PolyPublie
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Détermination d'un profil de l'apprenant des concepts fondamentaux de la programmation
Auteur: Author:	Anka Stoykova Mihaylova
Date:	2007
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Mihaylova, A. S. (2007). Détermination d'un profil de l'apprenant des concepts fondamentaux de la programmation [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7939/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/7939/
Directeurs de recherche: Advisors:	Yves Boudreault
Programme: Program:	Unspecified

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

« DÉTERMINATION D'UN PROFIL DE L'APPRENANT DES CONCEPTS FONDAMENTAUX DE LA PROGRAMMATION »

ANKA STOYKOVA MIHAYLOVA
GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
Avril 2007

© Anka Stoykova Mihaylova, 2007.

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-29243-3
Our file Notre référence
ISBN: 978-0-494-29243-3

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

DÉTERMINATION D'UN PROFIL DE L'APPRENANT DES CONCEPTS FONDAMENTAUX DE LA PROGRAMMATION

présenté par : MIHAYLOVA Anka Stoykova
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. DESMARAIS Michel, Ph.D., président
M. BOUDREAULT Yves, Ph.D., membre et directeur de recherche
M. ROBILLARD Pierre-N., Ph.D., membre

Dédicace

À mes enfants et à mon

mari

Avec tout mon amour

Remerciements

Mes premiers remerciements vont à ma famille. À mes merveilleux enfants et à mon mari qui pendants ces années assez difficiles pour eux aussi ont montré une telle grande patience et compréhension envers moi. Ils étaient toujours avec moi et me soutenaient dans mon travail. Je vous remercie, mes chéris!

Je tiens également à remercier chaleureusement mon directeur de recherche Yves Boudreault qui était toujours présent pour m'aider et me conseiller quand j'avais besoin et de me diriger en avant dans la recherche. C'est un grand plaisir de travailler avec vous, Yves. Je vous remercie beaucoup.

J'apprécie grandement l'aide des experts du Laboratoire de Robotique Pédagogique (LRP) de l'Université de Montréal et de M. Matthiew Khouzam. Leur aide nous a permis d'obtenir un montage robuste et pratique.

Résumé

Conscient des difficultés rencontrées par les étudiants dans leur apprentissage des concepts fondamentaux de la programmation, notre recherche a pour but de leur venir en aide. Pour ce faire, nous croyons que la première chose à faire est de mieux les connaître en identifiant les modèles psychologiques et les différentes caractéristiques de leur stratégie d'apprentissage. Nous identifierons cet ensemble de caractéristiques par le terme style d'apprentissage. Une fois cette information connue, nous désirons vérifier quel type d'activité sera le plus approprié selon le style d'apprentissage. Nous regarderons plus spécifiquement deux types d'activité : des exercices à caractère abstrait et des exercices utilisant un montage électronique. Nous observerons comment les étudiants se comportent du point de vue stratégique lors de la résolution des problèmes par l'analyse détaillée des résultats obtenus.

En clair, nous voulons vérifier quel style d'apprentissage est le mieux servi par un exercice à caractère abstrait et quel style d'apprentissage est le mieux servi par un exercice à caractère concret.

Nous désirons tout particulièrement vérifier si les étudiants du type réflexif vont avoir des meilleurs résultats en travaillant avec un exercice à caractère abstrait que les étudiants du type actif. Et au contraire, si les résultats des étudiants du type actif seront meilleurs que ceux des représentants du type réflexif en travaillant avec un exercice à caractère concret. Les types réflexif et actif auxquels nous faisons références ici sont ceux définis dans le modèle de Felder-Silverman.

Mots-clés : stratégies d'apprentissage, programmation, modèles psychologiques, C++, informatique.

Abstract

Being aware of the difficulties encountered by the students in their learning of the concepts of elementary programming, the purpose of our research is to come to them for assistance. With this intention, we believe that the first step is to know them better by identifying their style of learning. Once this information known, we wish to check which type of activity will be adapted the most according to the style of learning. We will more specifically look at two types of activity: exercises in abstract matter and exercises using an electronic assembly. We will observe how the students behave from the strategic point of view at the time of the resolution of the problems by the detailed analysis of the obtained results.

Basically, we want to check which style of learning is best suited by an abstract subject exercise and which style of learning is best suited by a concrete subject exercise.

We would like to particularly verify if the «reflective» students will have more success in working on an abstract subject exercise than the «active» students. On the other hand, we also investigate if the «active» student achieved better results in working on electronic assembly exercise comparing to their «reflexive» counterparts.

Key words: learning styles, programming, psychological models, C++, data processing.

Table des matières

Dédicace iv
Remerciements v
Résumé vi
Abstract vii
Table des matières viii
Liste des tableaux x
Liste des figures xi
Liste des sigles et abréviations xii
Liste des annexes xiii
CHAPITRE 1. Problématique et idée de recherche 1
1.1. Difficultés inhérentes au processus d'apprentissage de la programmation 1
1.1.1. Hétérogénéité des étudiants. 1
1.1.2. Caractère abstrait des concepts informatiques. 2
1.1.3. Détection, interprétation et correction de. 5
1.2. Idée de la recherche 6
CHAPITRE 2. Quelques moyens utilisés pour favoriser l'apprentissage de la programmation 8
2.1. Méthodes d'apprentissage 8
2.1.1. Traditionnelles 8
2.1.2. Non traditionnelles 9
2.2. Les environnements et les applications 13
2.3. « Minute paper » 13
CHAPITE 3. Les modèles psychologiques de l'apprentissage 15
3.1. MBTI (Myers-Briggs Type Indicator) 15
3.2. Modèle de Kolb 18
3.3. Herrmann Brain Dominance Instrument 20
3.4. Le modèle de Felder 21

CHAPITRE 4. Description de l'expérimentation 29
4.1. Test Felder. 29
4.2. Montage électronique. 30
4.3. Atelier de programmation. 31
4.3.1. Exercice : Structures booléennes et de décision 31
4.3.2. Exercice : Structures de répétition 33
4.3.3. L'étape de programmation 34
CHAPITRE 5. Mise à l'essai et les résultats 36
5.1. Évaluation du test 37
5.2. Évaluation des exercices 41
5.2.1. Évaluation de premier exercice 44
5.2.2. Évaluation de deuxième exercice 53
5.3. Évaluation de l'expérience 60
Conclusion 62
Bibliographie 66
Annexes 68

Liste des tableaux

Tableau 2.1 Exemple de questions d'un « Minute paper »....14
Tableau 3.1 Modèle psychologique de MBTI....18
Tableau 3.2 Modèle psychologique de Kolb....19
Tableau 3.3 Styles d'apprentissage préférés selon le modèle de Felder-Silverman....23
Tableau 3.4 Modèle psychologique de Felder-Silverman....27
Tableau 5.1 Réponses des candidats au questionnaire de Felder-Silverman....38
Tableau 5.2 Regroupement des résultats selon trois intensités....39
Tableau 5.3 Résultats sommaires des participants au test de Felder-Silverman....40
Tableau 5.4 Classification des résultats obtenus au test de Felder-Silverman....40
Tableau 5.5 Classification finale des résultats obtenus au test de Felder–Silverman .41
Tableau 5.6 Analyse des programmes, des étudiants de type REF de l'exercice 1....45
Tableau 5.7 Analyse des programmes des étudiants de type ACT de l'exercice 1....46
Tableau 5.8 Résultats des étudiants des différents types pour l'exercice 1....47
Tableau 5.9 Analyse des programmes des étudiants de type REF de l'exercice 2....55
Tableau 5.10 Analyse des programmes des étudiants de type ACT de l'exercice 2....56
Tableau 5.11 Résultats des étudiants des différents types pour l'exercice 2....54
Tableau 5.12 Réussite de l'exercice 1....60
Tableau 5.13 Réussite de l'exercice 2....61

Liste des figures

Figure 1.1 : Types d'erreur. 6
Figure 2.1 : Exemple d'une grille. 12
Figure 3.1 : Types d'apprentissage de Kolb. 19
Figure 3.2 : Types d'apprentissage de Hermann. 21
Figure 3.3 : Exemple d'un profil d'un style d'apprentissage 26
Figure 4.1 : Le montage électronique. 31
Figure 4.2 : Fichier où les étudiants doivent écrire leurs instructions. 35
Figure 5.1 : Diagramme du flux de contrôle (groupe REF) – exercice 1. 50
Figure 5.2 : Diagramme du flux de contrôle (groupe ACT) – exercice 1 51
Figure 5.3 : Diagramme du flux de contrôle (groupe ACT) – exercice 2 57
Figure 5.4 : Diagramme du flux de contrôle (groupe REF) – exercice 2. 58

Liste des sigles et abréviations

MBTI - Myers-Briggs Type Indicator

HBDI - Herrmann Brain Dominance Instrument

E – Extraverti (type de personnalité chez Myers-Briggs)

I – Introverti (type de personnalité chez Myers-Briggs)

S – Sensitif (type de personnalité chez Myers-Briggs)

N – Intuitif (type de personnalité chez Myers-Briggs)

T – Penseur (type de personnalité chez Myers-Briggs)

F – Sentimental (type de personnalité chez Myers-Briggs)

J – Jugeur (type de personnalité chez Myers-Briggs)

P – Penseur (type de personnalité chez Myers-Briggs)

SNS – sensoriel (type psychologique chez Felder)

INT – intuitif (type psychologique chez Felder)

VIS – visuel (type psychologique chez Felder)

VRB - verbal (type psychologique chez Felder)

ACT – actif (type psychologique chez Felder)

REF – réflexif (type psychologique chez Felder)

SEQ – séquentiel (type psychologique chez Felder)

GLO – global (type psychologique chez Felder)

Liste des annexes

Annexe A : Le questionnaire de Felder 68
Annexe B : Compilation des résultats du questionnaire 74
Annexe C : PWM 77
Annexe D : Formulaires concernant les normes d'éthique 78
Annexe E : Organisation de l'espace du travail 81
Annexe F : Les énoncés des exercices 83
Annexe G : Calculs statistiques 88

CHAPITRE 1. Problématique et idée de recherche

1.1. Difficultés inhérentes au processus d'apprentissage de la programmation

Un cours présentant les rudiments de la programmation procédurale est offert à l'ensemble des étudiants de l'École Polytechnique de Montréal et cela indépendamment de la discipline du génie choisie.

Le but de notre recherche est de déterminer les profiles psychologiques des étudiants et leurs styles d'apprentissage privilégiés afin de développer des dispositifs favorisant l'apprentissage mieux adaptés à ces caractéristiques.

Dans son processus d'apprentissage l'apprenant rencontre certaines difficultés qui sont d'origine diverses. Nous allons en mentionner quelques unes.

1.1.1. Hétérogénéité des étudiants.

L'hétérogénéité des étudiants est le premier problème perçu par les enseignants. Comme nous le mentionnons précédemment, un cours obligatoire présentant les rudiments de la programmation est au programme de tous les curriculums. Puisqu'ils sont nombreux, les étudiants sont divisés en plusieurs groupes. Très souvent dans un groupe il y a des étudiants qui ont déjà découvert le monde de la programmation

« ... was there a significant difference between
experienced and inexperienced students ... » [3]

et d'autres qui sont au début de cette découverte. La disparité des connaissances de ces étudiants entraîne d'inévitables problèmes. Ici le rôle du professeur est crucial, plus précisément comment il va réagir pour stimuler les « débutants » sans perdre l'intérêt des autres étudiants.

> *« ..the student groups are large and heterogeneous and thus it is difficult to design the instruction so that it would be beneficial for everyone. This often leads to high drop-out rates on programming courses. » [21]*

D'un autre côté, le niveau d'éducation générale des étudiants est aussi différent. Il y a des étudiants qui viennent des écoles où les exigences sont beaucoup plus élevées que d'autres.

Troisièmement, le niveau d'aisance d'utilisation de l'ordinateur peut être fort différent chez les étudiants. Il y a ceux qui utilisent facilement différents logiciels et d'autres qui s'initient à l'ordinateur et ses possibilités. Encore une fois, le professeur joue un rôle important en tentant de provoquer l'intérêt et la curiosité chez ses étudiants. Le professeur recherchera à diminuer l'écart entre les étudiants, conscient des craintes de certains et des attentes élevés des autres étudiants.

Étant donné ces considérations, nous estimons que les exercices doivent être choisis avec parcimonie et doivent être conçus en tirant profit au maximum des styles d'apprentissages privilégiés par la majorité des étudiants.

1.1.2. Caractère abstrait des concepts informatiques.

Un programme se compose de différents types de relation et niveaux d'abstraction [17] comme :

- La structure linéaire : la version textuelle du programme.
- Le flux de contrôle : l'ordre dans lequel des opérations sont exécutées. Elles sont décrites à l'aide des constructions : l'itération, la séquence et le test.
- Le flux de données : les données à l'entrée et leur transformation durant l'exécution du programme.

- Les plans et l'organisation hiérarchique des buts et sous-buts; le plan est une suite d'opérations qui vont amener jusqu'au but final ou au sous-but. Le plan est généralement partitionné en sous-plans.

Le caractère abstrait de la programmation est le premier obstacle que les étudiants doivent surmonter. Avant de créer un programme ils doivent comprendre ces aspects abstraits [17]. Pour concevoir un programme ils doivent avoir en plus certaines notions de base, comme les structures décisionnelles, les structures répétitives et les structures de données.

Les étudiants doivent être capables de détailler le problème en question. Après ils doivent savoir comment traduire chaque détail de ce problème en langage informatique. C'est une démarche assez difficile. C'est aussi l'un de ces aspects abstraits. Prenons l'exemple suivant :

Concevoir un programme qui calculera, selon la méthode de Cardan, les racines d'une équation de degré trois de la forme :

$$\mathbf{x^3 + ax^2 + bx + c = 0\ .}$$

Voici quelques unes des questions que les étudiants se poseront au départ.

- Quelles sont les données à considérer? (de quoi on a besoin)
  - des coefficients a, b, c.
- Quelle substitution doit-on faire pour obtenir la forme $\mathbf{y^3 + px + q = 0}$ **?**
  - $y = x + \frac{a}{3}$
- Quelle sera la nouvelle forme?
  - $y^3 + y\left(b - \frac{a^2}{3}\right) + \frac{a}{27}\left(2a^2 - 9b\right) + c = 0$
- Quelles seront les valeurs de **p** et **q**?

- $p = b - \frac{a^2}{3}$
- $q = \frac{a}{27}\left(2a^2 - 9b\right) + c$

- À l'aide des coefficients p et q on obtient quoi?
  - $D = 4p^3 + 27q^2$
- Comment calculer les racines?

En raffinant, les questions suivantes pourraient être :

- Comment obtenir les coefficients a, b, c? Ce sont des données à l'entrée ou à la sortie?
- Si a,b,c sont des données à l'entrée quelles sont celles à la sortie? Est-ce qu'il y a des résultats intermédiaires et quels sont-ils?
- Que fait-on si le discriminant est négatif? S'il est égal à zéro? Dans ces cas, quelles sont les possibilités?

D>0

$$x1 = u + v - \frac{a}{3}$$

$$x2 = -\frac{u}{2} - \frac{v}{2} - \frac{a}{3} + i\frac{\sqrt{3}}{2}(u - v)$$

$$x3 = -\frac{u}{2} - \frac{v}{2} - \frac{a}{3} - i\frac{\sqrt{3}}{2}(u - v)$$

D<0

Pour obtenir les racines réelles on peut utiliser la forme trigonométriques :

$$x_{1,2,3} = 2\sqrt{-\frac{p}{3}}\cos\frac{1}{3}\left(\arccos\left(-\frac{q}{2}\sqrt{-\frac{27}{p^3}}\right) + 2kp\right) - \frac{a}{3},$$

$$k = 1,2,3$$

D=0

$$x_1 = x_2 = -\frac{3q}{2p}$$

Une fois la méthode de Cardan démystifiée, l'étudiant est confronté à la question fondamentale :

- Comment décrire toutes ces expressions en C++? Quelles sont les règles?

L'étudiant est ici à la recherche d'une recette, d'une procédure. La structure des opérations permettant d'exprimer la solution à son problème est appelée un algorithme. Par la suite, l'étudiant devra traduire son algorithme sous la forme d'une séquence d'instructions respectant la syntaxe et les règles sémantiques d'un langage de programmation. Cet exemple illustre que la route menant de l'énoncé du problème à la rédaction du programme est parsemée d'embûches se situant à différents niveaux d'abstraction.

1.1.3. Détection, interprétation et correction de.

Pendant le processus de création de son programme l'étudiant commet des erreurs. Certaines erreurs sont détectées à la compilation ou à l'exécution du programme par l'environnement de développement qui réagit en conséquence. L'étudiant doit pouvoir identifier l'erreur et savoir de quelle façon elle peut être corrigée (erreurs de compilation ou d'exécution). Il y a par contre d'autres erreurs qui ne sont pas détectées directement par l'environnement et qui doivent être trouvées et interprétées par l'étudiant lui-même (erreurs dues à un mauvais algorithme). Plus précisément, nous distinguons deux groupes d'erreur :

1. Erreur de compilation – ces erreurs sont détectées durant la compilation par l'environnement de programmation.
2. Erreur d'exécution – toutes les autres erreurs – dynamique, logique, algorithmique.

Le schéma suivant (Figure 1.1) présente quelques exemples concernant les deux types d'erreurs.

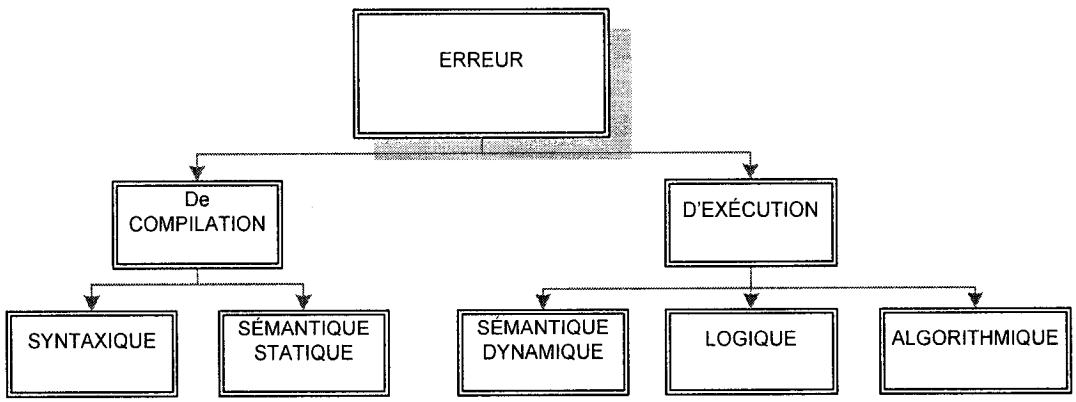


Figure 1.1 : Types d'erreur

Lors de la correction d'erreur, l'étudiant est dans une situation active de résolution de problèmes. De plus, puisqu'il développe une solution pour un problème issu d'un domaine particulier, il doit maîtriser à la fois des connaissances du domaine de l'application et du langage de programmation utilisé. La détermination des racines d'une expression de troisième degré, présentée précédemment, en est un bon exemple.

