

Projet de compilateur S³VG

Par Skrapits Cyriaque et Strambini Eddy

Objectifs

Les objectifs à atteindre étaient de pouvoir sortir un fichier SVG depuis notre compilateur via des méthodes que nous avons définies.

Ces méthodes, disponibles en détail dans le cahier des charges, permettent entre autre de dessiner les formes de bases d'un fichier SVG (cercles, rectangles etc.) ainsi que de définir des styles à appliquer aux éléments précédemment cités.

La gestion de la boucle « for » est également prévu, le but de créer, par exemple, très simplement une suite de formes.

Les variables ainsi que qu'un objet permettant de définir des coordonnées sont également gérés.

Fonctionnalités

Le compilateur gère toutes les commandes, objectifs, fixés auparavant pour le projet dont les formes de bases disponibles dans SVG..

Nous avons omis l'implémentation de la méthode « setUnit » (permettant de définir l'unité du SVG). On peut en effet préciser l'unité au cas par cas lors de la création d'une forme. L'idée de cette méthode est donc superflue.

L'utilisation dans les fichiers S3VG des opérateurs de bases (+ - / * et %) est possible, notamment pour la boucle « for ».

Langages et librairie

Le « langage » utilisé dans les fichiers de descriptions est décrit dans le cahier des charges. Pour que cela fonctionne, il faut tâcher de le respecter au maximum.

Pour la création des fichiers SVG, nous avons décidé d'utiliser une librairie externe : `svgwrite` (<https://pypi.python.org/pypi/svgwrite/>) afin de nous faciliter la tâche et de ne pas perdre un temps précieux à définir des méthodes écrivant le SVG lui-même, la part du travail consistant surtout de faire en sorte que nos méthodes soient quant à elles comprises par le compilateur.

Comment ça marche ?

Il suffit d'écrire un fichier au format S3VG qu'on nommera par exemple `test.s3vg` puis de la passer en premier argument du compilateur. Un fichier SVG de sortie devra également être entré comme deuxième paramètre.

Exemple : `"s3vg.py test.s3vg sortie.svg"`

Des exemples sont fournis avec le programme.

Remarques

Fonctionnalités non-implémentées

La spécification SVG contient également d'autres fonctionnalités telles que les dégradés ou les transformations d'objets (redimensionnement, rotation, etc.) mais nous n'avons pas pris le temps de les implémenter. Notre but était de créer un compilateur assez simple qui comprend les éléments de base du SVG.

Malgré cela, il se pourrait que le projet puisse être continué ultérieurement.

Boucles while

Nous avons remarqué qu'un autre groupe avait implémenté une boucle while. En termes de langage descriptif comme le nôtre, il nous semblait inutile d'implémenter un tel procédé d'autant plus qu'une boucle peut tourner infiniment, ce qui serait absurde dans notre cas. La méthode du for allant d'une valeur à une autre nous permet d'être sûr que notre boucle soit finie.