin que tout fonctionne correctement. Une fois ceci fait, on obtient le code final.

La dernière étape réside dans la division de ce code en modules. Les modules sont en fait les carrés noirs présents dans le Qr Code tel qu'on le connaît. Ça y est ! Le Qr Code est créé.

voici toutes les étapes qui ont été réalisées. On peut voir le résultat de notre travail sur l'image ci dessous.

```

clem@clem-Stealth-15M-A11UEK:~/Bureau/epita/prog/QrCode-Project$ ./main
===== Encode message =====
version = 1
mode indicator = 1
size of input string = 12
corection_level = M
Error code value : 239

0100000011000111001101101110111010101110100011001010110111001100001011011100110001101100101001000000110010000011101100
01100
=====
1 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1
1 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 1
1 0 1 1 1 0 1 0 0 0 0 1 1 0 1 0 1 1 1 0 1
1 0 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1
1 0 1 1 1 0 1 0 1 0 0 1 1 0 1 0 1 1 1 0 1
1 0 0 0 0 0 1 0 1 1 0 0 1 0 1 0 0 0 0 0 1
1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1
0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 1 1 0 0 1
0 1 1 1 0 1 0 1 0 1 1 0 1 0 0 0 1 1 0 1 0
0 0 1 0 0 0 1 0 1 1 1 0 1 0 1 0 1 0 1 1 0
1 1 1 0 1 0 0 0 0 1 0 1 1 1 0 0 1 0 0 1 1
1 1 0 0 0 0 1 1 1 1 1 1 0 1 0 0 0 1 0 1 1
0 0 0 0 0 0 0 0 1 0 1 0 1 1 1 0 1 1 1 0 0
1 1 1 1 1 1 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0
1 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0
1 0 1 1 1 0 1 0 1 1 1 0 1 1 0 1 1 1 0 0 0
1 0 1 1 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 1
1 0 1 1 1 0 1 0 0 1 1 1 0 1 0 0 1 1 0 0 0
1 0 0 0 0 1 0 0 1 1 1 0 0 0 0 1 0 0 0 0
1 1 1 1 1 1 1 0 1 1 0 0 1 1 0 1 0 0 0 1
=====

```

## 5.6 Transformation en image

Une fois la la matrice de module du QrCode créée, il fallait pouvoir l'afficher. Pour cela nous avons utiliser la bibliothèque SDL comme pour le projet réalisé au s3.

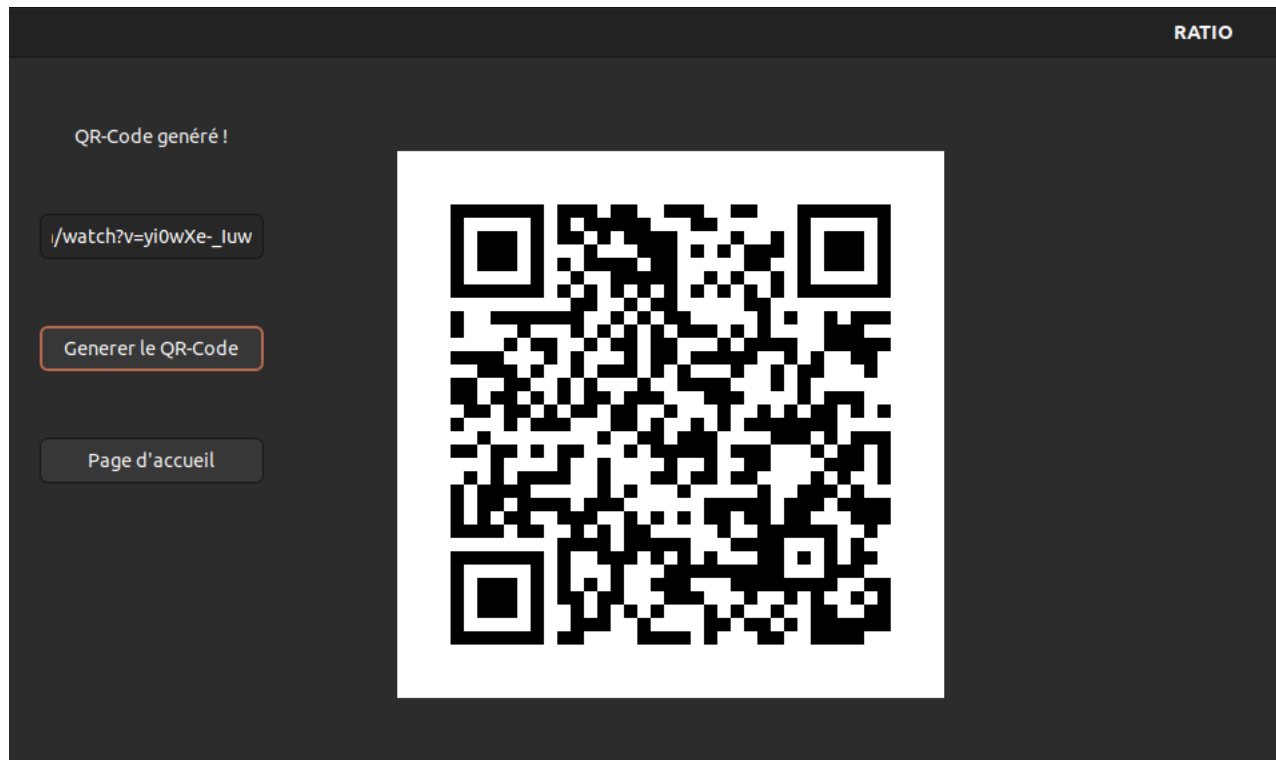
Nous avons commencé par calculer le nombre de module qu'il y avait dans le QrCode, pour ensuite connaitre la taille que fera l'image.