On a décrit dans les paragraphes précédents quelques obstacles qui compliquent le bon apprentissage de la programmation. Soulignons-les encore une fois :

1.1.1. Hétérogénéité des étudiants.

1.1.2. Caractère abstrait des concepts informatiques.

1.1.3. Détection, interprétation et correction de l'erreur.

1.2.Idée de la recherche.

Afin d'identifier les principaux styles d'apprentissage d'un apprenant de la programmation nous avons pensé soumettre nos étudiants à une expérience offrant deux types distincts d'activités. Dans un premier temps il s'agira de réaliser des exercices à caractère abstrait et dans un deuxième temps, il s'agira de réaliser des exercices à caractère concrets en utilisant un montage électronique.

Nous croyons que nous tirerons plusieurs informations intéressantes de la comparaison des résultats obtenus de chaque type d'activité. De plus nous identifierons le profil psychologique de chaque candidat de l'expérience en leur demandant de compléter un test permettant d'obtenir cette information.

CHAPITRE 2. Quelques moyens utilisés pour favoriser l'apprentissage de la programmation

2.1.Méthodes d'apprentissage

Dans ce chapitre nous décrirons différentes méthodes couramment utilisées pour favoriser l'apprentissage de la programmation. Nous proposerons une classification et décrirons les traits caractéristiques de chaque méthode.

2.1.1. Traditionnelles

Dans cette classe nous incluons les différentes méthodes traditionnelles utilisées dans le processus d'apprentissage de la programmation et sans lesquelles ce dernier est presque impensable.

Les cours

Cette bonne vieille méthode où le professeur est devant le tableau et explique les rudiments de la programmation continue d'être valable de nos jours. Bien sûr, la façon de présenter les notions de la programmation a changé. Le temps où le professeur écrivait au tableau et que les étudiants se dépêchaient à recopier sans réussir à suivre ses explications a évolué. Du moins, nous l'espérons.

Nous sommes maintenant à l'ère des technologies qui offrent différentes façons de présenter la matière [15]. De cette façon l'attention des étudiants est centrée sur l'information, présentée d'une façon captivante. Cette information étant disponible de différentes façons, l'étudiant peut aisément questionner le professeur sur les notions présentées sans craindre de perdre le fil de la présentation. De cette manière, il y a un effort de transformer l'étudiant d'écouteurs passif en participant actif [1].

En utilisant les nouvelles technologies on a la possibilité d'offrir plus de matériel et plus d'exemples qui peuvent éclaircir les notions complexes. Les étudiants ne perdent plus de

temps à copier en détail le texte du tableau. Leur attention est consacrée entièrement sur le contenu.

Les labos

Dans les heures passées au laboratoire l'étudiant montre, dans la pratique, quelle partie du matériel de cours est bien compris et laquelle non. C'est ici le bon moment de discuter de nouveau de certaines notions et définitions concernant la programmation et de quelle façon elles peuvent être utilisées dans l'élaboration de leurs programmes.

Dans le processus de création de différents programmes les étudiants voient leurs points faibles et leurs points forts. Ils prennent ainsi conscience des points qu'ils doivent approfondir le plus.

Les discussions pendant ces heures de laboratoire sont un moment important dans l'assimilation des notions de base de la programmation.

La littérature supplémentaire

Au début de chaque cours de programmation les professeurs recommandent certains livres aux étudiants dans le but de les aider à s'outiller adéquatement pour leurs activités personnelles d'apprentissage des concepts et de certaines techniques de la programmation. Chaque livre, chaque document ou site électronique est bienvenu s'il aide vraiment à la compréhension de ces notions de base.

2.1.2. Non traditionnelles

Qu'importe la panoplie des instruments et des méthodes d'apprentissage les étudiants continuent à éprouver des difficultés à assimiler certaines notions de base avec leur caractère abstrait.

Il est clair que dans certains cas les méthodes traditionnelles ne sont pas suffisantes. Il est nécessaire de chercher de nouveaux moyens, de nouvelles façons pour présenter

l'information plus clairement pour que les étudiants puissent assimiler ce que représentent les boucles et les tableaux par exemple.

Dans certaines universités on a introduit des heures supplémentaires dans le but d'aider les étudiants qui continuent à éprouver des difficultés à comprendre certaines notions ou pour les étudiants qui font pour la deuxième fois le même cours.

Utilisation de méthodes plus attractives pour expliquer certaines définitions de base

L'un des moyens est d'essayer de rendre les étudiants plus actifs, de participer plus vivement dans le processus d'apprentissage de la programmation. On essaie de travailler avec des participants actifs et non avec de récipients passifs [1]. On cherche un moyen qui va les impliquer dans le processus de création d'un programme.

Jenkins[1] propose une façon intéressante pour une meilleure compréhension des tableaux, rétrospectivement des *boucles*. Pour réaliser cela il utilise un groupe d'étudiants qui jouent le rôle d'un tableau. Chaque étudiant représente un élément de ce tableau. Le tableau est déclaré en choisissant un volontaire. Cet étudiant (c'est valable pour les autres étudiants qui participent aussi) possède une grande feuille qui montre son numéro (son *index*). Sur la feuille il y assez de place pour inscrire également la valeur de « cet élément ». Après on choisit « le deuxième élément » (un autre étudiant). De cette façon cette action se répète jusqu'à la définition de tous « les éléments du tableau ». Après la déclaration du tableau, c'est le tour du professeur d'expliquer que chaque élément peut sauvegarder seulement une valeur. Il explique aussi quelles opérations peuvent être effectuées sur les tableaux. Tout ça pourra être ensuite décrit en pseudo code ou en langage étudié.

Après ces explications on peut passer à des problèmes plus complexes comme par exemple de trouver l'élément minimal ou maximal dans un tableau, comment trier un tableau et d'autres.

De la même façon on procède avec la présentation des **sous-programmes** et surtout de leurs paramètres. Pour les étudiants il est difficile de comprendre la différence entre paramètres par valeur et paramètres par adresse.

De nouveau l'idée est d'expliquer à l'aide des étudiants comment fonctionnent les paramètres. On choisit un étudiant « sous-programme » et des étudiants « paramètres ». Dans le jeu on explique le fonctionnement des paramètres par valeur et des paramètres par adresse. De cette manière on introduit implicitement du même coup la notion de pointeur.

Introduction des grilles (réseaux)

Hagan [2] décrit une autre expérimentation en utilisant des grilles. Les étudiants sont divisés en quelques groupes. Chaque groupe doit répondre à un certain nombre de questions. Bien sûr, au début ils ont assez de temps pour discuter ou faire une analyse. Après on compare les résultats et on explique pourquoi on choisit telle réponse. S'il y a des fautes on les explique en discutant. La Figure 2.1 présente un exemple d'une telle grille [2].

Les étudiants par exemple doivent à partir du contenu de la grille, répondre aux questions suivantes [2] :

- Trouver et donner des exemples pour :
  - le nom d'une fonction;
  - une expression avec un opérande arithmétique;
  - une expression qui va changer la valeur d'une variable
- Quelles sont les deux instructions qui vont afficher le même résultat ?
- Est-ce qu'il y a des instructions identiques à celle de B1 ?

Les mimiques.

« Les mimiques » [2] est une technique où les étudiants suivent les réactions sur le visage du professeur pendant qu'il travaille sur un problème au tableau. Les étudiants

suivent ses explications et ses réflexions. Ils doivent ensuite reproduire les mêmes mimiques. Un étudiant commence sur le tableau, après quelques pas un autre peut continuer. Si quelqu'un fait une erreur on la discute (pourquoi elle surgit, quelles seront les conséquences, comment on peut l'éviter) et on continue de cette façon jusqu'à la rédaction de tout le programme.

	A	B	C	D
1	cout<<"H" <<"e"<<"l" <<"p";	sum = sum+2;	2*sum+2-sum;	cout <<"Help";
2	#include <iostream.h>	int main() { }	int pageNumber; pageNumber = 0; pageNumber = pageNumber+1;	sum = sum +3*5-13;
3	int pageNumber=1;	cin>>studentId;	Sum+2;	2*M_PI*radius;
4	char input; Input = "y";	int lineNumber; lineNumber = 9; cout<<"Line:" << lineNumber%2;	float total; total = 45.5; total/5; cout<<total;	double circleArea;

Figure 2.1 : Exemple d'une grille

Toutes ces techniques et méthodes visent une meilleure révision du matériel et à aider les étudiants à surmonter les difficultés concernant l'abstraction des notions.

Soulignons que ces expérimentations sont réalisées hors des cours réguliers soit dans des cours supplémentaires. Par contre, on ne peut pas toujours organiser ces séances de

cours supplémentaires, soit à cause d'un manque du temps, soit à cause d'un manque de ressources financières.

Notre question demeure : Comment dans les heures régulières (surtout les cours de labos) impliquer plus intensément l'étudiant dans son processus d'apprentissage de telle façon qu'ils puissent assimiler plus facilement les concepts de base de la programmation?

2.2.Les environnements et les applications

Une autre méthode utilisée pour aider les étudiants dans leurs premiers pas dans la programmation est l'utilisation des environnements de programmation. Des environnements comprenant, entre autres, un compilateur, des outils de diagnostic, un système d'animation du programme comme AnimPascal [4]. Satratzemi mentionne aussi les applications Thetis et DYNALAB [4]. Certains environnements enregistrent le travail des étudiants (design, développement, vérification, débogage). Muni de ces informations, le professeur peut intervenir en temps opportun tout en fournissant des explications mieux ciblées. De cette façon un tel environnement donne la possibilité aux professeurs d'obtenir une image plus claire et de pouvoir réagir à temps.

2.3.« Minute paper »

La technique « Minute paper » correspond à utiliser à la fin de chaque séance de cours de quelques minutes pour répondre à quelques questions telles que présentées au tableau 2.1. [22]

De cette façon, l'étudiant est invité à réfléchir sur le contenu du cours qui se termine de manière à vérifier s'il a réussi à saisir l'essentiel. Le plus important et de poser des questions concrètes sur le matériel qui risque d'être mal compris ou partiellement compris.

Tableau 2.1 Exemple de questions d'un « Minute paper »

1. Quelle est la notion la plus importante que vous avez apprise aujourd'hui?
2. Quelles sont les questions dans votre tête en ce moment?

CHAPITE 3. Les modèles psychologiques de l'apprentissage

Chaque étudiant a sa propre façon d'apprendre. Il a ses propres préférences et caractéristiques, il accepte à sa manière les différents faits et informations. C'est ce qu'on appelle « **styles d'apprentissage** » ou « **modèles psychologiques d'apprentissage** ».

> *« Students have different **learning styles** – characteristic strengths and preferences in the way they take in and process information. »* (Felder) [5]

Il existe plusieurs modèles psychologiques d'apprentissage. Chacun a ses priorités et ses avantages. Nous vous exposons sommairement quatre modèles qui, à la lumière de nos lectures, semblent les plus populaires.

3.1.MBTI (Myers-Briggs Type Indicator)

Cet instrument a été développé par Isabel Briggs Myers et sa mère Katharine Briggs. Ce modèle décrit un individu selon une notion type de personnalité ou type psychologique. Elles basent leurs recherches sur la théorie des types psychologiques de Carl Jung (psychologue suisse).

> *« Psychological type is a theory of personality developed by Swiss psychiatrist Carl G. Jung to explain the normal differences between healthy people. Based on his observations, Jung concluded that differences in behavior result from individuals' inborn tendencies to use their minds in different ways. As people act on these tendencies, they develop predictable patterns of behavior. Jung's psychological type theory defines eight different patterns, or types, and gives an explanation of how type develops. »[11]*

-Isabel B. Myers, Introduction to Type, 6th Edition, p.6
« Jung's therapeutic approach aimed at reconciling the diverse states of personality, which he saw divided not only into the opposites of introvert and extrovert, but also into those of sensing and intuiting, and of feeling and thinking. By understanding how the personal unconscious integrates with the collective unconscious, Jung theorized, a patient can achieve a state of individuation, or wholeness of self. » [19]

Carl Jung [25] « déduit de ses observations que l'esprit humain dispose de quatre fonctions psychologiques de base :

- deux fonctions **irrationnelles de Perception P**, lui permettant de recueillir de l'information de deux manières opposées : l'**iNtuition N** ou la **Sensation S**;
- deux fonctions **rationnelles de Jugement J**, lui permettant de traiter cette information pour aboutir à des conclusions : **la Pensée T** (Thinking) ou la **Sentiment F** (Feeling).

Il a aussi observé que les individus ont tendance à trouver leur énergie et à être dynamisés :

- soit par l'environnement extérieur, les activités et les expériences : attitude d'**Extraversion E**;
- soit par l'univers intérieur des idées, des souvenirs et des émotions : attitude d'**Introversion I**. »

« La Sensation (c.-à-d., le sentiment de perception) vous dit que quelque chose existe;
la réflexion vous dit ce que c'est;
le sentiment vous dit si c'est agréable ou pas;
et iNtuition vous dit d'où il vient et il va. » – C.G.Jung [25]

« L'apport de Briggs-Myers a été d'identifier une 4-ème préférence, entre justement la Perception et le Jugement; l'indicateur MBTI identifie alors 16 grands types de

personnalité à partir des deux préférences possibles sur chacune des quatre dimensions précédentes. » [25]

Les *Sensitifs* (**S**) acceptent l'information par leurs cinq sens. Ce sont des personnes pratiques, réalistes qui aiment les faits et comptent sur l'expérience réelle.

Les *iNtuitifs* (**N**) préfèrent les faits abstraits, ils s'intéressent aux relations qui existent parmi les choses. Ils sont plus imaginatifs. [25]

Selon la façon de prendre les décisions on peut distinguer des personnes qui réfléchissent et jugent en se basant sur une analyse profonde et objective, et d'autres qui réagissent d'une façon spontanée et basée sur les actions des autres personnes. Dans le premier cas ce sont les *Penseurs* (**T**) et dans le deuxième – les *Sentimentaux* (**F**). Les premiers aiment les règles et la vérité, les deuxièmes évaluent subjectivement en cherchant l'harmonie.

Extravertis (**E**) ceux qui réagissent d'abord et après réfléchissent, qui se sentent déprimés s'ils n'ont pas la possibilité de contacter avec le monde extérieur, qui créent très vite des nouvelles relations et *Introvertis* (**I**) – les personnes qui réfléchissent et après réagissent, qui ont besoin de rester seuls de temps en temps pour « recharger leurs batteries ». [13]

Les caractéristiques des tous ces 16 types psychologiques sont décrites en [12] [14] [25]. Par exemple un étudiant peut être **ENTP** (extraverti, intuitif, penseur, perceveur) et l'autre **ISTJ** (introverti, sensitif, penseur, jugeur).

Des études ont montré que les étudiants qui font des études en ingénierie sont en général de type **ST** (Sensitif, Penseur) et ceux en Informatique sont du type **NT** (Intuitif, Penseur) [20].

Myers - Briggs Type Instrument représente un questionnaire avec un nombre de questions permettant de déterminer le type d'un étudiant. Pour obtenir cet instrument on doit s'adresser aux organismes autorisés par MBTI Trust.

Le tableau 3.1 résume les traits caractéristiques du modèle MBTI [5].

Tableau 3.1 Modèle psychologique de MBTI

Préférences		Type	Caractéristiques
Orientation de l'énergie	E	***Extraverti ou***	Expose les choses à l'extérieur, vers le monde
	I	***Introverti***	Garde les choses en dedans, dans le monde des idées
Recueil de l'information	S	***Sensitif ou***	Plus pratique, orienté vers les détails
	N	***Intuitif***	Imaginatif, orienté vers les conceptions
Prise de décision	T	***Penseur ou***	Prend des décisions basées sur les règles et la logique
	F	***Sentimental***	Prend des décisions basées sur des considérations personnelles et humanitaires
Mode d'action	J	***Jugeur ou***	Suit toujours les agendas, termine le travail même s'il n'est pas prêt
	P	***Perceveur***	Essaie de changer les circonstances, ne termine pas le travail afin d'obtenir les résultats

3.2.Modèle de Kolb

Une autre méthode qui a pour but d'explorer les styles d'apprentissage est celle de Kolb. Selon Kolb, le style se base sur les préférences des étudiants qu'on peut diviser en deux groupes:

I. L'expérience concrète - La conceptualisation abstraite

II. L'expérimentation active - L'observation réflexive

Dans le premier groupe l'accent est sur « *Comment les étudiants acceptent l'information*? » et dans le deuxième « *Comment ils analysent cette information*? »

En disposant cette information dans un schéma nous obtenons une image plus claire des différents types d'apprentissage de Kolb (Figure 3.1) [22].

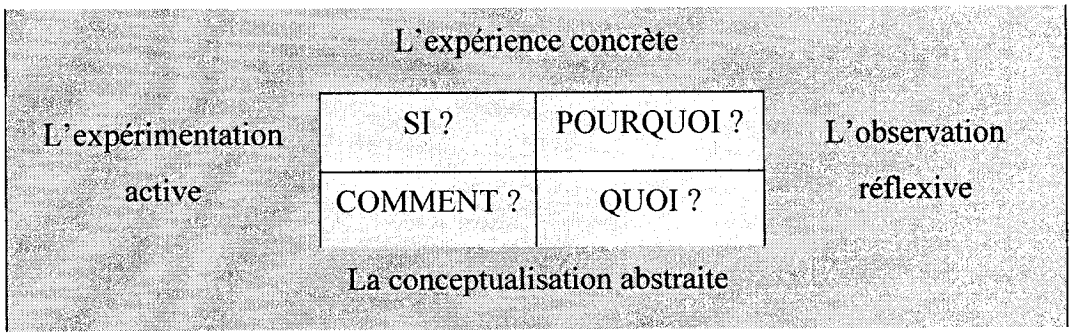


Figure 3.1: Types d'apprentissage de Kolb

Les quatre groupes d'étudiants sont décrits brièvement dans le tableau 3.2. [5],[10]:

Tableau 3.2 Modèle psychologique de Kolb

Type 1 – concret, réflexif (Pourquoi?)
La fonction du professeur – un motivateur
Les étudiants de ce groupe essaient toujours de relier leur expérience avec le matériel du cours et avec leurs futures intentions.
Type 2 – abstrait, réflexif (Quoi?)
La fonction du professeur - un expert
Les représentants de ce type préfèrent qu'on leur présente l'information d'une façon organisée, bien structurée et qu'on leur laisse un peu de temps pour réfléchir
Type 3 – abstrait, actif (Comment?)
La fonction du professeur – un entraîneur
Dans ce groupe on voit des étudiants qui apprennent dans la pratique, en utilisant la stratégie « par essais et erreurs ».
Type 4 – concret, actif (Si?)
La fonction de professeur – un surveillant
Les étudiants aiment utiliser les connaissances d'un cours dans des nouvelles situations pour résoudre des problèmes.

