

TD n° 1 — Interface graphique et événements

Pour réaliser des interfaces graphiques, nous allons utiliser une architecture MVC¹ :

- La vue s'occupera de l'affichage de notre application
- Le modèle contient les méthodes utiles pour le fonctionnement de l'application en elle même
- Le contrôleur liera la vue avec le modèle

Cette architecture nous permet de pouvoir séparer les différentes choses simplement. Mais Également, cela nous permet de pouvoir éventuellement changer l'aspect visuel sans modifier le fonctionnement du jeu (le modèle).

1 Construction du modèle – Classe ModelMorpion

```
1  /**
2   * Enumeration des différents types de cases
3   */
4  public enum TypeCase {
5      VIDE,
6      JOUEUR1,
7      JOUEUR2
8  }
9
10 /**
11  * Classe gérant les actions du jeu indépendamment de l'interface
12  *     MODELE
13  */
14 public class ModeleMorpion {
15     private TypeCase plateau[][];
16
17     public ModeleMorpion(){
18         this.plateau = new TypeCase[3][3];
19         this.initialiser();
20     }
21
22     public boolean casesAlignees(final int p_premiereCase, final int p_secondeCase)
23         throws Exception {
24     }
25
26     public TypeCase getValeurCase(final int x, final int y) throws Exception {
27         return (this.plateau[x][y]);
28     }
29
30     public void initialiser(){
31         for(int i = 0 ; i < 3 ; i++){
32             for(int j = 0 ; j < 3 ; j++){
33                 this.plateau[i][j] = TypeCase.VIDE;
34             }
35         }
36     }
37 }
```

1. Modèle Vue Contrôleur

```

35     }
36 }
37
38 public void setValeurCase(final int i, final int j, TypeCase p_type){
39     this.plateau[i][j] = p_type;
40 }
41
42
43 @Override
44 public String toString(){
45     String retour = "";
46
47     for(int i = 0 ; i < 3 ; i++){
48         for(int j = 0 ; j < 3 ; j++){
49             retour += " ";
50             switch(this.plateau[i][j]) {
51                 case VIDE:
52                     retour += " ";
53                     break;
54                 case JOUEUR1:
55                     retour += "X";
56                     break;
57                 case JOUEUR2:
58                     retour += "O";
59                     break;
60             }
61         }
62         retour += "\n";
63     }
64 }
65
66 }

```

Listing 1 – Classe ModeleMorpion

2 Construction de la vue – Classe VueMorpion

```

1  import java.awt.Panel;
2
3  /**
4   * Classe permettant d'afficher le jeu du morpion
5   *     VUE
6   */
7  public class VueMorpion extends Panel {
8      private Button bGrille[][];
9      private ModeleMorpion modele;
10     private Label etiq1;
11
12     public VueMorpion() {
13         Panel p1 = new Panel();
14         Panel p2 = new Panel();
15         Button b1 = new Button("Effacer");
16         this.bGrille = new Button[3][3];
17         this.etiq1 = new Label("Joueur 1");
18         Label etiq2 = new Label("Joueur 2");
19         ControleurBtGrille controleurBoutonGrille = new ControleurBtGrille();
20

```

```
21     this.modele = new ModeleMorpion();
22
23     this.setLayout(new BorderLayout());
24
25     this.add(p1, BorderLayout.NORTH);
26     this.add(p2, BorderLayout.CENTER);
27     this.add(this.etiq1, BorderLayout.SOUTH);
28
29     p1.setLayout(new GridLayout(1,2);
30     p1.add(b1);
31     b1.addActionListener(new ContoleurBtEffacer());
32     p1.add(etiq2);
33     p2.setLayout(new GridLayout(3,3);
34
35     for(int i = 0; i < 3 ; j++) {
36         for(int j = 0; j < 3; j++) {
37             this.bGrille[i][j] = new Button();
38             p2.add(this.bGrille[i][j]);
39             this.bGrille[i][j].addActionListener(controleurBoutonGrille);
40         }
41     }
42 }
43
44 private Couple coordeesBGrille(Button p_bouton) {
45     Couple coupleRetour;
46
47     for (int i = 0 ; i < 3 ; i++ ) {
48         for (int j = 0 ; j < 3 ; j++) {
49             if(bGrille[i][j] == p_bouton) {
50                 retour = new Couple(-1,-1);
51             }
52         }
53     }
54
55     return coupleRetour;
56 }
57
58
59 }
```