Nous avons choisis de faire des carrés de 7 pixels par module. Donc nous avons fait un parcours classique de matrice composé de 1 et de 0 et a chaque case de la matrice nous ajoutons a notre surface un carré de 7 par 7 pixel noir ou blanc. Une fois notre surface réalisé il a fallut la mettre sur une surface blanche que l'on appelle une zone silencieuse.

Cette zone est très utile pour détecter les QrCodes, c 'st d'ailleurs pour cela que tout les QrCodes scannable sont toujours entourés d'une epaisse zone blanche. Car les algorithmme de lecture sont juste calqué sur cette zone.

Une fois la zone silencieuse créer en fonction de la largeur de notre QrCode il a fallut coller notre surface de Qrcode sur la zone silencieuse.

Le resultat est ci-dessous :



## 6 Site Web

Dans cette section, où nous parlerons du site web, il sera explicité les étapes, les difficultés ainsi que les solutions apportées afin de mener ce mini projet à bien. Par chance la personne déléguée à la réalisation du site web possédait déjà des connaissances solides en la matière, nous n'avions donc pas besoin de faire des recherches ou seulement peu, ce qui nous fit gagner un temps précieux. Comme tout projet, la première étape importante est de faire un croquis de la solution finale, pour but d'avoir une ligne directrice afin de ne pas perdre du temps inutilement à essayer des choses qui ne nous plaisent pas. La seconde étape fut de faire un petit planning de l'avancement de ce site web.

essayer de mettre une image du croquis + planning

Après une après-midi de création artistique pour dessiner le site web et faire un planning cohérent, il était temps de se mettre au travail pratique. En commençant par la page d'accueil bien évidemment. Ici m'est apparu un problème d'ambition, où, il m'était passé par l'esprit de faire une animation à l'entrée du site web qui s'est avéré bien compliqué, aucunement utile, et, pas si belle que ça. J'ai donc fait le choix de revenir sur mes pas, pour finalement faire place à une page d'accueil on ne peut plus basique mais cependant très efficace.

