

# Projet

## Un interpréteur pour la création d'images

Il existe de nombreux logiciels permettant de créer des images, par exemple, Paint, Gimp, Photoshop. Pour tous ces exemples, l'utilisateur transmet ses instructions à la souris. La souris offre une prise en main intuitive mais ne permet pas d'automatiser les tâches que l'utilisateur souhaite effectuer ni de communiquer avec d'autres programmes.

Dans ce projet vous allez concevoir et implanter un interpréteur pour la création d'images. L'interpréteur va prendre ses instructions dans un fichier texte et dessiner à l'écran une image construite à partir de ces instructions. Les instructions permettent de déplacer verticalement ou horizontalement un curseur virtuel. Si le curseur est en position basse chaque déplacement laisse une trace noire, s'il est en position haute le déplacement n'écrit rien. Les instructions permettent aussi de tourner le curseur dans le sens trigonométrique (c'est à dire, contre le sens de rotation des aiguilles d'une montre), par un angle donné en degrés.

**La grammaire** La grammaire du langage est la suivante, les non-terminaux sont en gras et en italiques, les terminaux (jetons) en police code :

- *programme* → *declarations instruction*
- *declarations* → Var identificateur; *declarations* |  $\varepsilon$
- *instruction* → Avance *expression* | Tourne *expression* | BasPinceau | HautPinceau | identificateur = *expression* | Debut *blocInstruction* Fin
- *blocInstruction* → *instruction*; *blocInstruction* |  $\varepsilon$
- *expression* → nombre *expressionSuite* | identificateur *expressionSuite* | (*expression*) *expressionSuite*
- *expressionSuite* → + *expression* | - *expression* | \* *expression* | / *expression* |  $\varepsilon$

avec les définitions suivantes de jetons :

- identificateur → [a-z][a-zA-Z0-9]\*
- nombre → [1-9][0-9]\* | 0

**Exemple** Le programme suivant dessine ainsi un carré dont les côtés ont longueur 10.

```
Var angle;
Debut
angle = 0-90;
BasPinceau;
Avance 10;
Tourne angle;
Avance 10;
Tourne angle;
Avance 10;
Tourne angle;
Avance 10;
Fin
```

## 1 Première Étape

Il est conseillé d'écrire le programme en plusieurs étapes de difficulté croissante. Dans un premier temps, écrivez un analyseur lexical et syntaxique afin de construire l'arbre de syntaxe

abstraite d'un programme décrit par la grammaire ci-dessus. Ensuite, écrivez un interpréteur pour l'arbre construit.

Il faudra veiller à ce que toute variable utilisée soit bien déclarée au début du programme et vérifier que chaque variable est déclarée au plus une fois.

Nous choisissons aussi que le curseur est initialement placé aux coordonnées (0,0), c'est-à-dire un bas à gauche du canvas graphique, en orientation nord (c'est à dire vers le haut du canvas), et le pinceau en position haute.

**Gestion des erreurs** Votre programme doit traiter proprement toutes les erreurs qui peuvent se produire pendant l'interprétation, c'est à dire afficher un message explicite, détaillant l'erreur rencontrée, et puis terminer l'interpréteur. Les erreurs à capturer sont :

- une expressions fournie à Tourne n'est pas un multiple de 90 ;
- division par 0 ;
- le curseur sort du canevas.

## 2 Deuxième Étape

Ajoutez des instructions de type

Si *expression* Alors *instruction* Sinon *instruction*  
et  
Tant que *expression* Faire *instruction*

produites par le non-terminal *instruction*. Nous choisissons pour l'instruction **Si Alors Sinon** que l'instruction après le **Alors** est exécutée seulement si l'expression est différente de 0. Nous choisissons aussi pour l'instruction **Tant que** que la boucle s'arrête si l'expression est égale à 0 .

Vérifiez tout d'abord que la grammaire est bien LL(1) et modifiez-la le cas échéant. Modifiez votre code afin d'intégrer la nouvelle grammaire à votre interpréteur.

## 3 Troisième Étape

Nous proposons quelques pistes pour des extensions de votre programme. À vous de choisir les extensions que vous souhaitez réaliser :

- Ajouter une instruction permettant de modifier l'épaisseur des traits et leur couleurs.
- Modifier la grammaire afin d'autoriser en plus les conditions

**Si *expression* Alors *instruction***

sans **Sinon** derrière. Attention, assurez-vous qu'elle soit LL(1).

- Ajouter un mode interprété où l'utilisateur peut entrer des commandes et voir en même temps les résultats de ces commandes dans une fenêtre graphique.

## 4 Rendu

Il est conseillé de faire le projet en binôme (mais pas de trinômes!) Le projet est à rendre avant le dimanche 10 mai sur Didel dans la section travaux, pour le travail *infoK.projet* où K est le numéro de votre groupe. Vous devez rendre une archive contenant tous les fichiers de votre projet ainsi qu'un fichier texte expliquant comment les compiler et les exécuter.