3.3.Herrmann Brain Dominance Instrument

« Ned Herrmann, founder of Herrmann International and the originator of Whole Brain Thinking *first pioneered the study of the brain in the field of business while working as Manager of General Electric Corporation's Management Education. »[23]*

Sa théorie se base sur le fonctionnement du cerveau lors d'un processus d'apprentissage. Il « divise » le cerveau en quatre parties et explique que l'apprenant utilise une de ces parties selon le moment où il est rendu dans son processus d'apprentissage. Il précise que le processus d'apprentissage peut être différent dépendamment de la partie du cerveau utilisée.

Quelle partie est utilisée dépend de la personne, de ces propres préférences. Une personne peut utiliser plusieurs parties mais il y a toujours une qui est préférée (primaire) et une autre qui joue le rôle secondaire (supplémentaire). Il est possible d'introduire une troisième partie dans le processus.

La figure 3.2 présente quatre types ou quadrants définis par Hermann définit selon différentes caractéristiques suivantes [5]. En général la plupart des professeurs sont orientés vers le quadrant A, c.-à-d. qu'ils sont des personnes logiques, assez critiques et travaillent avec les faits et les données. Leur façon de travailler est plutôt orientée vers les représentants du groupe B (les étudiants bien organisés et méthodiques) mais elle est en désaccord avec les représentants des quadrants C et D. Environ à 20%-40% des étudiants sont des représentants des groupes C et D [5].

Hermann Brain Dominance Instrument est un questionnaire de 120 questions qui donne la possibilité d'apprécier le type de chaque personne ou étudiant [24]. Tous les droits de cet instrument sont réservés.

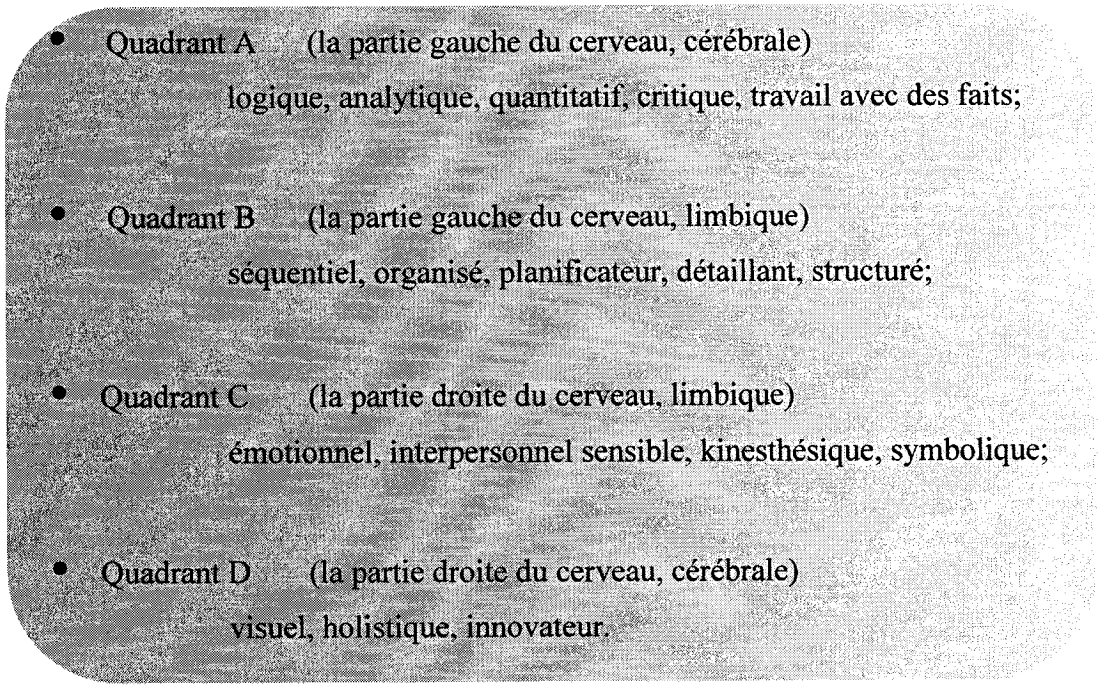


Figure 3.2 : Types d'apprentissage de Herman

3.4.Le modèle de Felder

D'après Felder [7] l'apprentissage est un processus qui peut être divisé en deux parties : la réception de l'information et son traitement. L'information peut être extérieure ou interne. Les étudiants font la sélection, c.-à-d. ils choisissent quoi apprendre et quoi ignorer. Le traitement peut être une simple mémorisation ou une analyse plus profonde ou une interaction avec les autres.

Le modèle de Felder classifie les étudiants selon les moyens pertinents qu'ils utilisent pour percevoir l'information et traiter cette information. C'est un modèle qui vise plus particulièrement des étudiants des sciences pures (ingénieurs, mathématiciens, chimistes, informaticiens).

Le style d'apprentissage d'un étudiant peut être défini à partir des réponses obtenues des questions suivantes [7] :

1. Quel type d'information l'étudiant perçoit-il plus facilement ?

Sensory (extérieur) – les images, les sons, les sensations physiques.

Intuitive (intérieur) – les possibilités, les suppositions.

2. Quels sont les canaux sensitifs pour percevoir l'information externe?

Visual – les images, les diagrammes, les graphes, les démonstrations.

Auditory – les mots, les sons.

3. Quelle organisation de l'information l'étudiant préfère-t-il?

Inductive – les faits et les observations sont présentés, il reste à soutirer les principes.

Deductive – les principes sont donnés, on doit apprécier les conséquences et les applications.

4. Comment les étudiants préfèrent traiter l'information?

Actively – dans un travail pratique.

Reflectively – par introspection.

5. Comment les étudiants apprennent-ils?

Sequentilly – successivement, pas à pas.

Globally – en grands sauts, en grands étapes.

Les dimensions des différents styles d'apprentissages sont résumées dans le tableau 3.2. [7].

Tableau 3.3 Styles d'apprentissage préférés selon le modèle de Felder-Silverman

Styles d'apprentissage préférés		
sensory	**(sensoriel)**	*perception*
intuitive	**(intuitif)**	
visual	**(visuel)**	*introduction (input)*
auditory	**(verbal)**	
inductive	**(inductif)**	*organisation*
deductive	**(déductif)**	
active	**(actif)**	*traitement*
reflective	**(réflexif)**	
sequential	**(séquentiel)**	*compréhension*
global	**(global)**	

Les étudiants *Sensoriel* **(SNS)** et *Intuitif* **(INT)**

D'après Felder [7] les étudiants du type sensoriel préfèrent les faits, les données et les expérimentations. Pour résoudre un problème ils préfèrent les méthodes standardisées. Ils n'aiment pas les surprises et les complications. Ils sont patients et mémorisent facilement les données. Ces étudiants sont très attentifs mais lents.

Les étudiants du type intuitif préfèrent les principes et les théories. Ils sont tentés par des innovations et ils détestent les répétitions. Ils se fatiguent vite par des détails mais ils agréent les complications, c'est un défi pour eux. Ils comprennent avec facilité les nouvelles conceptions. Ces étudiants sont rapides mais non-attentifs.

Les étudiants *Visuel* **(VIS)** et *Verbal* **(VRB)**

L'information s'introduit (pénètre) dans un être humain de différentes façons. On peut les diviser en trois parties [7]:

- Visuel – les vues, les images, les diagrammes
- Verbal – les mots dits, les sons

- Cinesthésique – les goûts, les contacts, les odeurs

Ce qui nous intéresse ce sont les deux premiers groupes. Le troisième présente un intérêt surtout pour des domaines comme la gastronomie, la médecine ou la pharmacie.

Les étudiants du type visuel apprennent mieux quand ils regardent les images, les diagrammes, les schémas, les démonstrations. Ils oublient plus facilement des faits qui sont dits et non pas vus.

Les étudiants du type verbal apprennent beaucoup quand ils écoutent et beaucoup plus quand ils écoutent et parlent sur le sujet en question. Ils préfèrent les discussions plutôt que les démonstrations visuelles et ils apprennent en expliquant des choses aux autres.

Les étudiants *Inductif* et *Déductif*

L'induction est un processus qui commence par les particularités (les observations, les mesures, les données) et qui finissent par les généralités (les règles, les lois, les théories)
La déduction est le contraire de l'induction – une chose qui déduit les conséquences.

D'après les recherches de Felder la plupart des étudiants en ingénierie sont inductifs. Ils préfèrent observer d'abord le phénomène et après d'accepter et comprendre la théorie qui va l'expliquer.

Les étudiants *Actifs* (**ACT**) et *Réflexifs* (**REF**)

Un étudiant actif est une personne qui se sent plus confortable avec une expérimentation active qu'avec une observation réflexive. Il n'apprend pas beaucoup dans des situations qui exigent de lui d'être passif. Les étudiants de ce groupe se sentent très à l'aise parmi les autres. Ils aiment travailler en équipe. Ils essaient toujours d'être des expérimentalistes.

Les étudiants du type réflexif préfèrent l'observation plutôt que l'expérimentation. Ils ne se sentent pas bien dans des situations qui ne leur permettent pas de réfléchir

tranquillement. S'ils ont le choix, ils vont choisir de travailler seuls. Ce sont des théoriciens.

Les étudiants *Séquentiels* (**SEQ**) et *Globaux* (**GLO**)

Il y a des étudiants qui se sentent confortable avec la façon d'enseigner, qui consiste à offrir le matériel d'une façon régulière, sujet après sujet, et que la difficulté augmente graduellement. Ce sont les étudiants séquentiels. Ils suivent un plan linéaire pour résoudre un problème. Ils peuvent travailler avec un matériel qui n'est pas complètement compris, seulement une partie. Ils sont bons dans la convergence de la réflexion et de l'analyse. Ils apprennent mieux quand on leur offre le matériel « pièce par pièce », quand la difficulté augmente progressivement.

Les étudiants du type global ne peuvent pas étudier de la manière décrite précédemment. Ils peuvent se sentir « perdus » pendant des jours et des semaines dans le labyrinthe du matériel et après, tout à coup, la lumière s'allume et le contenu est assimilé. Parfois ils agissent intuitivement sans réussir à expliquer pourquoi ils le font [7]. Pour plus détails vous pouvez consulter les ouvrages [6], [8], [9].

Pour apprécier le type d'un étudiant selon le modèle de Felder on doit posséder le questionnaire nécessaire. Il est distribué librement et on peut l'obtenir gratuitement. Vous trouverez un exemplaire à l'annexe A. Que représente-t-il? C'est un ensemble de 44 questions où la réponse possible à chaque question est binaire, soit a) ou soit b). Après qu'une personne ait complété le questionnaire c.-à-d. elle a précisé son choix a) ou b) à toutes les questions, elle peut commencer à compiler les résultats. Pour ce faire, elle doit suivre les instructions qui accompagnent le questionnaire (Annexe B).

Premièrement elle doit transmettre les réponses données dans un tableau avec 4 colonnes en suivant exactement les instructions données (voir annexe B). Après elle doit compter le nombre de réponses a) et le nombre de réponses b) pour chaque colonne. Ensuite il suffit de calculer la différence en valeur absolue entre le nombre de réponses a) et le

nombre de réponses b) pour chaque colonne. Finalement, on préfixe le résultat de la lettre du plus grand nombre de réponses. Après ces opérations on obtient 4 réponses comme résultat.

Par exemple : 7a - 3b – 11a – 5b.

Comment les interpréter?

On les dispose dans un autre tableau qui représente le profil de votre style d'apprentissage. Ce tableau représente 4 échelles aux extrémités desquelles sont marqués les différents styles. La figure 3.3 constitue un exemple où le résultat 7a - 3b – 11a – 5b. est consigné.

ACT			X										**REF**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
SNS								X					**INT**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
VIS	X												**VRB**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
SEQ									X				**GLO**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	

Figure 3.3 : Exemple d'un profil d'un style d'apprentissage

- Dans cet exemple on a une personne du type **ACT** (actif) avec des <u>préférences marquées</u> pour cette dimension.
- Elle est assez <u>bien équilibrée</u> dans l'échelle SNS – INT, mais elle est quand même du type **INT** (intuitif) avec ses caractéristiques. Elle peut se débrouiller assez facilement avec les méthodes de l'autre côté de cette échelle (dans notre cas SNS) si c'est nécessaire.
- Cette personne démontre des <u>préférences extrêmement marquées</u> dans le troisième cas. Elle est surtout une personne **VIS** (visuelle). Même parfois elle

pourra avoir des difficultés en utilisant les méthodes caractéristiques pour les personnes du type VRB (verbal).

- Dans le dernier cas elle a des préférences marquées vers le type **GLO** (global).

Précisons que lorsqu'une personne est d'un type particulier cela ne signifie pas qu'elle ne peut pas se débrouiller avec les méthodes du type opposé. Un individu pourrait toujours utiliser des méthodes appropriées d'un autre type pour résoudre un certain problème.

On souligne encore une fois qu'on divise les étudiants en différents groupes selon leurs **préférences**. Le tableau 3.3 suivant résume les types psychologiques selon le modèle de Felder.

Tableau 3.4 Modèle psychologique de Felder-Silverman

sensoriel (concret, pratique, orienté vers les faits et procédures) **ou** ***intuitif*** (conceptuel, innovateur, orientés vers les théories);	**SNS - INT**
visuel (préfère les représentations visuelles avec des images, diagrammes…) ou ***verbal*** (préfère les explications écrites ou verbales);	**VIS - VRB**
inductif (préfère les présentations qui commencent par le spécifique vers le général) ***ou déductif*** (préfère les présentations qui commencent par le général vers le spécifique);	-
actif (préfère travailler en équipe) ***ou réflexif*** (préfère travailler seul);	**ACT - REF**
séquentiel (linéaire, bien ordonné) ***ou global*** (holistique, penseur systématique);	**SEQ - GLO**

Le modèle de Felder a plusieurs points communs avec les autres modèles décrits précédemment. La première dimension sensing/intuition est la même que celle du

modèle de Myers-Briggs qui se base sur la théorie de Jung et la partie active-réflexive est empruntée de la théorie de Kolb et plus précisément la partie gauche et droite du schéma (voir 3.3) correspond à la classification Actif-Réflexif de Felder et la partie au bout et en bas à celle Sensoriel-Intuitif de Felder. Par contre, en ce qui concerne les autres parties de Felder, Visuel-Verbal et Global-Séquentiel, une telle correspondance directe manque [5].

CHAPITRE 4. Description de l'expérimentation

Rappelons, tout d'abord, que l'objectif de cette expérimentation est de vérifier comment les représentants d'un modèle psychologique particulier vont se débrouiller avec des exercices aux caractères différents. Avec quel type d'exercice ils vont avoir plus de difficultés. Avec quel type d'exercice ils seront plus à l'aise. Notre hypothèse est que les étudiants qui sont du groupe psychologique ACT vont mieux travailler avec l'exercice à caractère concret tandis que les représentants du groupe REF vont mieux se débrouiller avec l'exercice à caractère abstrait.

Nous allons initialement décrire les exercices que nous avons conçus. Nous allons ensuite discuter comment l'atelier de programmation s'est déroulé et quels sont les résultats obtenus.

4.1. Test Felder.

Dans le Chapitre 3 nous avons présenté quatre modèles psychologiques d'apprentissage (MBTI, Felder, Kolb et Herman brain dominance). Pour notre recherche on a choisi celui de Felder. C'est un modèle psychologique d'apprentissage très puissant.

- Il possède l'avantage d'exiger peu de questions au répondant, soit 44, par rapport à ceux de MBTI et HBDI. Il nécessite donc moins de temps à compléter ce qui est important pour nous.
- Il n'est pas nécessaire d'obtenir une permission spéciale pour l'utiliser. Cela nous a permis de gagner du temps et éviter de débourser des coûts.
- C'est un test qui est très utilisé en sciences pures. Il a déjà été utilisé à l'École Polytechnique par le département de Génie chimique.

Selon différentes recherches réalisées par Felder, on doit s'attendre à avoir des étudiants de types *visuel, sensoriel, inductif* et *actif*. Parfois les étudiants les plus créatifs sont du type *global*. [7].

« ...Many or most engineering students are visual, sensing, inductive and active, and some of the most creative students are global... » (Felder)

4.2.Montage électronique.

Pour rendre concret l'exercice de programmation nous avons recours à un montage électronique. La pièce maîtresse de ce montage est un microcontrôleur servant à manipuler des luminodiodes et un moteur. Pour ce faire, on utilise une plaque sur laquelle un **microcontrôleur PIC16F876** est installé. Ce dernier a l'avantage d'avoir une mémoire flash permettant de mémoriser à plusieurs reprises un programme. Ce programme est créé sur un ordinateur en utilisant le langage de programmation C. Après avoir été créé le programme est compilé et transféré en mémoire flash du microcontrôleur. Le transfert est réalisé en utilisant le port série. Ce microcontrôleur nous donne la possibilité de développer nos propres programmes et nous en avons profité. On utilise un **capteur optoélectronique** (une luminodiode et une cellule photoélectrique) à l'entrée du microcontrôleur et un **transistor** (monté en émetteur commun) pour manipuler le moteur. La place de ce dernier (le moteur) est près du capteur optoélectronique d'une telle manière qu'il permet à une **hélice** de passer au-dessus.

On utilise également un **condensateur** pour réguler l'alimentation. Il s'occupe du parasitage en sus du moteur et utilise la même alimentation que le microcontrôleur. Le plus grand avantage du microcontrôleur est sa facilité de programmation. Le montage électronique est assez simple et facile à utiliser (Figure 4.1). Il permet une grande panoplie d'exercices. La manipulation du moteur offre également la possibilité d'une variété d'exercices qui plaisent aux étudiants. On a la possibilité de changer la difficulté des exercices et d'introduire de nouveaux éléments.

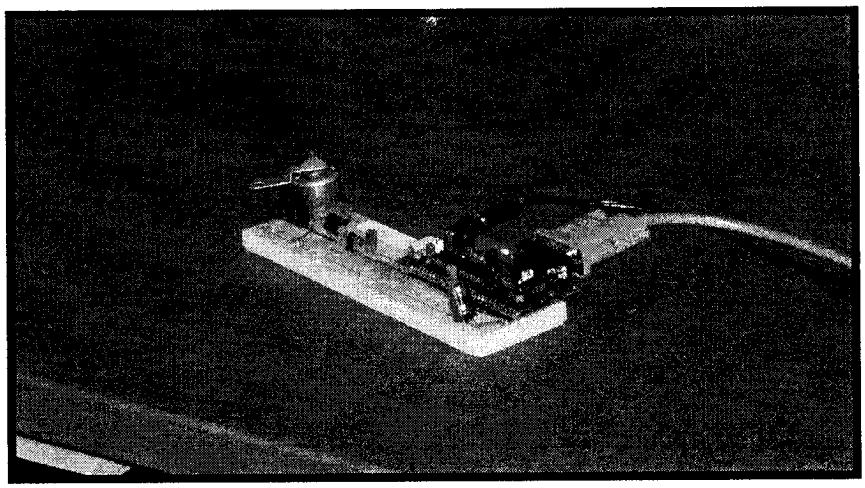


Figure 4.1 : Le montage électronique

Le montage permet d'observer comment les étudiants utilisant les différents styles d'apprentissage réagissent et comment ils résolvent les problèmes.

4.3.Atelier de programmation.