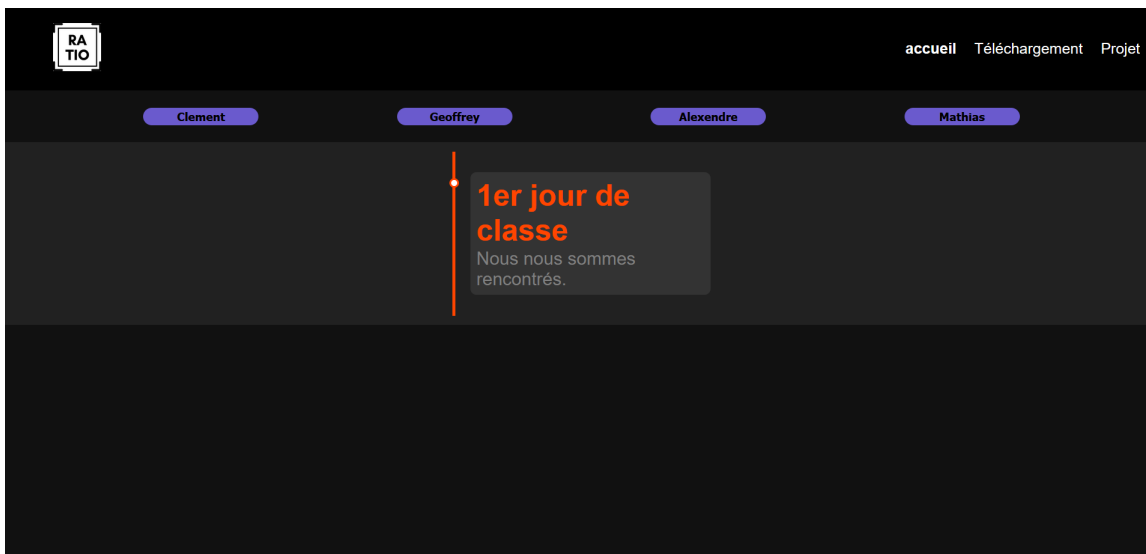




En suite vins le moment de s'attaquer à la page des téléchargement. cette page fut assez rapide à coder car la décision était de rester fidèle au croquis qui avait été proposer. Cependant la réalisation de l'image a été plus difficile que prévue.



finalement a la dernière partit voyant que l'avancement se faisait a pas de géant le temps n'était plus un problème. Nous n'avons donc pas tout à fait respecté la page qui développe l'avancement chronologique de chacun des membres. Le choix de complexifier la mise en page afin de rendre ce site web plus sérieux et professionnelle fut l'unanimité. Le but était d'utiliser toute les connaissance de CSS à notre disponibilité. Ce qui a donné à la frise chronologique que vous voyez là. De plus la première idée était de mettre tous les évènement dans une seule frise indépendamment de la ou les personnes aillant réalisé la tache, cependant, nous avons finalement préférer faire des phrase individuelle qui, selon nous, avait plus de sens vis à vis du déroulement de notre projet.



## 7 Planning Prévisionnel

Tâche à réaliser	Alexandre	Matthias	Geoffrey	Clément
Sélection Écran	O	X		
Détection QrCode			O	X
Analyse du QrCode		O	X	
Création du QrCode	X			O
Site Web	O			X

Table 3: Répartition des tâches (O : Suppléants)

Avancement Prévu	Soutenance 1	Soutenance 2	Soutenance 3
Sélection Écran	100%	100%	100%
Isolation du QrCode	90%	100%	100%
Analyse QrCode	20%	70%	100%
Création QrCode	20%	80%	100%
Site Web	50%	90%	100%

Table 4: Planning prévisionnel

## 8 Nos peines et joies

### 8.1 Clément

Je me suis occupé majoritairement du traitement matricielle et de l'isolation du QrCode. J'ai trouvé ça assez simple au départ malgré les petits problèmes des transformations de matrices lors de la première soutenance. j'ai aussi participé à la lecture du QrCode avec Goefrey qui représentait la partie la plus importante de notre projet. Nous avons plutôt bien réussi dans l'ensemble malgré un certains retard pris lors de la première soutenance, nous avons su le rattraper avec une deuxième soutenance ou la lecture nous permettait d'aller jusqu'à la version 4. Nous avons eu pas mal de problème d'index dans un premier temps mais au final avec plusieurs jours de debug les problèmes ont été résolut. Enfin pour cette dernière soutenance nous avons pas mal souffert par manque de temps avec les partiels. Mais nous sommes très content du résultat final. Je suis fière de ce que nous avons fait et de ce que nous avons appris tout au long de ce semestre.

## 8.2 Alexandre

Je me suis lancé dans un projet informatique passionnant, celui de créer un QR Code à partir d'une chaîne de caractères. Dès le départ, j'ai ressenti une immense joie en voyant comment les données prenaient vie et se transformaient en une matrice binaire. Chaque fois que j'écrivais une ligne de code, je me sentais tel un artiste qui donnait vie à une œuvre numérique.

