Gaëtan Limentour
Thomas Mignon

```
package main;

import java.io.IOException;
import java.util.Scanner;
import javax.swing.*;
import java.awt.*;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import javax.swing.JFrame;
import javax.swing.WindowConstants;

import dialogue.*;

public class main extends JPanel
{
	private static final Color BG_COLOR = new Color(0xbbada0);
	private static final String FONT_NAME = "Arial";
	private static final int TILE_SIZE = 64;
	private static final int TILES_MARGIN = 16;
	private static Jeu jeu = new Jeu();
	private static JLabel casee = new JLabel();
	private static JLabel[] tabcase = new JLabel[15];
	private static JPanel panel = new JPanel();
	private static JFrame game = new JFrame();



	public static void main(String[] args)
	{
	    game.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
	    game.setSize(340, 400);
	    game.setResizable(false);
	    panel.setLayout(new GridLayout(6, 5));
	    game.addKeyListener(gameKeyAdapter());
	    game.setContentPane(MainPanel());
	    game.repaint();
	       game.setVisible(true);
	       casee.setOpaque(true);
	       jeu.compterNbCase();




	}
	private static KeyAdapter gameKeyAdapter()
	 {
	  return new KeyAdapter()
	  {
	   @Override
	   public void keyReleased(KeyEvent e)
	   {
	          switch (e.getKeyCode()) {
	          case KeyEvent.VK_LEFT:
	               jeu.changementGrille();
	               panel.removeAll();
```

```
		game.setContentPane(RefreshPanel());
		if(jeu.getNbCase()/2==16)
		{
		if(jeu.gameover())
		{
			game.setTitle("Perdu !");
			panel.removeAll();
				game.setContentPane(LosePanel());
		}
		}
		if(jeu.maxValue()==2048)
		{
		 game.setTitle("Gagné !");
		 panel.removeAll();
			game.setContentPane(WinPanel());
		}
	break;
	case KeyEvent.VK_RIGHT:
		jeu.swapRight();
		panel.removeAll();
		game.setContentPane(RefreshPanel());
		if(jeu.getNbCase()/2==16)
		{
		if(jeu.gameover())
		{
			game.setTitle("Perdu !");
			panel.removeAll();
				game.setContentPane(LosePanel());
		}
		}
		if(jeu.maxValue()==2048)
		{
		 game.setTitle("Gagné !");
		 panel.removeAll();
			game.setContentPane(WinPanel());
		}
	break;
	case KeyEvent.VK_DOWN:
		jeu.swapDown();
		panel.removeAll();
		game.setContentPane(RefreshPanel());
		if(jeu.getNbCase()/2==16)
		{
		if(jeu.gameover())
		{
			game.setTitle("Perdu !");
			panel.removeAll();
				game.setContentPane(LosePanel());
		}
		}
		if(jeu.maxValue()==2048)
		{
		 game.setTitle("Gagné !");
		 panel.removeAll();
			game.setContentPane(WinPanel());
		}
	break;
	case KeyEvent.VK_UP:
		jeu.swapUp();
		panel.removeAll();
		game.setContentPane(RefreshPanel());
		if(jeu.getNbCase()/2==16)
```

```
                    {
                    if(jeu.gameover())
                    {
                           game.setTitle("Perdu !");
                           panel.removeAll();
                                  game.setContentPane(LosePanel());
                    }
                    }
                    if(jeu.maxValue()==2048)
                    {
                     game.setTitle("Gagné !");
                     panel.removeAll();
                           game.setContentPane(WinPanel());
                    }
                break;
            }
           }
          };
         }

       private static Container MainPanel() {
              jeu.randomSpawn();
              jeu.randomSpawn();
              jeu.compterNbCase();
              for(int i=0;i<=3;i++)
              {
                     for(int j=0;j<=3;j++)
                     {
                           int value = jeu.getLigne()[i].getCase()[j].getValue();
                           if(value != 1)
                           {
                                  JLabel lb2 = new JLabel("<html><h1>" + value +
"</h1></html>",JLabel.CENTER);
                                  lb2.setOpaque(true);
                                  lb2.setBackground(getBackground(value));
                                  panel.add(lb2);
                           }
                           else
                           {
                                  JLabel lb = new JLabel("   ",JLabel.CENTER);
                                  lb.setOpaque(true);
                                  lb.setBackground(getBackground(value));
                                  panel.add(lb);
                           }
                     }
              }
              panel.add(new JLabel("<html><h1>Score :</h1><html>"));
              panel.add(new JLabel(" "));
              panel.add(new JLabel(" "));
              panel.add(new JLabel(" "));
              return panel;
       }
       private static Container RefreshPanel() {
