Introduction

Background

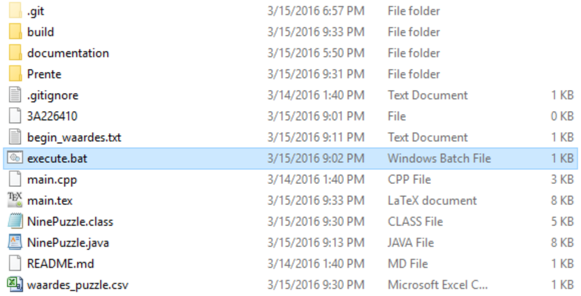
History

Literature Study

User Guide

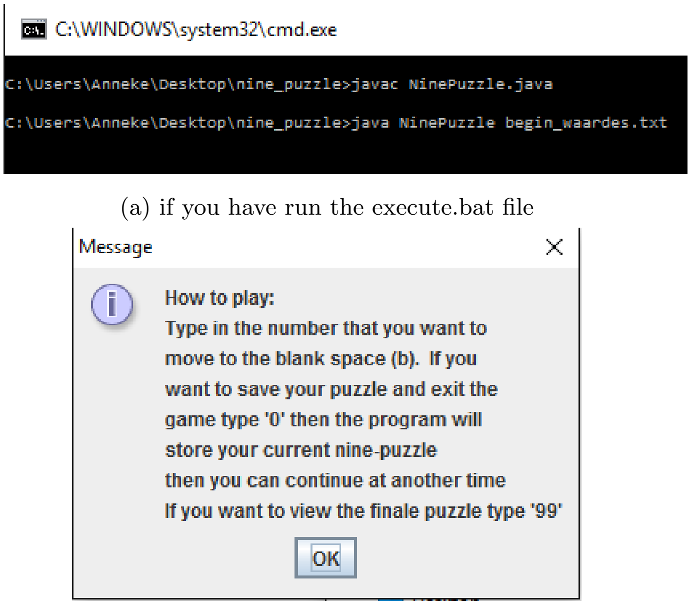
How to begin the Nine-puzzle

Step 1: Right click on the execute.bat file and open it with notepad. Change the name of the .csv file. Save the file and exit it. Double click on the file to run the file. Figure 1
Step 2: If the program has run you will see figure Figure 2



.git	3/15/2016 6:57 PM	File folder	
build	3/15/2016 9:33 PM	File folder	
documentation	3/15/2016 5:50 PM	File folder	
Prente	3/15/2016 9:31 PM	File folder	
.gitignore	3/14/2016 1:40 PM	Text Document	1 KB
3A226410	3/15/2016 9:01 PM	File	0 KB
begin_waardes.txt	3/15/2016 9:11 PM	Text Document	1 KB
execute.bat	3/15/2016 9:02 PM	Windows Batch File	1 KB
main.cpp	3/14/2016 1:40 PM	CPP File	3 KB
main.tex	3/15/2016 9:33 PM	LaTeX document	8 KB
NinePuzzle.class	3/15/2016 9:30 PM	CLASS File	5 KB
NinePuzzle.java	3/15/2016 9:13 PM	JAVA File	8 KB
README.md	3/14/2016 1:40 PM	MD File	1 KB
waardes_puzzle.csv	3/15/2016 9:30 PM	Microsoft Excel C...	1 KB

Figure 1



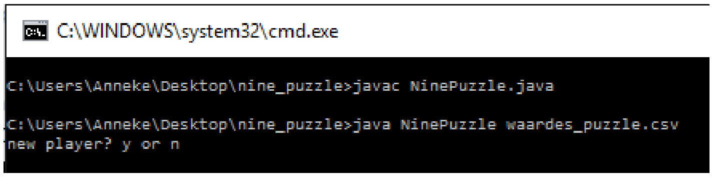
```
C:\Users\Anneke\Desktop\nine_puzzle>javac NinePuzzle.java

C:\Users\Anneke\Desktop\nine_puzzle>java NinePuzzle begin_waardes.txt
```

(a) if you have run the execute.bat file

How to play:
Type in the number that you want to
move to the blank space (b). If you
want to save your puzzle and exit the
game type '0' then the program will
store your current nine-puzzle
then you can continue at another time
If you want to view the finale puzzle type '99'

