



MARHERAOUI BERRAHOU Amin | 102

IBOUDA Yasser | 111

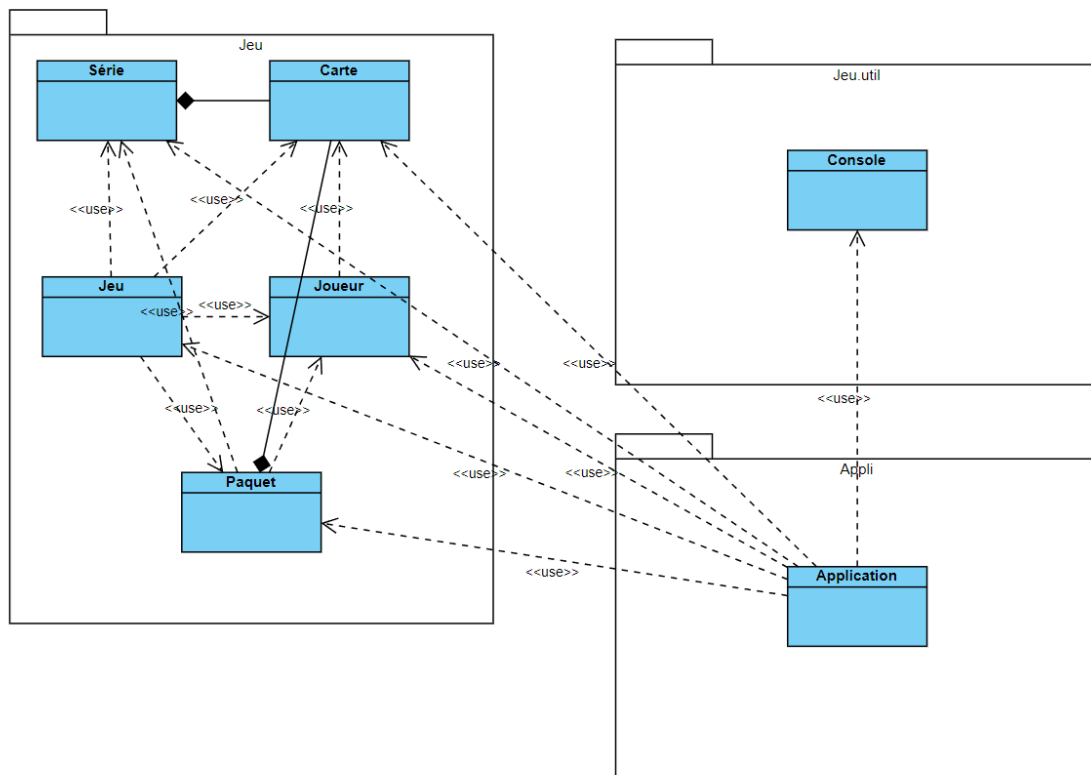
Table des matières

Introduction	3
Diagramme UML.....	3
Tests Unitaires	4
Code du projet	5
Config.txt	6
Joueur.java	6
Carte.java.....	8
Série.java	9
Paquet.java.....	10
Jeu.java	12
Application.java	16
Console.java	19
Conclusion.....	21

Introduction

Ce projet a pour but de produire un jeu de cartes qui s'appelle le jeu du 6 qui prend. Le jeu de cartes est composé de 104 cartes ayant chacune un nombre de têtes de bœuf basé sur le numéro de la carte. Notre but sera donc de programmer ce jeu et de permettre un nombre de joueurs compris entre 2 et 10. Chaque joueur a 10 cartes et on pose 4 cartes sur la table qui vont former les 4 séries dans lesquelles les joueurs pourront poser leur carte tant qu'elle est supérieure à la série.

