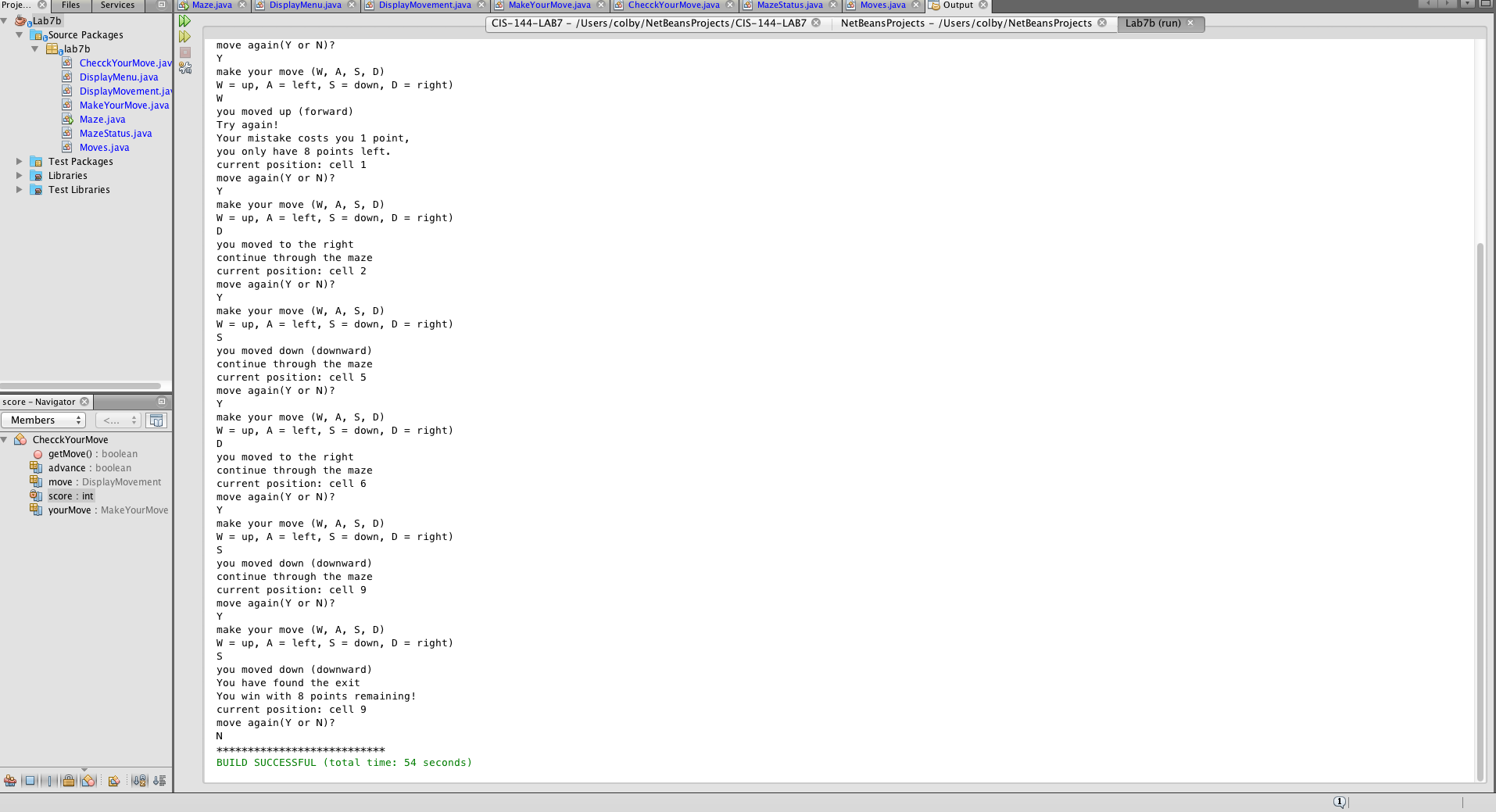