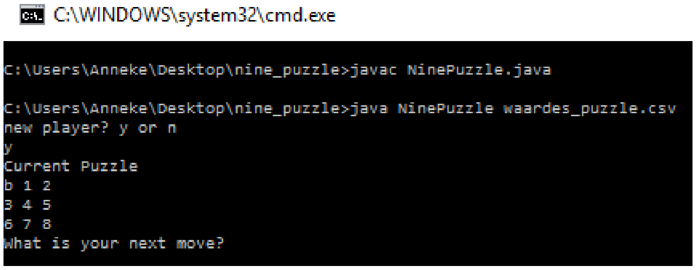
(b) The program will show this message directly after you have run the execute.bat file

Figure 2



```
C:\Users\Anneke\Desktop\nine_puzzle>javac NinePuzzle.java

C:\Users\Anneke\Desktop\nine_puzzle>java NinePuzzle waardes_puzzle.csv
new player? y or n
```

Figure 3: This figure asks if you are a new player or not



```
C:\Users\Anneke\Desktop\nine_puzzle>javac NinePuzzle.java

C:\Users\Anneke\Desktop\nine_puzzle>java NinePuzzle waardes_puzzle.csv
new player? y or n
y
Current Puzzle
b 1 2
3 4 5
6 7 8
What is your next move?
```

Figure 4

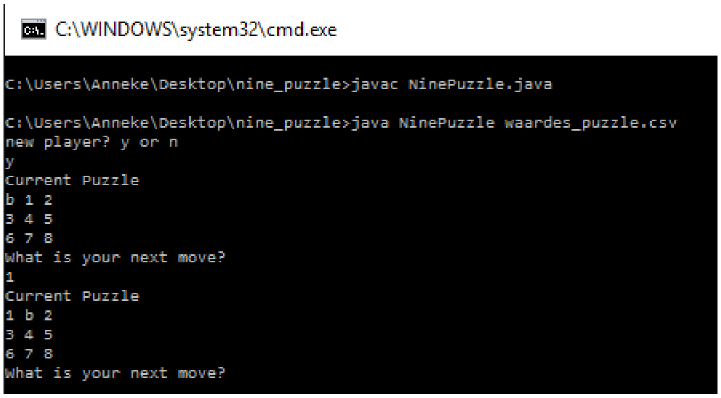
Step 3: Enter a y for you are a new player or n for you are not a new player Figure 3
Step 4: After step 3 is completed the begin puzzle will be given. Figure 4
Step 5: Enter the number that you want to move like in Figure 5. The game will move that number to the blank space(b) and give the updated puzzle again.

Functionality

How to stop save and exit the program: If you want to stop playing and play again later then you can type 0 then the puzzle will be saved and the game will exit. Figure 6

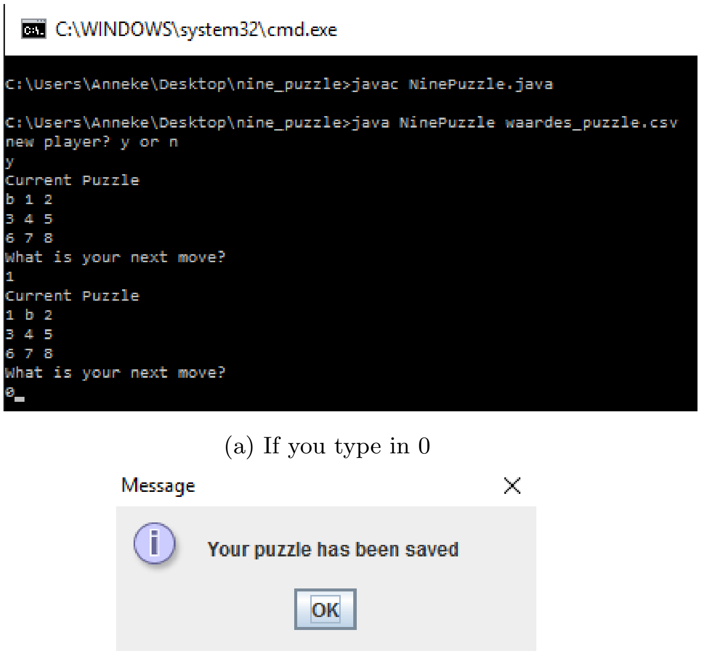
If you have won the game the program will save your work and show a message to congratulate you and show how many moves you have made to solve the puzzle. Figure 7

If you want to view the finale puzzle you type in 99. Figure 8



```
C:\Users\Anneke\Desktop\nine_puzzle>javac NinePuzzle.java

C:\Users\Anneke\Desktop\nine_puzzle>java NinePuzzle waardes_puzzle.csv
new player? y or n
y
Current Puzzle
b 1 2
3 4 5
6 7 8
What is your next move?
1
Current Puzzle
1 b 2
3 4 5
6 7 8
What is your next move?
```

