UE Informatique
Étape 4 - Les pioches des joueurs

Olivier Beaudoux (olivier.beaudoux@eseo.fr)

Semestre 4

Table des matières

1 Les pioches de planchettes **2**
1.1 Les exemplaires 2
1.2 La pioche 4
1.3 Représentation graphique de la pioche 4

2 Les joueurs **6**

1 Les pioches de planchettes

Le type abstrait *Pioche* est en toute logique une liste de planchettes. Cependant, le type *Exemplaires* est introduit afin d'éviter des doublons dans une telle liste [1]. La figure 1 fournit un modèle des 3 types abstraits *Pioche*, *Exemplaires* et *Planchette* à considérer dans cette étape.

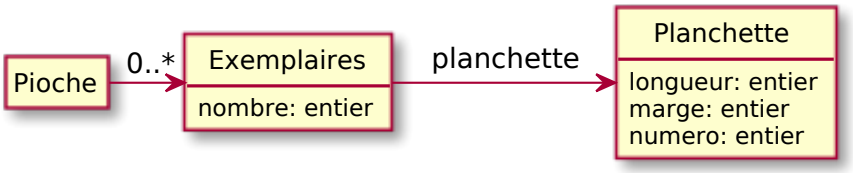

FIGURE 1 – Modèle des exemplaires

Une pioche est constituée de 0 à plusieurs exemplaires de planchettes. Le type *Exemplaires* précise le nombre d'exemplaires d'une *planchette* donnée [2]. Le type abstrait *Exemplaires* est abordé dans la section 1.1 et le type *Pioche* dans la section 1.2.

1.1 Les exemplaires

Introduction

Les pioches des joueurs sont composées de 27 planchettes dont les caractéristiques et les nombres sont précisés dans le tableau 1 de la notice 1.

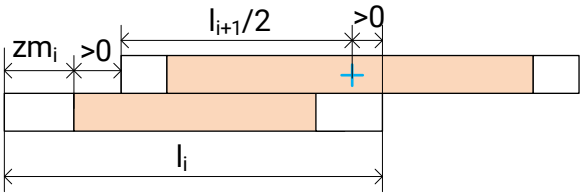

FIGURE 2 – Contraintes de dimensionnement des planchettes

Une telle répartition n'est cependant pas établie par hasard. En effet, comme l'illustre la figure 2, deux planchettes successives doivent avoir un premier espace de longueur positive pour respecter la règle de pose (pas de pose sur la marge de gauche dans l'exemple de la figure 2), ainsi qu'un seconde espace de longueur positive pour que la planchette supérieure reste à l'équilibre. On peut donc en déduire que :

$$zm_i + \frac{l_{i+1}}{2} < l_i$$

soit :

$$l_{i+1} \leq 2(l_i - zm_i)$$

1. La type *Planchette* représente en fait un modèle de planchette, lequel est identifié de manière unique par son numéro.
2. ou plutôt : d'un modèle de planchette donné.

d'où :

$$l_{i+1} - l_i \leq zp_i$$

Cette relation doit être vérifiée pour tout i, sinon il devient très facile au joueur j d'imposer au joueur $j+1$ une planchette qu'il sera impossible d'empiler dans déséquilibrer la pile. En conclusion, **l'écart maximal entre les longueurs des planchettes doit rester inférieur ou égal à la longueur minimale de leur zone de pose**. Les dimensions proposées pour les planchettes respectent toutes cette contrainte.

Spécification

La spécification du type abstrait *Exemplaires* est la suivante :

cree(planchette, nombre) : retourne une structure de données qui représente le *nombre* d'exemplaires d'une *planchette*.

planchette(exemplaires) : retourne la *planchette* considérée par les *exemplaires* en question.

nombre(exemplaires, valeur=None) : modifie le *nombre* d'occurrences des *exemplaires* considérés en lui donnant la nouvelle *valeur*, dans le cas où cette valeur est définie ; retourne le *nombre* d'occurrences des *exemplaires* considérés.

retireUn(exemplaires) : retire un exemplaire des *exemplaires* considérés et renvoie le nombre d'exemplaires restants.

versChaine(exemplaires) : retourne une chaîne de caractères représentant textuellement les *exemplaires* considérés. Le format est le suivant : « nxp » où n est le nombre d'exemplaires de la planchette numéro p. Par exemple, la chaîne « 2x464 » représente 2 exemplaires de la planchette 464.

Travail à faire

Étape 4.1.1

- Télécharger depuis le campus l'archive « Etape-4.zip » et en extraire les fichiers dans le répertoire « S4-Informatique ».
- Copier les fichiers de l'étape 3, hormis ceux contenant les modules de test, dans le répertoire « S4-Informatique/Etape-4 ».
- Compléter le fichier « Exemplaires.py ».
- Tester ce code en exécutant le module « TestExemplaires.py ».

Bien comprendre :

- **le principe des « exemplaires »,**
- **et notamment le rôle de la propriété modifiable nombre du type Exemplaires.**

1.2 La pioche

Cette section vise à implémenter les fonctions de base du type abstrait *Pioche*.

Spécification

La spécification du type abstrait *Pioche* est la suivante :

***cree()* :** retourne une liste d'exemplaires correspondant à une pioche complète, c'est-à-dire contenant 27 planchettes réparties comme précisé dans l'étape 1.

***nombrePlanchettes(pioche)* :** retourne le nombre total de planchettes présentes dans la *pioche*.

***versChaine(pioche)* :** retourne une chaîne de caractères représentant textuellement le contenu de la pioche. Par exemple, la représentation textuelle d'une pioche complète est : « `1x181 2x262 3x343 2x282 3x363 4x444 3x383 4x464 5x545` ».