              jeu.compterNbCase();
              for(int i=0;i<=3;i++)
              {
                     for(int j=0;j<=3;j++)
                     {
                           int value = jeu.getLigne()[i].getCase()[j].getValue();
                           if(value != 1)
                           {
                                  JLabel lb2 = new JLabel("<html><h1>" + value +
```

```
"</h1></html>",JLabel.CENTER);
                                lb2.setOpaque(true);
                                lb2.setBackground(getBackground(value));
                                panel.add(lb2);
                        }
                        else
                        {
                                JLabel lb = new JLabel("   ",JLabel.CENTER);
                                lb.setOpaque(true);
                                lb.setBackground(getBackground(value));
                                panel.add(lb);
                        }
                }
        }
        panel.add(new JLabel("<html><h1>Score :</h1><html>"));
        panel.add(new JLabel("<html><h1>"+jeu.getScore()+"</h1></html>"));
        panel.add(new JLabel(" "));
        panel.add(new JLabel(" "));
        return panel;
    }

    private static Container LosePanel() {
        jeu.compterNbCase();
        for(int i=0;i<=3;i++)
        {
                for(int j=0;j<=3;j++)
                {
                        int value = jeu.getLigne()[i].getCase()[j].getValue();
                        if(value != 1)
                        {
                                JLabel lb2 = new JLabel("<html><h1>" + value +
"</h1></html>",JLabel.CENTER);
                                lb2.setOpaque(true);
                                lb2.setBackground(getBackground(value));
                                panel.add(lb2);
                        }
                        else
                        {
                                JLabel lb = new JLabel("   ",JLabel.CENTER);
                                lb.setOpaque(true);
                                lb.setBackground(getBackground(value));
                                panel.add(lb);
                        }
                }
        }
        panel.add(new JLabel("<html><h1>Score :</h1><html>"));
        panel.add(new JLabel("<html><h1>"+jeu.getScore()+"</h1></html>"));
        panel.add(new JLabel("<html><h1>Perdu !</h1></html>"));
        panel.add(new JLabel(" "));
        return panel;
    }
    private static Container WinPanel() {
        jeu.compterNbCase();
        for(int i=0;i<=3;i++)
        {
                for(int j=0;j<=3;j++)
                {
                        int value = jeu.getLigne()[i].getCase()[j].getValue();
                        if(value != 1)
                        {
                                JLabel lb2 = new JLabel("<html><h1>" + value +
"</h1></html>",JLabel.CENTER);
```

```
                        lb2.setOpaque(true);
                        lb2.setBackground(getBackground(value));
                        panel.add(lb2);
                    }
                    else
                    {
                        JLabel lb = new JLabel("   ",JLabel.CENTER);
                        lb.setOpaque(true);
                        lb.setBackground(getBackground(value));
                        panel.add(lb);
                    }
                }
            }
            panel.add(new JLabel("<html><h1>Score :</h1><html>"));
            panel.add(new JLabel("<html><h1>"+jeu.getScore()+"</h1></html>"));
            panel.add(new JLabel("<html><h1>Gagné !</h1></html>"));
            panel.add(new JLabel(" "));
            return panel;
        }

         public static Color getBackground(int value) {
              switch (value) {
                case 2:    return new Color(0xeee4da);
                case 4:    return new Color(0xede0c8);
                case 8:    return new Color(0xf2b179);
                case 16:   return new Color(0xf59563);
                case 32:   return new Color(0xf67c5f);
                case 64:   return new Color(0xf65e3b);
                case 128:  return new Color(0xedcf72);
                case 256:  return new Color(0xedcc61);
                case 512:  return new Color(0xedc850);
                case 1024: return new Color(0xedc53f);
                case 2048: return new Color(0xedc22e);
              }
              return new Color(0xcdc1b4);
            }


}
```

```java
package dialogue;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import java.util.Scanner;
import java.io.IOException;

import dialogue.*;

public class Jeu extends Grille
{
	Grille grille = new Grille();
}
```

