

Question 15 :

```
import java.awt.*;
import javax.swing.*;

public class Jeu {
    private static final int TAILLE_CASE=30;
    private static final int NB_CASES=20;
    private Monde monde;
    private Avatar j1;
    private Avatar j2;

    public Jeu(Monde monde, Avatar j1, Avatar j2) {
        this.monde = monde;
        this.j1 = j1;
        this.j2 = j2;
    }

    private void init(int nbC, int nbA) {
        double rand;
        for (int i = 0; i < nbC; i++) {
            rand = Math.random();

            if (rand < 0.2) {
                monde.ajouterItem(new Glouton());
            }else {
                monde.ajouterItem(new Creature());
            }
        }

        for (int i = 0; i < nbA; i++) {
            rand = Math.random();
            if (rand < 0.05) {
                monde.ajouterItem(new PotionMagique());
            }else if (rand < 0.1) {
                Sac s = new Sac();
                while(rand < 0.5) {
                    s.ajouter(new Pomme());
                    rand = Math.random();
                }
                monde.ajouterItem(s);
            }else {
                monde.ajouterItem(new Pomme());
            }
        }
    }
}
```

```

        public void start(int nbCreatures, int nbAliments) throws
InterruptedException {
            init(nbCreatures,nbAliments);
            monde.repaint(); //Redessine le graphisme
            for (int i = 0; i < 5; i++) {
                Thread.sleep(1000); // Ralenti l'affichage
                j1.seDeplacer();
                j1.rencontrerVoisins();
                monde.repaint(); //Redessine le graphisme
                Thread.sleep(1000); // Ralenti l'affichage
                j2.seDeplacer();
                j2.rencontrerVoisins();
                monde.repaint(); //Redessine le graphisme
            }
            double scoreJ1 = j1.course();
            double scoreJ2 = j2.course();
            if (scoreJ1 > scoreJ2 ) {
                System.out.println(String.format("%s a gagne avec
%.2f points contre %.2f points", j1.getNom(), scoreJ1, scoreJ2));
            }else if (scoreJ2 > scoreJ1) {
                System.out.println(String.format("%s a gagne avec
%.2f points contre %.2f points", j2.getNom(), scoreJ2, scoreJ1));
            }else {
                System.out.println("EGALITE PARFAITE !!! Vous avez
tous les deux "+scoreJ1+" points !");
            }
        }

        public static void main(String[] args) throws
InterruptedException {
            //Creation denetre graphique et ses caracteristiques
            JFrame f = new JFrame();
            f.setLocationRelativeTo(null);
            f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            //Creation du monde (qui est un panneau)
            Monde m = new Monde(NB_CASES,TAILLE_CASE);
            f.setContentPane(m); //Ajout du monde a la fenetre
            f.pack(); // Adaptation de la fenetre au panneau
            f.setVisible(true);

            new Jeu(m,new Avatar("Faraan",50.0,m,new
Color(180,0,180)),new Avatar("Loann",60.0,m,new
Color(0,255,0))).start(5,10);
            Console.in.close();//Do not remove this line
        }
    }
}

```

J'ai modifié la classe Jeu pour que ce soit un objet, lorsqu'on crée un objet Jeu, on crée une nouvelle partie avec en paramètre, le monde dans lequel on joue ainsi que ses joueurs. La classe Jeu s'occupe ensuite de générer une nouvelle partie selon les règles de création constantes c'est à dire, un nombre fixé de Créatures, un nombre fixé d'Accessoires. Cependant chaque créature et chaque accessoire a une chance (également prédéfinie) d'être instancié. Cela crée donc une partie avec des objets et des créature, on utilise ensuite la méthode start pour lancer la partie, cette méthode permet de gérer la partie et d'afficher le gagnant a la fin de la partie .

Question 16 :

```
import java.awt.Color;
import java.awt.Graphics;

public class PotionMagique extends Acc implements Mangeable {
    private double force;

    public PotionMagique() {
        super("potion magique");
        force = Math.random()*100.0 - 25.0;
    }

    @Override
    public double getPoids() {
        if (force < 0) {
            return 0;
        }
        return force;
    }

    //Methods
    @Override
    public String toString() {
        return String.format("potion No %d %.2fkg",
this.getNumero(), this.getPoids());
    }

    @Override
    public void dessiner(Graphics g, Monde m) {
        int tc = m.getTailleCase();
        g.setColor(new Color(255,0,255));
        g.fillOval(getX()*tc, getY()*tc, tc, tc);
    }
}
```

Création d'un élément Mangeable, c'est une potion magique. Elle possède des propriétés étranges, en effet son poids peut être égal a 0, une potion avec un poids nul est un manque de chance, la créature qui la boit ne gagnera aucune force. Cependant la potion peut être très lourde, dans ce cas, si la créature la boit elle gagne énormément de force, mais si elle ne la bois pas, elle sera fortement handicapé lors de la course.

L'affichage de cet objet est un cercle de couleur violette.

Question 17 :

```
import java.awt.Color;
import java.awt.Graphics;

public class Glouton extends Creature {
    public Glouton() {
        super();
    }

    @Override
    public void manger() {
        double p = -1;
        while (p != getPoids()) {
            p = getPoids();
            mangerRec(getSac());
        }
    }

    @Override
    public void dessiner (Graphics g , Monde m) {
        int tc = m.getTailleCase();
        g.setColor(new Color(75 ,100 ,150)); //couleur courante
        //devient bleu
        g.fillRect(getX()*tc, getY()*tc, tc, tc); //carré plein
    }
}
```

La particularité des Gloutons c'est que la méthode manger a été redéfinie pour qu'ils mangent tout le contenu de leur inventaire d'un coup, il est donc rentable de lui donner pleins d'objets pendant la partie puisqu'ils finiront tous mangés. Leur affichage est un carré bleu plus foncé que les autres créatures .