

TP n°5 - C'est la fin

Site: [Université de Reims Champagne-Ardenne](https://cours.univ-reims.fr/)

Cours: Langages et compilation

Livre: TP n°5 - C'est la fin

Imprimé par: QUENTIN JUILLIARD

Date: lundi 13 mars 2023, 11:51

Table des matières

1. Pour commencer
2. Listes d'instructions
3. Fonctions et procédures

1. Pour commencer

Avec le code du TP précédent, nous sommes en mesure de créer des niveaux à l'aide d'instructions simples. Nous aurions pu ajouter aisément les conditionnelles. Vous pouvez d'ailleurs le faire puis vous arrêter là : votre projet ne serait pas parfait, mais s'il est fonctionnel (et que le rapport est correct), vous aurez déjà la moyenne (12/20 maximum).

Si vous souhaitez aller plus loin, il nous manque les éléments suivants : les fonctions, les procédures et les boucles (Pour et TantQue). Or, avec le code du TP précédent, ce n'est pas possible car les instructions sont exécutées lors de l'analyse, mais ce n'est pas possible pour ces éléments. La solution est donc de placer les instructions correspondantes dans une structure spécifique (une liste chaînée d'instructions) pour pouvoir l'exécuter autant de fois que nécessaire.

2. Listes d'instructions

Pour le moment, nous exécutons les instructions au fur-et-à-mesure. Mais comme [le projet](#) exploite un langage interprété, il est nécessaire de placer les instructions en mémoire pour pouvoir les exécuter plusieurs fois (dans le cas des boucles, par exemple, ou des fonctions).

1. Selon le sujet du projet, quelques instructions peut-on rencontrer ?

Spoil alert

2. Pour chacune, que doit-on stocker en mémoire ?

Spoil alert

3. Trouvez une structure permettant de stocker en mémoire les listes instructions.

4. Écrivez les fonctions nécessaires pour gérer les listes, les afficher (pour vérification) et les exécuter.

5. Modifiez votre grammaire pour que les instructions soient mises en mémoire puis exécutées une fois un niveau reconnu.

3. Fonctions et procédures

Pour les fonctions et les procédures, nous devons placer leurs instructions dans une liste d'instructions, récupérer la liste des paramètres puis ajouter le symbole correspondant à la fonction / procédure dans la table des symboles. Lorsqu'un appel à une fonction/procédure doit être exécuté, on recherche ce symbole et on exécute la liste des instructions associée.

La gestion de la table des symboles n'est pas si simple. En effet, pour le moment, on recherche les symboles dans la table du niveau courant. Cette table est construite au début du niveau et supprimée à la fin. Lors de l'exécution d'une fonction / procédure, la recherche de symboles doit s'effectuer dans une autre table des symboles. Pire, dans le cas d'un appel récursif, la modification d'une variable ne doit pas avoir d'impact sur le reste !

Techniquement, nous avons systématiquement une table des symboles globale pour les fonctions / procédures et les variables globales. Pour le reste, on peut utiliser une pile de tables de symboles. Lorsqu'on analyse un niveau, on empile une nouvelle table. Si on exécute une fonction / procédure, on empile encore une nouvelle table (qui contient par défaut les paramètres et leurs valeurs). Une fois la fonction / procédure exécutée, on dépile la table. Idem pour le niveau. La recherche d'un symbole doit donc être effectuée dans la table au sommet de la pile puis dans la table globale.