Université de Lille
IUT de Lille

Préparation au TD n°4

R1.01 - Initiation au développement
BUT Informatique
Yann.Secq@univ-lille.fr
Fabien.Delecroix@univ-lille.fr
2022–2023

Objectifs : Consolidation des boucles à compteur. Initiation aux boucles à événement.

Exercice 1 : Potato pirates : le retour

La semaine passée, vous avez pris en main les règles de base du jeu potato pirates et travaillé plusieurs notions fondamentales étudiées depuis le début du cours :

- la notion de variable et d'affectation (plus précisément d'accumulation et même de décrémentation),
- la notion de répétition avec les boucles à compteur (`répéter X fois`),
- la notion de condition (expression booléene),
- la notion d'alternative (`si (condition) alors ... sinon ...`)
- ... et bien sûr d'algorithme (mais assez court, pas plus de 3 cartes !).

Cette semaine, nous allons ajouter la notion de boucle à évènement (`tantque`) et quelques cartes de ludification (les interruptions pour bloquer une attaque, voler ou pirater un bateau ennemi !).

Le but du jeu change aussi légèrement, maintenant, l'objectif est de réunir toute la famille pirate (4 cartes). Ainsi, lorsque vous tirez une carte de la famille pirate, vous devez la poser à côté de vos bateaux (ce qui risque de vous transformer en cible ;)). Le but est maintenant de couler tous les bateaux d'un adversaire ayant une ou des cartes de la famille pirate afin de récupérer l'ensemble des cartes de sa main et de famille pirate.

Afin d'avoir des algorithmes un peu plus intéressants (ou violents selon le point de vue), nous nous autoriserons des algorithmes pouvant contenir jusqu'à 5 cartes au maximum ...

Exercice 2 : Ça fait répétition

Voici un programme utilisant une boucle `while`.

```
1  class ProgrammeWhile extends Program{
2    void algorithm(){
3      int indice = 0;
4      String chaine = "Une phrase composée de plusieurs mots";
5      while (indice<length(chaine)/2){
6        print(charAt(chaine,indice));
7        indice=indice+1;
8      }
9    }
10 }
```

1. Quelle est la valeur de `indice` au début du premier tour de la boucle `while` ?
2. Quelle est la valeur de `indice` à la fin du dernier tour de la boucle `while` ?
3. Combien de fois la fonction `print` sera-t-elle appelée lors de l'exécution de ce programme ?
4. Qu'affiche précisément ce programme ?
5. Modifier ce programme pour qu'il s'arrête dès qu'il a rencontré un espace.

Exercice 3 : Ça tourne en boucle

Lors du pré-TD précédent, on était passé du programme de gauche à celui de droite.

```
class ProgrammeRepetitif extends Program{
  void algorithm(){
    int i=1;
    println("affichage numéro "+i);
    i=i+1;
    println("affichage numéro "+i);
    i=i+1;
    println("affichage numéro "+i);
    i=i+1;
    println("affichage numéro "+i);
    i=i+1;
    println("affichage numéro "+i);
  }
}
```

```
class AvecBoucleFor extends Program{
  void algorithm(){
    for (int i=1;i<=5;i=i+1){
      println("affichage numéro "+i);
    }
  }
}
```

Écrivez un programme produisant le même résultat mais en utilisant cette fois une boucle while.

Exercice 4 : Contrôle de saisie : ah ces utilisateurs/utilisatrices !

Une tâche récurrente lorsque l'on écrit des programmes nécessitant des saisies consiste à vérifier que l'utilisateur ou l'utilisatrice a bien fait ce que l'on a demandé ... ce qui n'est pas toujours le cas ! Dans le programme ci-dessous, on souhaite saisir un prénom et l'on considère qu'une chaîne non vide peut convenir (discutable, mais soyons simples ;)). Complétez le code ci-dessous afin d'avoir un programme valide.

```
class ControleDeSaisie extends Program{
  void algorithm(){
    boolean saisieInvalide = ...; // à) compléter
    String prenom;
    while (saisieInvalide){
      print("Veuillez entrer votre prénom : ");
      // à compléter
    }
  }
}
```

Serait-il possible de se passer de la variable booléenne ?