

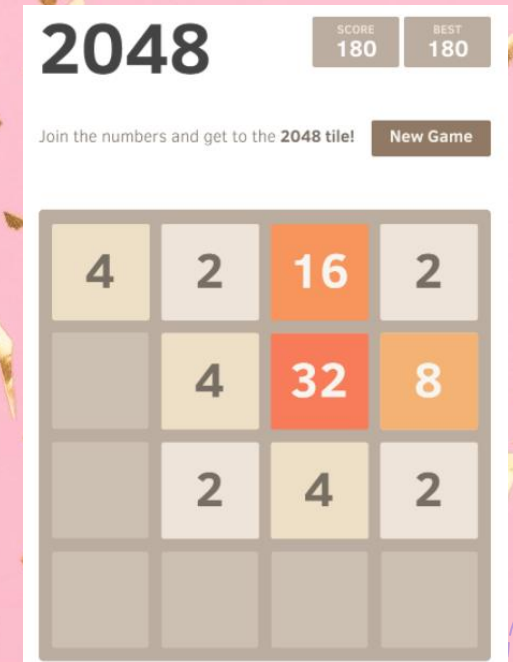
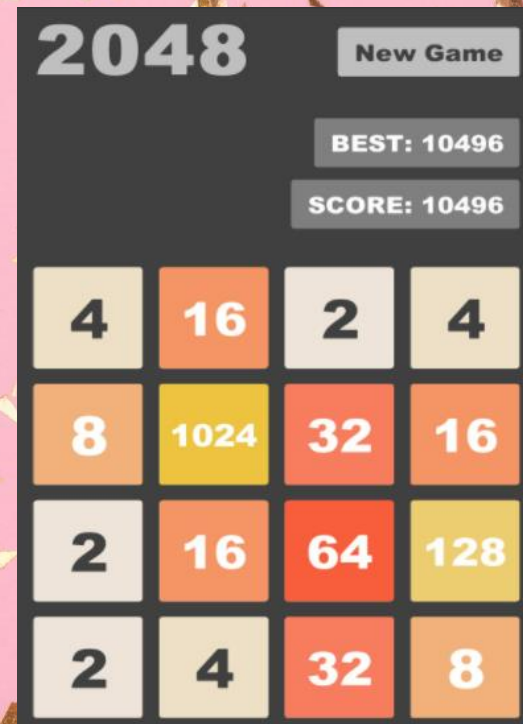
2048 Game

+

Mălina Diaconescu

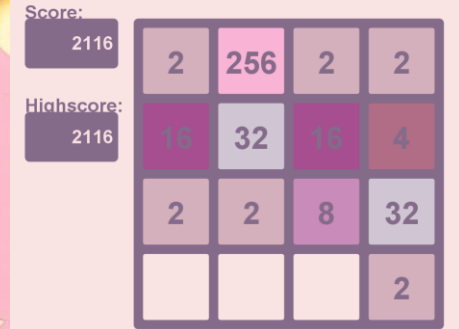
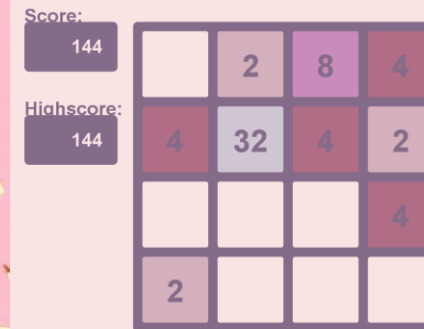
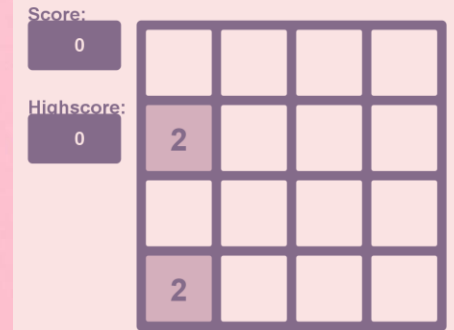
Motivation

- It was my childhood game
- I wanted to make my own version of it by my own imagination
- Considered a pretty good way of learning more Java, since it was my first game implemented in this language



How to play

- When it is your first time playing the game, the game will show you a welcome page. By clicking, you will start the game.
- You will be taken when the actual game starts (second image). You play by using the arrow keys and moving the tiles until you get to the final (2048 value).