***recherche(pioche, numero)* :** recherche dans la pioche la planchette donnée par son *numero*. Retourne l'indice correspondant si la planchette est trouvée, -1 sinon.

***contient(pioche, numero)* :** retourne vrai si la planchette donnée par son *numero* est présente dans la *pioche*, faux sinon. Équivalent à `recherche(pioche, numero) == -1`.

***retire(pioche, numero)* :** retire de la *pioche* un exemplaire la planchette donnée par son *numero*. Si la planchette n'est pas présente, aucune action n'est effectuée. Si le nombre d'exemplaires après retrait atteint 0, les exemplaires correspondant sont supprimés de la liste.

Travail à faire

Étape 4.1.2

- Compléter le fichier « Pioche.py ».
- Tester ce code en exécutant le module « TestPioche.py ».

Bien comprendre :

- **la gestion des exemplaires de la liste d'exemplaires que constitue la pioche,**
- **et notamment la gestion du retrait d'un exemplaire.**

1.3 Représentation graphique de la pioche

Introduction

Le module *VuePioche* gère la représentation graphique des pioches des joueurs. La figure 3 illustre l'agencement de cette représentation graphique, où L et H sont respectivement la largeur et la hauteur de la zone de dessin (ou « toile ») de la fenêtre de l'application. L'affichage de la pioche s'effectue à gauche ou à droite de la toile, en fonction du joueur. Les exemplaires de la pioche sont affichés de haut en bas, de exemplaires de longueur 10 cm à ceux de longueur 14 cm. Bien noter la symétrie dans les affichages de gauche et de droite.

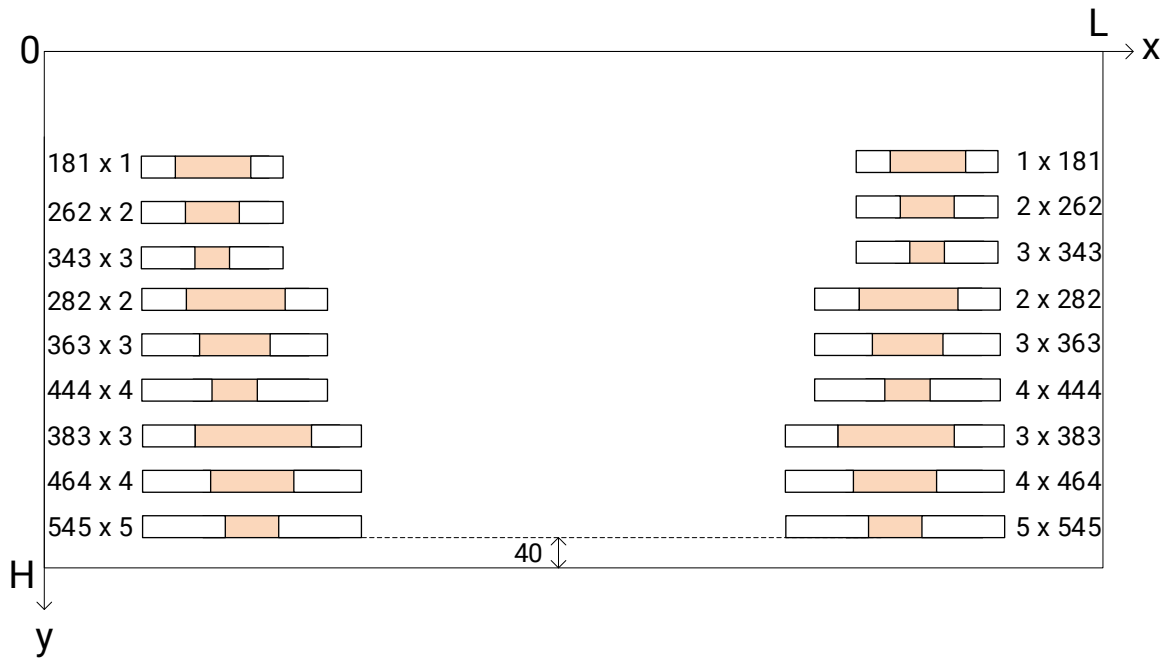

FIGURE 3 – Représentation graphique des pioches

Spécification

Le module *VuePioche* ne contient qu'une fonction définie comme suit :

dessine(fenetre, pioche, gauche) : dessine la *pioche* à gauche (si *gauche* vaut vrai) ou à droite (si *gauche* vaut faux) de la toile de la *fenetre*.

Travail à faire

Étape 4.1.3

- Compléter le fichier « VuePioche.py ». La fonction *dessine* devra définir différentes variables locales afin de rendre le code lisible. Bien entendu, le module *VuePlanchette* est à utiliser dans cette fonction.
- Tester ce code en exécutant le module « TestVuePioche.py ».

Bien comprendre :

- l'ordre de l'affichage de chacune des planchettes ;
- la symétrie de l'affichage gauche et droit.

2 Les joueurs

Chaque joueur est défini par son numéro (1 ou 2) est possède sa propre pioche de planchettes. Le type abstrait Joueur est donc relativement simple.

Spécification

La spécification du type abstrait *Joueur* est la suivante :

cree(numero) : retourne une structure de données qui représente un joueur par son *numero* et sa pioche complète de 27 planchettes.

numero(joueur) : retourne le numéro du *joueur* (1 ou 2).

nom(joueur) : retourne le nom du *joueur* représenté par la chaîne de caractères « `Joueur` n » où n est le numéro du joueur.

pioche(joueur) : retourne la pioche du *joueur*.

Travail à faire

Étape 4.2

- Compléter le fichier « Joueur.py ».
- Tester ce code en exécutant le module « TestJoueur.py ».

L'étape 4 est terminée. Il est maintenant possible de réaliser enfin le jeu : c'est le but de la dernière étape 5.