```java
package dialogue;

import java.util.Random;

public class Grille
{
	private Ligne[] lignes = new Ligne[4];
	private int CompteurLose=0;
	private int nbCase;
	private int score=0;

	public Grille()
	{
		for(int i=0;i<4;i++)
		{
			lignes[i]=new Ligne();
		}
	}

	//Getter
	public Ligne[] getLigne()
	{
		return this.lignes;
	}

	public int getNbCase()
	{
		return this.nbCase;
	}

	public void afficherGrille()
	{
		for(int i=0;i<4;i++)
		{
			lignes[i].afficherLigne();
			System.out.println("");
		}
	}

	public int getScore()
	{
		return this.score;
	}

	public void changementGrille()
	{
		int randomSpawn=0;
		for(int tour=0;tour<=3;tour++)//Première ligne
		{
			for(int j=1;j<=3;j++)//Examination de la ligne
			{


				if(this.getLigne()[tour].getCase()[j].getValue()!=1)//On
vérifie si la case est à 1 ou pas
				{
					for(int i=j-1;i>=0;i--)//On regarde la case derrière
					{
```

```
                                        if(this.getLigne()[tour].getCase()
[i].getValue()!=1)//Si elle différente de 1
                                        {
                                                if(this.getLigne()[tour].getCase()
[i].getValue() == this.getLigne()[tour].getCase()[j].getValue())//On vérifier si elle
est égale à la case à laquelle j est
                                                {

                                                        this.getLigne()[tour].getCase()
[i].setValue(this.getLigne()[tour].getCase()[i].getValue()*2);

      this.score=this.score+this.getLigne()[tour].getCase()[i].getValue();
                                                        this.getLigne()[tour].getCase()
[j].setValue(1);
                                                        if(randomSpawn==0)
                                                        {
                                                                randomSpawn();
                                                                randomSpawn++;
                                                        }
                                                }
                                                else //Sinon on arrête la boucle (pour
éviter de vérifier les cases encore derrière)
                                                {
                                                        i = -1;
                                                }
                                        }
                                        else //Sinon on remplace cette case à 1 par la
valeur de la case sur laquelle J est
                                        {
                                                this.getLigne()[tour].getCase()
[i].setValue(this.getLigne()[tour].getCase()[j].getValue());
                                                this.getLigne()[tour].getCase()
[j].setValue(1);
                                                j--; // On fais en sorte que J sois à la
même valeur pour revérifier la case qu'il y a derrière
                                                if(randomSpawn==0)
                                                {
                                                        randomSpawn();
                                                        randomSpawn++;
                                                }
                                        }
                                }
                        }
                }
        }

        public void swapDown()
        {
                Grille temp = new Grille();
                for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la droite
                {
                        for(int i=0;i<=3;i++)
                        {
                                temp.getLigne()[i].getCase()[x].setValue(this.getLigne()
[j].getCase()[i].getValue());
                        }
                }
                temp.changementGrille();
                this.score=this.score+temp.getScore();
                for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la gauche
                {
```

```
                        for(int i=0;i<=3;i++)
                        {
                                this.getLigne()[j].getCase()[i].setValue(temp.getLigne()
[i].getCase()[x].getValue());
                        }
                }
        }

        public void swapUp()
        {
                Grille temp = new Grille();
                for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la gauche
                {
                        for(int i=0;i<=3;i++)
                        {
                                temp.getLigne()[j].getCase()[i].setValue(this.getLigne()
[i].getCase()[x].getValue());
                        }
                }