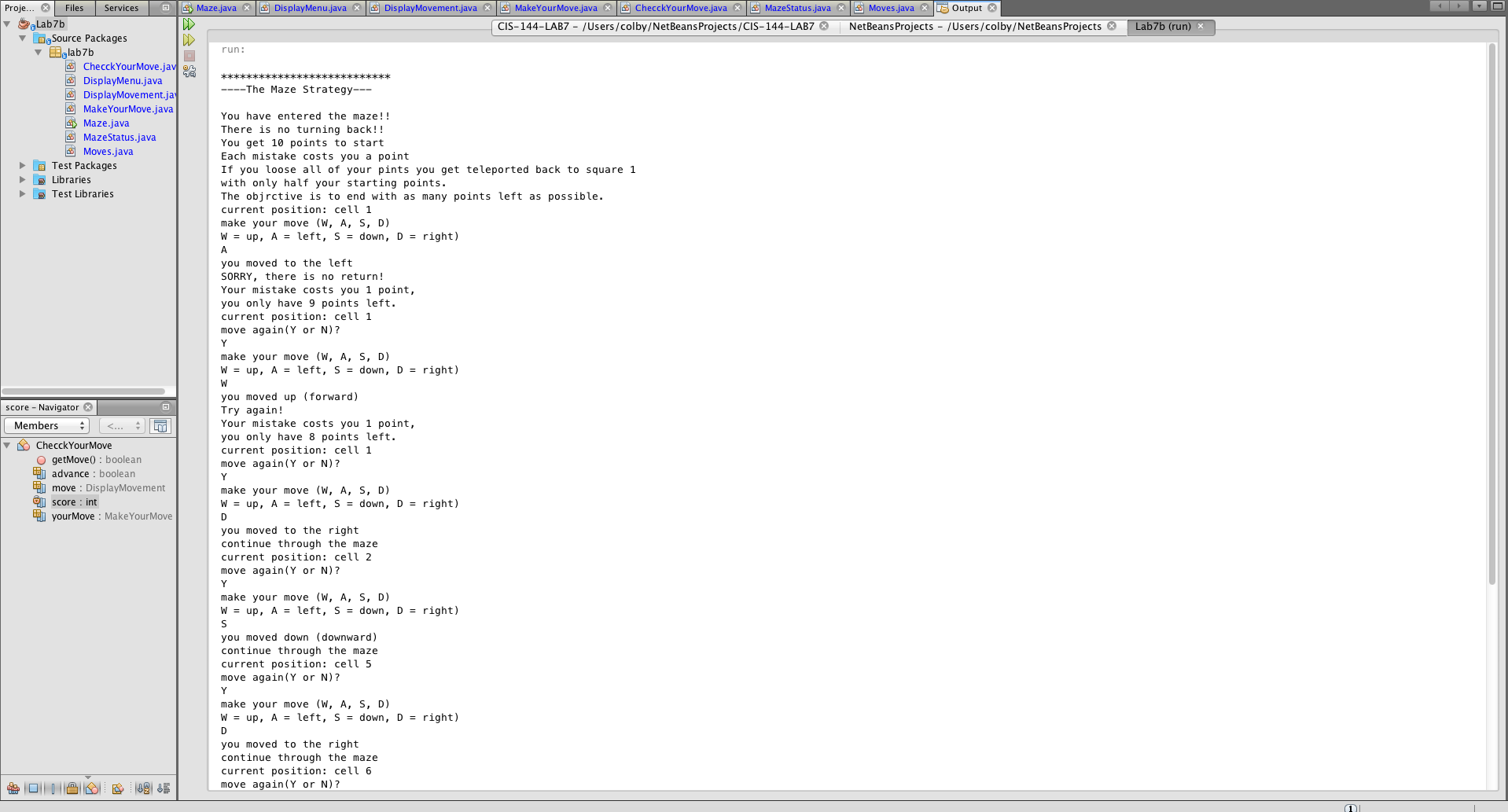
Colby Underhill