L'atelier propose deux exercices différents selon deux groupes psychologiques. L'un utilisant un montage électronique et l'autre sans utiliser le montage.

4.3.1. Exercice : Structures booléennes et de décision

Une partie de ces exercices a pour but de familiariser les étudiants avec l'expression booléenne et les structures de décision.

4.3.1.1. Exercice à caractère abstrait

Les étudiants doivent s'imaginer qu'ils manipulent un robot. L'exercice consiste à déplacer le robot dans un espace bien défini, par exemple un rectangle 20×50. Le robot peut se déplacer selon les quatre points cardinaux : nord, sud, est et ouest. L'objectif est de s'assurer que le robot demeure dans le rectangle.

Afin de faciliter le traitement, nous précisons la position initiale du robot et nous décrétons que les mouvements possibles du robot doivent toujours être précisés selon des coordonnées positives. En clair, le rectangle est comme une matrice d'indices 1..20×1..50 et les déplacements doivent être spécifiés par des entiers positifs de manière à préciser l'incrément selon la ligne et la colonne. De cette façon, les étudiants peuvent dès le début vérifier la position initiale du robot et calculer si les mouvements suivants sont possibles. À chaque requête de déplacement l'étudiant doit décider si le mouvement demandé peut être effectué, c.-à-d. calculer la nouvelle position et vérifier si le robot demeure dans les coordonnées 1..20×1..50 .

4.3.1.2. Exercice à caractère concret

Dans cette partie, les étudiants doivent manipuler quelques luminodiodes (led) situées sur le montage électronique, c.-à-d. les allumer et les éteindre. Cependant, le processus d'allumer une led est en fonction de l'état des autres leds. Par exemple :

on peut allumer la led jaune uniquement si la led rouge est déjà allumée;
la led orange peut être allumée si les leds jaune et rouge sont déjà allumées, etc.

Des conditions similaires doivent être vérifiées pour éteindre une led.

Nous n'oublions pas qu'il s'agit de débutants en programmation et pour cette raison nous leur proposons de découvrir préalablement la bonne séquence pour chaque montage. Pour réaliser ça on leur offre un programme qui les aidera à mieux comprendre l'énoncé de l'exercice et leur montrera certaines particularités entre l'ordinateur et le microcontrôleur. Plus précisément :

- la vitesse du processeur de l'ordinateur (3.00GHz et plus) mesurée en milliards d'instructions par seconde;
- la vitesse du microcontrôleur (20MHz) mesurée en millions d'instructions par seconde;
- la vitesse du port série mesurée en bauds.

L'étudiant doit prendre conscience de ces particularités, sinon il risque d'obtenir une commande ignorée ou une lecture erronée causée par deux commandes successives envoyées presque en même temps au microcontrôleur ou par une lecture trop rapide qui ne permet pas au microcontrôleur d'envoyer toute l'information nécessaire.

Le travail que l'étudiant doit accomplir est de trouver la bonne séquence pour allumer et éteindre les leds et de rédiger ensuite le programme correspondant. La manipulation initiale devrait permettre de concevoir mentalement l'algorithme menant à la solution.

4.3.2. Exercice : Structures de répétition

En introduisant des exigences supplémentaires dans les différents exercices nous imposons l'utilisation des structures de répétition.

4.3.2.1. Exercice à caractère abstrait

Les étudiants sont conviés à améliorer leur programme qui manipule le robot. Premièrement ils doivent prévoir un dialogue, un menu, qui invite l'usager au fur et à mesure à choisir la direction préférée pour le déplacement du robot jusqu'à spécifier TERMINER pour mettre fin aux déplacements. Deuxièmement ils doivent sauvegarder et afficher à la fin tous les déplacements effectués et non effectués.

4.3.2.2. Exercice à caractère concret

Dans cet exercice les étudiants manipulent le montage électronique. Ils doivent créer un programme qui manipule un moteur, c.-à-d. de le faire tourner et de l'arrêter.

Pour cet exercice, les étudiants doivent trouver deux impulsions – l'une qui démarre le moteur et l'autre qui l'arrête. On a créé une fonction PWM (voir annexe C) qu'on a inscrite sur le processeur et qui donne l'impression que la vitesse du moteur change. Quand le moteur démarre, l'hélice débute à tourner et passe au-dessus d'un capteur. Le capteur transmet un signal au microcontrôleur. En analysant ce signal on peut constater l'état du moteur – s'il est démarré ou non. Comme dans l'exercice avec les leds, les

étudiants ont la possibilité, avant de créer leurs propres programmes, de découvrir le coefficient de PWM qui déclenche le moteur et celui-ci qui l'arrête.

4.3.3. L'étape de programmation

Pour l'exercice 2 les étudiants étaient invités à ajouter leurs instructions dans un programme existant. On a choisi cette stratégie pour éviter certaines explications supplémentaires qui pourraient alourdir le travail. Dans ce programme se trouvent toutes les fonctions nécessaires à la manipulation du montage électronique :

- initialiser le port série;
- allumer une led sans condition;
- éteindre une led sans condition;
- démarrer le moteur;
- activer le moteur selon une vitesse;
- obtenir la vitesse du moteur et etc.

D'autres fonctions sont prévues pour faciliter la réalisation d'exercices dans le futur. La Figure 4.2 montre le fichier en question. Ici les étudiants écrivent leurs propres instructions en remplaçant les deux fonctions *DemarrerArreterMoteur()* et *AllumerEteindreLed()*. En exécutant ces deux fonctions les étudiants ont la possibilité de se familiariser avec le montage électronique et de bien comprendre les exercices à réaliser

```
#include "stdafx.h"
#include "serial.h"
#include "utilitaires.h"

#include <iostream>
#include <cstring>
#include <ctime>

using namespace std;


/*-------------------------------------------------------------------*/
/* DESCRIPTION: Fonction principale                                  */
/*              Fonction offrant une interface permettant d'allumer*/
/*              et d'éteindre les leds                               */
/* FONCTIONS :                                                       */
/*              DemarrerArreterMoteur() pour manipuler le moteur     */
/*              AllumerEteindreLed() pour manipuler les leds         */
/* VALEUR DE RETOUR: Aucune                                          */
/* REMARQUE:         Aucune                                          */
/*-------------------------------------------------------------------*/

void main()
{
CSerial Port;

if (Port.Open(1,19200))
{
   DemarrerArreterMoteur(Port);
   AllumerEteindre(Port);
}
    else
       cout << "Probleme d'ouverture du port!";
}
```

Figure 4.2 : Fichier où les étudiants doivent écrire leurs instructions

L'exercice 2 montre assez clairement aux étudiants que l'ordinateur est une machine (un automate) qui exécute des commandes. Ils ont la possibilité de visualiser l'effet de certaines instructions sur le montage.

CHAPITRE 5. Mise à l'essai et les résultats

Nous avons réalisé notre étude dans le cadre du cours INF1005C Programmation procédurale de l'École Polytechnique de Montréal à l'automne 2005 et l'hiver 2006.Ce cours présente les notions de base de la programmation procédurale en C++. Les étudiants qui ont participé à notre étude sont inscrits dans les programmes de Génie logiciel ou Génie Informatique.

Au début de notre rencontre avec les étudiants on leur a expliqué le but de l'expérience et quels avantages ils pouvaient en tirer. Nous avons souligné l'intérêt d'identifier leur style d'apprentissage et de connaître leur niveau d'assimilation des concepts de base de la programmation de manière à améliorer efficacement ses habiletés dans le domaine.

Malheureusement parmi tous les étudiants seulement 18 ont accepté de participer à cette étude à la fin de la session. Ces étudiants ont été priés de remplir les formulaires nécessaires (Annexe D) afin de démontrer notre respect des principes d'éthique de la recherche utilisant des individus.

Dans ce chapitre nous allons présenter comment s'est déroulé l'expérimentation, quels sont les résultats du test et des deux exercices.

5.1. Évaluation du test.

Avant de commencer les exercices concernant la programmation les étudiants étaient invités à répondre aux questions du « test » de Felder. On leur a remis les tests avec les explications concernant l'évaluation. Après avoir complété le test, les étudiants nous ont remis leurs résultats.

Toutes les mesures pour garder l'anonymat étaient prévues. Avant le commencement du test on a remis à chaque étudiant deux étiquettes avec des numéros, choisis aléatoirement. Chaque étudiant devait coller l'une des étiquettes sur la feuille résultat. Il devait conserver l'autre étiquette afin de ne pas oublier son numéro puisque celui-ci constituait le nom du fichier de son programme. Avant de nous les remettre on leur a rappelé de vérifier si les exigences étaient bien suivies. De cette façon, à l'analyse des données nous pouvions faire la correspondance entre les résultats du test et les programmes rédigés.

Avec les résultats (comme à la Figure 3.3) en mains, nous avons commencé notre analyse. D'abord on les a transposés dans un tableau qui correspond à celui des instructions de compilation des résultats (Annexe B). Le nombre inscrit dans chaque cellule du tableau 5.1 correspond au nombre d'étudiants qui obtiennent cette préférence sur l'échelle correspondante.

Tableau 5.1 Réponses des candidats au questionnaire de Felder-Silverman

ACT	-	**1**	-	**1**	**2**	**3**	**2**	**6**	**2**	-	-	**1**	**REF**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
SNS	**1**	**1**	**1**	**4**	**1**	**3**	**2**	**2**	**2**	-	**1**	-	**INT**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
VIS	**1**	**2**	**4**	**4**	**3**	**4**	-	-	-	-	-	-	**VRB**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
SEQ	-	**1**	-	**1**	**2**	**5**	**2**	**4**	**1**	**2**	-	-	**GLO**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	

Il nous faut maintenant expliquer, interpréter ces résultats.

- Que signifie 1a, 3a,..,5b et etc.?

C'est l'intensité qui montre le degré des préférences dans une catégorie.

Si le résultat, que ça soit ***a*** ou ***b,*** se situe entre (voir Annexe B):

  - 1 et 3 on estime que les étudiants sont bien équilibrés *« dans les deux dimensions de cette échelle »*
  - 5 et 7 on dit que les étudiants ont une préférence vers l'un des deux côté de cette échelle et ils apprennent mieux *« dans un environnement d'apprentissage qui est favorable à cette dimension ».*
  - 9 et 11 – les étudiants montrent une préférence assez bien marquée. Dans certains cas ils peuvent éprouver une difficulté *« à apprendre dans un environnement qui n'est pas adapté à cette préférence ».*

- Nous avons précisé au chapitre 3.4 la signification des termes ACT/REF, SNS/INT, VIS/VRB, SEQ/GLO

Considérant l'interprétation de l'intensité décrit précédemment, nous avons regroupé dans le tableau 5.2 les résultats selon trois intensités : 1-3, 5-7 et 9-11.

Tableau 5.2 Regroupement des résultats selon trois intensités

ACT	**1**		**1**		**5**		**8**		**2**		**1**		**REF**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
SNS	**2**		**5**		**4**		**4**		**2**		**1**		**INT**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
VIS	**3**		**8**		**7**		**-**		**-**		**-**		**VRB**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
SEQ	**1**		**1**		**7**		**6**		**3**		**-**		**GLO**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	

Nous remarquons qu'une partie des étudiants qui ont participé à notre expérimentation sont des personnes équilibrées selon l'échelle **ACT/REF** et **SEQ/GLO**. Cela signifie que ces étudiants sont des représentants d'un groupe, ils ont leurs propres préférences mais en cas de nécessité ils peuvent se débrouiller avec des méthodes utilisées par les représentants de l'autre groupe.

Une autre observation qui nous saute aux yeux est que tous les étudiants sont des représentants du groupe **VIS** (visuel) indépendamment qu'ils sont avec des préférences équilibrées ou bien marquées. Cela veut dire qu'ils apprennent et acceptent mieux en regardant des diagrammes, des images, des films, etc.

Le tableau 5.3 présente le nombre d'étudiants présents de chaque côté (a ou b) de l'échelle de préférence. Nous remarquons, dans notre groupe de volontaires, la présence de représentants des deux côtés de l'échelle de Felder Des étudiants qui suivent le processus linéaire (SEQ) pour résoudre un problème et d'autres qui cherchent de nouvelles méthodes (GLO). Des étudiants qui se jettent directement dans le travail (ACT) et d'autres qui aiment d'abord réfléchir et après réagir (REF).

Tableau 5.3 Résultats sommaires des participants au test de Felder-Silverman

ACT	**7**	**11**	**REF**
SNS	**11**	**7**	**INT**
VIS	**18**	**-**	**VRB**
SEQ	**9**	**9**	**GLO**

Le tableau 5.4 présente les résultats de nos étudiants en le regroupant selon les profils obtenus. Les caractères gras soulignent l'intensité de la préférence correspondante.

Tableau 5.4 Classification des résultats obtenus au test de Felder-Silverman

ACTIF						**REFLEXIF**			
4 étud.		**2** étud.		**1** étud.		**6** étud.		**5** étud.	
ACT		**ACT**		**ACT**			**REF**		**REF**
SNS		SNS			INT		INT	SNS	
VIS		VIS		VIS		VIS		VIS	
SEQ			**GLO**		**GLO**		**GLO**	**SEQ**	

Puisque le nombre d'étudiants n'est pas très grand nous avons décidé de travailler et d'observer uniquement deux groupes et deux sous-groupes de chacun. Dans le groupe avec des étudiants du type Actif on a trois sous-groupes. L'un est composé seulement d'un seul étudiant. Nous avons décidé d'ajouter cet étudiant au sous-groupe de deux étudiants de préférences ACT-GLO marquées. Les groupes finaux sont montrés dans le tableau 5.5.

Soulignons encore une fois les modèles psychologiques de nos étudiants :

- Nous avons des étudiants **actifs** et **réflexifs**. Tous sont **visuels**.
- Les étudiants actifs sont soit **séquentiels** soit **globaux**.
- Les étudiants réflexifs sont soit **séquentiels** et sensitifs soit **globaux** et intuitifs.

Tableau 5.5 Classification finale des résultats obtenus au test de Felder-Silverman

ACTIF				**REFLEXIF**			
4 étud.		**3** étud.		**6** étud.		**5** étud.	
ACT		**ACT**			**REF**		**REF**
SNS		SNS			INT	SNS	
VIS		VIS		VIS		VIS	
SEQ			**GLO**		**GLO**	**SEQ**	

5.2. Évaluation des exercices

Après avoir complété le test, les étudiants sont prêts à débuter la rédaction de leurs programmes. Les instructions concernant l'organisation de l'espace de travail, de l'environnement de programmation étaient inscrites dans un fichier. (Annexe E).

Les fichiers nécessaires pour organiser leur espace de travail sont :

MainSerial.CPP	Le programme existant
Utilitaires.h	En-têtes des fonctions utilitaires
Serial.h	Déclaration de la classe CSerial
StdAfx.h	Déclaration utilisée par la classe CSerial
CSerial.LIB	Bibliothèques des fonctions utilitaires
CSerial.DLL	Exécutables des fonctions utilitaires

Les énoncés des exercices étaient disponibles dans un autre fichier (Annexe F). Une période de deux heures était prévue pour la rédaction des deux programmes. Malheureusement le temps prévu ne fut pas suffisant pour terminer les deux exercices. On a décidé d'inviter les étudiants à finir leurs travaux à un moment de leur choix. Chaque étudiant avait à sa disposition le même montage électronique avec lequel il travaillait pendant les deux premières heures. Chaque étudiant a travaillé seul. De cette façon on s'est assuré d'éviter les équipes avec des représentants de deux groupes psychologiques opposés.

Pour le premier exercice les étudiants devaient déplacer un robot selon certaines règles. Ils devaient vérifier les valeurs initiales et après, au fur et à mesure, introduire la direction de déplacement du robot. Ils devaient déterminer quels sont les mouvements permis et ceux interdits. À la fin, les mouvements effectués et non-effectués devaient être affichés d'une façon convenable.

Une fois cet exercice terminé les étudiants s'attaquent au deuxième en utilisant un montage électronique (robotisé). Il est divisé en deux parties. La première consiste à faire démarrer le moteur ainsi que de l'arrêter. Pour réaliser cette tâche ils doivent trouver, d'une certaine façon, l'intensité électrique nécessaire. Initialement ils ont la possibilité d'explorer ces valeurs grâce au programme fourni. La deuxième partie consiste à allumer et éteindre les leds dans un ordre déterminé. Une fois l'ordre trouvé à l'aide du programme fourni ils doivent reproduire le même programme en utilisant les fonctions mises à leur disposition.

À la fin de leur travail les étudiants nous ont remis leurs programmes rédigés. Nous avons analysé les programmes en se basant sur certaines caractéristiques. Les caractéristiques retenues sont décrites ci-après.

- Le nombre de lignes de code de chaque programme.

- Le niveau d'imbrication des structures de décision et des structures de répétition. Si le programme contient une structure de contrôle et une autre dans celle-ci on considère que le niveau d'imbrication est égal à un. Par exemple :

```
if (coord_X>0)                    if (coord_X>0)
{                                 {
   while (ch<>'T')                   if (ch<>'T')
   {                        ou       {
      .....                             .....
   }                                 }
}                                 }
```

En fait, le niveau d'imbrication correspond au nombre de structures de contrôle inscrites dans la première structure.

- La formulation des expressions booléennes. Si les étudiants sont à l'aise à utiliser des opérateurs booléens (AND, OR, NOT) ou des variables de type booléen. Une expression booléenne est dite simple si elle contient seulement des opérateurs de comparaison comme > ,< ,<> ,<= , >=. Une expression booléenne est qualifiée de composée si elle contient un opérateur booléen.
- La préférence de structure de contrôle. Il est intéressant de voir si les étudiants d'un groupe psychologique ont certaines préférences vers certaines structures de contrôle :
    - S'ils utilisent plus la structure de décision IF que IF...ELSE
    - S'ils utilisent les structures de décision d'une façon imbriquées à la place de l'instruction SWITCH
    - S'ils préfèrent plus la structure de répétition WHILE que DO... WHILE
- La simplicité du programme. On apprécie la simplicité d'un programme avec une note jusqu'à 3.
    - Si le programme est assez simple on lui donne 1
    - S'il contient des procédures supplémentaires simples ou un algorithme un peu plus lourd on lui donne 2

- Si on utilise des pointeurs ou des procédures plus complexes on lui attribue 3.

- L'exécution du programme. Le programme s'exécute correctement ou non.

5.2.1. Évaluation de premier exercice

Tous les résultats de l'analyse de premier exercice qu'on a obtenus sont présentés aux tableaux 5.6 et 5.7. Cet exercice qui consiste à manipuler un robot imaginaire se résout en écrivant la logique procédurale transcrite en langage C++. La manipulation de ce « robot » s'exprime naturellement sous la forme des énoncés **if... else** et une structure de répétition. Le défi est de bien agencer ces structures de décision de manière à bien contrôler les différents mouvements du robot.

