

# PIM : Mini-projet 1

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

## Raffinages

**R0 :** Affronter la machine au jeu de nim avec 13 allumettes avec un niveau demandé à l'utilisateur.

**Exemples :** On ne mettra pas d'exemples.

**R1 :** Comment "Affronter la machine au jeu de nim à 13 allumettes avec un niveau demandé par l'utilisateur" ?

```
Demander le niveau de jeu à l'utilisateur          Niv_jeu : out Caractère
Demander si l'utilisateur veut commencer la partie  Util_deb : in Caractère,
Joueur_act : out Caractère
Nbr_al <= 13
TANT QUE Nbr_al>0 FAIRE
    Afficher le nombre d'allumettes restantes Nbr_al : in Entier
    Jouer le tour du joueur actuel au niveau demandé Nbr_al : in out Entier
FIN TANT QUE
Afficher le gagnant Joueur_act : in Caractère
```

**R2 :** Comment "Demander le niveau de jeu à l'utilisateur" ?

```
Écrire ("Niveau de l'ordinateur (n)aïf, (d)istrait, (r)apide ou (e)xpert ?")
Lire (Niv_jeu)
Afficher niveau de jeu Niv_jeu : in Caractère
```

**R2 :** Comment "Demander si l'utilisateur veut commencer" ?

**Écrire** (“Est-ce que vous commencez (o/n) ?”)  
**Lire** (Util\_deb)  
**SELON** Util\_deb **FAIRE**  
     **SI** ‘o’|‘O’ **FAIRE** Joueur\_act <- ‘u’  
     **SI AUTRES FAIRE** Joueur\_act <- ‘o’  
**FIN SELON**

**R2 :** Comment “Jouer le tour du joueur actuel au niveau demandé ” ?

**SI** Joueur\_act=‘u’ **FAIRE**  
     Jouer le tour de l'utilisateur Nbr\_al : **in out Integer**, Nal\_tour : **in out Integer**  
     Joueur\_act <= ‘o’  
  
**SINON**  
     Jouer le tour de l'ordinateur au niveau demandé  
     Joueur\_act <= ‘u’  
     **Ecrire**(“J’en prend”)  
     **Ecrire**(Nal\_tour)  
     Nbr\_al <= Nbr\_al - Nal\_tour  
**FIN SI**

**R2 :** Comment “Afficher le gagnant” ?

**SI** Joueur\_act=‘u’ **FAIRE**  
     **Écrire**(“Vous avez gagné”)  
**SINON FAIRE**  
     **Écrire**(“J’ai gagné”)  
**FIN SI**

**R2 :** Comment “Afficher le nombre d'allumettes restantes” ?

**POUR** i allant de 1 à 3 **FAIRE**  
     Afficher les allumettes par paquet de maximum 5 Nbr\_al : **in Caractère**,  
     **Nouvelle\_ligne**()  
**FIN POUR**

**R3 :** Comment “Afficher le niveau de jeu” ?

**Écrire**(“mon niveau est :”)  
**SELON** Niveau de jeu **FAIRE**

```

    SI naif FAIRE
        Écrire("naïf")
    SINON SI rapide FAIRE
        Écrire("rapide")
    SINON SI distrait FAIRE
        Écrire("distract")
    SINON SI expert FAIRE
        Écrire("expert")
    FIN SELON

```

**R3 :** Comment "Jouer le tour de l'utilisateur" ?

```

    Écrire("Combien d'allumettes prenez-vous ?")
    Lire(Nal_tour)
    TANT QUE Nal_tour <= 0 OU Nal_tour > 3 OU (Nbr_al - Nal_tour < 0) FAIRE
        Contrôler le nombre d'allumettes à retirer Nal_tour : in out Entier, Nbr_al : in
Entier
    FIN TANT QUE
    Nbr_al <= Nbr_al - Nal_tour

```

**R3 :** Comment "Calculer le nombre d'allumettes à retirer au niveau demandé" ?

```

    SELON Niveau jeu FAIRE
        SI naif FAIRE
            Jouer le tour de l'ordinateur en niveau naïf Nbr_al : in out Entier,
            Nal_tour : in out Entier
        SINON SI rapide FAIRE
            Jouer le tour de l'ordinateur en niveau rapide Nbr_al : in out Entier
            Nal_tour : in out Entier
        SINON SI distrait FAIRE
            Jouer le tour de l'ordinateur en niveau distrait Nbr_al : in out Entier
            Nal_tour : in out Entier
        SINON FAIRE
            Jouer le tour de l'ordinateur en niveau expert Nbr_al : in out Entier
            Nal_tour : in out Entier
    FIN SELON

```

**R3 :** Comment "Afficher les allumettes par paquet de maximum 5" ?

```

    POUR i DE 1 à Nbr_al FAIRE

```

```

    SI i=5 ou i=10 FAIRE
        Ecrire('|')
        Ecrire(' ')
    SINON FAIRE
        Ecrire('|')
    FIN SI
FIN POUR

