Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

FileLoader	`public void loadFile(String fileName)`
	`{`
	`try`
	`{`
	`BufferedReader in = new BufferedReader(new FileReader(fileName));`
	`String x;`
	`int lineNum=0;`
	`while (( x = in.readLine()) != null)`
	`{`
	`MatrixLoader(x,lineNum);//pass the Matrix loader method the line and the line number for parsing.`
	`lineNum++;//we will use the line number later in this class`
	`}`
	`}//end try`
	`catch (IOException e)`
	`{`
	`JFrame frame = new JFrame("Alert");`
	`JOptionPane.showMessageDialog(frame, "Ooops IOException error, i did it again!"`
	`+ e.getMessage());`
	`}//end catch`
	`}//end load file method`
		2	0

FileLoader	`public void MatrixLoader(String fileTextLine, int lineNum)throws gameFileError`
	`{`
	`// exitCount=0;//we must reset our variables to zero for the next level.`
	`int sum=0;`
	`char textVar;`
	`if(lineNum == 0)//it is the first line of the maze file, create The Matrix based on first line of the maze file`
	`{`
		7	2

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	for(int i=0; i<fileTextLine.length();i++)
	{
	if(fileTextLine.charAt(i) ==' ')//find blank area on first line number
	sum+=1;//how many blank spaces between the size of the matrix aka 4 6 or 5 7
	}
	int locationOfSpace = fileTextLine.indexOf(" ");//still handling that possible blank space in the matrix size in the file
	String c1=fileTextLine.substring(0,locationOfSpace);//see above
	String r1=fileTextLine.substring(locationOfSpace+sum,fileTextLine.length());//see above
	column = Integer.parseInt(c1);
	row = Integer.parseInt(r1);
	GameMatrix=new String[row][column];//create new matrix based on the size from the file
	}//end if
	else
	for(int i=0; i< fileTextLine.length();i++)//it is not the first line of the maze file
	{
	textVar = fileTextLine.charAt(i); //grab the individual charaters from the string.
	if(textVar == '.')//change . to N, so we dont have any goofy file system problems
	textVar='N';
	String textVar1= "" + textVar;
	if(textVar == 'E')//log the position of the exit for later use
	{
	exitXCord = lineNum-1;
	exitYCord =i;
	// textVar='W';
	textVar1= "" + textVar;//turn the exit into a wall
	}
	GameMatrix[lineNum-1][i]=textVar1;//load the matrix with values,aka N,W,D,H,etc
	}//end for loop
	}//end matrixloader method		

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

FileLoader	`public String[][] getGameMatrix()`
	`{`
	`int exitCount=0;`
	`int i1=0;`
	`int j1=0;`
	`int playerCount=0;`
	`for (int i = 0; i < GameMatrix.length; i++) {`
	`for (int j = 0; j < GameMatrix[i].length; j++) {`
	`if(GameMatrix[i][j].equals("P"))`
	`{`
	`playerCount+=1;`
	`}`
	`else if(GameMatrix[i][j].equals("E"))`
	`{`
	`exitCount+=1;`
	`i1=i;`
	`j1=j;`
	`}`
	`System.out.println(playerCount + "playerCount");`
	`System.out.println(exitCount + "playerCount");`
	`}}//end double for loop`
	`if(playerCount >1 || exitCount>1)`
	`{`
	`throw new gameFileError();`
	`}`
	`else`
	`GameMatrix[i1][j1]="W";`
	`return GameMatrix;`
	`}//end getGameMatrix method`
		6	3

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

FileLoader	`public int getMatrixSizeColumn()//return the matrixsize-column`
	`{`
	`  return column;`
	`}`
		1	0

FileLoader	`public int getMatrixSizeRow()//return the matrix size-row`
	`{`
	`  return row;`
	`}`
		1	0

FileLoader	`public int ExitXCord() //return the X cordinates for the Exit`
	`{`
	`  return exitXCord;`
	`}`
		1	0

FileLoader	`public int ExitYCord()//return the Y cordinates for the Exit`
	`{`
	`  return exitYCord;`
	`}`
		1	0