La signification des symboles utilisés dans les deux tableaux suivants :

*	mauvais contrôle/affichage
**	sans boucle
***	tableau
√	présence
---	absence
OUI*	marche mais pas correctement (beaucoup de remarques)

Tableau 5.6 Analyse des programmes des étudiants de type REF de l'exercice 1

N	POINTS OBSERVÉS	REF + SEQ					REF + GLO					
		5 étudiants					*6 étudiants*					
		1er	2ème	3ème	4ème	5ème	1er	2ème	3ème	4ème	5ème	6ème
1.	**Contrôle** des données à l'entrée	√	√	√*	√	√	√**	---	√	√	√	√**
2.	Utilisation de **AND, OR, NOT** ou d'une variable de type **booléenne**	5	1	1	0	1	2	0	5	0	1	2
3.	Utilisation de boucles du type **WHILE**	1	0	0	0	0	0	3	4	1	2	1
4.	Utilisation de boucles du type **DO ..... WHILE**	3	2	2	2	2	1	0	2	1	0	0
5.	Utilisation de boucles du type **FOR**	3	0	0	2	0	0	1	0	0	0	0
6.	Niveau **d'imbrication**	2	1	2	2	1	1	1	5	2	2	5
7.	**Affichage** des résultats	√	---	√	√	√	√*	√	√	√	√	---
8.	**Simplicité** du programme	2	1	1	3	2	2	2	3	1	1	1
9.	Utilisation de fonctions **créées par les étudiants**	0	0	0	2	1	2	1	0	0	0	0
10.	Utilisation de fonctions **intégrées**	0	1	0	0	0	0	0	0	1	1	0
11.	Type des données (différents de INTEGER, STRING, CHAR)	t***	bool		enum struct	bool		t*** bool	pointeur struct		bool	
12.	Utilisation de **IF .... ELSE**	2	---	2	2	---	---	4	4	2	2	4
13.	Utilisation de la commande **SWITCH**	---	---	---	√	---	---	√	---	---	---	----
14.	Nombre de lignes	114	94	95	144	86	85	123	140	75	87	87
15.	Le programme s'exécute ou non	OUI	NON	OUI*	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI

Tableau 5.7 Analyse des programmes des étudiants de type ACT de l'exercice 1

N	POINTS OBSERVÉS	ACT + SEQ				ACT + GLO		
		4 étudiants				*3 étudiants*		
		1er	2ème	3ème	4ème	1er	2ème	3ème
1.	**Contrôle** des données à l'entrée	---	---	√**	√	√*	---	√*
2.	Utilisation de **AND, OR, NOT** ou d'une variable de type **booléenne**	0	0	0	0	0	0	1
3.	Utilisation de boucles du type **WHILE**	0	1	0	2	0	1	2
4.	Utilisation de boucles du type **DO ..... WHILE**	1	0	0	0	2	0	0
5.	Utilisation de boucles du type **FOR**	0	0	0	2	0	2	0
6.	Niveau **d'imbrication**	2	2	1	2	1	2	6
7.	**Affichage** des résultats	√	---	---	√	---	√	√
8.	**Simplicité** du programme	1	1	1	1	2	1	2
9.	Utilisation de fonctions **créées par les étudiants**	0	0	0	0	1	0	0
10.	Utilisation de fonctions **intégrées**	0	0	0	0	0	0	0
11.	Type des données (différents de INTEGER, STRING, CHAR)	---	enum	---	t***	---	t***	---
12.	Utilisation de **IF .... ELSE**	2	---	1	2	---	2	7
13.	Utilisation de la commande **SWITCH**	---	√	---	√	√	---	---
14.	Nombre de lignes	80	48	78	84	65	90	90
15.	Le programme s'exécute ou non	OUI*	NON	NON	NON	OUI*	OUI*	OUI*

Tous les étudiants ont présenté des programmes qui montrent une bonne compréhension du sujet. Il y a des programmes qui montrent une logique précise et d'autres dans lesquels il y a des blocs mal conçus ou inutiles. Pour nous le plus important était de voir comment les représentants de différents groupes psychologiques ont travaillé.

Les étudiants qui sont plus **réflexifs (REF)** ont des meilleurs résultats que les étudiants actifs (ACT). Tous leurs programmes s'exécutent d'une façon correcte (sauf un). Tous les étudiants, sauf un, de ce groupe font un contrôle des données à l'entrée. Neuf parmi eux font un affichage des résultats. La plupart parmi eux ont utilisé des expressions booléennes complexes. Rappelons que ces personnes aiment bien réfléchir avant de réagir. Ce sont des personnes qui préfèrent travailler seuls. Ils ont probablement apprécié de travailler seuls.

Les étudiants du groupe **ACT (actif)** montrent une préférence vers utilisation de la structure IF plutôt que IF .... ELSE. Ils ont des problèmes avec l'exécution des programmes (ou s'exécute mais pas correctement ou ne s'exécutent pas). Peut-être parce qu'ils sont plus impulsifs et impatients. Dans certains programmes de ces étudiants on observe, soit un manque du contrôle à l'entrée, soit un manque d'affichage à la fin. Mais cependant leurs programmes sont bien construits – le nombre de lignes de code est plus petit que celui des étudiants réflexifs.

Dans le tableau 5.8 sont décrites certaines particularités remarquées dans les programmes des deux groupes et les sous-groupes.

Tableau 5.8 Résultats des étudiants des différents types pour l'exercice 1

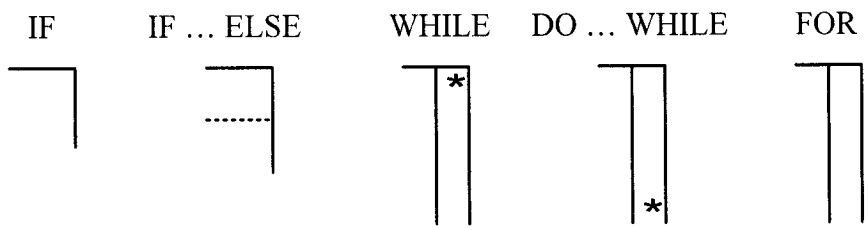
REF (Réflexif)	ACT (Actif)
• À l'aise avec IF ...ELSE • Grand nombre lignes de code • Contrôle des données à l'entrée en	• Évite à utiliser IF ... ELSE. Préfère IF • Plus petit nombre ligne de code • Manque dans certains programmes du

général prévu • Affichage des résultats en général prévus • Les programmes s'exécutent • Il y a présence des données des types structure, bool, pointeur • Fonctions crées par les étudiants (4)		contrôle à l'entrée • Manque dans certains programmes de l'affichage des résultats • Les programmes ne s'exécutent pas ou avec des remarques • Types standards • Fonctions crées par les étudiants (1)	
SEQ	**GLO**	**SEQ**	**GLO**
• Plus grande utilisation de DO … WHILE • Utilise niveau d'imbrication plus petit	• Plus grande utilisation de WHILE • Niveau d'imbrication plus grand	• La plupart des programmes ne s'exécute pas • Un algorithme plus simple • Plus disposé vers SWITCH	• Tous les programmes s'exécutent (avec remarques) • Un algorithme plus compliqué • Présence de fonction crée par étudiant.

Les graphiques des Figure 5.1 et Figure 5.2 illustrent les programmes rédigés par chaque étudiant. Ces graphiques nous révèlent le niveau d'imbrication que chaque étudiant a utilisé. En réalité si une graphique s'étend en largeur la possibilité des erreurs logiques augmente car la complexité de la logique accroit.

Ces structures nous montrent l'habileté des étudiants à transposer la solution dans le langage C++.

Schémas utilisée pour représenter les structures de contrôle



Sur ces graphiques on voit la variété des idées que les étudiants ont utilisées. Chacun a travaillé à sa façon.

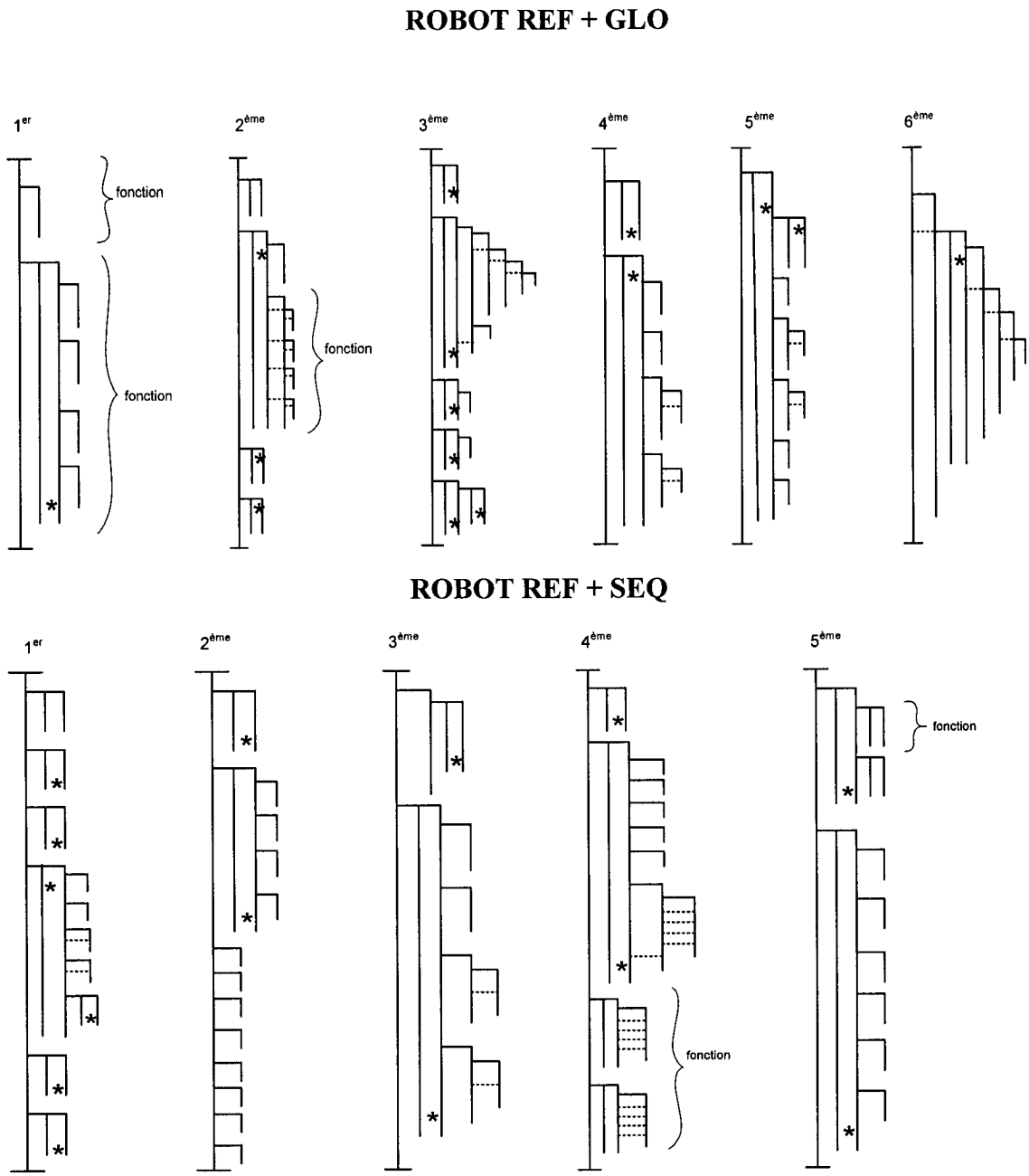
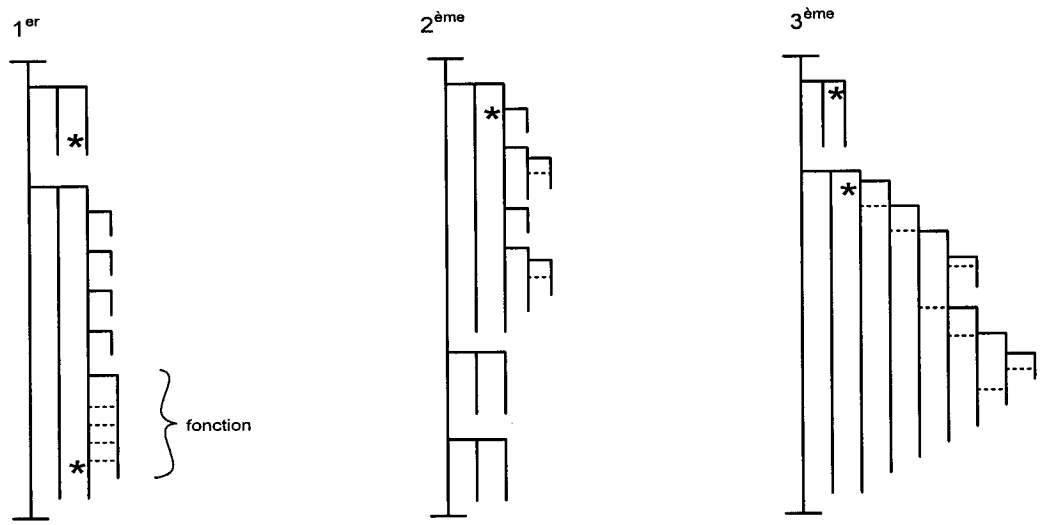


Figure 5.1 : Diagramme du flux de contrôle (groupe REF) – exercice 1.

ROBOT ACT + GLO



ROBOT ACT + SEQ

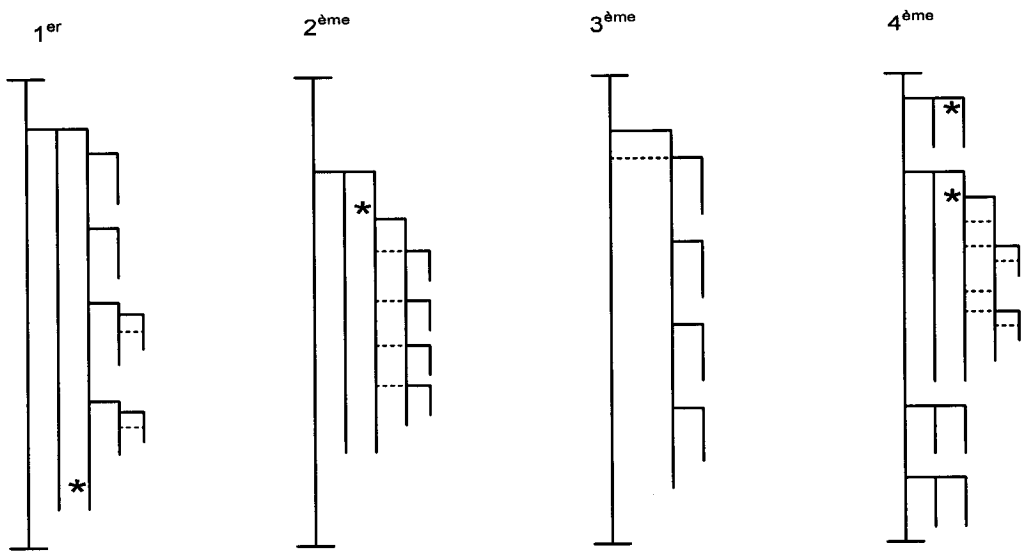


Figure 5.2 : Diagramme du flux de contrôle (groupe ACT) – exercice 1.

Pendant l'analyse et l'exécution des programmes nous avons observé certaines erreurs :

ACT+SEQ

1. *Déclaration de variables inutilisées dans le programme.*
2. Mélange des types des données (on déclare un type énumération et on veut lire une donnée de ce type du clavier ou on lit une valeur entière mais on la compare incorrectement avec la valeur numérique qui lui correspond dans l'énumération).
3. *Mauvais contrôle de la condition de sortie de la boucle – boucle infinie (on ne change pas la valeur de la variable impliquée dans cette condition)*
4. Mauvaise formulation d'une condition
   - mauvaise comparaison (on essaie à comparer des valeurs de différents types)
   - mauvais signe pour la comparaison (=) à la place de (= =)

ACT+GLO

1. Un appel de trop à une fonction à la sortie du programme.
2. Déclaration d'un type qu'on n'utilise pas.
3. Mélange des index de tableaux différents.

REF+SEQ

1. *Boucle infinie*
2. Déclaration de paramètres (dans des fonctions) dont on n'a pas besoin

REF+GLO

1. *Déclaration de variables inutiles.*
2. Déclaration des paramètres (dans des fonctions) lesquelles on n'utilise pas

On voit qu'il y a des erreurs qui sont communes, elles sont soulignées, et d'autres qui sont propres à un type donné.

5.2.2. Évaluation de deuxième exercice

Comme dans le premier exercice tous les résultats sont présentés dans les tableaux 5.9 et 5.10. Cet exercice est composé de deux parties. La première partie consiste à faire tourner un moteur sur un montage électronique. On doit trouver la valeur qui démarre le moteur et celle qui l'arrête. Ces deux tâches se réalisent en C++ en utilisant des énoncés If ...Else et des structures de répétition. En même temps ils doivent utiliser des fonctions qui leur sont déjà fournies. La deuxième partie consiste à allumer les leds dans un ordre précis. Ils doivent organiser tout ça dans un dialogue convenable. Pour réaliser cette partie, ils doivent de nouveau se servir des énoncés If ... Else, des structures de répétition. Ici, ils ont même l'opportunité à la place des certaines commandes If ... Else d'utiliser la structure SWITCH.

Une fois les programmes remis nous avons effectué leur analyse. Malheureusement, quatre étudiants n'ont pas fait cet exercice. Puisque l'expérimentation était anonyme nous n'avons pas eu la possibilité de les contacter pour connaître la raison de cette absence. De plus, la deuxième partie de l'exercice manquait, celle avec les leds, pour deux des programmes obtenus. Le plus grand manque des programmes est attribué à trois représentants du type réflexif (REF) et un de type actif (ACT).

En ce qui concerne l'exécution d'un programme on l'a divisée en deux parties – la partie concernant le moteur et celle concernant les leds. On observe ainsi distinctement l'exécution de chaque partie. Cette information apparaît à la ligne 14 des tableaux 5.9 et 5.10. Par exemple, (O/N) signifie que la partie du programme qui fait tourner le moteur s'exécute correctement (la première lettre de cette signification, O pour oui) et que la partie du programme qui allume les leds ne s'exécute pas dans cet exemple (la deuxième lettre de signification utilisée, N pour non).
De nouveau les étudiants REF sont disposés à concevoir des programmes avec un nombre de code de lignes plus grand. L'affichage chez les étudiants REF est mieux organisé que chez les ACT. Chez les ACT il y des manques d'affichage des résultats. Étonnamment, il n'y a peu de différence entre les deux groupes.

Les résultats sont résumés dans le tableau 5.11.