                temp.changementGrille();
                this.score=this.score+temp.getScore();
                for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la droite
                {
                        for(int i=0;i<=3;i++)
                        {
                                this.getLigne()[i].getCase()[x].setValue(temp.getLigne()
[j].getCase()[i].getValue());
                        }
                }
        }

        public void swapRight()
        {
                Grille temp = new Grille();
                Grille temp2 = new Grille();
                for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la gauche 1
                {
                        for(int i=0;i<=3;i++)
                        {
                                temp.getLigne()[j].getCase()[i].setValue(this.getLigne()
[i].getCase()[x].getValue());
                        }
                }
                for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la gauche 2
                {
                        for(int i=0;i<=3;i++)
                        {
                                temp2.getLigne()[j].getCase()[i].setValue(temp.getLigne()
[i].getCase()[x].getValue());
                        }
                }
                temp2.changementGrille();
                this.score=this.score+temp2.getScore();
                for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la droite 1
                {
                        for(int i=0;i<=3;i++)
                        {
                                temp.getLigne()[i].getCase()[x].setValue(temp2.getLigne()
[j].getCase()[i].getValue());
                        }
                }
                for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la droite 2
```

```
		{
			for(int i=0;i<=3;i++)
			{
				this.getLigne()[i].getCase()[x].setValue(temp.getLigne()
[j].getCase()[i].getValue());
			}
		}
	}

	public void randomSpawn()
	{
		int nbCaseVide=0;
		Random r = new Random();
		Case[] TabCaseVide = new Case[16];
		for(int i=0;i<=3;i++)
		{
			for(int j=0;j<=3;j++)
			{
				if(this.getLigne()[i].getCase()[j].getValue()==1)
				{
					TabCaseVide[nbCaseVide] = this.getLigne()[i].getCase()
[j];
					nbCaseVide++;
				}
			}
		}
		int nbRandom1 = r.nextInt(nbCaseVide - 0);
		TabCaseVide[nbRandom1].setValue(Math.random() < 0.9 ? 2 : 4);
	}

	public void compterNbCase()
	{
		this.nbCase=16;
		for(int i=0;i<=3;i++)
		{
			for(int j=0;j<=3;j++)
			{
				if(this.getLigne()[i].getCase()[j].getValue()==1)
				{
					this.nbCase--;
				}
				else
				{
					this.nbCase++;
				}
			}
		}
	}

	/*public int LoseOrNot() {
		//cas gauche
		CompteurLose=0;
		for(int i = 0;i<=3;i++)
		{
			for(int j = 1;j<=3;j++)
			{
				if(this.getLigne()[i].getCase()[j].getValue() !=
this.getLigne()[i].getCase()[j-1].getValue())
				{
					this.CompteurLose++;
					i=4;
					j=4;
```

```
                        }
                  }
            }
            //cas haut
            Grille temp = new Grille();
            for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la gauche
            {
                  for(int i=0;i<=3;i++)
                  {
                        temp.getLigne()[j].getCase()[i].setValue(this.getLigne()
[i].getCase()[x].getValue());
                  }
            }
            for(int i = 0;i<=3;i++)
            {
                  for(int j = 1;j<=3;j++)
                  {
                        if(temp.getLigne()[i].getCase()[j].getValue() !=
temp.getLigne()[i].getCase()[j-1].getValue())
                        {
                              CompteurLose++;
                              i=4;
                              j=4;
                        }
                  }
            }
            Grille temp2 = new Grille();
            //cas droite
            for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la gauche
            {
                  for(int i=0;i<=3;i++)
                  {
                        temp2.getLigne()[j].getCase()[i].setValue(temp.getLigne()
[i].getCase()[x].getValue());
                  }
            }
            for(int i = 0;i<=3;i++)
            {
                  for(int j = 1;j<=3;j++)
                  {
                        if(temp2.getLigne()[i].getCase()[j].getValue() !=
temp2.getLigne()[i].getCase()[j-1].getValue())
                        {
                              CompteurLose++;
                              i=4;
                              j=4;
                        }
                  }
            }
            Grille temp3 = new Grille();
            temp3=temp2;
            //cas Bas
            for(int j=0, x=3;j<=3;j++,x--) //Tourner la grille vers la gauche