FileLoader	`public int dimondCount()`
	`{`
	`  int totalDimonds=0;`
	`  for (int i = 0; i < GameMatrix.length; i++){`
	`  for(int j = 0; j < GameMatrix[i].length; j++){`
	`  if(GameMatrix[i][j].equals("D") || GameMatrix[i][j].equals("H"))`
	`  totalDimonds+=1;`
	`  }}`
	`  return totalDimonds;`
	`  }`
		4	2

gameFileError	`public gameFileError()`
	`  {`
	`  JFrame frame = new JFrame("Alert");`
	`  JOptionPane.showMessageDialog(frame, "Your maze file ether had more than one player, or more than one exit.");}`
		1	0

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

TimeCalculator	`public void calcTimeforMaze(int totalDimonds, int xSize, int ySize)`
	`{`
	`if(xSize/ySize < 1)`
	`{`
	`minutes+=(ySize/xSize)*1+1;`
	`}`
	`else`
	`minutes+=(ySize/xSize)*1+1;`
	`if(totalDimonds >6 && totalDimonds*.10 + seconds <= 60)`
	`minutes+=(ySize/xSize)*1+1;`
	`else`
	`{`
	`minutes+=1;`
	`}`
	`if(minutes ==0)`
	`minutes=2;`
	`}`
		4	0

TimeCalculator	`public int getMinutes()`
	`{`
	`return minutes;`
	`}`
		1	0

TimeCalculator	`public int getSeconds()`
	`{`
	`return seconds;`
	`}`
		1	0

TimeKeeper	`public void TimeKeeper(int min, int sec)`
	`{`
	`if(sec + seconds <=60)`
	`{`
	`minutes+=min1`
	`seconds=sec+seconds;`
	`}`
	`else`
	`{`
		2	0

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	minutes+=min;
	minutes+=1*((sec+seconds)/60);
	seconds=(sec+seconds)%60;
	}
	}		

TimeKeeper	public int getMinutes()
	{
	return minutes;
	}
		1	0

TimeKeeper	public int getSeconds()
	{
	return seconds;
	}
		1	0

TheArchitect	public void setExit(int x, int y)
	{
	WallXCord=x;
	WallYCord=y;
	}
		1	0

TheArchitect	public void showWall()
	{
	updatedMatrix[WallXCord][WallYCord]="E";
	}
		1	0

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

TheArchitect	`public void playerMove(int xScale, int yScale, String[][] currentMatrix,int totalDimonds)throws StupidAssMove`
	`{`
	`int x=0;`
	`int y=0;`
	`int found=0;`
	`globalTotalDimonds=totalDimonds; //use this later for the gui dimond count`
	`nextLevel(false); //dont go to the next level yet.`
	`String[][] junkMatrix=currentMatrix;//we will be updating currentMatrix`
	`for (int i = 0; i < currentMatrix.length; i++) //for loop will find were the player is now`
	`{`
	`for (int j = 0; j < currentMatrix[i].length; j++)`
	`{`
	`if(currentMatrix[i][j].equals("P"))//we found the player`
	`{`
	`x=i;//record the players position`
	`y=j;`
	`found = 1;`
	`break;`
	`}`
	`}}//end both for loops`
	`if(currentMatrix[x+xScale][y+yScale].equals("H"))//its a hidden dimond`
	`{`
	`currentMatrix[x][y]="N";`
	`currentMatrix[x+xScale][y+yScale]="P";`
	`currentMatrix[x][y]="N";`
	`collected+=1;//we got a hidden dimond! wow!`
	`}`
	`else if(currentMatrix[x+xScale][y+yScale].equals("D"))//its a dimond`
	`{`
	`currentMatrix[x][y]="N";`
	`currentMatrix[x+xScale][y+yScale]="P";`
	`collected+=1;//we got a dimond`
	`}`
	`else if(currentMatrix[x+xScale][y+yScale].equals("M") && currentMatrix[x+(xScale*2)][y+(yScale*2)].equals("N"))`
	`{`
	`currentMatrix[x][y]="N";`
	`currentMatrix[x+xScale][y+yScale]="P";`
	`currentMatrix[x+(xScale*2)][y+(yScale*2)]="M";`
	`}`
	`else if (currentMatrix[x+xScale][y+yScale].equals("N"))//normal move foward onto nothing`
	`{`
	`currentMatrix[x][y]="N";`
		10	4