Figure 5



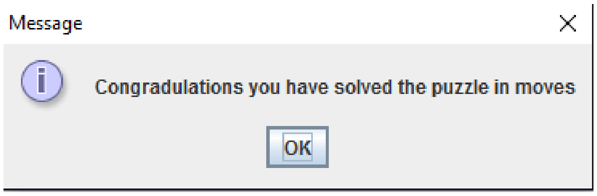
```
C:\Users\Anneke\Desktop\nine_puzzle>javac NinePuzzle.java

C:\Users\Anneke\Desktop\nine_puzzle>java NinePuzzle waardes_puzzle.csv
new player? y or n
y
Current Puzzle
b 1 2
3 4 5
6 7 8
What is your next move?
1
Current Puzzle
1 b 2
3 4 5
6 7 8
What is your next move?
0
```

(a) If you type in 0

Your puzzle has been saved

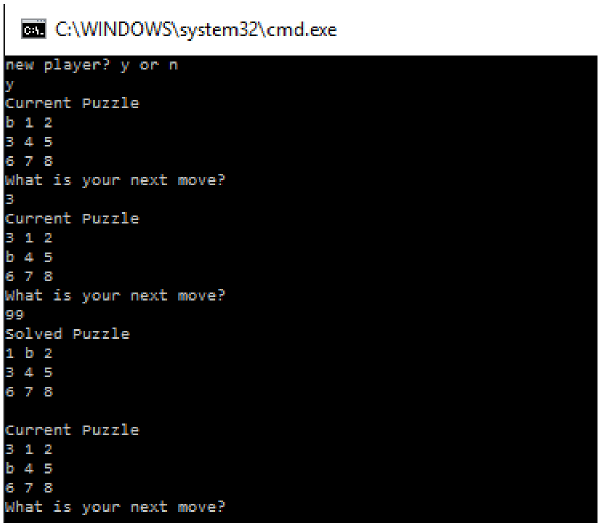
(b) The message will show that you work has been saved

Figure 6



Congradulations you have solved the puzzle in moves

Figure 7



```
new player? y or n
y
Current Puzzle
b 1 2
3 4 5
6 7 8
What is your next move?
3
Current Puzzle
3 1 2
b 4 5
6 7 8
What is your next move?
99
Solved Puzzle
1 b 2
3 4 5
6 7 8

Current Puzzle
3 1 2
b 4 5
6 7 8
What is your next move?
```

