

PIM : Mini-projet 1

Auteur 1 (Exercice 1 & 3) : Rahma

Auteur 2 (Exercice 2) : Myriam

Raffinages exercice 1	1
Les raffinages	1
Evaluation par les étudiants	2
Remarques diverses	2
Raffinages exercices 2	2
Les raffinages	2
Evaluation par les étudiants	3
Remarques diverses	3
Raffinages exercices 3	4
Les raffinages	4
Evaluation par les étudiants	4
Remarques diverses	4
Exercice 4	5
Bilan	5
Annexe : Le code complet	5

Raffinages exercice 1

Les raffinages

R0 : Faire deviner à l'utilisateur le nombre choisi par l'ordinateur.

Exemples :

J'ai choisi un nombre compris entre 1 et 999.

Proposition 1 : 20

Trop petit.

Proposition 2 : 100

Trop grand.

Proposition 3: 70

Trop grand

Proposition 4 : k

Vous devez saisir un nombre entre 1 et 999

Proposition 5 : 60

Trop petit

Proposition 6 : 63

Trouvé !

Bravo. Vous avez trouvé 63 en 6 essais.

R1 : Comment "Faire deviner à l'utilisateur le nombre choisi par l'ordinateur"

Choisir un nombre aléatoire entre 1 et 999

Ecrire("J'ai choisi un nombre aléatoire entre 1 et 999")

Demander à l'utilisateur de deviner

Afficher le nombre d'essais réalisés par l'utilisateur après avoir deviné le nombre.

R2 : Comment "Demander à l'utilisateur de deviner"?

Compteur <- 0

- -variable compteur du nombre d'essais

Compteur : out

Répéter

Lire le nombre choisi par l'utilisateur

Déterminer si le nombre deviné est plus petit, plus grand ou égal au nombre choisi

Compteur <- Compteur + 1

Compteur : in out

Jusqu'à Nombre_uti = Nombre_orði

Nombre_uti, Nombre_orði : in

R2 : Comment : “ Afficher le nombre d’essais réalisés par l’utilisateur après avoir deviné le nombre.”?

```

Ecrire (“Bravo. Vous avez trouvé” )
Ecrire (Nombre_ordi)           Nombre_ordi : in
Ecrire(“ en ”)
Ecrire (Compteur)              Compteur : in
Ecrire(“essais”)

```

R3 : Comment : “ Lire le nombre choisi par l’utilisateur” ?

```

Ecrire (“Proposition “)
Ecrire (Compteur)              Compteur : in
Lire (Nombre_uti)              Nombre_uti : out

```

R3 : Comment “Déterminer si le nombre deviné est plus petit ou plus grand que le nombre choisi” ?

```

Si Nombre_uti > Nombre_ordi Alors           Nombre_uti, Nombre_ordi :in
    Ecrire (“Trop grand”)
SinonSi Nombre_ordi > Nombre_uti Alors
    Ecrire(“Trop petit”)
SinonSi Nombre_ordi = Nombre_uti Alors
    Ecrire (“Trouvé ! “)
Sinon
    Ecrire(“Vous devez saisir un nombre entre 1 et 999”)
FinSi

```

Evaluation par l'autre étudiant

		Evaluation Etudiant (I/P/A/+)	Justification / commentaire	Evaluati on Enseign ant (I/P/A/+)
Forme (D-21)	Respect de la syntaxe Ri : Comment "... une action complexe ..." ? des actions combinées avec des structures de controle	+	Les règles sont toujours respectées. Le comment est là, les actions complexes sont	