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	currentMatrix[x+xScale][y+yScale]="P";
	}
	else if (currentMatrix[x+xScale][y+yScale].equals("E"))//its an exit
	{
	currentMatrix[x][y]="N";
	currentMatrix[x+xScale][y+yScale]="P";
	nextLevel(true);//allow the next level to be loaded.
	}
	else
	throw new StupidAssMove("Ass Hole hit wall!");
	
	if(collected==totalDimonds)//if we have all the dimonds give the player the exit
	showWall();
	
	updatedMatrix=currentMatrix; //we will return updatedMatrix for the gui
	}//end method		

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

TheArchitect	`public void nextLevel(boolean tOrF)//true we go to next level, false we update current level's gui`
	`{`
	`level=tOrF;`
	`}`
		1	0

TheArchitect	`public boolean getLevel()//returs level true or false`
	`{`
	`return level;`
	`}`
		1	0

TheArchitect	`public int getDimondsLeft()`
	`{`
	`return globalTotalDimonds-collected;//for GUI JLabel, show how many dimonds are left to be collected`
	`}`
		1	0

TheArchitect	`public String[][] getUpdatedMatrix()//returns the updated matrix for the gui to display`
	`{`
	`return updatedMatrix;`
	`}`
		1	0

StupidAssMove	`public StupidAssMove(String event)`
	`{`
	`JFrame frame = new JFrame("Warning");`
	`JOptionPane.showMessageDialog(frame, "You Stupid Ass, Ran into something did you?");`
	`}`
		1	0

HighScore	`public void addHighScore(String name, int min, int sec,int level)`
	`{`
	`try{`
	`String outData="PlayerName: "+name+" Total Time for Levels:"+min+":"+sec+ "(Minutes:Seconds)"+`
	`"Level Reached:*" + level;`
	`PrintWriter out = new PrintWriter(new FileOutputStream("scores.txt",true));`
	`out.println("");`
	`out.println(outData);`
	`out.close();`
	`}//prints the highscore data to scores.txt`
	`catch(Exception ex){`
	`System.out.println(ex);`
	`}//end catch`
	
	`}//end addHighScore`
		1	0

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

MyKeyHandler	`public void keyPressed (KeyEvent theEvent)`
	`{`
	`switch (theEvent.getKeyCode())`
	`{`
	`case KeyEvent.VK_UP:`
	`{`
	`theArc.playerMove(-1,0,scrapMatrix,fl.dimondCount());//let the Architect know we moved, along with the current matrix`
	`loadMatrixGui("updateLoad");//reload the gui to show the move`
	`if (theArc.getLevel()==true)`
	`{`
	`nextLevelLoad();//if the player hit an exit door, load the next level`
	`}`
	`break;`
	`}`
	`case KeyEvent.VK_DOWN:`
	`{`
	`theArc.playerMove(1,0,scrapMatrix,fl.dimondCount());//see above`
	`loadMatrixGui("updateLoad");//see above`
	`if (theArc.getLevel()==true)//see above`
	`{`
	`nextLevelLoad();//see above`
	`}`
	`break;`
	`}`
	`case KeyEvent.VK_LEFT:`
	`{`
	`theArc.playerMove(0,-1,scrapMatrix,fl.dimondCount());//see above`
	`loadMatrixGui("updateLoad");//see above`
	`if (theArc.getLevel()==true)//see above`
	`{`
		8	2

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	nextLevelLoad();//see above
	}
	break;
	}
	case KeyEvent.VK_RIGHT:
	{
	theArc.playerMove(0,1,scrapMatrix,fl.dimondCount()); //see above
	loadMatrixGui("updateLoad");//see above
	if (theArc.getLevel()==true)
	{
	nextLevelLoad();//see above
	}
	break;
	}
	}//end switch
	JLabel mainLabel=new JLabel("Total Dimonds Left to Collect"+theArc.getDimondsLeft()+"", JLabel.CENTER);//show how many dimonds are left to collect on the gui!
	JPanel dimondsPanel = new JPanel();
	dimondsPanel.add(mainLabel);
	cp.add(dimondsPanel,BorderLayout.SOUTH);
	}//end method		

