

Faculté des Sciences et Techniques de Saint-Étienne
Laboratoire Hubert Curien, UMR CNRS 5516
Université Jean Monnet, Saint-Étienne

Programmation Orientée Objets (POO - Java)

Mathias Géry

Mathias.Gery@univ-st-etienne.fr

Projet : interface ICarte

```
package wargame;
import java.awt.Graphics;
public interface ICarte {
    Element getElement(Position pos);
    Position trouvePositionVide(); // Trouve aléatoirement une position vide
                                // sur la carte
    Position trouvePositionVide(Position pos); // Trouve une position vide choisie
                                                // aléatoirement parmi les 8 positions
                                                // adjacentes de pos
    Heros trouveHeros(); // Trouve aléatoirement un héros sur la carte
    Heros trouveHeros(Position pos); // Trouve un héros choisi aléatoirement
                                    // parmi les 8 positions adjacentes de pos
    boolean deplaceSoldat(Position pos, Soldat soldat);
    void mort(Soldat perso);
    boolean actionHeros(Position pos, Position pos2);
    void jouerSoldats(PanneauJeu pj);
    void toutDessiner(Graphics g);
}
```

Projet : interface IConfig

```
package wargame;
import java.awt.Color;
public interface IConfig {
    int LARGEUR_CARTE = 25; int HAUTEUR_CARTE = 15; // en nombre de cases
    int NB_PIX_CASE = 20;
    int POSITION_X = 100; int POSITION_Y = 50; // Position de la fenêtre
    int NB_HEROS = 6; int NB_MONSTRES = 15; int NB_OBSTACLES = 20;
    Color COULEUR_VIDE = Color.white, COULEUR_INCONNU = Color.lightGray;
    Color COULEUR_TEXTE = Color.black, COULEUR_MONSTRES = Color.black;
    Color COULEUR_HEROS = Color.red, COULEUR_HEROS_DEJA_JOUE = Color.pink;
    Color COULEUR_EAU = Color.blue, COULEUR_FORET = Color.green, COULEUR_ROCHER =
Color.gray;
}
```

Projet : classe Position

```
package wargame;

public class Position implements IConfig {
    private int x, y;
    Position(int x, int y) { this.x = x; this.y = y; }
    public int getX() { return x; }
    public int getY() { return y; }
    public void setX(int x) { this.x = x; }
    public void setY(int y) { this.y = y; }
    public boolean estValide() {
        if (x<0 || x>=LARGEUR_CARTE || y<0 || y>=HAUTEUR_CARTE) return false;
    else return true;
    }
    public String toString() { return "("+x+","+y+")"; }
    public boolean estVoisine(Position pos) {
        return ((Math.abs(x-pos.x)<=1) && (Math.abs(y-pos.y)<=1));
    }
}
```

Projet : classe abstraite Element

```
package wargame;  
public abstract class Element (...)
```

Projet : classe Obstacle

```
package wargame;
import java.awt.Color;
import java.awt.Graphics;
public class Obstacle extends Element {
    public enum TypeObstacle {
        ROCHER (COULEUR_ROCHER), FORET (COULEUR_FORET), EAU (COULEUR_EAU);
        private final Color COULEUR;
        TypeObstacle(Color couleur) { COULEUR = couleur; }
        public static TypeObstacle getObstacleAlea() {
            return values()[ (int) (Math.random()*values().length) ];
        }
    }
    private TypeObstacle TYPE;
    Obstacle(TypeObstacle type, Position pos) { TYPE = type; this.pos = pos; }
    public String toString() { return ""+TYPE; }
}

// utilisation dans la classe Carte :
(...)
Obstacle o = new Obstacle(Obstacle.TypeObstacle.getObstacleAlea(), new Position(...))
(...)
```

Projet : interface ISoldat

```
package wargame;  
  
public interface ISoldat {  
    int getPoints(); int getTour(); int getPortee();  
    void joueTour(int tour);  
    void combat(Soldat soldat);  
    void seDeplace(Position newPos);  
  
    (...)
```

Projet : interface ISoldat

```
(...)  
static enum TypesH {  
    HUMAIN (40,3,10,2), NAIN (80,1,20,0), ELF (70,5,10,6),  
                                                    HOBBIT (20,3,5,2);  
    private final int POINTS_DE_VIE, PORTEE_VISUELLE,  
                                                    PUISSANCE, TIR;  
    TypesH(int points, int portee, int puissance, int tir) {  
        POINTS_DE_VIE = points; PORTEE_VISUELLE = portee;  
        PUISSANCE = puissance; TIR = tir;  
    }  
    public int getPoints() { return POINTS_DE_VIE; }  
    public int getPortee() { return PORTEE_VISUELLE; }  
    public int getPuissance() { return PUISSANCE; }  
    public int getTir() { return TIR; }  
    public static TypesH getTypeHAlea() {  
        return values()[ (int) (Math.random()*values().length) ];  
    }  
}
```


Projet : interface ISoldat

// Utilisation de TypeH dans la classe Carte :

```
Heros h = new Heros(this, Soldat.TypesH.getTypeHAlea(), "Y", new Position(...));
```

// Dans le constructeur de la classe Heros :

```
public Heros(Carte carte, TypesH type, String nom, Position pos) {  
    super(carte, type.getPoints(), type.getPortee(),  
          type.getPuissance(), type.getTir(), pos);  
    NOM = nom; TYPE = type;  
}
```

// Dans le constructeur de la classe abstraite Soldat :

```
public abstract class Soldat extends Element implements ISoldat {  
    private final int POINTS_DE_VIE_MAX, PUISSANCE, TIR, PORTEE_VISUELLE;  
    private int pointsDeVie;  
    (...)  
    Soldat(Carte carte, int pts, int portee, int puiss, int tir, Position pos) {  
        POINTS_DE_VIE_MAX = pointsDeVie = pts;  
        PORTEE_VISUELLE = portee; PUISSANCE = puiss; TIR = tir;  
        this.carte = carte; this.pos = pos;  
    }  
}
```