PIM : Mini-projet 1

Labiade Omar

Raffinages

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

R0 : jouer le jeu des 13 allumettes avec l'utilisateur

R1 : comment <jouer le jeu des 13 allumettes avec l'utilisateur>
Initialiser le jeu --Initialisation du jeu avec le niveau de l'ordinateur et le tour de l'utilisateur
Niveau:**OUT caractère**
Tour_utilisateur:**OUT booléen**
Nombre_allumettes_restantes:**OUT entier**

Tantque Nombre_allumettes_restantes > 0 **Faire**
Afficher l'état du jeu Nombre_allumettes_restantes:**IN entier**
Si Tour_utilisateur
Traiter la tour de l'utilisateur
Nombre_allumettes_a_retirer:**OUT entier**
Nombre_allumettes_restantes:**IN OUT entier**
Prise_utilisateur_valide = **OUT booléen**

Sinon
Jouer un tour Nombre_a_prendre:**OUT entier**
Niveau:**IN caractère**
Nombre_allumettes_restantes:**IN OUT entier**
Prise_ordinateur_valide:**OUT booléen**

Finsi
FinTQ
Afficher le gagnant -- Afficher le gagnant en fonction du dernier tour (utilisateur ou ordinateur)
Tour_utilisateur:**IN booléen**