```

**R4:** Comment “Contrôler le nombre d’allumettes à retirer ” ?

```

SI Nal_tour<=0 FAIRE
    Écrire(“Arbitre : Il faut prendre au moins une allumette.”)
    Écrire(“Combien d'allumettes prenez-vous ?”)
    Lire(Nal_tour)
SINON SI Nal_tour>3 FAIRE
    Écrire(“Arbitre : Il est interdit de prendre plus de 3 allumettes.”)
    Écrire(“Combien d'allumettes prenez-vous ?”)
    Lire(Nal_tour)
SINON
    Contrôler qu'il reste assez d'allumettes pour la demande de l'utilisateur Nbr_al :
in Entier, Nal_tour : out Entier
FIN SI

```

**R4 :** Comment “Calculer le nombre d’allumettes à retirer au niveau naïf” ?

```

SI Nbr_al>=3 FAIRE
    Nal_tour <= Alea(1,3)
SINON FAIRE
    Nal_tour <=(1,Nbr_al)
FIN SI

```

**R4 :** Comment “Calculer le nombre d’allumettes à retirer au niveau distrait” ?

```

Nal_tour <= alea(1,3)
TANT QUE Nbr_al-Nal_tour<0 FAIRE
    Contrôler qu'il reste assez d'allumettes pour la demande de l'ordinateur Nbr_al :
in Entier, Nal_tour : out Entier
FIN TANT QUE

```

**R4 :** Comment “Calculer le nombre d’allumettes à retirer au niveau rapide” ?

**SI** Nbr\_al>=3 **FAIRE**

Nal\_tour <= 3

**SI** Nbr\_al=2 **FAIRE**

Nal\_tour <= 2

**SINON FAIRE**

Nal\_tour <= 1

**FIN SI**

**R4 :** Comment “Calculer le nombre d’allumettes à retirer au niveau expert” ?

**SI** (Nbr\_al-1) mod 4=0 **FAIRE**

Nal\_tour <= alea(1,3)

Contrôler qu’il reste assez d’allumettes pour la demande de  
l’ordinateur Nbr\_al : **in Entier**, Nal\_tour : **out Entier**

**SINON SI** Nbr\_al<=4 **FAIRE**

Nal\_tour <= Nbr\_al -1

**SINON FAIRE**

Choisir un nombre d’allumettes de sorte que le reste soit un multiple de 4  
+1 Nal\_tour : **in out Entier**, Nbr\_al : **in out Entier**

**FIN SI**

**R5:** Comment “Contrôler qu’il reste assez d’allumettes pour la demande de l’utilisateur” ?

**SI** Nbr\_al=2 **FAIRE**

**Ecrire**(Arbitre : Il reste deux allumettes.)

**Lire**(Nal\_tour)

**SINON FAIRE**

**Ecrire**(Arbitre : Il reste une seule allumette.)

**Lire**(Nal\_tour)

**FIN SI**

**R5: Comment** “Contrôler qu’il reste assez d’allumettes pour la demande de l’ordinateur” ?

**SI** Nbr\_al=2 **FAIRE**

**Ecrire**(Arbitre : Il reste deux allumettes.)

    Nal\_tour <= alea(1,2)

**SINON FAIRE**

**Ecrire**(Arbitre : Il reste une seule allumette.)

    Nal\_tour <= 1

**FIN SI**

**R5: Comment** “Choisir un nombre d’allumettes de sorte que le reste soit un multiple de 4 +1” ?

Nal\_tour <= 1

**TANT QUE NON ((Nbr\_al-Nal\_tour-1) mod 4=0) FAIRE**

    Nal\_tour <= Nal\_tour + 1

**FIN TANT QUE**

## 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	A		
	Ri : Comment "... une action complexe ..." ? des actions combinées avec des structures de controle			
	Rj : ...			
	Verbe à l'infinitif pour les actions complexes	A		
	Nom ou équivalent pour expressions complexes	P		
	Tous les Ri sont écrits contre la marge et espacés	A		
	Les flots de données sont définis	A		
	Une seule structure de contrôle par raffinage	A		
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	P		
	Le raffinage d'une action décrit complètement cette action	A		
	Le raffinage d'une action ne décrit que cette action	P		
	Les flots de données sont cohérents	I		
	Pas de structure de contrôle déguisée	P		
	Qualité des actions complexes	I		

## Remarques diverses

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

## Évaluation du code

		<b>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".</b>	
<b>Commentaire</b>	<b>Etudiant (O/N)</b>	<b>Règle</b>	<b>Enseignant (O/N)</b>
		Le programme ne doit pas contenir d'erreurs de compilation.	
		Le programme doit compiler sans messages d'avertissement.	
		Le code doit être bien indenté.	
		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.	
		On doit utiliser les structures de contrôle adaptées (Si/Selon/TantQue/Répéter/Pour)	
		Utiliser des constantes nommées plutôt que des constantes littérales.	
		Les raffinages doivent être respectés dans le programme.	



		Les actions complexes doivent apparaître sous forme de commentaires placés AVANT les instructions correspondantes, avec la même indentation	
		Une ligne blanche doit séparer les principales actions complexes	
		Le rôle des variables doit être explicité à leur déclaration (commentaire).	