

PIM : Mini-projet 1

TODO : Vous ne pouvez pas modifier ce document. Vous devez en **créer une copie** (Fichier / Créer une copie) que vous **partagerez avec votre enseignant de TP** (c'est lui qui corrigera votre mini-projet).

TODO : Nommer votre document PIM-MP1-X-Nom où X est la lettre de votre groupe de TP (A, B, C...) et Nom votre nom.

TODO : Les TODO doivent être enlevés quand ils sont traités.

Raffinages	1
Évaluation des raffinages par l'étudiant	2
Remarques diverses	2
Évaluation du code	3

Raffinages

TODO : écrire ici les raffinages en suivant les règles présentées en cours. On ne donnera pas d'exemples.

R0 :<<jouer le jeu a 13 allumette>>

R1:comment<< jouer le jeu a 13 allumette>>?

Demander le niveau du jeu niveau : out Caractère

Afficher le niveau du jeu niveau : in Caractère

Demander à l'utilisateur s'il veut jouer Reponse : out caractere

Jouer une parti

Afficher le gagnant

R2: Comment <<demander le niveau du jeu>>

Ecrire("Niveau de l'ordinateur (n)aiif, (d)istrail, (r)apide ou (e)xpert ?")

Lire (niveau)

R2: Comment << afficher le niveau du jeu>>

Ecrire ("Mon niveau est")

Selon niveau faire

'n', "N" => Ecrire("naïf")

'd', " D" => Ecrire("distrail")

**'r' , 'R' => Ecrire("Rapide")
Autre => Ecrire("expert")**

R2: comment <<Demander a l'utilisateur s'il veut jouer >> ?

**Ecrire ("Est-ce que vous commencez (o/n) ?")
Lire(Reponse)
Si Reponse ="O" ou Reponse="o" Alors**

Tour <- Vraie Tour : out Booléen

Sinon

Tour <- Faux Tour : out Booléen

FinSi

R2: comment 'jouer une partie' ?

**Nombre_Allumette <- 13 Nombre_Allumette : out Entier
 Nombre_Choisi: out Entier
gagne<- Vraie gagne:out Booléen
 Tour:out Booléen**

TantQue Nombre_Allumette >0 Faire

Afficher le nombre d'allumette

Si tour Alors

Demander a l'utilisateur de choisir un nombre

Gagne <- Faux Gagne: out Booléen

Nombre_Allumette <- Nombre_Allumette - Nombre_Choisi

Afficher le nombre d'allumette

FinSi

Si Nombre_Allumette>0 Alors

Choisir un nombre selon le niveau du jeu

Nombre_Allumette <- Nombre_Allumette - Nombre_Choisi

Gagne <- Vraie Gagne: out Booléen

Tour <- vraie tour : out Booléen

Fin Si

FinTQ

R2:comment<< afficher le gagnant>>

Si gagne alors

Ecrire("l'utilisateur a gagné")

Sinon

Ecrire("j'ai gagné")

R 3: Comment Afficher ('le nombre d'allumette')

Pour j de 1 a 3 Faire

Groupe <- Nombre_Allumettes Groupe: out Entier

TantQue Groupe>5 Faire

Ecrire("| | | |")

Groupe=Groupe-5

FinTQ

Pour i de 1 a Groupe Faire

Ecrire(" | ")

Fin Pour

Sauter une ligne

FinPour

R 3: Comment<< Demander a l'utilisateur de choisir un nombre >>?

Repeter

Ecrire("Combien d'allumettes prenez-vous ?")

Lire(Nombre_Choisi)

Si Nombre_Choisi > 3 Alors

Ecrire("Arbitre : Il est interdit de prendre plus de 3 allumettes. ")

Sinon Si Nombre_Choisi <1 Alors

Ecrire("")

Sinon Si Nombre_Choisi>Nombre_Allumette alors

Ecrire("Il reste seulement ",Nombre_Allumette,"Allumette")

Sinon

Rien

Fin Si

**JusquA Nombre_Choisi <= 3 et Nombre_Choisi >=1 et Nombre_Choisi
<=Nombre_Allumette**

R3: comment choisir un nombre selon le niveau du jeu ?

Selon niveau Faire

‘N’, ‘n’ => jouer avec le niveau naïf
‘D’, ‘d’ => jouer avec le niveau distrait
‘R’, ‘r’ => jouer avec le niveau rapide
Autres => jouer avec le niveau expert
FinSelon

R3 :Comment Afficher le gagnant ?

Si Gagne Alors

Ecrire(“l'utilisateur a gagné”)

Sinon

Ecrire(“j'ai gagné”)

R4 : Comment jouer avec le niveau naïf ?

Si Nombre_Allumette>3 Alors

Donner à Nombre_Choisi une valeur entre 1 et 3 par hasard

Sinon

Donner à Nombre_Choisi une valeur entre 1 et Nombre_Allumette par hasard

FinSi

R4 : Comment jouer avec le niveau distrait ?

Repeter

Donner à Nombre_Choisi une valeur entre 1 et 3 par hasard

Jusqu'à Nombre_Choisi<=Nombre_Allumette

R4 : Comment jouer avec le niveau rapide ?

Si Nombre_Allumette<= 3 Alors

Nombre_Choisi<-Nombre_Allumette

Sinon

Nombre_Choisi <- 3

FinSi

R4: comment jouer avec le niveau expert?

Si Nombre_Allumette>4:

Nombre_Choisi<- 3

Sinon Si Nombre_Allumette=4

Nombre_Choisi <-3

Sinon Si Nombre_Allumette=3

Nombre_Choisi<- 2

Sinon

Nombre_Choisi<-1
FinSi

Exemples : On ne mettra pas d'exemples.

...

Évaluation des raffinages par l'étudiant

		Evaluation Etudiant (I/P/A/+)	Justification / commentaire	Evaluation Enseignant (I/P/A)
Forme (D-21)	Respect de la syntaxe	p		
	Ri : Comment "... une action complexe ..." ? des actions combinées avec des structures de controle			
	Rj : ...			
	Verbe à l'infinitif pour les actions complexes	+		
	Nom ou équivalent pour expressions complexes	p		
	Tous les Ri sont écrits contre la marge et espacés	+		
	Les flots de données sont définis	+		
	Une seule structure de contrôle par raffinage	p		
Fond (D21-D 22)	Pas trop d'actions dans un raffinage (moins de 6)	A		
	Bonne présentation des structures de contrôle	A		
	Le vocabulaire est précis	+		
	Le raffinage d'une action décrit complètement cette action	+		
	Le raffinage d'une action ne décrit que cette action	P		
	Les flots de données sont cohérents	+		
	Pas de structure de contrôle déguisée	A		
	Qualité des actions complexes	A		

Remarques diverses

TODO : Indiquer ici ce qui est utile à l'enseignant pour comprendre les raffinages. Cette partie peut être vide.

Évaluation 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)
	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...	
		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.	
	O	Les raffinages doivent être respectés dans le programme.	
	N	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).	