Lab 7

10/27/2016

CIS 144 JL



I decided to try my hand at object oriented programming, and split this up into individual class files. It’s a bit of a mess right now but it works, and was a good learning experience. …

I also added a scoring system. The player starts out with 10 points and looses a point for every mistake they make. If they reach 0 points they grt transported back to the start and are givrn only half their original points this time. The objective is to end the game loosing as few points as possible.

package lab7b;

/\*\*

\*

\* @author colby

\*/

import java.util.Scanner;

//Colby Underhill, Programmer

public class Maze

{

static DisplayMenu menu = new DisplayMenu();

static Scanner sc = new Scanner(System.in);

static MazeStatus status = new MazeStatus();

static Moves moves = new Moves();

static DisplayMovement move = new DisplayMovement();

static MakeYourMove yourMove = new MakeYourMove();

static ChecckYourMove checkMove = new ChecckYourMove();

static char myMove = '\0';

static boolean checkThis = false;

public static void main(String args[])

{

char answer = 'Y';

menu.getMenu();

while(answer == 'Y' || answer == 'y')

{

move.getMovement();

yourMove.getYourMove();

checkThis = checkMove.getMove();

status.status();

System.out.println("move again(Y or N)?");

answer = sc.next().charAt(0);

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}// end main() method

}

package lab7b;

/\*\*

\*

\* @author colby

\*/

public class DisplayMenu

{

public void getMenu()

{

System.out.println("");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("----The Maze Strategy---");

System.out.println("");

}// end method displayMenu()

}

package lab7b;

/\*\*

\*

\* @author colby

\*/

public class DisplayMovement

{

static int score = 10;

static int currentCell = 0;

boolean advance = true;

static MazeStatus status = new MazeStatus();

public int getScore()

{

return score;

}

public void setScore(int points)

{

score = points;

}

public int getCurrentCell()

{

return currentCell;

}

public void setCurrentCell(int cell)

{

currentCell = cell;

}

public boolean getAdvance()

{

return advance;

}

public void setAdvance(boolean go)

{

advance = go;

}

public void getMovement()

{

if(currentCell == 0)

{

System.out.println("You have entered the maze!!");

System.out.println("There is no turning back!!");

System.out.println("You get " + getScore() +" points to start");

System.out.println("Each mistake costs you a point");

System.out.println("If you loose all of your pints you get teleported back to square 1" );

System.out.println("with only half your starting points." );

System.out.println("The objrctive is to end with as many points left as possible.");

setCurrentCell(1);

status.status();

setAdvance(true);

}

System.out.println("make your move (W, A, S, D)");

System.out.println("W = up, A = left, S = down, D = right)");

}// end method displayMovement()

}

package lab7b;

import java.util.Scanner;

/\*\*

\*

\* @author colby

\*/

public class MakeYourMove

{

static Moves move = new Moves();

static Scanner sc = new Scanner(System.in);

static char myMove = '\0';

public char getMyMove()

{

return myMove;

}

public void setMyMove(char move)

{

myMove = move;

}

public void getYourMove()

{

setMyMove(sc.next().charAt(0));

switch(myMove)

{

case 'W': { move.MoveUp(); break; }

case 'A': { move.MoveLeft(); break; }

case 'S': { move.MoveDown(); break; }

case 'D': { move.MoveRight(); break; }

}

}// end method makeYourMove()

}

package lab7b;

/\*\*

\*

\* @author colby

\*/

public class ChecckYourMove

{

static DisplayMovement move = new DisplayMovement();

static MakeYourMove yourMove = new MakeYourMove();

static boolean advance=true;

private static int score = 10;

public boolean getMove()

{// Advance on correct move, subtract points on wrong move.