Diagramme UML des classes



Code java des tests unitaires

```
}  
  
import static org.junit.Assert.*;  
  
import java.util.ArrayList;  
  
import org.junit.Test;  
  
import Jeu.Jeu;  
import Jeu.Joueur;  
import Jeu.Paquet;  
  
public class Test6quiprend {  
  
    @Test  
    public void test() { //Test si le paquet contient bien 104 cartes  
        Paquet test = new Paquet();  
        assertEquals(104, test.getPaquet().size());  
    }  
  
    @Test  
    public void test1() { // Test si il y a 4 séries  
        Jeu test = new Jeu();  
        Paquet p = new Paquet();  
        test.creer_et_distrib_serie(p);  
        assertEquals(4, test.getTab_serie().size());  
    }  
  
    @Test  
    public void test2() { // Test si la fonction lire lit et crée bien les  
joueurs qui sont dans le fichier config.txt  
        Jeu test = new Jeu();  
        ArrayList<Joueur> lecture = test.lire();  
        assertEquals(2, lecture.size());  
    }  
  
    @Test  
    public void test3() { //Test si les joueurs ont bien 10 cartes  
        Paquet p = new Paquet();  
        Jeu jeu = new Jeu();  
        jeu.lire();  
        jeu.distribuer(p);  
        assertEquals(10, jeu.getTab_j().get(0).getCarte_joueur().size());  
    }  
  
    @Test  
    public void test4() { //Test si chaque série contient 1 carte à sa création  
à l'aide de la fonction ajouter_serie  
        Jeu test = new Jeu();  
        Paquet p = new Paquet();  
        test.creer_et_distrib_serie(p);  
        assertEquals(1, test.getTab_serie().get(0).getSerie_cartes().size());  
        assertEquals(1, test.getTab_serie().get(1).getSerie_cartes().size());  
        assertEquals(1, test.getTab_serie().get(2).getSerie_cartes().size());  
        assertEquals(1, test.getTab_serie().get(3).getSerie_cartes().size());  
    }  
  
    @Test
```

```

    public void test5() { // Test si la fonction retirer peut retirer une carte
    du paquet
        Jeu jeu = new Jeu();
        Paquet p = new Paquet();
        p.retirer();
        assertEquals(103, p.getPaquet().size());
    }

    @Test
    public void test6() { // Test si la fonction ajouter ajoute bien 10 cartes
    aux joueurs
        Jeu jeu = new Jeu();
        Paquet p = new Paquet();
        jeu.lire();
        p.ajouter(jeu.getTab_j().get(0));
        assertEquals(10, jeu.getTab_j().get(0).getCarte_joueur().size());
    }

    @Test
    public void test7() { // Test si la fonction retirer peut retirer une carte
    du paquet
        Jeu jeu = new Jeu();
        Paquet p = new Paquet();
        p.retirer();
        assertEquals(103, p.getPaquet().size());
    }

    @Test
    public void test8() { // Test si la fonction verif_nbjeueur fonctionne
        Jeu jeu = new Jeu();
        jeu.lire();
        boolean nbjeueur = jeu.verif_nbjeueur(jeu.getTab_j());
        assertEquals(true, nbjeueur);
    }
}

```

Code du programme

Config.txt

Yasser

Amin

Joueur.java

```
package Jeu;
import java.util.ArrayList;

public class Joueur {
    private String prénom;
    private static ArrayList<Carte> cartes_joueur = new ArrayList<Carte>();

    public Joueur() {

    }

    public Joueur(String string) {
        this.prénom = string;
    }

    public String getPrénom() {
        return prénom;
    }

    public void setPrénom(String prénom) {
        this.prénom = prénom;
    }

    @Override
    public String toString() {
        for(int i = 0; i < cartes_joueur.size(); ++i) {
            if(i < cartes_joueur.size()-1)
                System.out.print(cartes_joueur.get(i)+", ");
            else
                System.out.print(cartes_joueur.get(i));
        }
        return "";
    }

    public void setCartes_joueur(ArrayList<Carte> deck) {
        this.cartes_joueur = deck;
    }
}
```

```
public static ArrayList<Carte> getCarte_joueur() {  
    return cartes_joueur;  
}  
  
}
```

Carte.java

```
package Jeu;

public class Carte {
    private int numéro_de_carte;
    private int tetes_de_boeuf;
    private static int numerocpt = 1;

    public Carte() {
        this.numéro_de_carte = numerocpt++;
        this.tetes_de_boeuf =
initialisation_tete_de_boeuf(this.numéro_de_carte);
    }

    public int initialisation_tete_de_boeuf(int numéro_de_carte) {// initialise
la valeur des tetes de boeuf de chaque carte
        int val_tetes_de_boeuf = 0;

        if (numéro_de_carte==55)
            val_tetes_de_boeuf = 7;

        else if (numéro_de_carte%10 == 5)
            val_tetes_de_boeuf = 2;

        else if (numéro_de_carte%10 == 0)
            val_tetes_de_boeuf = 3;

        else if (numéro_de_carte%11 == 0)
            val_tetes_de_boeuf = 5;

        else
            val_tetes_de_boeuf = 1;

        return val_tetes_de_boeuf;
    }

    public int getTetes_de_boeuf() {
        return tetes_de_boeuf;
    }

    @Override
    public String toString() {
        if (tetes_de_boeuf != 1) {
            return numéro_de_carte + " (" + tetes_de_boeuf+"");
        }
        else {
            return numéro_de_carte+"";
        }
    }
}
```


