Fonctionnement général d'un programme

Prérequis : 200_IntroductionALaProgrammaionAvec1InterpreteGraphiqueX.doc

Prérequis : Il faut d'abord voir : 200_IntroductionALaProgrammaionAvec1InterpreteGraphiqueX.doc

Il traite :

Instruction, modalité, contexte local et global, boucle d'interaction.

Introduction

Pour aider les débutants, ce polycopié donne un aperçu global du fonctionnement d'un programme en s'appuyant sur des expériences de la vie quotidienne.

Pour le professeur, la difficulté est d'éclairer le domaine, sans entrer dans les détails, car le risque est de s'y enliser.

Notion d'instruction

L'ordre de faire une action

Au début peut être vu comme un verbe d'action à l'impératif :

Exemples : avance, tourne, additionne, imprime.

Notion de paramètre

En raffinant, on précise que l'ordre ne se réduit pas seulement à un verbe d'action.

Il faut préciser les compléments de l'action :

Faire quelque chose : à qui, à quoi, avec quoi, où, comment.

Exemples : avance de 10 pas, tourne à gauche, additionne ce nombre avec 5, imprime le nom et le prénom de cette personne..

Notion de programme

Exécution de plusieurs instruction

Un programme est une liste d'instructions.

C'est la donnée de plusieurs instruction à exécuter :

Exemple 1 : répète 4 fois ceci : Avance de 20 pas et tourne à droite d'un angle droit.

Exemple 2 : Une recette de cuisine

- Peser 400g de farine.
- La mettre dans un saladier.
- Creuser un puits au milieu et y casser un œuf …

Déroulement séquentiel d'un programme

Cela va sans dire, mais ça va tellement mieux en le disant :

Un programme s'exécute selon les conventions de lecture d'un texte :

De la gauche vers la droite.

puis, de haut en bas.

Et enfin de page en page.

Ruptures dans le déroulement séquentiel d'un programme

Définition

Quand le parcours de la mémoire cesse de se faire séquentiellement, mais qu'il y a un saut brutal ailleurs on dit qu'on a une rupture de séquence.

Exemples de branchement dans le déroulement séquentiel d'un programme

Notion d'instruction conditionnelle.

Notion d'alternative.

Notion de commutateur (switch)

Notion de condition à une action

La notion d'instruction conditionnelle signifie que sous certaines conditions, on fait une action.

L'évaluation de ces conditions implique de pratiquer des tests afin de prendre la décision de faire ou non l'action.

Ces tests sont de plusieurs sortes :

- Comparaison entre nombres ou symboles.
- Nature et propriétés des objets concernés par le programme.
- Recherche de mots clés (début fin).

Ruptures dans le déroulement séquentiel d'un programme

Ruptures de séquences naturelles, elles permettent les actions conditionnelles.

Faire ou non une action : L'action conditionnelle

En fonction du résultat d'un test, on décide ou non, de faire un traitement : Soit on l'effectue, soit on le saute.

Le SAS (l'alternative)

Il s'agit d'un aiguillage du cheminement du processeur : selon le résultat du test, on fait l'action « alors » ou bien l'action « sinon » .

L'aiguillage multiple (Le commutateur, dispatcheur, switch)

C'est un peu comme dans une gare de triage, le train arrive par une voie, mais, selon sa destination, il sera orienté vers une seule, choisie parmi N autres.

Parmi tous les traitements à effectuer, on n'effectue que celui qui est sélectionné en fonction d'un mot clé.

Accident dans le déroulement séquentiel d'un programme

Plantage

Pour l'utilisateur, ça semble la fin de traitement séquentiel des instructions : L'ordinateur ne répond plus aux commandes, on a perdu la main, le programme est planté:

. Quand on bouge la souris, son curseur demeure immobile sur l'écran.

. Les commandes du clavier n'ont aucun effet.

. L'ordinateur s'est perdu dans le traitement inutile d'autres instructions et ne traite plus les nôtres. Souvent, il tourne en rond.

Rupture de séquence artificielle : piratage et détournement

Détournement momentané de l'exécution d'un programme par virus, qui en profite pour exécuter quelques instructions à son bénéfice, et au détriment de l'utilisateur attitré.

Aspect effectif d'un programme

La grande question en relation avec l'état de l'univers, c'est : Comment le programme produit-il un résultat tangible, effectif ?

Analyse : ne change pas l'état de l'univers

Il analyse les informations qu'il possède, mais n'en change pas le contenu. A partir des informations dont il dispose, il examine plus précisément la situation et retourner une information :

- Oui ou non.
- Un nombre : La quantité de cette matière est 123.
- Une relation : Ce conteneur est vide.
- Un conseil, un ordre(une action à faire) : Il faut tourner à droite et marcher jusqu'à B.

Action

Entrée au moyen de la perception : Ce sont les capteurs :

Le sens d'ajustement va du monde vers l'esprit

Quand il fait une action en entrée, l'ordinateur ne change pas l'état de l'univers extérieur. Il l'observe et déduit des relations supplémentaires, qui s'ajoute à l'idée qu'il se fait de l'univers externe.

Exemple :

- Lecture au clavier.
- Clic de la souris sur une zone de l'écran.
- Mesure de paramètres physique : pression, vitesse, accélération …
- Image en provenance d'une caméra.