Listing 2 – Classe VueMorpion

3 Construction des contrôleurs

3.1 Classe controleurBtGrille

```

1  /**
2   * Controleur du bouton de la grille
3   * Inclut dans la classe VueMorpion
4   */
5  private class ControleurBtGrille implements ActionListener {
6      @Override
7      public void actionPerformed (ActionEvent e) {
8          Button bouton = (Button) e.getSource();
9          Couple coordonnees VueMorpion.coorreesBGrille(bouton);
10         int i = coordonnees.getPremier();
11         int j = coordonnees.getSecond();
12         char numeroJoueur = VueMorpion.numeroJoueur();
13
14         if(VueModele.modele.getValeurCase(i, j) == TypeCase.VIDE) {
15             if (numeroJoueur == '1'){
16                 b.setLabel("X");
17                 VueMorpion.modele.setValeurCase(i, j, TypeCase.Joueur1);
18                 VueMorpion.etiq2.setText("Joueur 2");
19                 VueMorpion.etiq1.setText("Le joueur "+numeroJoueur+" à gagné");
20             }
21         }
22     }
23 }

```

Listing 3 – Classe VueMorpion

3.2 Classe contoleurBteffacer

```

1  /**
2   * Controleur du bouton de la grille
3   * Inclut dans la classe VueMorpion
4   */
5  private class ControleurBtGrille implements ActionListener {
6      @Override
7      public void actionPerformed (ActionEvent e) {
8          Button bouton = (Button) e.getSource();
9          Couple coordonnees VueMorpion.coorreesBGrille(bouton);
10         int i = coordonnees.getPremier();
11         int j = coordonnees.getSecond();
12         char numeroJoueur = VueMorpion.numeroJoueur();
13
14         if(VueModele.modele.getValeurCase(i, j) == TypeCase.VIDE) {
15             if (numeroJoueur == '1'){
16                 b.setLabel("X");
17                 VueMorpion.modele.setValeurCase(i, j, TypeCase.Joueur1);
18                 VueMorpion.etiq2.setText("Joueur 2");
19                 VueMorpion.etiq1.setText("Le joueur "+numeroJoueur+" à gagné");
20             }
21         }
22     }
23 }

```

Listing 4 – Classe VueMorpion

4 Amélioration – Centralisation des traitements

4.1 Traitement 1

- Initialiser l'étiquette joueur courant à joueur 1
- Initialiser la grille (boutons et modèle)
- Initialiser à vide le libellé du résultat

4.2 Traitement 2

- Positionner "X" dans le bouton cliqué
- Positionner "JOUEUR1" dans la grille du modèle
- Si le joueur à gagné ou non :

```
si 3 cases alignees alors
    mettre à jour le résultat;
sinon
    positionner dans l'étiquette le joueur 2 au joueur;
fin si;
```

4.3 Traitement 3

- Positionner "O" dans le bouton cliqué
- Positionner "JOUEUR2" dans la grille du modèle
- Si le joueur à gagné ou non :

```
si 3 cases alignees alors
    mettre à jour le résultat;
sinon
    positionner dans l'étiquette le joueur 1 au joueur;
fin si;
```