Série.java

```
package Jeu;
import java.util.ArrayList;

public class Série {
    private ArrayList<Carte> série_cartes = new ArrayList<Carte>();
    private int numéro_de_série;
    private static int numerocpt = 1;

    public Série() {
        this.numéro_de_série = numerocpt++;
    }

    public ArrayList<Carte> getSérie_cartes() {
        return série_cartes;
    }

    public void setSérie_cartes(ArrayList<Carte> carte_série) {
        this.série_cartes = carte_série;
    }

    @Override
    public String toString() {
        String chaine = " - Série n° "+numéro_de_série+" : ";
        for (int i = 0; i < série_cartes.size(); i++) {
            chaine += série_cartes.get(i);
        }
        return chaine;
    }
}
```

Paquet.java

```
package Jeu;

import java.util.ArrayList;
import java.util.Collections;

public class Paquet {

    private ArrayList<Carte> paquet = new ArrayList<Carte>();

    public Paquet() {

        for(int i = 1; i < 105; ++i) {

            Carte c = new Carte();

            paquet.add(c);

        }

    }

    public void melanger() { // mélange les cartes du paquet au hasard

        Collections.shuffle(paquet);

    }

    public void ajouter(Joueur j) { // ajoute 10 cartes à chaque joueur

        ArrayList<Carte> deck = new ArrayList<Carte>();

        for(int i = 0; i < 10; ++i)

            deck.add(retirer());

        j.setCartes_joueur(deck);

    }

    public Carte retirer() { // retire une carte du paquet

        Carte c = new Carte();
```

```

        c = paquet.get(0);

        paquet.remove(0);

        return c ;
    }

    public void ajouter_série(Série s) { // ajoute une carte à la série

        ArrayList<Carte> carte_série = new ArrayList<Carte>();

        carte_série.add(retirer());

        s.setSérie_cartes(carte_série);
    }

    @Override

    public String toString() {

        return "" + paquet;
    }

    public ArrayList<Carte> getPaquet() {

        return paquet;
    }
}

```

Jeu.java

```
package Jeu;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;
import java.util.List;
import java.util.Scanner;

public class Jeu {

    private static final int MAX = 10;
    private static final int MIN = 2;
    static Joueur[] tab = new Joueur[0];
    private ArrayList<Joueur> tab_j = new ArrayList<Joueur>(Arrays.asList(tab));
    private ArrayList<Série> tab_série = new ArrayList<Série>();

    public Jeu() {

    }

    public ArrayList<Joueur> getTab_j() {
        return tab_j;
    }

    public void setTab_j(ArrayList<Joueur> tab_j) {
        this.tab_j = tab_j;
    }
}
```

```

public ArrayList<Série> getTab_série() {
    return tab_série;
}

public void setTab_série(ArrayList<Série> tab_série) {
    this.tab_série = tab_série;
}

public ArrayList<Joueur> lire() { // lit et prend les joueurs du fichier
config.txt pour créer des objets Joueurs
    try {
        Scanner a = new Scanner(new FileInputStream("config.txt"));
        while(a.hasNext()) {
            Joueur j = new Joueur(a.next());
            tab_j.add(j);
        }
    }
    catch (FileNotFoundException e) {
        System.out.println("le fichier est introuvable");
    }
    return tab_j;
}

public void distribuer(Paquet p) { // distribue 10 cartes à chaque joueur à
l'aide de la fonction ajouter
    for(Joueur j1 : tab_j) {
        p.ajouter(j1);
    }
}

```

```
public void creer_et_distrib_serie(Paquet p) { // crée 4 séries et ajoute à
chacune 1 carte du paquet
```

```
    for(int i = 0; i < 4; ++i) {
        Série s = new Série();
        tab_serie.add(s);
        p.ajouter_serie(s);
    }
}
```

```
public boolean a_carte(Carte c, Joueur j) { //teste si le joueur possède la
carte entrée en argument
```

```
    boolean a_la_carte = false;
    for(Carte carte_j : j.getCarte_joueur()) {
        if(c == carte_j) {
            a_la_carte = true;
        }
        else {
            a_la_carte = false;
        }
    }
    return a_la_carte;
}
```

```
public boolean verif_nbjeuuer(ArrayList<Joueur> tab) { // vérifie que le
nombre de joueurs est respecté
```

```
    if(tab.size()>=MIN && tab.size()<=MAX) {
        return true;
    }
    else
        return false;
}
```