if(move.getCurrentCell() == 1 && advance == true)

{

if (yourMove.getMyMove() == 'W')

{

move.setAdvance(false);

score -=1;

System.out.println("Try again!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'A')

{

move.setAdvance(false);

score -=1;

System.out.println("SORRY, there is no return!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'D')

{

move.setCurrentCell(2);

move.setAdvance(true);

System.out.println("continue through the maze");

return advance;

}

if (yourMove.getMyMove() == 'S')

{

move.setAdvance(false);

score -=1;

System.out.println("Try again!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

}

if(move.getCurrentCell() == 2 && advance == true)

{

if (yourMove.getMyMove() == 'W')

{

move.setAdvance(false);

score -=1;

System.out.println("Try again!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'A')

{

move.setAdvance(false);

score -=1;

System.out.println("SORRY, there is no return!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'D')

{

move.setAdvance(false);

score -=1;

System.out.println("Try again!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'S')

{

move.setCurrentCell(5);;

move.setAdvance(true);

System.out.println("continue through the maze");

return advance;

}

}

if(move.getCurrentCell() == 5 && advance == true)

{

if (yourMove.getMyMove() == 'W')

{

move.setAdvance(false);

score -=1;

System.out.println("SORRY, there is no return!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'A')

{

move.setAdvance(false);

score -=1;

System.out.println("Try again!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'D')

{

move.setCurrentCell(6);

move.setAdvance(true);

System.out.println("continue through the maze");

return advance;

}

if (yourMove.getMyMove() == 'S')

{

move.setAdvance(false);

score -=1;

System.out.println("Try again!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

}

if(move.getCurrentCell() == 6 && advance == true)

{

if (yourMove.getMyMove() == 'W')

{

move.setAdvance(false);

score -=1;

System.out.println("SORRY, there is no return!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'A')

{

move.setAdvance(false);

score -=1;

System.out.println("SORRY, there is no return!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'D')

{

move.setAdvance(false);

score -=1;

System.out.println("Try again!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'S')

{

move.setCurrentCell(9);

move.setAdvance(true);

System.out.println("continue through the maze");

return advance;

}

}

if(move.getCurrentCell() == 9 && advance == true)

{

if (yourMove.getMyMove() == 'W')

{

move.setAdvance(false);

score -=1;

System.out.println("SORRY, there is no return!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'A')

{

move.setAdvance(false);

score -=1;

System.out.println("SORRY, there is no return!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'D')

{

move.setAdvance(false);

score -=1;

System.out.println("Try again!");

System.out.println("Your mistake costs you 1 point,");

System.out.println("you only have " + score + " points left.");

return advance;

}

if (yourMove.getMyMove() == 'S')

{

move.setCurrentCell(9);

move.setAdvance(true);

System.out.println("You have found the exit");

System.out.println("You win with " + score + " points remaining!");

return advance;

}

}

//If player makes too many mistakes, they get transported back to the start

// and get just 5 points to start with this time.

if (score <= 0)

{

move.setCurrentCell(1);

advance = true;

System.out.println("You have lost all points");

System.out.println("You have been transported back to square 1");

System.out.println("You get 5 points to work with this time.");

score = 5;

return advance;

}

return advance;

// end program menu

}// end method checkYourMove()

}

package lab7b;

/\*\*

\*

\* @author colby

\*/

public class MazeStatus

{

static DisplayMovement movement = new DisplayMovement();

public void status()

{

System.out.println("current position: cell " + movement.getCurrentCell());

}//end method

}

package lab7b;

/\*\*

\*

\* @author colby

\*/

public class Moves

{

public void MoveLeft()

{

System.out.println("you moved to the left");

}//end method

public void MoveRight()

{

System.out.println("you moved to the right");

}//end method

public void MoveUp()

{

System.out.println("you moved up (forward)");

}//end method

public void MoveDown()

{

System.out.println("you moved down (downward)");

}//end method

}