Tableau 5.11 Résultats des étudiants des différents types pour l'exercice 2

<table>
<tr><th colspan="2">REF (Réflexif)</th><th colspan="2">ACT (Actif)</th></tr>
<tr><td colspan="2">• 3 programmes manquants
• Plus grand nombre de lignes du code
• Utilisation de IF … ELSE
• Simplicité moyenne des programmes
• Niveau d'imbrication plus grand
• Affichage des résultats</td><td colspan="2">• 1 programme manquant
• Plus petit nombre du code de lignes
• Utilisation de IF … ELSE
• Programmes simples
• Niveau d'imbrication plus petit
• Manque d'affichage dans certains programmes</td></tr>
<tr><th>SEQ</th><th>GLO</th><th>SEQ</th><th>GLO</th></tr>
<tr><td>• Programme manquant
• Utilise plus Do … While</td><td>• Deux programmes manquants
• Utilise plus While
• Utilisation de SWITCH</td><td>• N'éteignent pas les leds
• La plupart des programmes ne s'exécutent pas</td><td>• Éteignent les leds
• Les programmes s'exécutent
• Utilisation des fonctions créées par les étudiants</td></tr>
</table>

MOTEUR-LED

Tableau 5.9 Analyse des programmes des étudiants de type REF de l'exercice 2

N	POINTS OBSERVÉS	REF + SEQ					REF + GLO					
		5 étudiants					*6 étudiants*					
		1er	2ème	3ème	4ème	5ème	1er	2ème	3ème	4ème	5ème	6ème
1.	**Éteinte** des leds	√	MANQUANT	---	---	√	MANQUANT	√	---	MANQUANT	√	√
2.	**Arrêt** du moteur	√		√	---	√		√	√		√	√
3.	Utilisation de boucles du type **WHILE**	2		0	0	0		2	0		2	3
4.	Utilisation de boucles du type **DO ..... WHILE**	1		2	2	5		0	1		0	0
5.	Utilisation de boucles du type **FOR**	2		0	0	0		2	0		0	0
6.	Niveau **d'imbrication**	2		1	5	6		4	5		6	4
7.	Utilisation de fonctions supplémentaires, **créées par les étudiants**	0		0	0	2		0	1		2	0
8.	**Affichage** des résultats	√		√	√	√		√	---		√	√
9.	**Simplicité** du programme	1		1	1	2		1	2		2	1
10.	**Type** des données (différents de INTEGER, STRING, CHAR)	---		---	---	---		t***	---		---	---
11.	Utilisation de **IF .... ELSE**	0		0	4	5		3	5		6	4
12.	Utilisation de la commande **SWITCH**	---		---	---	---		---	---		√	---
13.	Nombre de lignes	127		109	143	194		119	155		260	174
14.	Le programme s'exécute ou non	N/N		N/N	N/O	O/O		N/O	N/---		O/O	O/O

* mauvais contrôle/affichage √ présence

** sans boucle --- absence

*** tableau O/N (O-oui, marche correctement pour le moteur, /N-non, ne marche pas pour les leds)

MOTEUR-LED

Tableau 5.10 Analyse des programmes des étudiants de type ACT de l'exercice 2

N	POINTS OBSERVÉS	ACT + SEQ				ACT + GLO		
		4 étudiants				*3 étudiants*		
		1er	2ème	3ème	4ème	1er	2ème	3ème
1.	**Éteinte** des leds	√	---	---	---	MANQUANT	√	√
2.	**Arrêt** du moteur	√	---	√	---		√	---
3.	Utilisation de boucles du type **WHILE**	0	1	2	0		1	2
4.	Utilisation de boucles du type **DO ….. WHILE**	4	0	0	1		2	0
5.	Utilisation de boucles du type **FOR**	1	1	0	0		0	0
6.	Niveau **d'imbrication**	3	2	1	4		3	3
7.	Utilisation de fonctions supplémentaires, **créées par les étudiants**	0	0	0	0		0	3
8.	**Affichage** des résultats	---	---	√	---		√	√
9.	**Simplicité** du programme	1	1	1	1		1	2
10.	**Type** des données (différents de INTEGER, STRING, CHAR)	t***	t***	---	---		t***	---
11.	Utilisation de **IF …. ELSE**	1	3	---	---		1	---
12.	Utilisation de la commande **SWITCH**	√	---	---	---		---	√
13.	Nombre de lignes	88	94	71	73		115	182
14.	Le programme s'exécute ou non	N/N	N/N	O/---	N/N		O/O	O/O

* mauvais contrôle/affichage √ présence

** sans boucle --- absence

*** tableau O/N (O-oui, marche correctement pour le moteur, /N-non, ne marche pas pour les leds)

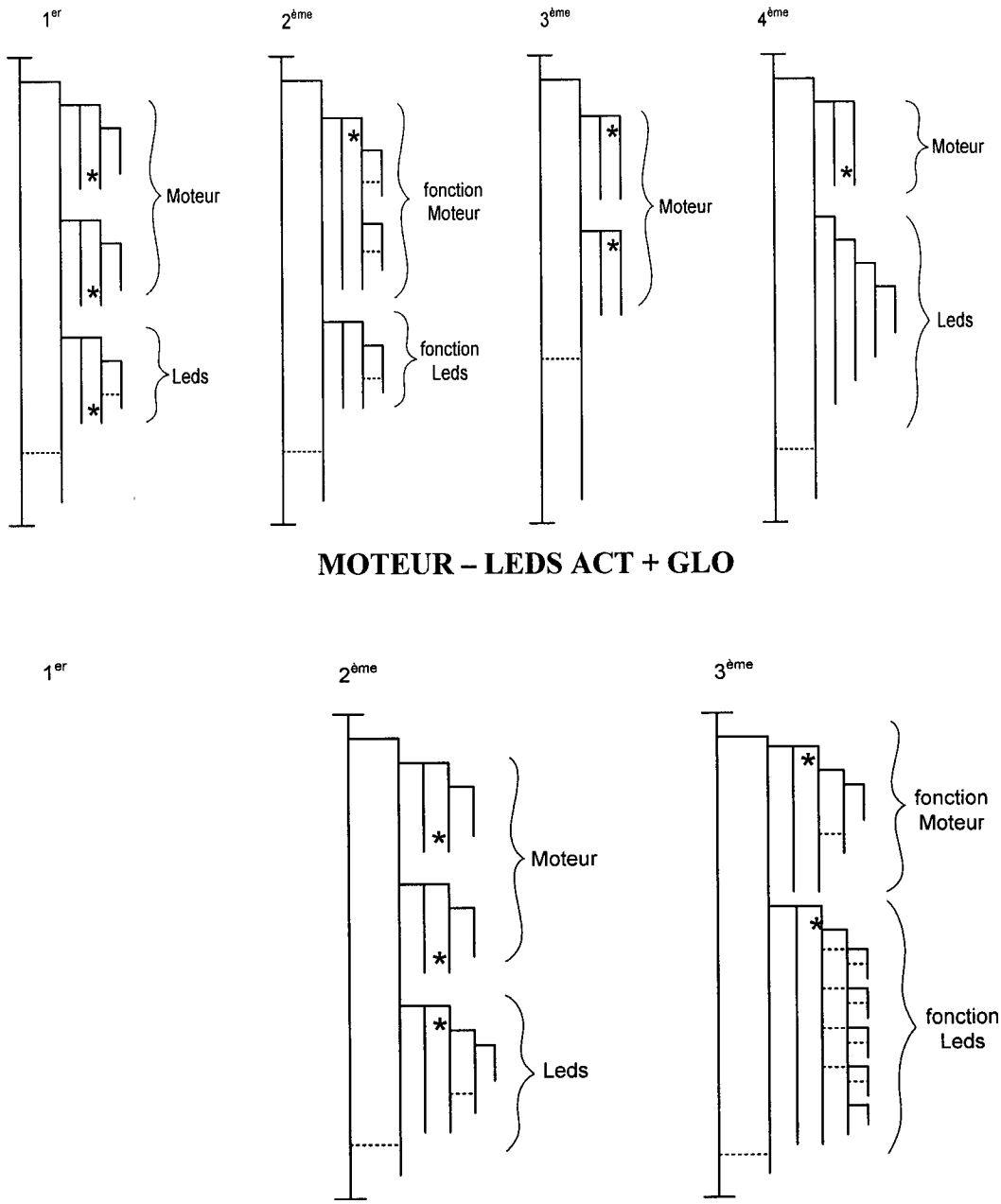


Figure 5.3 : Diagramme du flux de contrôle (groupe ACT) – exercice 2.

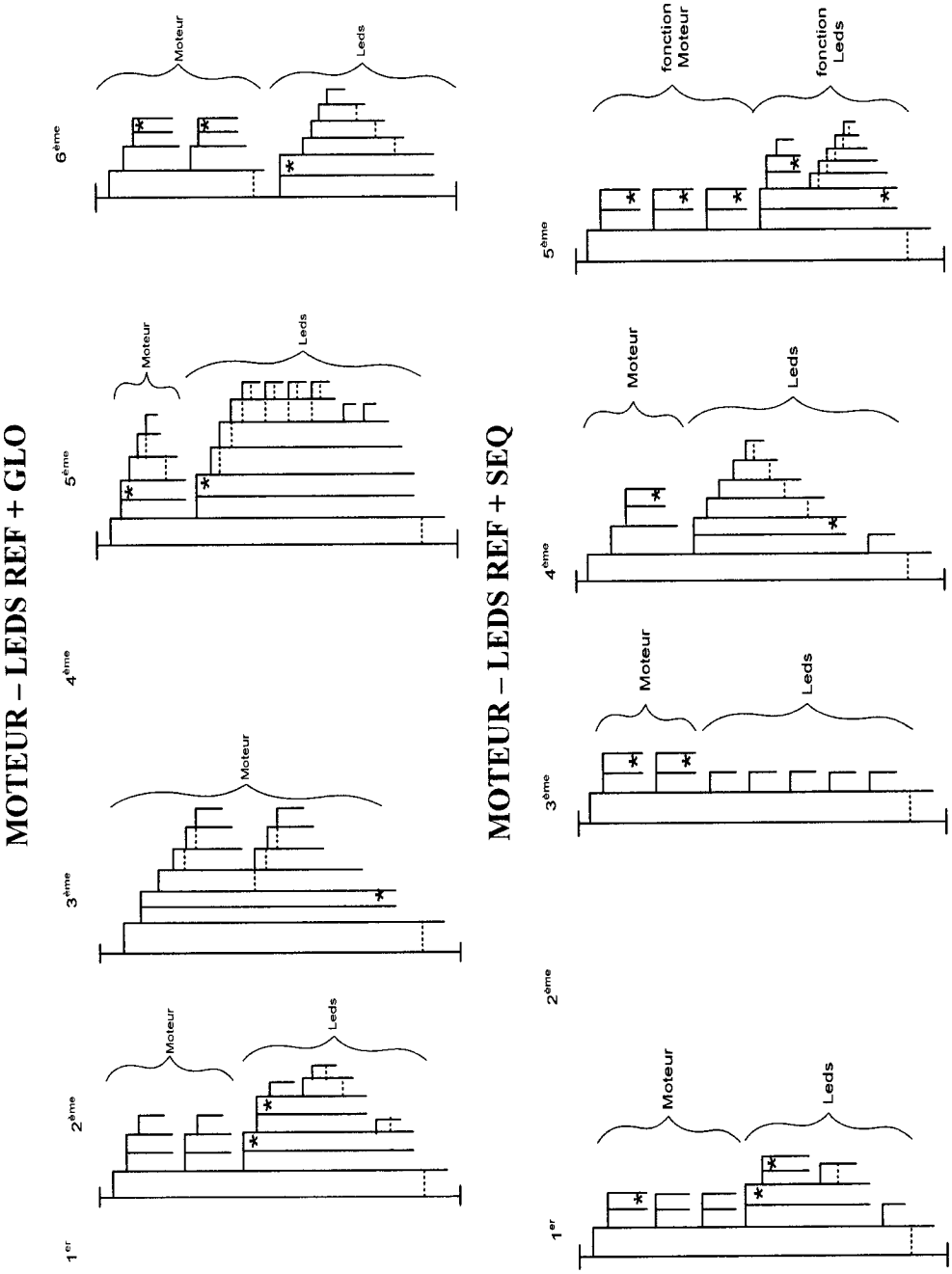


Figure 5.4 : Diagramme du flux de contrôle (groupe REF) – exercice

Remarques sur les programmes MOTEUR ET LEDS

Globalement, nous observons que dans cet exercice les étudiants des deux groupes principaux semblent rencontrer les mêmes difficultés et ont commis des fautes similaires.

REF+SEQ

1. *mauvaise condition pour sortir de la boucle (boucle infinie)*
2. *manque de Sleep*
3. *mauvaise construction de la boucle for et while*
4. *erreurs syntaxiques*
5. *mauvais choix des valeurs initiales*
6. mauvaise combinaison des leds

REF + GLO

1. *manque ou mauvaise place de Sleep*
2. utilisation de break
3. *mauvaise condition pour sortir de la boucle*
4. *mauvaise valeur initiale pour la boucle for*

ACT + GLO

1. *mauvaise place de Sleep*
2. warning- variable non utilisée

ACT + SEQ

1. *mauvaise place de Sleep*
2. combinaison pour allumer les leds mal trouvée
3. *initialisation et réinitialisation à la mauvaise place*
4. *erreurs syntaxiques (les accolades mal fermées, end à la place de* endl)
5. manque d'appel des fonctions.

5.3. Évaluation de l'expérience.

Après les analyses c'est le moment d'évaluer l'expérience.
Une fois l'expérience terminée on a demandé aux étudiants si l'expérience leur a plu. La réponse (malheureusement pas documentée) était OUI. La partie qui leur a plu le plus est celle avec le montage électronique.

Pour le premier exercice, exercice à caractère abstrait, on peut dire que les représentants du groupe **REF** ont mieux travaillé et que leurs résultats sont meilleurs que ceux du groupe **ACT**. Nous avons inscrit dans le tableau 5.12, le pourcentage des programmes qui s'exécutent correctement selon le type de préférence. Les réflexifs remportent de toute évidence la palme. Le pourcentage des programmes qui s'exécutent correctement du groupe **REF** est 91% tandis que celui chez les **ACT** est de 57%.

ROBOT

Tableau 5.12 Réussite de l'exercice 1

REF (11 étud.)	***SEQ*** *(5 étud.)*	OUI	80%	***OUI***	**91%**
		NON	20%		
	GLO *(6 étud.)*	OUI	100%	***NON***	**9%**
		NON	0%		
ACT (7 étud.)	***SEQ*** *(4 étud.)*	OUI	25%	***OUI***	**57%**
		NON	75%		
	GLO *(3 étud.)*	OUI	100%	***NON***	**43%**
		NON	0%		

En ce qui concerne le deuxième exercice cette différence diminue beaucoup. Ces résultats sont inscrits dans le tableau 5.13. Nous voyons que les deux groupes ont des résultats similaires et comparables. Dans cet exercice à caractère concret, les différences entre les deux groupes principaux (ACT et REF) ne sont pas tellement évidentes.

MOTEUR et LEDS

Tableau 5.13 Réussite de l'exercice 2

REF (8 etud.)	***SEQ*** *(4 étud.)*	OUI	37.5%	***OUI***	**50%**
		NON	62.5%		
	GLO *(4 étud.)*	OUI	62.5%	***NON***	**50%**
		NON	37.5%		
ACT (6 etud.)	***SEQ*** *(4 étud.)*	OUI	12.5%	***OUI***	**42%**
		NON	87.5%		
	GLO *(2 étud.)*	OUI	100%	***NON***	**58%**
		NON	0%		

Conclusion

« À la lecture des ACTES du 19e Colloque AIPU 2002 dont le thème central était: Les méthodes actives dans l'enseignement supérieur, un consensus ressort à travers plusieurs articles : l'apprentissage s'accroît lorsque l'étudiant est sollicité et actif. »[17]

Le temps où les professeurs enseignaient avec une voix monotone, avec des étudiants qui écoutaient sans participer est révolu. De nos jours le professeur doit jouer le rôle d'organisateur incitant ainsi ses étudiants à participer activement au processus d'apprentissage. Pour qu'il réussisse il est préférable de bien connaître ses étudiants et leurs modèles psychologiques d'apprentissage. De cette façon l'enseignant pourrait ajuster sa méthode d'enseignement afin de tirer profit au maximum des particularités de ces groupes.

Pour nous, il est nécessaire de savoir à quels groupes psychologiques appartiennent nos étudiants. On considère que cette information est extrêmement utile pour les étudiants eux-mêmes. Sachant quel style d'apprentissage il préfère, l'étudiant peut miser sur ses forces ou tenter de développer d'autres qualités qu'il a sous-estimées jusqu'à présent.

Nous désirons développer des situations d'apprentissage qui vont favoriser l'apprentissage de la programmation.

« Motivate learning... Provide a balance of concrete information and abstract concepts... Balance material that emphasizes practical problem-solving methods with material that emphasizes fundamental understanding... » (Felder-Silverman)

Nous avons observé comment les étudiants utilisant différents styles d'apprentissage se sont débrouillés et quelles difficultés ils ont rencontrées. Nous avons distingué des différences entres les différents groupes psychologiques.

Il est important de souligner qu'étant donné le peu de candidats qui ont participé à l'expérience nous avons opté pour des statistiques plutôt descriptives. Nous sommes également conscients des dangers ou des réserves qu'il faut avoir sur les données obtenues. Par exemple, selon un intervalle de confiance de 95%, les candidats du groupe REF ont réussi le premier exercice à 86.36363636 ±21.74007801 et les candidats du groupe ACT ont réussi le même exercice à 28.57142857±24.74873734. Toujours selon un intervalle de confiance de 95%, les candidats du groupe REF ont réussi le deuxième exercice à 42.85714286±29.6465839 et ceux du groupe ACT ont réussi à 57.14285714±49.49747468. Les détails de ces calculs sont présentés à l'annexe G.

Le premier exercice qu'on a proposé aux étudiants était avec un sujet abstrait. Pour créer le programme final les étudiants avaient besoin de beaucoup plus du temps qu'on a prévu. La raison peut-être était qu'ils ont consacré plus du temps pour bien réfléchir et bien comprendre l'énoncé – ce qui est permis et ce qui non, de se bien imaginer comment « le robot » va être manipulé. Ils devaient faire la liaison avec un domaine d'application qu'ils ne connaissaient pas.

On sait qu'une des difficultés qu'un étudiant éprouve est celle d'intégrer à la fois ses notions de programmation et celles concernant le domaine de l'application. Pour cette raison nous avons proposé un exercice qui ne nécessite même pas de connaissances du domaine de l'application. Au début de l'expérimentation les étudiants ont observé comment le montage électronique fonctionne et après de concevoir leurs propres programmes, qui devaient réaliser le même comportement. Ils devaient décrire et prédire le comportement de leur objet robotisé.

On a remarqué que dans cet exercice les différences entre les groupes psychologiques ne sont pas tellement évidentes. Les fautes commises sont presque du même type. Le pourcentage des programmes qui s'exécutent chez les deux groupes est pareil. Comment on peut expliquer ça?

Nous pensons que le fait d'avoir la possibilité d'observer au début comment le programme fonctionne a aidé beaucoup les étudiants. Il n'était pas nécessaire de perdre du temps pour bien comprendre le contenu. Même s'ils n'avaient aucunes connaissances dans le domaine concret après la démonstration ils pouvaient se mettre tout de suite au travail. Leur attention était concentrée surtout sur la création du programme.

En observant comment l'environnement robotisé marche, l'étudiant visualise et comprend mieux ce qu'il doit faire. Il a la possibilité de se construire déjà un algorithme abstrait (en se basant sur ses observations) comportant des notions propres de la programmation.