GameGui	public GameGui()
	{
	super("Maze, a game of wondering"); //call super to initilize title bar of G.U.I.
	cp=getContentPane();
	shagLabel = new JLabel("",new ImageIcon("yeababyyea.jpg"),JLabel.LEFT);//GUI background for initial load
	cp.add(shagLabel);
	//Add Exit & New Game Menu Items
	itemExit = new JMenuItem("Exit");
		1	0

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	itemExit.setAccelerator (KeyStroke.getKeyStroke (KeyEvent.VK_X, KeyEvent.CTRL_MASK));//press CTRL+X to exit if you want
	itemSaveScore = new JMenuItem("Save High Score");
	itemSaveScore.setAccelerator (KeyStroke.getKeyStroke (KeyEvent.VK_S, KeyEvent.CTRL_MASK));//press CTRL+S to save high score if you want
	itemHighScore=new JMenuItem("High Score");
	itemHighScore.setAccelerator (KeyStroke.getKeyStroke (KeyEvent.VK_H, KeyEvent.CTRL_MASK));//press CTRL+H to view high score if you want
	itemEnterName = new JMenuItem("Enter Player Name");
	itemEnterName.setAccelerator (KeyStroke.getKeyStroke (KeyEvent.VK_N, KeyEvent.CTRL_MASK));//press CTRL+N to enter your name if you want
	newGameItem = new JMenuItem("New Game");
	openFileItem = new JMenuItem("Open Maze File.");
	openFileItem.setAccelerator (KeyStroke.getKeyStroke (KeyEvent.VK_O, KeyEvent.CTRL_MASK));//press CTRL+O to open a level if you want
	newGameItem.setActionCommand("New Game");
	newGameItem.addActionListener(this);
	itemEnterName.setActionCommand("EnterName");
	itemEnterName.addActionListener(this);
	itemSaveScore.setActionCommand("SaveScore");
	itemSaveScore.addActionListener(this);
	itemHighScore.setActionCommand("HighScore");
	itemHighScore.addActionListener(this);
	itemExit.setActionCommand("Exit");
	itemExit.addActionListener(this);
	openFileItem.setActionCommand("Open");
	openFileItem.addActionListener(this);
	newMenu = new JMenu("File");
	newMenu.add(newGameItem);
	newMenu.add(itemEnterName);
	newMenu.add(openFileItem);		

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	newMenu.add(itemHighScore);
	newMenu.add(itemSaveScore);
	newMenu.add(itemExit);
	//Add Exit Menu Item
	//Add Menu Bar
	menuBar = new JMenuBar();
	menuBar.add(newMenu);
	setJMenuBar(menuBar);
	//Add Menu Bar
	newPanel = new JPanel();
	hs = new HighScore();
	tk=new TimeKeeper();
	pack();
	setVisible (true);//show our menu bar and shagLabel.. Yea baby Yea! Whoa.. to much java.
	}//end constructor		

GameGui	public void actionPerformed(ActionEvent e)
	{
	if (e.getActionCommand().equals("Exit"))//exit on the menu bar
	{
	new Timer(1000, updateCursorAction).stop();
	System.exit(0); //exit the system.
	}
	else if (e.getActionCommand().equals("New Game"))//new game on the menu bar
	{
	return; //maybe implent this feature later
	}//end New Game Command
	else if(e.getActionCommand().equals("EnterName"))//Allows user to enter their name for high score
	{
	JOptionPane optionPane = new JOptionPane();
	playerName=optionPane.showInputDialog("Please Enter your Earth Name");
	}
		5	5

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	else if(e.getActionCommand().equals("HighScore"))//Displays the high scores
	{
	ScoreGui sg = new ScoreGui();
	sg.ScoreGui();
	}
	else if(e.getActionCommand().equals("SaveScore"))//allows the user to save their score at any time.
	{
	hs.addHighScore(playerName,tk.getMinutes(),tk.getSeconds(),levelNum);
	}
	else if(e.getActionCommand().equals("Open"))//to start the game you have to open a maze file. this is on the menu
	{
	JFileChooser chooser = new JFileChooser();
	int returnVal = chooser.showOpenDialog(this);
	if(returnVal == JFileChooser.APPROVE_OPTION)
	{
	fl.loadFile(chooser.getSelectedFile().getName());//load the file we need
	theArc.setExit(fl.ExitXCord(),fl.ExitYCord());
	loadMatrixGui("newLoad");
	}
	}
	}//end actionPerformed method		

