

## Cahier des charges pour projet compilation

---

Titre: **GraphicCode Generator**

Etudiants: **Schaffo Raphaël, Gygi Damien**

Professeur: **François Tièche**

---

### Situation initiale

Dans le cadre du cours de compilation de 3ème année dans la filière informatique développement et multimédia, il nous est demandé de réaliser compilateur ou interpréteur de structure de données. Nous avons choisi de créer un interpréteur générant des images.

### Buts du projet

Traduire un code source fourni (une suite d'instruction respectant des règles prédéfinis dans notre implémentation) en une image finale. Cela à l'aide de matrices. Celle-ci seront créées et adaptées d'après les opérations du fichier source.

L'image finale sera en noir et blanc.

#### Objectifs principaux :

- Création d'un dictionnaire de matrice (chaque caractère possédera sa propre matrice)
- Interpréter les instructions pour initialiser des matrices et les modifier
- Exporter ces matrices en une image 2D (noir/blanc)

#### Objectifs secondaires :

- Gérer les erreurs

### Démarche

Pour réaliser ce projet, nous avons avant tout accès à de nombreux exemple de code traité en classe. Nous avons aussi créé, codé un interpréteur et un compilateur ensemble durant les premières leçons de l'année qui nous servirons de file rouge.

### Contraintes

- Le langage utilisé sera du Python
- Délai pour les livrables 24 janvier 2017

### Délivrables

- La partie code du projet
- Rapport final

## Exemple de code

```
Var a ='a' ;           // Assigne la matrice correspondant à b dans le dictionnaire à Var1
Var b ='b' ;           // Assigne la matrice correspondant à a dans le dictionnaire à Var2
Var c = a mul 10 ;      // Assignation de 10x la matrice Var1 à Var3
Line(a+ b) ;           // Assigne la matrice Var1+Var2 en tant que première ligne de l'image
Line(c) ;              // Assigne la matrice Var3 en tant que deuxième ligne de l'image
a->'a'->c               // Remplir la variable a avec le caractère 'a' tant que la variable a est plus
                       // petite que la taille (dimension de celle de c)
```