Updates and Changes from the Demo Version

- Added the score and highscore board that update in real time when the user is playing
- The highscore board saves when you lose or win the game
- Also, I added a new event on the keyboard: when you press Q the game saves itself and exits from the application
- When you open the game again, you will be put in the exact same state that you exit from and everything is saved (including score and highscore).

Score:

2116

Highscore:

2116

Score:

0

Highscore:

2536

```
124         case KeyEvent.VK_Q:
125             try {
126                 FileOutputStream myObj2 = new FileOutputStream("tabla.ser");
127                 JocuCurrent joc=new JocuCurrent(highscore,score,highest,tiles);
128                 // for(int i=0;i<4;i++)
129                 //     for(int j=0;j<4;j++)
130                 //         writer.write(tiles[i][j]);
131                 ObjectOutputStream out=new ObjectOutputStream(myObj2);
132                 out.writeObject(joc);
133                 myObj2.close();
134                 File myObj=new File("score.txt");
135                 FileWriter reader=new FileWriter(myObj);
136                 reader.write("1");
137                 reader.close();
138                 System.out.println("AAAAAAA");
139             } catch (IOException e1) {
140                 System.out.println("An error occurred.");
141                 e1.printStackTrace();
142             }
143             System.out.println(terminari);
144             System.exit(0);
145             break;
146         }
```


Technical Part

- 2048 game implementation – it functions by the exact same algorithm
- Implemented in Java this semester, by using JPanel as the tool for designing the actual game/the part that users interact with
- It has 4 actual states that can be in (as seen in the screenshot) each of them corresponds to the actual name of it
- It also uses the Random class in Java for generating random new tiles in the game
- Uses AWT Event for the user's interaction with the game

```
19 public class Game2048 extends JPanel {
20
21     enum State {
22         start, won, running, over //stari ale jocului
23     }
```

```
108     addKeyListener(new KeyAdapter() {
109         @Override
110         public void keyPressed(KeyEvent e) {
111             switch (e.getKeyCode()) {
112                 case KeyEvent.VK_UP: //sazul in sus
113                     moveUp();
114                     break;
115                 case KeyEvent.VK_DOWN: //sazul in jos
116                     moveDown();
117                     break;
118                 case KeyEvent.VK_LEFT: //sazul in stanga
119                     moveLeft();
120                     break;
121                 case KeyEvent.VK_RIGHT: //dreapta
122                     moveRight();
123                     break;
124                 case KeyEvent.VK_Q:
125                     try {
126                         FileOutputStream myObj2 = new FileOutputStream("tabla.ser");
127                         JoculCurent joc=new JoculCurent(highscore,score,highest,tiles);
128                         // for(int i=0;i<4;i++)
129                         // for(int j=0;j<4;j++)
130                         // writer.write(tiles[i][j]);
131                         ObjectOutputStream out=new ObjectOutputStream(myObj2);
132                         out.writeObject(joc);
133                         myObj2.close();
134                         File myObj=new File("scoruri.txt");
135                         FileWriter reader=new FileWriter(myObj);
136                         reader.write("1");
137                         reader.close();
138                         System.out.println("AAAAAAA");
139                     } catch (IOException e1) {
140                         System.out.println("An error occurred.");
141                         e1.printStackTrace();
142                     }
143                     System.out.println(terminari);
144                     System.exit(0);
145                     break;
146             }
147         }
148     });
```

```
301     private void addRandomTile() {
302         int pos = rand.nextInt(side * side); //generam un nr. random intre 0 si 15 la care vom adauga unu
303         int row, col;
304         do {
305             //calculam pozitia acestuia
306             pos = (pos + 1) % (side * side);
307             row = pos / side;
308             col = pos % side;
309         } while (tiles[row][col] != null);
310
311         int val = rand.nextInt(10) == 0 ? 4 : 2;
312         tiles[row][col] = new Tile(val);
313     }
```

The background is a solid pink color. It is decorated with numerous small, irregular gold-colored shapes that resemble confetti or crushed paper, scattered across the entire surface. Overlaid on this are several thin, white, wavy lines that meander across the background, creating a sense of movement and depth. In the bottom right corner, there is a single, solid orange line that curves upwards and to the right.

Thank you!