Figure 8

Bibliography

Code

```
import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import javax.swing.JOptionPane;

public class NinePuzzle {
  public static void main(String[] args) throws FileNotFoundException {
    // scanner to read out of txt file
    Scanner input = new Scanner(args[0]);
    // gebruiker se skuiwe
    Scanner gebruiker = new Scanner(System.in);
    boolean solved = false;
    boolean save = false;
    int count = 0;
    // create arrays
    int[] current = new int[9];
    int[] finaal = new int[9];
    // array for the string read in csv file
    String[] userInputs = new String[1];
    String filename = input.nextLine();
    // call method that reads the csv file
    setUpArray(current, finaal, userInputs, filename);
    // display how the game is played
    String message = "How to play: \n" +
                 "Type in the number that you want to \n" +
                     "move to the blank space (b).  If you \n"+
 "want to save your puzzle and exit the \n"+
 "game type '0' then the program will \n"+
 "store your current nine-puzzle \n" +
                     "then you can continue at another time \n"+
 "If you want to view the finale puzzle type '99'";
    JOptionPane.showMessageDialog(null, message);
    // while loop to see if the user has finished playing
System.out.println("new player? y or n");
String newUser = gebruiker.nextLine();
String yn = newUser.substring(0, 1);
if (yn.equals("y") ||  yn.equals("Y")) {
  userInputs[0] = "";
      writeArrayToFile(current, finaal, userInputs[0], filename);
}
    while ( !solved && !save) {
      System.out.println("Current Puzzle");
      puzzle(current); // print current puzzle
      int lees = 0;
      // ask user what his next move is going to be
      System.out.println("What is your next move?");
      lees = gebruiker.nextInt();
      // test if the input from user is 0 or any other number between 1 - 8
      if ( lees != 0 ) {
        if(lees == 99) {
          System.out.println("Solved Puzzle");
          puzzle(finaal);
          System.out.println("");
          } else {
          // move the index of the tiles
          move_index(lees, current);
          count++;
          solved = compare_solution(finaal, current);
          userInputs[0] += Integer.toString(lees) + ",";
          }
      } else { //if the user enters 0
        //write current puzzle to csv file
        writeArrayToFile(current, finaal, userInputs[0], filename);
        //stoor na txt file as begin waardes
        save = true;
      }//end if
      //show the final puzzle to the user
    }// end while
    // show message that say he has solved  the puzzle
    if ( solved ) {
      String message1 = "Congradulations you have solved the puzzle in moves";
      JOptionPane.showMessageDialog(null, message1);
      System.out.println("jy het" + " " + count + " "
      + "skuiwe gemaak om die puzzle klaar te maak");
    }
if(save){
  String message2 = "Your puzzle has been saved";
      JOptionPane.showMessageDialog(null, message2);
}
}//end main

  public static void writeArrayToFile(int [] current,
                                       int [] finaal,
                                  String userinputs,
                                  String filename)
                throws FileNotFoundException {
  try {
    PrintWriter outputStream = new PrintWriter(filename);
    for ( int k = 0; k < 9; k++ ) {
      if ( k != 8 ) {
     // write the current puzzle to file
        if(current[k] == 0) {
          outputStream.print("b,");
        } else {
          outputStream.print(current[k] + ",");
        }
      } else {
          outputStream.print(current[k] + "\r\n");
      }
    }
  for(int h = 0; h < 9; h++ ) {
    if (h != 8) {
      if (finaal[h] == 0) {
        outputStream.print("b,");
      } else {
          outputStream.print(finaal[h] + ",");
      }
    } else {
      outputStream.print(finaal[h] + "\r\n");
    }//end else
  }
  if (userinputs.length() > 0) {
    String user_inputs = userinputs.substring(0, userinputs.length() - 1) + "\r\n";
    outputStream.println(user_inputs);
  }
  outputStream.close();
  } catch (FileNotFoundException e) {
      e.printStackTrace();
  }//end catch
}
//read csv file into an array
public static void setUpArray(int [] current, int [] finaal, String [] userinputs, String filename) throws FileNotFoundException {
  File File = new File(filename);
  try {
    Scanner inputStream = new Scanner(File);
    int lines_read = 0;
    while (inputStream.hasNext() ) {
      String data = inputStream.next();//gets a whole line
      String [] values = data.split(",");
      if ( lines_read == 0 ) {
        for ( int v = 0; v < 9; v++ ) {
          if (values[v].equals("b")) {
            current[v] = 0;
          } else {
            current[v] = Integer.parseInt(values[v]);
          }
        }
      } else if ( lines_read == 1 ) {
               for ( int w = 0; w < 9; w++ ) {
                 if (values[w].equals("b")) {
                   finaal[w] = 0;
                 } else {
                     finaal[w] = Integer.parseInt(values[w]);
                 }
               }
      } else if ( lines_read == 2 ) {
       if (!data.equals(null) || !data.equals("")) {
                 userinputs[0] = data + ",";
       } else {
           userinputs[0] = "";
       }
      }
    lines_read++;
  }
    inputStream.close();
  }
  catch(FileNotFoundException e) {
    e.printStackTrace();
  }
}//end setuparray
//print puzzle
public static int puzzle(int [] current) {
  for(int k = 0; k < 9; k++) {
    if (current[k] == 0) {
      System.out.print("b ");
    } else {
        System.out.print(current[k] +" ");
    }
    if ( ((k + 1) % 3) == 0 ) {
      System.out.print("\n");
    }
  }//end for
  return 0;
}//end puzzle

//swap the index of the tile that has been moved
```

```
  public static boolean move_index(int lees, int [] current) {
    int[][] lookup = {{1, 3, -1, -1},
                      {0, 2, 4, -1},
                      {1, 5, -1, -1},
                      {0, 4, 6, -1},
                      {1, 3, 5, 7},
                      {2, 4, 8, -1},
                      {3, 7, -1, -1},
                      {4, 6, 8, -1},
                      {5, 7, -1, -1}};
    int zero_position = get_index(0, current);
    int value_position = get_index(lees, current);
    for ( int i = 0; i < 4; i++) {
      if ( lookup[zero_position][i] == value_position ) {
        current[value_position] = 0;
        current[zero_position] = lees;
    return true;
      }
    }
  return false;
  }//end move_index
  //get the index of the blank space and the number that the user want to move
  public static int get_index(int lees, int [] current) {
    for ( int i = 0; i < 9; i++ ) {
      if ( lees == current[i] ) {
        return i;
      }
    }
  return 0;
  }
  //compare wether the current puzzle has the same values of the final puzzle
  public static boolean compare_solution(int [] finaal, int [] current) {
    for ( int i = 0; i < 9; i++) {
      if ( finaal[i] != current[i] ) {
        return false;
      }
    }
    return true;
  }//end compare
}//end class
```