	Rj : ...		données, le raffinage est indenté...	
	Verbe à l'infinitif pour les actions complexes	+	Ces règles sont toujours respectées. Les verbes sont toujours à l'infinitif.	
	Nom ou équivalent pour expressions complexes	A	Les expressions complexes sont bien nommées.	
	Tous les Ri sont écrits contre la marge et espacés	+	Ces règles sont toujours respectées. Les Ri sont contre la marge et 2 lignes blanches les séparent.	
	Les flots de données sont définis	A	Les flots de données sont définis pour toutes les actions complexes.	
	Une seule décision ou répétition par raffinage	A	Une seule décision par raffinage.	
	Pas trop d'actions dans un raffinage (moins de 6)	A	Aucun raffinage ne dépasse les 6 actions.	
	Bonne présentation des structures de contrôle	A	Toutes les structures de contrôle sont présentées comme en cours.	
Fond (D21-D22)	Le vocabulaire est précis	+	Le raffinage est facilement compréhensible.	
	Le raffinage d'une action décrit complètement cette action	+	Chaque raffinage décrit bien et complètement	

			l'action complexe.	
	Le raffinage d'une action ne décrit que cette action	+	Il ne semble pas y avoir d'actions en trop dans les décompositions.	
	Les flots de données sont cohérents	+	Les in et les out semblent correctes et au bons endroits	
	Pas de structure de contrôle déguisée	+	Il n'y a pas de structure de contrôle déguisée.	
	Qualité des actions complexes	+	Les actions sont explicites.	

Raffinages exercices 2

Les raffinages

R0 : Faire deviner à l'ordinateur un entier entre 1 et 999 choisi par l'utilisateur.

Exemple :

Avez-vous choisi un nombre compris entre 1 et 999 (o/n)?

n

J'attends...

Avez-vous choisi un nombre compris entre 1 et 999 (o/n)?

o

Proposition 1: 900

Trop (g)rand, trop (p)etit, (t)rouvé?

g

Proposition 2 : 450

Trop (g)rand, trop (p)etit, (t)rouvé?

G

Proposition 3 : 225

Trop (g)rand, trop (p)etit, (t)rouvé?

x

Je n'ai pas compris,veuillez saisir :

“g”/”G” si ma proposition est trop grande , “p”/”P” si trop petite , “t”/”T” si j'ai trouvé le nombre.

Trop (g)rand, trop (p)etit, (t)rouvé?

P

Proposition 4 : 337

Trop (g)rand, trop (p)etit, (t)rouvé?

p

Proposition 5 : 393

Trop (g)rand, trop (p)etit, (t)rouvé?

g

Proposition 6 : 367

Trop (g)rand, trop (p)etit, (t)rouvé?

T

J'ai trouvé le nombre 367 en 6 essais !

R1 : Comment : “Faire deviner à l’ordinateur un entier entre 1 et 999 choisi par l'utilisateur?.

-- On considère que 1 et 999 sont compris dans l'intervalle

Ecrire(“Avez-vous choisi un nombre compris entre 1 et 999 (o/n)?”)

Lire(Reponse)

Reponse : out

Demander à l'utilisateur si il a choisi un nombre.

Deviner le nombre choisi par l'utilisateur entre Mini et Maxi

R2 : Comment : “Demander à l'utilisateur si il a choisi un nombre ?

Mini <- 1

Mini : out

Maxi <-999

Maxi : out

Tant que Réponse <> “o” Faire

Reponse : in

Ecrire (“J’attends...”)

Ecrire(“Avez-vous choisi un nombre compris entre 1 et 999 (o/n)?”)

Lire(Reponse)

Reponse : out

FinTQ

R2 : Comment : “Deviner le nombre choisi par l'utilisateur entre Mini et Maxi” ?

Compteur <- 0

Compteur : out

Repeter

Proposer un Nombre_ordi

Demander une indication à l'utilisateur;

Mettre à jour Mini et Maxi suivant l'indication

Compteur <- Compteur + 1

Compteur : in out

Jusqu'à (Indication = 't') ou (Indication='T') ou (Mini>Maxi)

Donner le résultat -- En cas de triche, un message apparaît

R3 : Comment “ Proposer un Nombre_ordi”?

Milieu <- (Mini+Maxi)/2

Milieu : out

Mini,Maxi : in

Nombre-ordi <- Milieu

Ecrire(Nombre_ordi)

Nombre_ordi : out

R3 : Comment “ Demander une indication à l'utilisateur” ?