            {
                  for(int i=0;i<=3;i++)
                  {
                        temp.getLigne()[j].getCase()[i].setValue(this.getLigne()
[i].getCase()[x].getValue());
                  }
            }
            for(int i = 0;i<=3;i++)
            {
```

```
                    for(int j = 1;j<=3;j++)
                    {
                          if(temp.getLigne()[i].getCase()[j].getValue() !=
temp.getLigne()[i].getCase()[j-1].getValue())
                          {
                                CompteurLose++;
                                i=4;
                                j=4;
                          }
                    }
              }
              return CompteurLose;
       }// Verification que deux case adjacente ne sont pas égale*/

       public boolean gameover()
       {
              Grille temp = new Grille();
              int compteur=0;
              for (int i=0;i<=3;i++)//Recopiage de la grille actuelle
              {
                    for (int j=0;j<=3;j++)
                    {
                          temp.getLigne()[i].getCase()[j].setValue(this.getLigne()
[i].getCase()[j].getValue());
                    }
              }
              temp.changementGrille();//Vérification à gauche
              for (int i=0;i<=3;i++)
              {
                    for (int j=1;j<=3;j++)
                    {
                          if(temp.getLigne()[i].getCase()
[j].getValue()==this.getLigne()[i].getCase()[j].getValue())
                          {
                                compteur++;
                          }
                          if(compteur==12)
                          {
                                this.CompteurLose++;
                          }
                    }
              }
              compteur=0;
              temp.swapRight();//Verification à droite
              for (int i=0;i<=3;i++)
              {
                    for (int j=1;j<=3;j++)
                    {
                          if(temp.getLigne()[i].getCase()
[j].getValue()==this.getLigne()[i].getCase()[j].getValue())
                          {
                                compteur++;
                          }
                          if(compteur==12)
                          {
                                this.CompteurLose++;
                          }
                    }
              }
              compteur=0;
              temp.swapDown();//Vérification en bas
              for (int i=0;i<=3;i++)
```

```
		{
			for (int j=1;j<=3;j++)
			{
				if(temp.getLigne()[i].getCase()
[j].getValue()==this.getLigne()[i].getCase()[j].getValue())
				{
					compteur++;
				}
				if(compteur==12)
				{
					this.CompteurLose++;
				}
			}
		}
		compteur=0;
		temp.swapUp();//Vérification en haut
		for (int i=0;i<=3;i++)
		{
			for (int j=1;j<=3;j++)
			{
				if(temp.getLigne()[i].getCase()
[j].getValue()==this.getLigne()[i].getCase()[j].getValue())
				{
					compteur++;
				}
				if(compteur==12)
				{
					this.CompteurLose++;
				}
			}
		}
		if(CompteurLose==4)
		{
			return true;
		}
		else
		{
			this.CompteurLose=0;
			return false;
		}

	}
	public int maxValue()
	{
		int max=0;
		for(int i=0;i<=3;i++)
		{
			for(int j=0;j<=3;j++)
			{
				if(this.getLigne()[i].getCase()[j].getValue()>max)
				{
					max=this.getLigne()[i].getCase()[j].getValue();
				}
			}
		}
		return max;
	}
}
```

```java
package dialogue;

public class Ligne
{
	private Case[] cases = new Case[4];

	public Ligne()
	{
		for(int i=0;i<4;i++)
		{
			cases[i]=new Case();
			cases[i].setValue(1);
		}
	}

	//Getteur
	public Case[] getCase()
	{
		return this.cases;
	}
	public void afficherLigne()
	{
		for(int i=0;i<4;i++)
		{
			cases[i].afficherCase();
			System.out.print(" ");
		}
	}
}
```

```java
package dialogue;

import java.awt.Color;

public class Case
{
	private int value;

	public Case()
	{
		this.value=1;
	}

	public int getValue()
	{
		return this.value;
	}

	public void setValue(int value)
	{
		this.value=value;
	}

	public void afficherCase()
	{
		System.out.print(this.value);
	}

	public Color getBackground(int value) {
		switch (value) {
		  case 2:    return new Color(0xeee4da);
		  case 4:    return new Color(0xede0c8);
		  case 8:    return new Color(0xf2b179);
		  case 16:   return new Color(0xf59563);
		  case 32:   return new Color(0xf67c5f);
		  case 64:   return new Color(0xf65e3b);
		  case 128:  return new Color(0xedcf72);
		  case 256:  return new Color(0xedcc61);
		  case 512:  return new Color(0xedc850);
		  case 1024: return new Color(0xedc53f);
		  case 2048: return new Color(0xedc22e);
		}
		return new Color(0xcdc1b4);
	 }
}
```