Sortie au moyen de l'action : Ce sont les actionneurs

- Écrire une information sur l'écran, un disque dur.
- Imprimer un texte.
- Faire un bip.
- Faire de la musique.

Appeler un sous-programme

Regrouper des actions sous un nom

Dans la vie courante, dès qu'on identifie un groupe d'actions caractéristiques, on lui donne un nom. Ex :

- Aller au marché.
- Faire la vaisselle ou le ménage.
- Poster une lettre.
- Reconduire un ami.

Ainsi quand on parle d'une action connue de tous, cette démarche évite de détailler toutes les actions élémentaires à faire, et chacun comprend.

Notion de sous-programme

En informatique, on procède pareillement. Dès qu'on repère un groupe d'actions qui est utilisé en plusieurs endroit, on le nomme en lui attribuant un nom significatif, et on ne l'évoque au moyen de ce nouveau nom.

En conclusion, un sous-programme est composé de deux choses :

- La donnée de son nom de baptême.

- Son expansion, cad la liste des actions élémentaires qui le composent.

Notion de primitive

Les instructions que l'ordinateur sait faire de façon native sont les primitives. On dit aussi qu'elles appartiennent au noyau.

Branchement à un sous-programme

Introduction

Si dans une recette de cuisine, on rencontre les instructions :

- Cuire les œufs pour qu'ils soient durs.

- Les éplucher.

On délaisse la recette principale, on n'exécute pas l'instruction "Les éplucher", mais on fait les actions nécessaires pour "cuire les œufs", on se reporte à la recette correspondante et quand on a des œufs durs, on en revient à notre recette principale, qui dit de les éplucher.

Départ vers un sous-programme

Quand l'ordinateur exécute une séquence d'instructions, si au lieu de tomber sur une primitive, il rencontre un sous-programme, il interrompt sa tâche en court (le programme principal), et exécute les instructions correspondant au sous-programme.

Retour d'un sous-programme

Quand il a finit, il retourne au programme principal.

Ensuite il continue d'exécuter le programme principal.

Les deux sortes de sous-programmes

Il existe deux sortes de sous programme. On les distingue par leur façon d'agir sur l'univers.

Fonction

De façon un peu caricaturale, on peu dire que la fonction, n'agit pas sur le monde. Elle est un peu comme un intellectuel dans sa tour d'ivoire, elle ne manipule que des idées et ne modifie pas l'univers. Elle retourne simplement une nouvelle analyse des propriétés de l'univers.

Par exemple elle dit :

- 3 + 4 a pour valeur 7.

- Oui, cet atome appartient à cette liste.

- La couleur de cette table est bleue.

Sa seule façon d'agir est donc de remonter une nouvelle relation à propos des propriétés de l'univers.

Procédure

Une procédure en entrée analyse l'état du monde avec des capteurs et change l'image que le programme a enregistrée du monde.

Une procédure en sortie agit sur ses actionneurs et change l'état du monde.

Le piège de la définition d'un sous-programme

Quand je voudrais que ma petite sœur fasse la vaisselle à ma place, je peux lui commander de le faire. Si elle me répond qu'elle ne sait pas comment le faire, je dois lui expliquer.

Quand un programmeur constate que l'action qu'il aimerait bien commander à l'ordinateur n'est pas présente dans la liste de ses primitives, il doit donc expliquer lui expliquer comment faire cette action nouvelle. Ca s'appelle définir un nouveau sous-programme.

Quand ma petite sœur sais comment faire la vaisselle, ce n'est pas pour cela que la vaisselle est faite. Il faut encore que je lui ordonne de la faire.

Quand un ordinateur accepte la définition d'un nouveau sous-programme, ce n'est pas pour ça qu'il est effectué, il faut encore lui commander de le faire.

Définir un sous-programme à un ordinateur est difficile car il faut respecter la syntaxe du langage. Quand il l'a enfin accepté, l'erreur très fréquente chez les élèves est de dire : ben, pourquoi il ne le fait pas. La réponse est claire :

- Oui, l'ordinateur a acceptée ta définition de sous-programme.

- Si tu veux qu'il le fasse, ordonne le lui !

Passage de paramètres à un sous-programme

Nous avons déjà vu comment on nomme un groupe d'actions élémentaires pour en faire une action notoire, mais il faut préciser qu'elle peut requérir des précisions :

- Pour dessiner une marche on doit connaître ses dimensions.

- Pour dessiner un carré, on doit connaître la taille de son côté.

- Si l'ordinateur doit jouer une note, il a besoin de connaître sa hauteur et sa durée.

Ainsi, certains sous-programme doivent recevoir des paramètres.

Notion de contexte local et global

Notion de contexte local

Ces informations que l'on fournit à un sous-programme, lui sont dédiés, et ne sont visibles localement que par lui. Elles constituent le contexte local du sous-programme.

Notion de contexte local et global

Les informations qui sont valables durant tout le programme : La date, l'heure, le nom de l'utilisateur, les caractéristiques de l'ordinateur, etc. sont nommées contexte global.

Selon les langages de programmation le contexte global est visible ou non à l'intérieur d'un sous-programme.