}

}

Application.java

```
package Appli;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;

import Jeu.Jeu;
import Jeu.Joueur;
import Jeu.Paquet;
import Jeu.Série;

import static Jeu.util.Console.clearScreen;
import static Jeu.util.Console.pause;

public class Application {

    public static void main(String[] args) {

        Jeu Jeu = new Jeu();

        Jeu.lire();

        if (Jeu.verif_nbjeoueur(Jeu.getTab_j()) == true) {

            System.out.print("Les " + Jeu.getTab_j().size() + " joueurs
sont");

            int i = 0;

            for (Joueur Joueur1 : Jeu.getTab_j()) {

                System.out.print(" " + Joueur1.getPrénom());

                if (i == Jeu.getTab_j().size()-2) {

                    System.out.print(" et");

                    ++i;
                }
            }
        }
    }
}
```



```

    }

    else if(i < Jeu.getTab_j().size()-2) {

        System.out.print(",");

        ++i;

    }

    //affichage de la premiere ligne (qui sont les joueurs)
}

System.out.println(". Merci de jouer à 6 qui prend ! ");


Paquet Paquet = new Paquet();

Paquet.melanger();

Paquet.ajouter(Jeu.getTab_j().get(0));

Jeu.distribuer(Paquet);

Jeu.creer_et_distrib_serie(Paquet);

for (int k=0; k<Jeu.getTab_j().size();++k) { //affichage du
joueur à qui c'est le tour de jouer

    System.out.println("A " +
Jeu.getTab_j().get(k).getPrénom() + " de jouer.");

    pause();

    for (Série Serie : Jeu.getTab_serie()) { // affichage
des séries

        System.out.println(Serie);

    }

    System.out.print("Vos cartes : ");

    System.out.println(Jeu.getTab_j().get(k)); //affichage
des cartes du joueur qui doit jouer

    Scanner carte_a_jouer = new Scanner(System.in); //entrée
clavier de la carte que le joueur veut poser

    System.out.println("Choisissez votre carte : ");

    clearScreen();

```

```
        }  
    }  
    else if(Jeu.verif_nbjeu(Jeu.getTab_j()) == false){ //vérification  
du nombre de joueurs  
        System.out.println("Problème de nombre de joueurs, il y a trop  
ou il n'y a pas assez de joueurs");  
    }  
}  
}
```

Console.java

```
package Jeu.util;

import java.io.IOException;

public class Console {
    private static final ProcessBuilder CLEANER_PROCESS;
    private static final ProcessBuilder PAUSE_PROCESS;

    private static final String MSG_PAUSE = "Appuyez sur une touche pour continuer..." + System.LineSeparator();

    static {
        if (System.console() != null) {
            String[] cdeClean;
            String[] cdePause;
            if (System.getProperty("os.name").contains("Windows")) {
                cdeClean = new String[] { "cmd", "/c", "cls" };
                cdePause = new String[] { "cmd", "/c", "pause" };
            }
            else {
                cdeClean = new String[] { "clear" };
                cdePause = new String[] { "read", "-n1", "-rsp",
MSG_PAUSE };
            }
            CLEANER_PROCESS = new ProcessBuilder(cdeClean).inheritIO();
            PAUSE_PROCESS = new ProcessBuilder(cdePause).inheritIO();
        } else
            CLEANER_PROCESS = PAUSE_PROCESS = null;
    }

    private static final String MSG_C = "<clearScreen>";

    public static void clearScreen() {
        if (CLEANER_PROCESS != null)
            try {
                CLEANER_PROCESS.start().waitFor();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            } catch (IOException e) {
                System.out.println(MSG_C);
            }
        else
            System.out.println(MSG_C);
    }

    private static final String MSG_P = "<pause>";

    public static void pause() {
        if (PAUSE_PROCESS != null)
            try {
                PAUSE_PROCESS.start().waitFor();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            } catch (IOException e) {
                System.out.println(MSG_P);
            }
    }
}
```

```
        }  
        else  
            System.out.println(MSG_P);  
    }  
    private Console() {  
    }  
}
```

Conclusion

Pour conclure, nous avons réalisé partiellement le projet car nous avons rencontré des difficultés au cours du projet, tel que comment faire poser une carte aux joueurs dans la série.

Nous n'avons pas non plus réussi à utiliser l'entrée clavier (le Scanner) afin de retirer la carte entrée du jeu de cartes du joueur.

En revanche nous avons pu donner à chaque joueur un jeu de cartes composé de 10 cartes, créé un paquet de 104 cartes avec pour chaque carte une valeur de tête de bœuf lui correspondant.

De plus, nous avons créé 4 séries contenant au départ 1 carte chacune.

Les points à améliorer sont :

- la lisibilité du programme causée par notre surutilisation des "getters" et "setters"
- l'optimisation du programme