4.4 Améliorations du code

```
1  /**
2   * Classe permettant d'afficher le jeu du morpion
3   */
4  public class VueMorpion extends Panel {
5      private EtatControleur {JOUEUR1, JOUEUR2} etat;
6
7      private ModeleMorpion modele;
8
9      public VueMorpion(){
10         Controleur controleur;
11         btEffacer.addActionListener(controleur);
12     }
13
14     private class Controleur implements ActionListener {
15         private EtatControleur etat;
16
17         public Controleur(){
18             this.etat = EtatControleur.JOUEUR1;
19         }
20
21         public void actionPerformed(ActionEvent e){
22             Button bouton = (Button) e.getSource();
23
24             if(bouton.getLabel().equals("effacer")) {
```

```

25     VueMorpion.etiq2.setText("Joueur 1");
26     VueMorpion.effacerBtGrille();
27     VueMorpion.modele.initialiser();
28     VueMorpion.etiq1.setText("");
29     this.etat = EtatControleur.JOUEUR1;
30 } else {
31     Couple couple = VueMorpion.coordBtGrille(bouton);
32     int i = couple.getPremier();
33     int j = couple.getSecond();
34
35     if(VueMorpion.modele.getValeurCase(i, j) == TypeCase.VIDE) {
36         switch(this.etat){
37             case JOUEUR1 :
38                 bouton.setLabel("X");
39                 ModeleVue.modele.setValeurCase(i, j, TypeCase.JOUEUR1);
40                 if(ModeleVue.modele.caseAlignees(i, j) {
41                     ModeleVue.etiq1.setText("Le joueur 1 à gagné");
42                 } else {
43                     ModeleVue.etiq1.setText("Joueur 2");
44                 }
45                 this.etat = EtatControleur.JOUEUR2;
46                 break;
47             case JOUEUR2 :
48                 bouton.setLabel("O");
49                 ModeleVue.modele.setValeurCase(i, j, TypeCase.JOUEUR2);
50                 if(ModeleVue.modele.caseAlignees(i, j) {
51                     ModeleVue.etiq1.setText("Le joueur 2 à gagné");
52                 } else {
53                     ModeleVue.etiq1.setText("Joueur 1");
54                 }
55                 this.etat = EtatControleur.JOUEUR1;
56                 break;
57         }
58     }
59 }
60 }
61 }
62 }

```

Listing 5 – Améliorations dans la classe VueModele

5 Application du jeu de Morpion

```

1 public class Application {
2     public static void main(String [] args){
3         Frame fenetre = new Frame();
4         fenetre.setTitle("Jeu du morpion");
5         fenetre.setLayout(new GridLayout(1, 1));
6         fenetre.add(new VueMorpion());
7         fenetre.setValeurCase(true);
8         fenetre.pack();
9         fenetre.setSize(50,70);
10    }
11 }

```

Listing 6 – Classe Application

6 Applet du jeu de Morpion

6.1 Applet

```
1 public class AppletMorpion extends Applet {  
2     public void init(){  
3         this.setLayout(new GridLayout(1,1));  
4         this.add(new VueMorpion());  
5     }  
6 }
```

Listing 7 – Classe AppletMorpion

6.2 Page HTML

```
1 <html>  
2     <body>  
3         <applet code="AppletMorpion.class" width="50" height="70"> </applet>  
4     </body>  
5 </html>
```

Listing 8 – Page HTML

A Le code complet

A.1 Enumeration TypeCase

```
1 /**
2  * Enumeration des différents types de cases
3  */
4 public enum TypeCase {
5     VIDE,
6     JOUEUR1,
7     JOUEUR2
8 }
```

Listing 9 – Énumération TypeCase

A.2 Classe ModelMorpion

```
1 /**
2  * Classe gérant les actions du jeu indépendamment de l'interface
3  *     MODELE
4  */
5 public class ModeleMorpion {
6     private TypeCase plateau[][];
7
8     public ModeleMorpion(){
9         this.plateau = new TypeCase[3][3];
10        this.initialiser();
11    }
12
13    public boolean casesAlignees(final int p_premiereCase, final int p_secondeCase)
14        throws Exception {
15    }
16
17    public TypeCase getValeurCase(final int x, final int y) throws Exception {
18        return (this.plateau[x][y]);
19    }
20
21    public void initialiser(){
22        for(int i = 0 ; i < 3 ; i++){
23            for(int j = 0 ; j < 3 ; j++){
24                this.plateau[i][j] = TypeCase.VIDE;
25            }
26        }
27    }
28
29    public void setValeurCase(final int i, final int j, TypeCase p_type){
30        this.plateau[i][j] = p_type;
31    }
32
33
34    @Override
35    public String toString(){
36        String retour = "";
37
38        for(int i = 0 ; i < 3 ; i++){
39            for(int j = 0 ; j < 3 ; j++){
40                retour += " ";
```