Pourtant, malgré cette joie initiale, j'ai rapidement été confronté à des peines. L'une de mes premières peines a été de constater que les ressources et les informations précises sur la création d'un QR Code étaient rares et difficiles à trouver sur internet. Je me suis senti démuni face à cette lacune, cherchant désespérément des réponses à mes questions sans trouver de solutions claires.

Néanmoins, je n'ai pas laissé ces obstacles me décourager. Au contraire, j'ai fait preuve de persévérance et j'ai décidé de m'investir pleinement dans mes recherches. J'ai exploré des sources variées, consulté des forums spécialisés et sollicité l'aide de personnes compétentes pour combler les lacunes dans mes connaissances. Chaque petite victoire, chaque découverte, renforçait ma détermination à avancer malgré les difficultés rencontrées.

La joie s'est à nouveau intensifiée lorsque j'ai commencé à comprendre les algorithmes de correction d'erreurs. J'ai été fasciné par

la façon dont ces algorithmes permettent d'ajouter de la redondance aux données, garantissant ainsi la fiabilité du QR Code. Chaque fois que j'implémentais un nouvel algorithme et que je constatais son efficacité, je ressentais une profonde satisfaction et une fierté immense d'avoir surmonté un nouvel obstacle.

Cependant, de nouvelles peines ont fait surface lors de mon projet. La complexité croissante des étapes de création m'a parfois déconcerté. J'ai dû faire face à des erreurs de codage, à des problèmes de compatibilité et à des défis techniques qui semblaient insurmontables. J'ai ressenti une grande frustration lorsque mes efforts ne portaient pas immédiatement leurs fruits et que je devais retourner à la case départ pour trouver des solutions.

Mais dans ces moments de doute et de découragement, j'ai trouvé du réconfort dans ma passion pour l'informatique. Chaque fois que je parvenais à résoudre un problème complexe, à surmonter un obstacle ou à comprendre une nouvelle subtilité du processus, j'éprouvais une joie immense et une satisfaction profonde. Ces moments de triomphe étaient comme des rayons de soleil perçant à travers les nuages sombres de la frustration.

Finalement, la réalisation de mon projet informatique a été une expérience enrichissante. Les joies que j'ai ressenties en voyant mes idées prendre forme, en surmontant les obstacles et en développant mes compétences ont surpassé les peines rencontrées en chemin. Chaque

difficulté était une occasion d'apprendre, de grandir et de me dépasser.

Au-delà du simple fait de créer un QR Code, ce projet m'a permis de découvrir ma passion pour la résolution de problèmes, la créativité et l'apprentissage continu. J'ai réalisé que les peines sont souvent le reflet des défis qui nous poussent à repousser nos limites et à explorer de nouveaux horizons. Et c'est dans ces moments de joie et de satisfaction que je trouve ma véritable motivation, celle qui me pousse à continuer à apprendre et à créer, peu importe les difficultés que je rencontre sur mon chemin.

### **8.3 Matthias**

Je n'ai pas fait le projet.

## 8.4 Geoffrey

En tant que participant à la lecture des QR Codes dans ce projet, j'ai vécu à la fois des moments de difficulté et de satisfaction. Permettez-moi de vous parler de mes peines et de mes joies tout au long du processus.

Dans un premier temps, il me semblait que le décodage d'un QR Code n'était pas un défi de taille. De plus, lors de mes années de secondaire, il m'avait été permis d'étudier l'encodage et le décodage des QR Codes de petite taille. Si vous pensez que cette initiation ne se limitait qu'au théorique, détrompez-vous. Nous avons bel et bien développé sur Excel un lecteur et un générateur de QR Codes. Mais très vite, lors de mes recherches préliminaires, je me suis aperçu que mes souvenirs de mon lycée n'étaient plus les plus récents. Surtout, je n'avais pas eu besoin de traitement d'image lors de ma première étude sur les QR Codes car nous avons directement le QR Code numérisé.

Après avoir relu et compris toute la documentation nécessaire, j'étais enfin prêt à commencer. C'est donc à ce moment-là que commence mon premier problème. Il me fallait, afin de commencer le traitement des données, convertir une matrice de pixels en une matrice de modules, les cases du QR Code. Ce problème m'a amené jusqu'à la première soutenance où seules des recherches avaient été effectuées. Peu avant cette soutenance, nous avons compris que cette



partie était en fait du traitement d'image et non des données, ce qui ne correspondait pas à la partie qui m'avait été attribuée. J'ai donc laissé à mon camarade Clément la joie de pouvoir se confronter à ce problème.

