Université De La Rochelle

Master Ingénierie des Contenus Numériques

Blason UML
Niveau 2

Rapport remis par RCAB :

BENAOU Mohammed
RASFA Mohammed
ABDELLAOUI Anass
CHERKAOUI walid

Date de remise 05/10/2017

1 Exercice I

1.1 Diagramme de classe

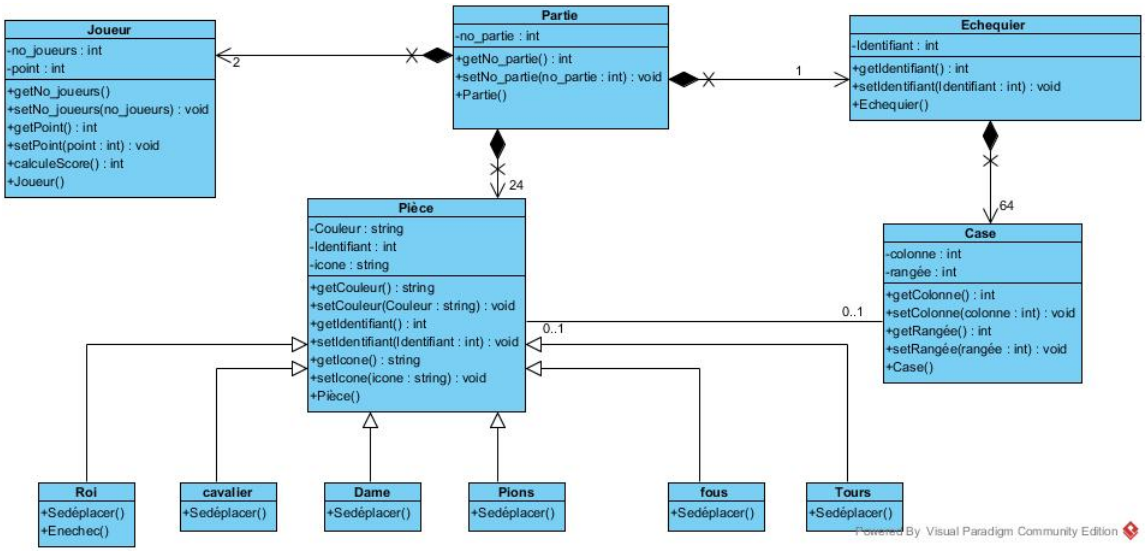

FIGURE 1 – Diagramme de classe

1.2 Les classes java

La class Case :

```
1       public class Case {
2
3   private int colonne;
4   private int range;
5
6   public int getColonne() {
7     return this.colonne;
8   }
9   /**
10   *
11   * @param colonne
12   */
13  public void setColonne(int colonne) {
14    this.colonne = colonne;
15  }
16
17  public int getRange() {
18    return this.range;
19  }
20
21  /**
22   *
23   * @param range
```

```
   */
  public void setRange(int range) {
    this.range = range;
  }

  public Case() {
    // TODO - implement Case.Case
    throw new UnsupportedOperationException();
  }

}
```

La class cavalier :

```
      public class cavalier extends Pice {

  public void Sedplacer() {
    // TODO - implement cavalier.Sedplacer
    throw new UnsupportedOperationException();
  }
}
```

La class Dame :

```
  public class Dame extends Pice {

  public void Sedplacer() {
    // TODO - implement Dame.Sedplacer
    throw new UnsupportedOperationException();
  }

}
```

La classe Echequier :

```
    public class Echequier {

  private int Identifiant;

  public int getIdentifiant() {
    // TODO - implement Echequier.getIdentifiant
    throw new UnsupportedOperationException();
  }

  /**
   *
   * @param Identifiant
   */
  public void setIdentifiant(int Identifiant) {
    // TODO - implement Echequier.setIdentifiant
    throw new UnsupportedOperationException();
  }

  public Echequier() {
    // TODO - implement Echequier.Echequier
    throw new UnsupportedOperationException();
  }
```

```
23
24 }
```

La classe fous :

```
1       public class fous extends Pice {
2
3   public void Sedplacer() {
4     // TODO - implement fous.Sedplacer
5     throw new UnsupportedOperationException();
6   }
7
8 }
```