GameGui	public void loadMatrixGui(String event)
	{
	if (event == "newLoad")
	{
	remove(newPanel);//remove the previous level's game from the screen
	if(progBarPanel !=null)//remove the progress bar from the gui as long as its already been created.
	remove(progBarPanel);
		8	2

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	String[][] temp = fl.getGameMatrix();
	scrapMatrix = new String[fl.getMatrixSizeRow()][fl.getMatrixSizeColumn()];
	for (int i = 0; i < scrapMatrix.length; i++){
	for (int j = 0; j < scrapMatrix[i].length; j++){
	scrapMatrix[i][j]= temp[i][j];//create a new matrix so we dont have a refrence to another objects matrix!
	}
	}//end double for loop
	timeCalc = new TimeCalculator();//create the time calculator used to determine how much time each level is given.
	
	timeCalc.calcTimeforMaze(fl.dimondCount(),fl.getMatrixSizeRow(),fl.getMatrixSizeColumn());//let time calculator know the parameters of the game
	timeLeft=timeCalc.getMinutes();//get the minutes allowed for the level
	ix=timeCalc.getSeconds();//get the seconds allowed for the level;
	jx=0;//reset the variable used for keeping time to zero since its a new level
	timely = new Timer(1000,updateCursorAction);//create a timer to update the progress bar
	timely.start();//start the timer
	progBarPanel = new JPanel();//panel for progress bar
	progressBar = new JProgressBar(0, timeCalc.getMinutes()*100);//minutes returns a single digit, we have to multiply it for Bar.
	progressBar.setStringPainted(true);
	progBarPanel.add(progressBar);
	cp.add(progBarPanel,BorderLayout.NORTH);
	newPanel = new JPanel();
	newPanel.setLayout(new GridLayout(fl.getMatrixSizeRow(),fl.getMatrixSizeColumn()));//set our panel for the game to the size of the matrix
	labelMatrix=new JLabel[fl.getMatrixSizeRow()][fl.getMatrixSizeColumn()];
	newPanel.addKeyListener(new MyKeyHandler());		

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	}//end if
	else
	if(event =="updateLoad")//every time the player moves the gui must be updated.
	{
	scrapMatrix = theArc.getUpdatedMatrix();//get the new matrix to be displayed from the architect
	remove(newPanel);//remove the old game
	newPanel = new JPanel();
	newPanel.setLayout(new GridLayout(fl.getMatrixSizeRow(),fl.getMatrixSizeColumn()));
	newPanel.addKeyListener(new MyKeyHandler());
	newPanel.grabFocus();
	}
	for (int i = 0; i < labelMatrix.length; i++){
	for (int j = 0; j < labelMatrix[i].length; j++){
	labelMatrix[i][j]= mo=new mazeObject(scrapMatrix[i][j]);//add our maze images into the gui
	}
	}//end double for loop
	cp.add(newPanel);
	remove(shagLabel);//remove the constructors initial background
	System.gc();//force java to clean up memory use.
	pack();
	setVisible (true);
	newPanel.grabFocus();
	}//end loadMatrixGui method		

GameGui	public void nextLevelLoad()
	{
	levelNum+=1;
		1	0

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	tk.TimeKeeper(timeLeft,ix);//The TimeKeeper object keeps a running tab of the total time the player has used.(for high score)
	timely.stop();//dont count while we are loading the next level.
	theArc = new TheArchitect();//flush everything from TheArchitect so we dont get goffee results
	catFileName+=01;//the next file to be loaded (number)
	String fileName="level"+catFileName+".maz";
	System.gc();
	fl.loadFile(fileName);//load the file we need
	scrapMatrix=fl.getGameMatrix();//get the new matrix from the fileloader for the next level.
	theArc.setExit(fl.ExitXCord(),fl.ExitYCord());
	loadMatrixGui("newLoad");
	}		