Il était pour nous d'une grande importance que chaque ligne de code soit authentique, que nous en soyons les auteurs. C'est pourquoi, dans ma partie, aucune librairie n'a été utilisée. J'étais maître de chaque ligne, ainsi que de chaque erreur possible. Ce qui nous amène au second problème qui fut un dépassement de données lors de l'application du masque, dû à une très mauvaise gestion des variables. Ce fut corrigé malgré le temps démesuré perdu sur un problème aussi rudimentaire.

Cependant, malgré ces défis, il y a eu de nombreux moments de joie, de satisfaction et de soulagement tout au long de notre travail de lecture des QR Codes. Lorsque j'ai rendu à mon camarade sa partie de traitement d'image, le sentiment de soulagement, de ne plus avoir à combattre ce monstre, était presque aussi grand que le problème lui-même. De plus, chaque étape, chaque fonction, même chaque ligne a été une satisfaction grandissante du début jusqu'à la fin de ce projet qui se voit être le point final de mon cycle préparatoire.

En fin de compte, malgré les défis rencontrés, j'ai trouvé l'expérience de m'exercer à la lecture des QR Codes dans ce projet enrichissante. Cela m'a permis d'approfondir ma compréhension des techniques

de décodage de l'information, tout autant que l'enrichissement personnel dû à l'expérience humaine de travailler en cohésion avec des camarades qui m'ont soutenu tout au long de l'année, tant sur le plan du travail que sur le plan émotionnel.

Je suis fier d'avoir contribué à cette partie du projet et je suis reconnaissant d'avoir travaillé avec une équipe où la cohésion, l'entraide et l'amitié étaient omniprésentes. Ensemble, nous avons surmonté les obstacles, célébré nos succès et créé un programme capable de lire efficacement les QR Codes.

Ces moments de peine et de joie ont fait partie intégrante de l'expérience, et ils ont contribué à notre croissance en tant qu'équipe et en tant qu'individus.

## 9 Conclusion

Je suis ravi de présenter la conclusion de notre projet QrCode, au cours duquel nous avons travaillé avec passion pour créer un programme permettant de générer et de lire des QRcodes. Je suis fier de dire que tout s'est bien passé et que nous sommes extrêmement satisfaits du travail accompli.

Tout au long de ce projet, notre objectif était de développer une solution efficace et conviviale pour la création et la lecture de QRcodes. Nous avons consacré des heures de recherche, de conception et de développement pour aboutir à un produit final solide et fonctionnel.

L'une de nos principales réussites est la création d'une interface utilisateur intuitive qui permet à l'utilisateur de générer facilement des QRcodes personnalisés. Nous avons également veillé à ce que notre programme puisse générer des QRcodes de haute qualité, tout en garantissant une compatibilité avec les lecteurs de QRcodes les plus courants.

En ce qui concerne la lecture des QRcodes, nous avons mis en place un mécanisme robuste qui permet à notre programme de détecter et d'interpréter correctement les informations contenues dans les QRcodes. Nous avons testé notre programme avec différents types de QRcodes et dans des conditions variées, et il s'est avéré être fiable

et précis.

Nous tenons à remercier tous les membres de l'équipe pour leur dévouement et leur travail acharné tout au long de ce projet. Chacun a apporté ses compétences et son expertise, ce qui a permis de surmonter les défis rencontrés et de réaliser un programme de grande qualité.

En conclusion, notre projet QrCode a été une réussite. Nous avons réussi à créer un programme efficace et convivial pour la génération et la lecture de QRcodes. Nous sommes fiers du résultat obtenu et confiants dans sa capacité à répondre aux besoins des utilisateurs. Ce projet nous a permis d'acquérir de nouvelles compétences et connaissances, et nous sommes impatients d'explorer davantage les possibilités offertes par les QRcodes à l'avenir.

Merci à tous ceux qui ont contribué à ce projet et qui ont soutenu notre équipe tout au long du processus. Nous sommes reconnaissants de cette expérience enrichissante et nous sommes impatients de relever de nouveaux défis dans nos futurs projets.