Répéter

Ecrire (“Trop (g)rand, trop (p)etit, (t)rouvé? “)

Lire(Indication)

Indication : out

jusqu'à (Indication='t') ou (Indication='T') ou (Indication ='p') ou (Indication='P') ou (Indication ='g') ou (Indication ='G')

R3 : Comment “mettre à jour Mini et Maxi suivant l'indication” ?

Si Indication = ('p' ou 'P') Alors

Mini <- Milieu + 1

SinonSi Indication = ('g' ou 'G') Alors

Maxi <- Milieu - 1

Fin Si

Indication : in

Maxi : out

Milieu : in

Mini : out

Milieu :in

R3 : Comment “Donner le résultat”?

Si Mini>Maxi Alors

Ecrire(“Vous trichez ! “)

Sinon

Afficher que le Nombre est trouvé

FinSi

Mini, Maxi : in

R4: Comment “Afficher que le nombre est trouvé” ?

Ecrire(“J’ai trouvé le nombre “)

Ecrire(Nombre_ordi)

Ecrire(“en”)

Ecrire(Compteur)

Ecrire(“Etapas”)

Nombre_ordi :in

Compteur : in

Evaluation par l'autre étudiant

		Evaluation Etudiant (I/P/A/+)	Justification / commentaire	Evaluati on Enseign ant (I/P/A/+)
Forme (D-21)	Respect de la syntaxe Ri : Comment "... une action complexe ..." ? des actions combinées avec des structures de controle Rj : ...	+	Les règles sont toujours respectées.	
	Verbe à l'infinitif pour les actions complexes	+	Les verbes sont toujours à l'infinitif.	
	Nom ou équivalent pour expressions complexes	A	Les expressions complexes sont	

			bien nommées	
	Tous les Ri sont écrits contre la marge et espacés	+	Ces règles sont toujours respectées. Les Ri sont contre la marge et 2 lignes blanches les séparent.	
	Les flots de données sont définis	A	Les flots de données sont définis pour toutes les actions complexes	
	Une seule décision ou répétition par raffinage	A	Une seule décision par raffinage.	
	Pas trop d'actions dans un raffinage (moins de 6)	A	Aucun raffinage ne dépasse les 6 actions.	
	Bonne présentation des structures de contrôle	A	Toutes les structures de contrôle sont présentées comme en cours.	
Fond (D21-D 22)	Le vocabulaire est précis	+	Le raffinage est facilement compréhensible.	
	Le raffinage d'une action décrit complètement cette action	+	Chaque raffinage décrit complètement l'action complexe.	
	Le raffinage d'une action ne décrit que cette action	+	Il n'y a pas d'actions en trop dans les décompositions.	
	Les flots de données sont cohérents	+	Les in et les out sont correctes et aux bons endroits	
	Pas de structure de contrôle déguisée	+	Il n'y a pas de	

			structure de contrôle déguisée.	
	Qualité des actions complexes	+	Les actions sont explicites.	

Raffinages exercices 3

Les raffinages.

R0 : “Jouer au jeu du Devin”

Exemple :

Choisissez votre mode de jeu :

1 - L'ordinateur choisit un nombre et vous le devinez

2 - Vous choisissez un nombre et l'ordinateur le devine

0 - Quitter le programme

Votre Choix: 1

...

R1 : **Comment** “Jouer au jeu du Devin” ?

Afficher le menu du jeu

Appliquer le mode de jeu choisi par l'utilisateur

R2 : **Comment** “Afficher le menu du jeu” ?

Ecrire(“Choisissez votre mode de jeu : 1 si l'ordinateur choisit le nombre et vous devinez, 2 si c'est l'inverse et 0 si vous voulez quitter le jeu”)

Lire(Mode_choisi)

Mode_choisi : out

R2 : **Comment** “Appliquer le mode de jeu choisi par l'utilisateur” ?

Tant que Mode_choisi <> 0 **Faire**

Si Mode_choisi = 1 **Alors**

Mode_choisi : in

Faire deviner à l'utilisateur le nombre choisi par l'ordinateur.