GameGui	public void actionPerformed(ActionEvent e)throws SlowAssPlayer //this inner class generates an exeption if the player takes to long to finish a level
	{
	ix-=1;
	jx+=1;
	if(ix<0)
	{
	ix=60;
	timeLeft-=1;
	}
	if(timeLeft==0 && ix==0)
	{
	timely.stop();
	JLabel yousuckLabel = new JLabel("",new ImageIcon("yousuck.jpg"),JLabel.LEFT);
	cp.add(yousuckLabel);
	remove(newPanel);
		5	1

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	remove(progBarPanel);
	pack();
	setVisible (true);
	timely.stop();
	catFileName-=01;
	if(catFileName<01)
	throw new SlowAssPlayer("Slow ass took to long.");
	else
	loadMatrixGui("newLoad");
	}//end first if
	progressBar.setValue(jx);
	progressBar.setString(timeLeft+":"+ix);
	}//end actionPerformed		

mazeObject	public mazeObject(String fileName)
	{
	fileName+=".png";
	JLabel fancyLabel;
	fancyLabel = new JLabel("",new ImageIcon(fileName),JLabel.LEFT);
	newPanel.add(fancyLabel);
	}
		1	0

SlowAssPlayer	public SlowAssPlayer(String event)
	{
	//the game is over, here we must tell our high score method to recond the details.
	hs.addHighScore(playerName,tk.getMinutes(),tk.getSeconds(),levelNum);
	JFrame frame = new JFrame("Warning");
	JOptionPane.showMessageDialog(frame, "You Stupid Ass, Did you eat to much for dinner? Move Faster!");//the entire game has ended.
	}
		1	0

ScoreGui	public ScoreGui() {	1	0

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento
	super(); }		

ScoreGui	public void ScoreGui()//the ScoreGui Method displays the scores in order from lowest to highest.
	{
	Container cp = getContentPane();
	JButton ok = new JButton("OK");
	ok.setActionCommand("OK");
	ok.addActionListener(this);
	int lineNum=0;
	cp.add(ok,BorderLayout.SOUTH);
	try{
	String line = "";
	String[] myScoreArray = new String[100];
	for(int i=0; i<myScoreArray.length;i++)
	myScoreArray[i]=" ";
	String line1="";
	BufferedReader br1 = new BufferedReader(new InputStreamReader(new FileInputStream("scores.txt")));//read in the scores data
	int recordsCount=0;
	while((line = br1.readLine()) != null)
	{
	line = br1.readLine();
	if(line!="")
	{
	recordsCount+=1;
	int tempPOS = line.indexOf("*");//use the star to indicate the next charator is going to be the maze level which we will sort by.
	String pos=line.substring(tempPOS+1);
	int index = Integer.parseInt(pos);
	if(myScoreArray[index] == " ")
	myScoreArray[index]=line;//add in the score to the array.
		9	3

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	else
	{
	for(int i=0; i<myScoreArray.length;i++)
	{
	if(index+i<myScoreArray.length)//prevent array out of bounds errors.
	{
	if(myScoreArray[index+i].equals(" "))
	{
	myScoreArray[index+1]=line;//add in a score to the next available area of the array
	}
	}//end first if
	}//end for loop
	}//end else
	JPanel scorePanel = new JPanel();
	scorePanel.setLayout(new GridLayout(recordsCount,recordsCount));
	for(int i=0; i<myScoreArray.length;i++)
	{
	if(myScoreArray[i]!=" ")
	{
	mainLabel=new JLabel(myScoreArray[i], JLabel.LEFT);//display the score on the screen
	scorePanel.add(mainLabel);
	}
	}//end for loop
	cp.add(scorePanel);
	}//end very first if
	}//end first while loop
	}//end try
	catch(IOException ex) {
	JFrame frame = new JFrame("Alert");		

Nombre de la clase	Cuerpo de la función	Complejidad ciclomática	Nivel de anidamiento

	JOptionPane.showMessageDialog(frame, "Problem with scores.txt file. Cant load high Scores");
	}//end catch
	pack();
	setVisible (true);
	}//end constructor		

ScoreGui	public void actionPerformed(ActionEvent e)
	{
	dispose();
	}
		1	0