

Feuille de travaux pratiques

Des dés

Exercice 1 (Dés)

On veut simuler une suite de lancers de dés aléatoires, réalisés de la façon suivante : on stocke un score qu'on augmente en fonction de chaque lancer. On commence avec un score de 0. On lance un dé et on additionne le résultat obtenu au score. Si on fait un 5 ou un 6, on relance le dé pour augmenter son score, sinon on s'arrête.

1. Écrire une fonction `lancer_de` de type `unit -> int` qui renvoie un entier aléatoire entre 1 et 6. On pourra utiliser la bibliothèque `Random`¹ de la façon suivante : on initialise la graine avec `Random.self_init ()`, on tire un entier aléatoire entre 0 et n avec `Random.int n`.
2. Écrire une fonction récursive `jouer`, de type `unit -> int` qui permet de jouer au jeu décrit ci-dessus et renvoie le score atteint. On prendra soin d'afficher après chaque lancer de dé le résultat du dé et le score courant.
3. Écrire un programme permettant de jouer plusieurs tours de jeu. On initialisera le générateur aléatoire une seule fois au début du programme.
4. Tester ce programme en s'assurant d'avoir couvert tous les cas possibles.

Exercice 2

On veut jouer au jeu de fléchettes du 501, adapté aux dés. Chaque joueur part avec 501 points, son objectif étant d'arriver à 0 exactement. Chacun son tour, chaque joueur tire trois dès, et, si le total de ces dès est plus petit que son score courant, il le retire à son score courant. Sinon, il garde son score courant. Le premier joueur à atteindre 0 a gagné.

1. Est-ce qu'un des joueurs peut se retrouver bloqué ? Et si oui, comment ?
2. Écrire une fonction `tour_501` de type `int -> int` qui prend en entrée le score actuel d'un joueur, et simule un tour de jeu pour ce joueur, en retournant le score après ce tour. Vous vous assurerez que la fonction termine toujours.
3. Écrire une fonction `jouer_501` de type `int -> unit` qui prend en entrée le nombre de joueurs, et simule une partie complète de 501, en imprimant le joueur gagnant avant de terminer.

1. Voir <https://caml.inria.fr/pub/docs/manual-ocaml/libref/Random.html>