- *-Le raffinage est donc celui de l'exercice 1*

Afficher le menu du jeu - - A la fin du jeu 1 on réaffiche le menu

SinonSi Mode_choisi = 2 **Alors**

Deviner le nombre choisi par l'utilisateur - -le raffinage est donc celui de l'exercice 2

Afficher le menu du jeu - - A la fin du jeu 2 on réaffiche le menu

FinSi

FinTQ

Quitter

R3 : Comment "Quitter"?

Rien

Evaluation par l'autre étudiant

		Evaluation Etudiant (I/P/A/+)	Justification / commentaire	Evaluation Enseignant (I/P/A/+)
Forme (D-21)	<p>Respect de la syntaxe</p> <p>Ri : Comment "... une action complexe ..." ? des actions combinées avec des structures de controle</p> <p>Rj : ...</p>	+	Les règles sont toujours respectées. Le comment est là, les actions complexes sont données, le raffinage et indenté...	
	Verbe à l'infinitif pour les actions complexes	+	Ces règles sont toujours respectées. Les verbes sont toujours à l'infinitif.	
	Nom ou équivalent pour expressions complexes	A	Les expressions complexes sont bien nommées.	
	Tous les Ri sont écrits contre la marge et espacés	+	Ces règles sont toujours respectées. Les Ri sont contre la marge et 2 lignes	

			blanches les séparent.	
	Les flots de données sont définis	A	Les flots de données sont définis pour toutes les actions complexes	
	Une seule décision ou répétition par raffinage	A	Une seule décision par raffinage.	
	Pas trop d'actions dans un raffinage (moins de 6)	P	Dans le raffinage R2 il y a 7 actions.	
	Bonne présentation des structures de contrôle	A	Toutes les structures de contrôle sont présentées comme en cours.	
Fond (D21-D 22)	Le vocabulaire est précis	+	Le raffinage est facilement compréhensib le.	
	Le raffinage d'une action décrit complètement cette action	+	Chaque raffinage décrit bien et complètement l'action complexe.	
	Le raffinage d'une action ne décrit que cette action	+	Il ne semble pas y avoir des actions en trop dans les décompositio ns.	
	Les flots de données sont cohérents	+	Les in et les out semblent correctes et au bons endroits	
	Pas de structure de contrôle déguisée	+	Il n'y a pas de structure de	

			contrôle déguisée.	
	Qualité des actions complexes	+	Les actions sont explicites.	

Bilan

Ce mini-projet a permis de mieux assimiler certaines notions théoriques du cours, notamment avec le raffinage. De plus, il nous a permis de nous familiariser et de maîtriser un minimum le langage Ada.

Annexe : Le code complet

Evaluation du code

		Consigne : Mettre O (oui) ou N (non) dans la colonne Etudiant suivant que la règle a été respectée ou non. Une justification peut être ajoutée dans la colonne "commentaire".	
Commentaire	Etudiant (O/N)	Règle	Enseignant (O/N)
Plusieurs tests ont montré que le programme fonctionne correctement.	O	Le programme ne doit pas contenir d'erreurs de compilation.	
	O	Le programme doit compiler sans messages d'avertissement.	
	O	Le code doit être bien indenté.	
	O	Les règles de programmation du cours doivent être respectées : toujours un Sinon pour un Si, pas de sortie au milieu d'une répétition...	

	O	Pas de code redondant.	
	O	On doit utiliser les structures de contrôle adaptées (Si/Selon/TantQue/Répéter/Pour)	
	O	Utiliser des constantes nommées plutôt que des constantes littérales.	
Les commentaires dans le code délimitent les différentes étapes du raffinage.	O	Les raffinages doivent être respectés dans le programme.	
	O	Les actions complexes doivent apparaître sous forme de commentaires placés AVANT les instructions correspondantes, avec la même indentation	
	O	Une ligne blanche doit séparer les principales actions complexes	
	O	Le rôle des variables doit être explicité à leur déclaration (commentaire).	