Nous pensons qu'en observant l'ouverture des leds dans un ordre spécifique ça va lui donner l'idée d'utiliser des structures conditionnelles. Le tournoiement du moteur va lui suggérer l'idée d'utiliser des structures répétitives.

Dans ce milieu l'étudiant a eu la possibilité de visualiser concrètement un algorithme. Nous espérons que cela l'aidera à produire un algorithme abstrait qui incorporera les structures de base en programmation. On a supposé qu'en allumant et éteignant plusieurs leds l'étudiant aura l'idée d'utiliser une structure conditionnelle. Ou bien le démarrage et l'arrêt d'un moteur va lui suggérer l'idée d'utiliser une structure de répétition et de mieux les comprendre.

En offrant aux étudiants la possibilité de travailler avec un objet concret on leur a montré que dans la pratique tout est lié. Même si on étudie quelque chose de spécifique (la programmation, les math et etc.) on doit toujours chercher la liaison avec les autres

matières, de quelle façon on peut utiliser ces connaissances pour résoudre des problèmes dans les autres domaines.

Dans une partie de cette expérimentation les étudiants devaient manipuler des leds et un moteur en C++. Mais en apprenant les bases de la programmation les étudiants ont déjà la possibilité de se poser la question : De quelle façon je peux moi-même réaliser une telle tâche? De quelles connaissances encore ai-je besoin?
En prenant par exemple le cours INF1995 les étudiants ont la possibilité de continuer à enrichir leurs connaissances dans la programmation en programmant un robot réel et de réaliser que « la pure » programmation n'existe pas. Elle est toujours liée avec un domaine d'application.

On peut dire que cet atelier était utile pour les étudiants qui ont participé dans cette expérimentation. L'utilisation d'un exercice avec un sujet concret les a aidés de voir la vraie liaison entre un programme et une tâche réelle.

En réponse à notre question « Quel style d'apprentissage est le mieux servi par un exercice à caractère abstrait et quel style d'apprentissage est le mieux servi par un exercice à caractère concret?» nous pouvons dire que selon les résultats obtenus, les étudiants du groupe REF ont mieux travaillé avec l'exercice à caractère abstrait que les étudiants du groupe ACT. Cependant, les deux groupes ont montré des résultats similaires lorsqu'il s'agit de l'exercice à caractère concret. On ne peut donc pas prétendre que les étudiants du groupe ACT se sont mieux débrouillés que leurs collègues du groupe REF. Notre hypothèse de départ est donc vérifiée qu'à moitié.

.

On pense que si on a toujours la possibilité de continuer à travailler dans cette direction et au futur les seuls gagnants seront les étudiants.

Bibliographie

[1]. JENKINS T. : *A participative Approach to Teaching Programming.* ITiCSE '98 Dublin, Ireland

[2]. HAGAN D., SHEARD J. : *The Value of Discussion Classes for Teaching Introductory Programming.* ITiCSE '98 Dublin, Ireland

[3]. HAGAN D., MARKHAM S. : *Does It Help to Have Some Programming Experience Before Beginning a Computing Degree Program?.* ITiCSE '2000 Helsinki, Finland

[4]. SATRATZEMI M., DAGDILELIS V., EVAGELIDIS G.: *A System For Program Visualization and Problem-Solving Path Assessment of Novice Programmers* ITiCSE '2001 Canterbery, UK

[5]. FELDER R. M. : *Matters of Styles* ASEE Prism, 6(4), 18-23 (December 1996).

[6]. CARDELLINI L. : *An Interview with Richard M. Felder.* Journal of Science Education, 3(2). 62-65 (2002)

[7]. FELDER R. M. : *Learning and teaching Styles.* Engr. Education, 78(7), 674-681 (1998)

[8]. http://chat.carleton.ca/~tblouin/Felder/felder.html

[9]. http://www.ncsu.edu/felder-public/

[10]. *Felder-Silverman Instrument.* BAGLEY College of Engineering, November 2, 2004

[11]. http://www.mbtitoday.org/model.html

[12]. http://www.personalitypage.com/portraits.html

[13]. http://www.personalitypathways.com/MBTI_intro.html

[14]. http://www.typelogic.com/

[15]. VIAU R., PRÉGENT R., FOREST L. : *Les façons d'apprendre des étudiantes et des étudiants de baccalauréat à l'École Polytechnique de Montréal.* Résultats de l'enquête menée à l'automne 2003, Juin 2004

[16]. BOUDREAULT Y., GUERFALI W., *C et un peu + : résolution de problèmes et programmation en C++,* École polytechnique de Montréal, 2001

[17]. BOUDREAULT Y., *Thèse « Conception et développement d'un environnement favorisant l'apprentissage des concepts fondamentaux de la programmation »*, Novembre 2003.

[18]. *C Comliler Reference Manual*, Octobre 2005

[19]. http://encarta.msn.com/encnet/refpages/RefArticle.aspx?refid=761578837

[20]. DITIBERIO J. K., HAMMER A.L. : *Introduction to type in college*

[21]. LAHTINEN E., ALA-MUTKA K., JARVINEN H. : *A study of the Difficulties of Novice Programmers* ITiCSE '2005,JUNE 27-29, Monte de Caparica, Portugal.

[22]. HOWARD R., CURTIS A., WILLIAM D. LANE : *Felder's Learning styles, Bloom's Taxonomy, and the Kolb Learning Cycle: Tying all together in the CS2 Course,* SIGCSE '96 2/96 Philadelphia, PA USA

[23]. http://www.hbdi.com

[24]. http://www.hopellc.com/hbdi.pdf ou
http://www.accsys-corp.com/HBDI_FAQ/hbdi.pdf

[25]. http://16types.free.fr/modele-MBTI-0-introduction.html

Sites expliquant la méthode de Cardan :

http://fr.wikipedia.org/wiki/M%C3%A9thode_de_Cardan

http://villemin.gerard.free.fr/ThNbDemo/Eqa3dres.htm

http://www.vttoth.com/cardano.htm

Site expliquant les calculs statistiques :

http://helios.bto.ed.ac.uk/bto/statistics/tress3.html

Annexe A : Le questionnaire de Felder

Questionnaire d'identification des styles d'apprentissage

(d'après Richard Felder et Barbara Soloman, Department of Chemical Engineering, North Carolina State University)

Consigne : Veuillez compléter chacun des énoncés ci-dessous en encerclant un seul choix de réponse. Dans l'éventualité où «a» et «b» s'appliquent, choisissez la proposition qui s'applique le plus fréquemment.

1. Je comprends mieux quelque chose après :

   a) l'avoir essayé.
   b) y avoir réfléchi.

2. Je suis considéré comme plutôt :

   a) réaliste.
   b) innovateur.

3. Lorsque je pense à ce que j'ai fait hier, je m'en souviens sous la forme :

   a) d'images.
   b) de mots.

4. J'ai tendance à :

   a) comprendre les détails d'un sujet sans en saisir la structure globale.
   b) comprendre la structure globale d'un sujet sans en comprendre les détails.

5. Quand j'aborde un nouveau sujet, je fais un meilleur apprentissage si :

   a) j'en parle à quelqu'un.
   b) j'y réfléchis seul.

6. Si j'étais professeur, je préférerais donner un cours :

   a) qui traite de faits réels et de situations de la vie courante.
   b) qui traite d'idées et de théories.

7. Je préfère obtenir de l'information sous la forme:

   a) d'images, de figures, diagrammes, graphiques ou schémas.
   b) de directives écrites ou d'explications verbales.

8. Une fois que je comprends :

   a) chacune des parties d'un sujet, j'arrive à saisir l'ensemble.
   b) l'ensemble d'un sujet, j'arrive à saisir comment les parties sont interreliées.

9. Dans un groupe d'étude travaillant sur un sujet complexe, j'ai tendance à :

   a) m'impliquer et exprimer mes idées.
   b) rester muet et écouter.

10. Je trouve plus facile :

   a) d'apprendre des faits.
   b) d'apprendre des concepts.

11. Dans un livre contenant beaucoup d'images et de tableaux, j'ai plutôt tendance à :

   a) examiner particulièrement les images et tableaux.
   b) concentrer mon attention sur le texte écrit.

12. Lorsque je résous des problèmes de mathématiques :

   a) j'utilise une démarche systématique, en franchissant une étape à la fois jusqu'à ce que je parvienne à la réponse.
   b) j'entrevois souvent la solution, mais je dois faire des efforts pour trouver les étapes qui mènent à celle-ci.

13. Dans les cours que j'ai suivis :

   a) j'ai cherché à connaître plusieurs des étudiants.
   b) j'ai rarement cherché à connaître les étudiants.

14. Lorsque je lis de la documentation, je préfère :

   a) un texte qui m'apprend de nouveaux faits ou qui m'apprend comment faire quelque chose.
   b) un texte qui me donne de nouvelles idées suscitant la réflexion.

15. J'aime les professeurs qui :

   a) montrent beaucoup de schémas et d'images au tableau ou à l'écran.
   b) passent beaucoup de temps à donner des explications verbales.

16. Lorsque j'analyse un récit ou un roman :

a) je me rappelle les différents épisodes et péripéties et, à partir de là, je tente de dégager les principaux thèmes et lignes directrices de l'œuvre.
b) au terme de ma lecture, j'ai bien saisi les principaux thèmes et lignes directrices, mais je dois revenir en arrière pour me rappeler les épisodes et péripéties qui les illustrent.

17. Lorsque je commence à résoudre un problème qui m'est soumis dans le cadre d'un devoir, j'ai tendance à :

a) démarrer en travaillant immédiatement sur la solution.
b) tenter d'abord de comprendre parfaitement le problème.

18. Je préfère l'idée de :

a) la certitude.
b) la théorie.

19. Je me souviens plus facilement de :

a) ce que je vois.
b) ce que j'entends.

20. Pour moi, il est plus important qu'un professeur :

a) présente la matière dans un ordre séquentiel clair.
b) donne d'abord une vue globale de la matière et fasse ensuite des liens avec d'autres sujets.

21. Je préfère étudier :

a) en groupe.
b) seul.

22. On considère que je suis avant tout :

a) soucieux des détails lorsque j'effectue un travail.
b) créatif en ce qui concerne ma façon d'effectuer le travail.

23. Lorsqu'on me donne des indications pour me rendre à un nouvel endroit, je préfère :

a) un plan.
b) des instructions écrites.

24. J'apprends :

a) en étudiant plutôt régulièrement. Si j'étudie beaucoup, je sais que je vais finir par comprendre.
b) en étudiant de façon intermittente, par bourrées de travail ; par moment, je peux être complètement perdu, mais soudain, tout tombe en place et je saisis l'ensemble de la matière.

25. Généralement, je préfère d'abord :

a) essayer directement des choses.
b) réfléchir à la façon dont je vais m'y prendre pour faire les choses.

26. Lorsque je lis pour le plaisir, je préfère des auteurs qui :

a) énoncent clairement ce qu'ils veulent dire.
b) énoncent les choses de façon créative et intéressante.

27. Lorsque je vois un graphique ou un dessin en classe, j'ai tendance à me rappeler surtout :

a) l'image.
b) ce que le professeur a dit à propos de l'image.

28. Lorsque je fais face à une grande quantité d'information, habituellement :

a) je me concentre plutôt sur les détails et laisse de côté de la vue d'ensemble.
b) j'essaie de me donner une vue d'ensemble avant d'entrer dans les détails.

29. Je me rappelle plus facilement :

a) une chose que j'ai faite.
b) une chose à laquelle j'ai beaucoup réfléchi.

30. Lorsque j'ai une tâche à réaliser, je préfère :

a) maîtriser une seule façon de l'accomplir.
b) découvrir de nouvelles façons de l'accomplir.

31. Lorsque quelqu'un me montre des données, je préfère voir :

a) des graphiques et des tableaux.
b) un texte résumant les résultats.

32. Lorsque je rédige un texte, je préfère :

a) travailler (y penser ou l'écrire) sur le début du texte et enchaîner avec les parties qui suivent.
b) travailler (y penser ou l'écrire) sur différentes parties du texte et les ordonner par la suite.

33. Lorsque j'ai un travail à réaliser en équipe, je veux tout d'abord :

a) faire un «remue-méninges» en groupe où chacun énonce ses idées.
b) réfléchir individuellement et par la suite revenir en équipe et comparer les idées.

34. Je considère qu'il est préférable d'être quelqu'un :

a) de raisonnable.
b) d'imaginatif.

35. Lorsque je rencontre une personne dans une soirée, je me souviens plus facilement de :

a) son aspect physique.
b) ce qu'elle a dit à propos d'elle-même.

36. Lorsque j'apprends un nouveau sujet, je préfère :

a) rester concentré sur le sujet et l'approfondir le plus possible.
b) tenter de faire des liens entre ce sujet et des sujets connexes.

37. On me considère comme :

a) plutôt extraverti.
b) plutôt réservé.

38. Je préfère les cours qui mettent l'accent sur :

a) des contenus concrets (faits, données).
b) des contenus abstraits (concepts, théories).

39. Pour me divertir, je préfère :

a) regarder la télévision.
b) lire un livre.

40. Certains professeurs commencent leurs cours en présentant le plan de la matière à couvrir. Ce plan a pour moi :

a) une certaine utilité.
b) une grande utilité.

41. Faire un travail d'équipe avec une seule note pour le groupe :

a) me plaît.
b) me déplaît.

42. Lorsque j'effectue de longs calculs :

a) j'ai tendance à refaire chacune des étapes et à vérifier mon travail en détail.
b) je trouve pénible de devoir vérifier ma démarche et mes calculs, et je dois me forcer pour le faire.

43. J'ai tendance à me remémorer les endroits où je suis allé :

a) facilement et avec exactitude.
b) avec difficulté et avec peu de détails.

44. Lorsque je résous un problème en équipe, j'ai tendance à :

a) penser aux étapes de la résolution.
b) penser aux applications ou conséquences possibles de la solution dans des domaines variés.

Pour compiler et interpréter vos résultats, reportez-vous au document *Compilation des résultats du questionnaire sur les styles d'apprentissage*.

Annexe B : Compilation des résultats du questionnaire

Compilation des résultats du questionnaire sur les styles d'apprentissage

Pour compiler vos réponses au questionnaire sur les styles d'apprentissage, remplissez les d tableaux ci-dessous de la façon décrite ici.

1. Dans les cases du tableau 1 correspondant à chacun des 44 énoncés du questionnaire, in un «X» vis-à-vis du choix de réponse que vous avez sélectionné (par exemple, si vou choisi «a» à l'énoncé n° 3, écrivez un «X» dans la colonne «a» pour l'énoncé n° 3.).

2. Faites le décompte des «X» pour chaque colonnes et écrivez-en le total dans les espaces à cet effet.

3. Pour chacune des quatre échelles (ACT/REF, SNS/INT, VIS/VRB SEQ/GLO), soustr plus petit total du plus grand. Inscrivez la différence (de 1 à 11) et la lettre (a ou b) total était plus grand sur la ligne prévue à cet effet.

   Par exemple, si sous «ACT/REF», vous avez 4 «a» et 7 «b», vous écrivez «3b» dernière ligne du tableau.

Tableau 1 - Compilation des réponses

ACT/REF Q a b	SNS/INT Q a b	VIS/VRB Q a b	SEQ/GLO Q a b
1 ___ ___	2 ___ ___	3 ___ ___	4 ___ ___
5 ___ ___	6 ___ ___	7 ___ ___	8 ___ ___
9 ___ ___	10 ___ ___	11 ___ ___	12 ___ ___
13 ___ ___	14 ___ ___	15 ___ ___	16 ___ ___
17 ___ ___	18 ___ ___	19 ___ ___	20 ___ ___
21 ___ ___	22 ___ ___	23 ___ ___	24 ___ ___
25 ___ ___	26 ___ ___	27 ___ ___	28 ___ ___
29 ___ ___	30 ___ ___	31 ___ ___	32 ___ ___
33 ___ ___	34 ___ ___	35 ___ ___	36 ___ ___
37 ___ ___	38 ___ ___	39 ___ ___	40 ___ ___
41 ___ ___	42 ___ ___	43 ___ ___	44 ___ ___
Total (total des «X» dans chaque colonne)			
ACT/REF a b ___ ___	**SNS/INT** a b ___ ___	**VIS/VRB** a b ___ ___	**SEQ/GLO** a b ___ ___
(Plus grand nombre – plus petit nombre) + Lettre du plus grand nombre (voir exemple ci-dessous *)			
_____	_____	_____	_____

* **Exemple** : Si vous avez un total de 3 pour «a», et de 8 pour «b», vous inscrivez «5b» dan l'espace situé en dessous.

4. Transférez maintenant vos résultats dans le tableau 2 ci-dessous en inscrivant des «X» au-dessus des pointages qui correspondent à vos résultats pour chacune des quatre échelles. Le tableau 3 donne la liste des appellations utilisées pour chaque échelle.

Tableau 2 – Profil de vos styles d'apprentissage

ACT													**REF**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
SNS													**INT**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
VIS													**VRB**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
SEQ													**GLO**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	

Tableau 3 - Appellations des différentes échelles de styles d'apprentissage

Appellation abrégée	**Anglais**	**Français**
ACT/REF	active/reflexive	actif/réflexif
SNS/INT	sensing/intuitive	sensation/intuition
VIS/VRB	visual/verbal	visuel/verbal
SEQ/GLO	sequential/global	analytique/global

Interprétation des résultats

Si votre pointage pour une échelle se situe entre 1 et 3, vous êtes assez bien équilibré dans les deux dimensions de cette échelle.

Si votre pointage pour une échelle est de 5 ou de 7, vous démontrez une préférence marquée pour une dimension de l'échelle et vous apprendrez plus facilement dans un environnement d'apprentissage qui est favorable à cette dimension.

Si votre pointage pour une échelle est de 9 ou de 11, vous démontrez une préférence très marquée pour une dimension de l'échelle. Vous pourriez avoir une réelle difficulté à apprendre dans un environnement qui n'est pas adapté à cette préférence.

Voir l'article «Learning Styles and Strategies» de Richard Felder et de Barbara Soloman pour l'explication détaillée des différents profils.

5. Recopiez à la page suivante les résultats compilés dans le tableau 2, et remettez cette page au professeur.

Copie pour le professeur

Nom de l'étudiant : _____

Profil de mes styles d'apprentissage d'après le questionnaire de Felder et Soloman.