R2 : comment <Initialiser le jeu>
Ecrire (" choisir Niveau de l'ordinateur: (n:naïf,d:distrait,r:rapide ou e:expert ?")
lire(Niveau)
Selon Niveau **Faire**
'n', 'N' = > **Ecrire**(Niveau naif)
'd', 'D' = > **Ecrire**(Niveau distrait)
'r', 'R' = > **Ecrire**(Niveau rapide)

```
            Autres  = > Ecrire(Niveau expert)
        Ecrire("Est-ce que vous commencez (o/n) ?")
        Lire(reponse)
        Tour_utilisateur ← reponse = 'o' ou reponse = 'O'

        Nombre_allumettes_restantes←13 ---- Le jeu commence avec 13 allumettes
        FinSi

R2 : Comment <Afficher l'état du jeu>
        --Afficher les allumettes restantes, avec un formatage en groupes de 5
        Pour i de 1 a 3
            Pour j de 1  a Nombre_allumettes_restantes
                Si j mod5 = 0 Alors
                    Ecrire("|    ") --laisser 3 espaces
                Sinon
                    Ecrire("| ")   --laisser 1 espaces
                FinPour
            Retourner à la ligne
            FinPour
        FinPour

R2 : comment<jouer un tour>
        Selon Niveau Faire
            'n' ou 'N' = >joue naïve          Nombre_a_prendre:OUT entier
                                              Nombre_allumettes_restantes:IN OUT entier
            'd ou 'D' = >joue distrait
                                              Nombre_a_prendre:OUT entier
                                              Nombre_allumettes_restantes:IN OUT entier
                                              Prise_ordinateur_valide:OUT  booléen
            'r' ou 'R' = >joue rapide
                                              Nombre_a_prendre:OUT entier
                                              Nombre_allumettes_restantes:IN OUT entier
            Autres = > joue expert
                                              Nombre_a_prendre:OUT entier
                                              Nombre_allumettes_restantes:IN OUT entier

R2 : comment<Traiter la tour de l'utilisateur>
        Repeter
            Ecrire("Combien d'allumettes prenez-vous ?")
            Lire(Nombre_allumettes_a_retirer)
            Traiter le choix de l'utilisateur  Nombre_allumettes_a_retirer:IN entier
                                               Nombre_allumettes_restantes:IN OUT entier
                                               Prise_utilisateur_valide = OUT booléen
        JusquÀ Prise_utilisateur_valide

R2 : comment<Afficher le gagnant>
        Si Tour_utilisateur
            Ecrire("bravo,vous avez gagné  ")
        Sinon
            Ecrire("J'ai gagné ")
        FinSi
            FinSi
```

```
R3 : comment<Traiter le choix de l'utilisateur>
        -- Vérifier si la prise d'allumettes est valide
        Si Nombre_allumettes_a_retirer< = 3 et Nombre_allumettes_a_retirer> = 1
                Si Nombre_allumettes_restantes<Nombre_allumettes_a_retirer
                        Ecrire("Arbitre : Il reste seulement ", Nombre_allumettes_restantes, " allumettes.")
                        Prise_utilisateur_valide = Faux
                Sinon
                        Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_allumettes_a_retirer
                        Prise_utilisateur_valide = Vrai
                FinSi
        Sinon Si  Nombre_allumettes_a_retirer = 0
                        Ecrire("Arbitre : Il faut prendre au moins une allumette.")

        Sinon
                Ecrire("Arbitre : Il est interdit de prendre plus de 3 allumettes")
                Prise_utilisateur_valide = Faux
        FinSi

R3 : comment<joue naive>
        Si Nombre_allumettes_restantes = 1
                Nombre_a_prendre = 1
                Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre

        SinonSi Nombre_allumettes_restantes = 2
                Nombre_a_prendre = hasard(1,2)
                Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre
        Sinon
                Nombre_a_prendre = hasard(1,3)
                Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre
        FinSi
        Ecrire("Je prends",Nombre_a_prendre," allumettes.")

R3 : comment<joue distrait>
        Repeter
                Nombre_a_prendre = hasard(1,3) –hasard(n,m) renvoi un entier aléatoire entre n et m avec n< = m
                Si Nombre_allumettes_restantes> = Nombre_a_prendre
                        Ecrire("Je prends",Nombre_a_prendre," allumettes.")
                        Prise_ordinateur_valide = Vrai
                Sinon Si
                        Ecrire("Je prends",Nombre_a_prendre," allumettes.")
                        Ecrire("Arbitre : Il reste seulement ",Nombre_allumettes_restantes," allumettes.")
                        Prise_ordinateur_valide = Faux
                FinSi
        JusquÀ Prise_ordinateur_valide
        Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre
        Ecrire('Je prends',Nombre_a_prendre,' allumettes.')

R3  : comment <joue rapide>
        Si Nombre_allumettes_restantes = 1 ou Nombre_allumettes_restantes = 2
```

```
            Nombre_a_prendre = 1
            Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre
        SinonSi Nombre_allumettes_restantes = 3
            Nombre_a_prendre = 2
            Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre
        Sinon
            Nombre_a_prendre = 3
            Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre
        FinSi
        Ecrire('Je prends',Nombre_a_prendre,' allumettes.')

R3 : comment <joue expert>
        Si Nombre_allumettes_restantes mod4 = 1
            Nombre_a_prendre = hasard(1,3)
            Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre
        SinonSi Nombre_allumettes_restantes mod4 = 0
            Nombre_a_prendre = 3
            Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre
        SinonSi Nombre_allumettes_restantes mod4 = 2
            Nombre_a_prendre = 1
            Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre
        Sinon
            Nombre_a_prendre = 2
            Nombre_allumettes_restantes←Nombre_allumettes_restantes-Nombre_a_prendre
        FinSi
        Ecrire('Je prends',Nombre_a_prendre,' allumettes.')
```

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 Ri : Comment "... une action complexe ..." ? des actions combinées avec des structures de controle Rj : ...	A	.	
	Verbe à l'infinitif pour les actions complexes	A	.	
	Nom ou équivalent pour expressions complexes	A	.	
	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	P	.	
	Pas trop d'actions dans un raffinage (moins de 6)	P	.	
	Bonne présentation des structures de contrôle	A	.	
Fond (D21-D22)	Le vocabulaire est précis	A	.	
	Le raffinage d'une action décrit complètement cette action	A	.	
	Le raffinage d'une action ne décrit que cette action	A	.	
	Les flots de données sont cohérents	A	.	
	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)
		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).	