La classe joueur :

```
1       public class Joueur {
2
3   private int no_joueurs;
4   private int point;
5
6   public void getNo_joueurs() {
7     // TODO - implement Joueur.getNo_joueurs
8     throw new UnsupportedOperationException();
9   }
10
11  /**
12   *
13   * @param no_joueurs
14   */
15  public void setNo_joueurs(int no_joueurs) {
16    this.no_joueurs = no_joueurs;
17  }
18
19  public int getPoint() {
20    return this.point;
21  }
22
23  /**
24   *
25   * @param point
26   */
27  public void setPoint(int point) {
28    this.point = point;
29  }
30
31  public int calculeScore() {
32    // TODO - implement Joueur.calculeScore
33    throw new UnsupportedOperationException();
34  }
35
36  public Joueur() {
37    // TODO - implement Joueur.Joueur
38    throw new UnsupportedOperationException();
39  }
40
```

```
}
```

La classe Partie :

```
      public class Partie {

  private int no_partie;

  public int getNo_partie() {
    return this.no_partie;
  }

  /**
   *
   * @param no_partie
   */
  public void setNo_partie(int no_partie) {
    this.no_partie = no_partie;
  }

  public Partie() {
    // TODO - implement Partie.Partie
    throw new UnsupportedOperationException();
  }

}
```

La classe Pièce :

```
      public class Pice {

  private string Couleur;
  private int Identifiant;
  private string icone;

  public string getCouleur() {
    // TODO - implement Pice.getCouleur
    throw new UnsupportedOperationException();
  }

  /**
   *
   * @param Couleur
   */
  public void setCouleur(string Couleur) {
    // TODO - implement Pice.setCouleur
    throw new UnsupportedOperationException();
  }

  public int getIdentifiant() {
    // TODO - implement Pice.getIdentifiant
    throw new UnsupportedOperationException();
  }

  /**
   *
```

```
28      * @param Identifiant
29      */
30     public void setIdentifiant(int Identifiant) {
31       // TODO - implement Pice.setIdentifiant
32       throw new UnsupportedOperationException();
33     }
34
35     public string getIcone() {
36       return this.icone;
37     }
38
39     /**
40      *
41      * @param icone
42      */
43     public void setIcone(string icone) {
44       this.icone = icone;
45     }
46
47     public Pice() {
48       // TODO - implement Pice.Pice
49       throw new UnsupportedOperationException();
50     }
51
52 }
```

La classe Pions :

```
1        public class Pions extends Pice {
2
3     public void Sedplacer() {
4       // TODO - implement Pions.Sedplacer
5       throw new UnsupportedOperationException();
6     }
7
8 }
```

La classe Roi :

```
1        public class Roi extends Pice {
2
3     public void Sedplacer() {
4       // TODO - implement Roi.Sedplacer
5       throw new UnsupportedOperationException();
6     }
7
8     public void Enechec() {
9       // TODO - implement Roi.Enechec
10      throw new UnsupportedOperationException();
11    }
12
13 }
```

La classe Tours :

```
1        public class Tours extends Pice {
2
```

```
3     public void Sedplacer() {
4       // TODO - implement Tours.Sedplacer
5       throw new UnsupportedOperationException();
6     }
7
8   }
```

2 Exercice II

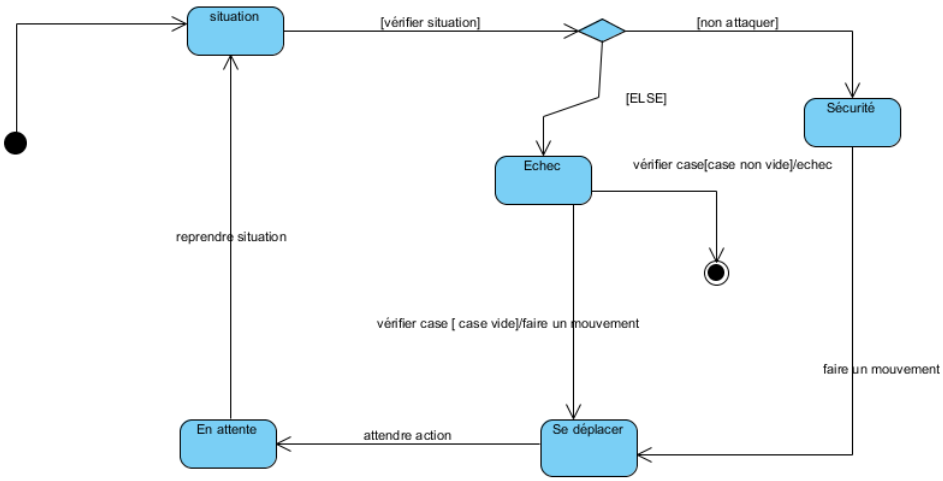

FIGURE 2 – Diagramme d'état transition