ACT													**REF**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
SNS													**INT**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
VIS													**VRB**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	
SEQ													**GLO**
	11a	9a	7a	5a	3a	1a	1b	3b	5b	7b	9b	11b	

Annexe C : PWM

« La PWM (Pulse Width Modulation ou Modulation de largeur d'impulsion) est un mode de commande des moteurs à courant continu par un signal carré de fréquence (de période) constante mais dont le rapport cyclique (duty cycle en Anglais) est variable. Le rapport cyclique est défini comme étant le rapport entre la durée de l'état haut du signal et la période de ce signal. »

Retiré de :
http://www.electroniquepratique.com/article.asp?mag=MR&num=4&article=8

Annexe D : Formulaires concernant les normes d'éthique

ÉCOLE
MONTRÉAL

Comité d'éthique de la recherche avec des sujets humains

Adresse civique
Campus de
(Université de Montréal)
2900, boul. Édouard-Montpetit
École Polytechnique
2500, chemin de Polytechnique
Montréal (Québec) H3T 1J4

Adresse postale
C. P. 6079, succursale Centre-ville
Montréal (Québec) H3C 3A7

Téléphone : (514) 340-4801
Télécopieur : (514) 340-4992

École affiliée à l'Université de Montréal

Membres du comité

Ghyslain Dennoncourt, IRSST
Daniel Jodoin, MAGI
Bernard Lacharité, éthicien
André Phaneuf, médecin-chercheur
Pierre Savard, ISR, président

Jean Choquette, secrétaire

CERTIFICAT D'ACCEPTATION D'UN PROJET DE RECHERCHE PAR LE COMITÉ D'ÉTHIQUE DE LA RECHERCHE AVEC DES SUJETS HUMAINS DE L'ÉCOLE POLYTECHNIQUE

Montréal, le 2 décembre 2005.

Mme Anka-Stoykova MIHAYLOVA
Étudiante à la maîtrise
Département de génie informatique
École Polytechnique de Montréal

N/Réf : Dossier CÉR-05/06-14

Cher Mme Mihaylova,

J'ai le plaisir de vous informer que le Comité d'éthique de la recherche avec des sujets humains de l'École Polytechnique a approuvé, lors de sa réunion du 2 décembre 2005, votre projet de recherche intitulé « *Détermination d'un profil de l'apprenant des concepts fondamentaux de la programmation* » que vous réalisez sous la direction du professeur Yves Boudreault.

Il est entendu que le présent certificat est valable pour le projet tel que soumis au Comité d'éthique de la recherche avec sujets humains. Les membres du Comité d'éthique de la recherche avec des sujets humains devront être informés de toute modification qui pourrait être apportée ultérieurement au protocole expérimental, de même que de tout problème imprévu pouvant avoir une incidence sur la santé et la sécurité des personnes impliquées dans le projet de recherche (sujets, professionnels de recherche ou chercheurs).

Nous vous prions également de nous faire parvenir **un bref rapport annuel** ainsi qu'un avis à la fin de vos travaux.

Je vous souhaite bonne chance dans vos travaux de recherche,

Pierre Savard
Président
Comité d'éthique de la recherche avec des sujets humains

c.c. : Yves Boudreault, GI
Jean Choquette, DRI

FORMULAIRE DE CONSENTEMENT

RESPONSABLE DU PROJET		
NOM : Mihaylova	PRÉNOM : Anka-Stoykova	TITRE : Étudiante à la maîtrise
DÉPARTEMENT : Génie informatique	TÉLÉPHONE : COURRIEL : anka-stoykova.mihaylova@polymtl.ca	
PROJET DE RECHERCHE		
TITRE du projet : Détermination d'un profil de l'apprenant des concepts fondamentaux de la programmation		

DÉCLARATION

Je, (nom en lettres moulées du sujet) _____ déclare avoir pris connaissance des documents ci-joints dont j'ai reçu copie, en avoir discuté avec (nom du chercheur en lettres moulées) Anka-Stoykova Mihaylova et avoir compris le but, la nature, les avantages, les inconvénients et les risques de cette recherche.

Après réflexion, je consens librement à participer à cette recherche. Je sais que je peux me retirer de celle-ci en tout temps, sans aucune forme de préjudice. De plus, tous les renseignements obtenus me concernant seront strictement confidentiels, à moins d'une autorisation de ma part ou d'une disposition législative. Pour ce faire, ces renseignements seront codés (ex.: codés, sous clé, etc.) et conservés à l'École Polytechnique de Montréal et à (dans le cas d'une recherche multicentre), pour une durée de deux années après la fin de la recherche. Par la suite, tous ces renseignements seront détruits.

Cependant, je consens à ce qu'aux fins de vérification de la saine gestion de cette recherche, les personnes suivantes puissent consulter les informations s'y rattachant : un membre du comité d'éthique de la recherche (C.É.R.), et/ou un délégué des autorités gouvernementales de Santé Canada, et/ou un ministère concerné du Québec, et/ou un délégué des organismes commanditaires. Par ailleurs, j'accepte que les résultats de cette recherche puissent être publiés, dans une revue technique ou scientifique, ou communiqués dans le cadre d'un congrès technique ou scientifique, sans que mon anonymat en soit compromis.

_____ _____

Signature du sujet Date

FORMULAIRE DE DESCRIPTION DES AVANTAGES, DES INCONVÉNIENTS ET DES RISQUES DE LA PARTICIPATION COMME SUJET AU PROJET DE RECHERCHE

RESPONSABLE DU PROJET		
NOM : Mihaylova	PRÉNOM : Anka-Stoykova	TITRE : Étudiante à la maîtrise
DÉPARTEMENT : Génie informatique	TÉLÉPHONE : COURRIEL : anka-stoykova.mihaylova@polymtl.ca	
PROJET DE RECHERCHE		
TITRE du projet : Détermination d'un profil de l'apprenant des concepts fondamentaux de la programmation		

DÉCLARATIONS

Je, (nom en lettres moulées du responsable du projet) Anka-Stoykova Mihaylova déclare avoir expliqué le but et la nature de ce projet ainsi qu'avoir dénoncé les inconvénients et les risques suivants : prendre conscience de l'ampleur de ses propres difficultés d'apprentissage

_____

À : Nom en lettres moulées du sujet

_____ _____

Signature du chercheur Date

JE RECONNAIS AVOIR ÉTÉ INFORMÉ DE CES INCONVÉNIENTS ET DE CES RISQUES

_____ _____

Signature du sujet Date

_____

Nom en lettres moulées du témoin (il s'agit d'une personne non associée à cette recherche)

_____ _____

Signature du témoin Date

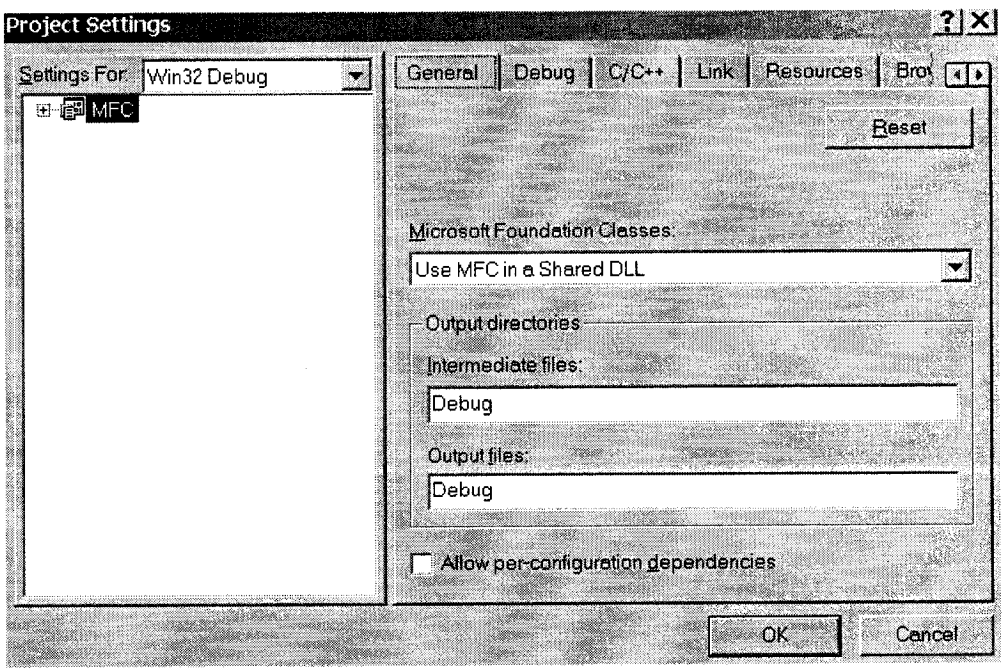
Annexe E : Organisation de l'espace du travail

PROCÉDURE D'INSTALLATION DE L'ESPACE DE TRAVAIL

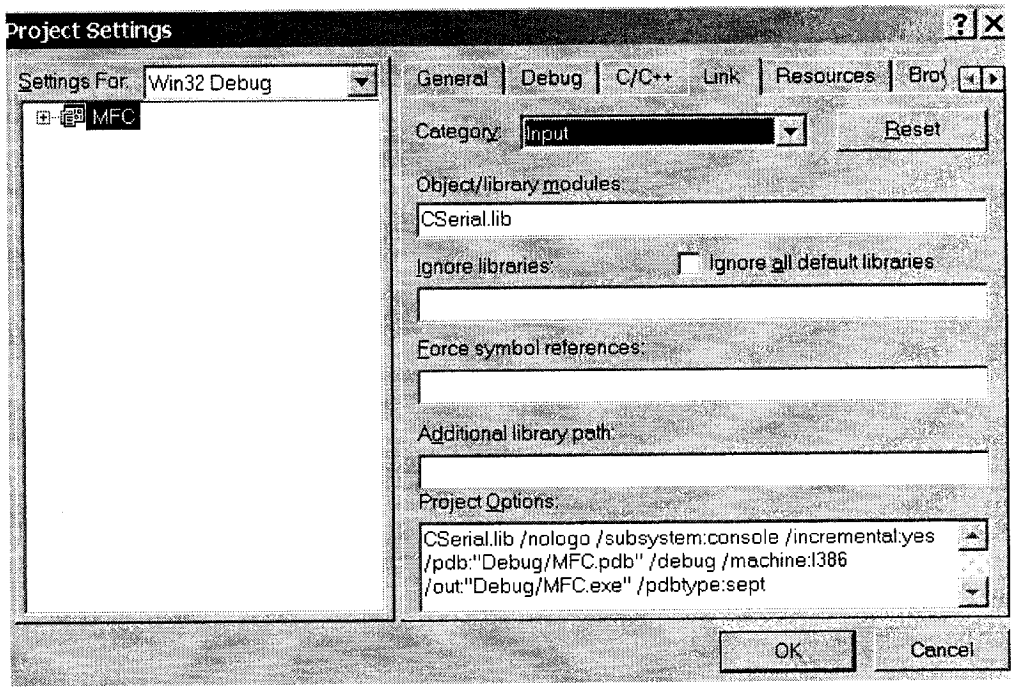
1. Copier les fichiers : MainSerial.CPP,
   Utilitaires.h, Serial.h, StdAfx.h,
   CSerial.LIB et CSerial.DLL
   dans votre répertoire courant du travail, i.e. celui que vous avez créé pour l'exercice.

2. Ajouter les fichiers MainSerial.CPP et Utilitaires.h dans votre projet (Source Files, Header Files).

3. Ouvrir la boite de dialogue « Project settings » du menu « Project ». Sélectionner l'onglet « General ». Sélectionner l'item «Use MFC in a Shared DLL» du bouton déroulant « Microsoft Foundation Classes ».



4. Ouvrir la boite de dialogue « Project settings » du menu « Project ». Sélectionner l'onglet « Link ». Sélectionner l'item « Input » du bouton déroulant « Category ». Ajouter dans la fenêtre de texte « Object/library » : CSerial.lib



5. Compiler et lier le projet.

Annexe F : Les énoncés des exercices

ATELIER

Instructions de décision

Instructions de répétition

Objectifs : permettre à l'étudiant de vérifier ses connaissances dans le domaine de la programmation en langage C et C++.

Durée: une séance de laboratoire (3 cours).

Remise du travail: à la fin de la troisième séance de laboratoire.

Documents à remettre: Le listing de tous les exercices dans le répertoire inf1005_depot\Expérience Anka.

Exercice 1

Soit un robot effectuant des déplacements selon les indications qui lui sont fournies à partir du clavier.

Les commandes possibles sont : HAUT, BAS, DROITE, GAUCHE et TERMINER. Elles correspondent aux déplacements suivants dans un plan cartésien :

HAUT	-	déplacement de plus une unité selon y
BAS	-	déplacement de moins une unité selon y
DROITE	-	déplacement de plus une unité selon x
GAUCHE	-	déplacement de moins une unité selon x
TERMINER	-	aucun déplacement (fin des commandes).

Écrire un programme qui demande et lit coordonnées initiales. Une vérification est nécessaire puisque les coordonnées doivent être positives.

Le programme demande et lit ensuite les commandes jusqu'à la lecture de TERMINER. Pour chaque commande, le programme effectue les vérifications et réalise le déplacement souhaité. Si le déplacement est dans la zone négative il ne sera pas effectué.

Finalement, le programme affiche la position finale ainsi que tous les déplacements effectués et non effectués.

Exemple : (le soulignement identifie une entrée de l'usager)

Valeur initiale

X = 1

Y = 1

Déplacement : BAS

Déplacement : BAS

Déplacement : HAUT

Déplacement : HAUT

Déplacement : HAUT

Déplacement : GAUCHE

Déplacement : TERMINER

Valeur finale

X = 0

Y = 3

Déplacement effectué : BAS HAUT HAUT HAUT GAUCHE

Déplacement non effectué : BAS

Exercice 2 UTILISATION DU MONTAGE ÉLECTRONIQUE

Écrire le numéro du montage électronique dans l'en-tête du programme ainsi que votre numéro d'identification.

1. Exécuter la fonction DemarrerArreterMoteur(Port)

Mettre cette fonction en commentaire et compléter le code qui réalisera la même opération, c'est-à-dire qui calcule et affiche :

- le coefficient le plus proche de la valeur de démarrage du moteur sur le montage électronique (le nombre de tours/sec >1);
- le coefficient le plus proche de la valeur d'arrêt du moteur (le nombre de tours/sec<=1).

Chaque fois il faut introduire un entier (1<entier<100) et obtenir la nouvelle vitesse (nombres de tours/sec).

2. Exécuter la fonction AllumerEteindre(Port) pour trouver la séquence qui permettra d'allumer toutes les luminodiodes (leds).

Mettre cette fonction en commentaire et compléter le code qui réalisera la même opération, c'est-à-dire qui allume les leds en respectant une séquence précise. La couleur des leds est introduite au clavier. Vous devez faire une vérification des entrées. Utiliser les symboles suivants :

O/o - pour la couleur orange J/j - pour la couleur jaune
R/r - pour la couleur rouge V/v - pour la couleur verte
T/t - pour terminer.

À la fin éteindre les leds en introduisant n'importe quelle séquence après un certain délai.

Annexe G : Calculs statistiques

Exercice à caractère abstrait (ROBOT) -REF		
échantillon	valeur	explication
1	100	Ce sont les résultats des étudiants du type REF concernant leurs programmes – s'ils s'exécutent correctement (100), s'ils s'exécutent avec remarques (50) ou s'ils ne s'exécutent pas (0).
2	0	
3	50	
4	100	
5	100	
6	100	
7	100	
8	100	
9	100	
10	100	
11	100	
Σ	950	
n	11	number
X	86.36363636	mean=Σ/n
$\Sigma(X^2)$	92500	
$(\Sigma X)^2$	902500	
$\Sigma(d^2)$	10454.54545	$\Sigma d^2 = \Sigma x^2 - \frac{(\Sigma x)^2}{n}$
σ^2	1045.454545	$\frac{\Sigma d^2}{n-1}$
σ	32.33348953	$= \sqrt{\sigma^2}$
σ_n	9.748913904	standart error= $\sigma / \sqrt{n}$
$X \pm \sigma_n$	86.36363636 ±9.748913904	

95% confidence	86.36363636 ±21.74007801	degree of freedom =10

Exercice à caractère abstrait (ROBOT) -ACT		
échantillon	valeur	explication
1	50	Ce sont les résultats des étudiants du type ACT concernant leurs programmes – s'ils s'exécutent correctement (100), s'ils s'exécutent avec remarques (50) ou s'ils ne s'exécutent pas (0)
2	0	
3	0	
4	0	
5	50	
6	50	
7	50	
Σ	200	
n	7	number
X	28.57142857	mean=Σ/n
$\Sigma(X^2)$	10000	
$(\Sigma X)^2$	40000	
$\Sigma(d^2)$	4285.714286	$\sum d^2 = \sum x^2 - \frac{(\sum x)^2}{n}$
σ^2	714.2857143	$\frac{\sum d^2}{n-1}$
σ	26.72612419	$= \sqrt{\sigma^2}$
σ_n	10.10152545	standart error $= \sigma / \sqrt{n}$
$X \pm \sigma_n$	28.57142857 ±10.10152545	
95% confidence	28.57142857 ±24.74873734	degree of freedom =6

Exercice à caractère concret (MOTEUR ET LEDS)-REF		
échantillon	valeur	explication
1	0	Ce sont les résultats des étudiants du REF concernant leurs programmes – s'ils s'exécutent correctement (100) ou s'ils ne s'exécutent pas (0). Chaque programme est divisé en deux parties : 1. la partie concernant les leds 2. la partie concernant le moteur. Chaque partie est estimée séparément s'il s'exécute (100) ou ne s'exécute pas (0). Pour cette raison le nombre est *2.
2	0	
3	0	
4	0	
5	0	
6	100	
7	100	
8	100	
9	0	
10	100	
11	0	
12	0	
13	100	
14	100	
Σ	600	
n	14	number
X	42.85714286	mean=Σ/n
$\Sigma(X^2)$	60000	
$(\Sigma X)^2$	360000	
$\Sigma(d^2)$	34285.71429	$\sum d^2 = \sum x^2 - \frac{(\sum x)^2}{n}$
σ^2	2637.362637	$\frac{\sum d^2}{n-1}$
σ	51.3552591	$= \sqrt{\sigma^2}$
σ_n	13.72527033	standart error $= \sigma / \sqrt{n}$

$X \pm \sigma_n$	42.85714286 ±13.72527033	
95% confidence	42.85714286 ±29.6465839	degree of freedom =13

Exercice à caractère concret (MOTEUR ET LEDS) -ACT		
échantillon	valeur	explication
	0	Ce sont les résultats des étudiants du type ACT concernant leurs programmes – s'ils s'exécutent correctement (100) ou s'ils ne s'exécutent pas (0). Chaque programme est divisé en deux parties : 1. la partie concernant les leds 2. la partie concernant le moteur. Chaque partie est estimée séparément s'il s'exécute (100) ou ne s'exécute pas (0). Pour cette raison le nombre est *2.
	0	
	0	
	0	
	100	
1	0	
2	0	
3	0	
4	100	
5	100	
6	100	
7	100	
Σ	400	
n	7	number
X	57.14285714	mean=Σ/n
$\Sigma(X^2)$	40000	
$(\Sigma X)^2$	160000	
$\Sigma(d^2)$	17142.85714	$\sum d^2 = \sum x^2 - \frac{(\sum x)^2}{n}$

σ^2	2857.142857	$\frac{\sum d^2}{n-1}$
σ	53.45224838	$= \sqrt{\sigma^2}$
σ_n	20.20305089	standart error $= \sigma / \sqrt{n}$
$X \pm \sigma_n$	57.14285714 ±20.20305089	
95% confidence	57.14285714 ±49.49747468	degree of freedom =6