CS 48 Computer Science Project

G02: Puzzle Me

Final Project

Winter, 2013

Charles Weng

Aura Roy

Cameron McNair

Julio Lemus

Zhuocheng Yu

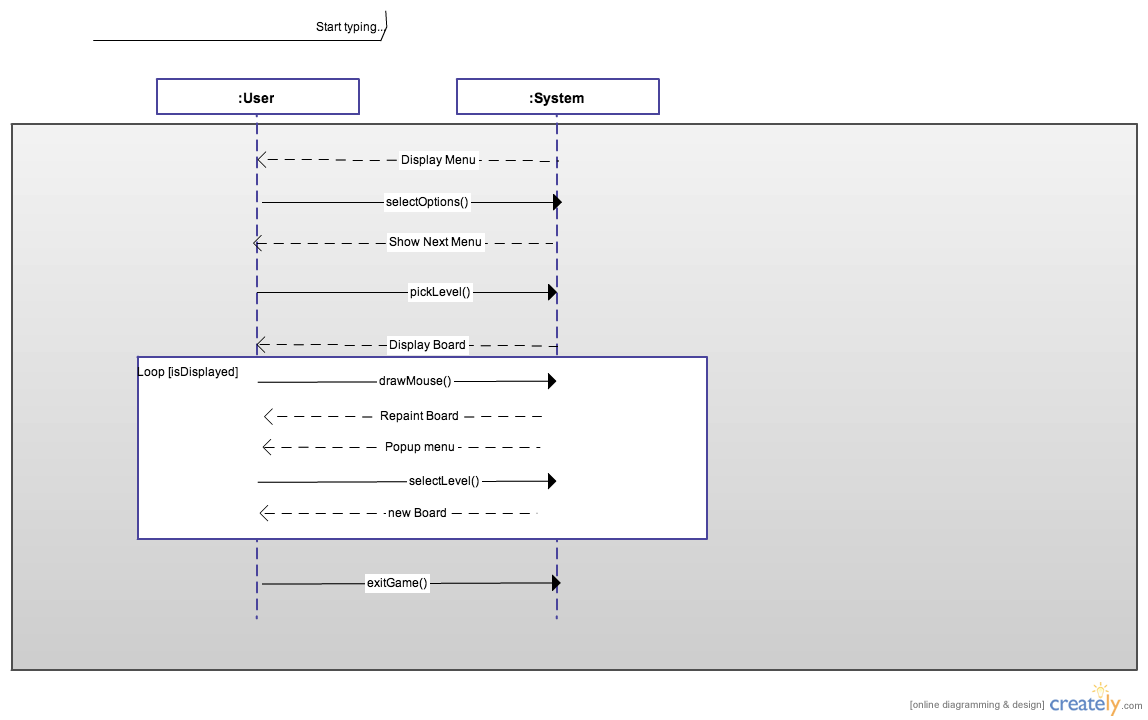
**TA: Cetin Sahin**

**Project Description**:

Our group is PuzzleMe and we made the game Connect The Dots. It is similar to the game Flow Free. It is a game where there are different colored dots on the board and the user must connect the same colored dots on the board and fill up the entire board. Our vision was to make a game that was thoughtful and entertaining without having to have violence, as a lot of games nowadays do. We decided to do Connect The Dots because any age group can play, and we can have different levels that could match different abilities.

**Domain Analysis:**  
  
This is our System Sequence Diagram. This shows our only use case.

System Sequence Diagram:



These are our Contracts. They describe the pre-conditions and the post-conditions of the functions used in our program.

***System Operations of the Connect the Dots Use Case***

Class Name: Board

Contract CO1: Board()

|  |
| --- |
| **Operation**; Board()  **Cross Reference**: Use Case: Connecting the Dots  **Preconditions**: None  **Postconditions**: - Board Class instance variables were initialized.  -object of the Class Maps was used within the operation to initialize  instance variable fbf. (association formed)  -object of Class Line was initialized withing operation with  Board Class instance variables.(association formed) |

Contract CO2: paintComponet(Graphics g)

|  |
| --- |
| **Operations:** paintComponent()  **Cross Reference:**  Use Case: Connecting the Dots  **Preconditions:** Game is started    **Postconditions:** -painted the board,dots, and lines on the grid.  -used method eraseBox method from line class to enable back dragging  when line is drawn by user.(association formed)  -drawHorizontal method from line was used if drawSinal instance  variable from line class is equal to 0 and drawVerticle method was used  if drawSignal is equal to 1. |

Contract CO3: drawDots()

|  |
| --- |
| **Operations:** Draw\_Dots()  **Cross Reference:** Use Case:Connecting the Dots  **Preconditions:** None  **Postconditions:** -used Map Class' object fbf to access the instance variable do\_counter.  (association formed)  - used operation convertToPixels(int num) to draw dots at specific  location(association formed)  - object of class Dot was initialized within the Draw\_Dots()  operation and object called operation drawObject from the Dot  class .(association formed) |

Class Name: Music

Contract CO1: run()

|  |
| --- |
| **Operation:** run()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game has been started  **Postconditions: -**string was created to hold audio file name.  -AudioStream was created.  -object input was used to open AudioStream by object clip of type  AudioStream.getClip(). Clip was also usded to start a loop of the  audio file. |

Class Name: Maps

Contract CO1: Maps()

|  |
| --- |
| **Operation:** Maps()  **Cross Reference:** Use Case: Connect the Dots  **Preconditions:** Game has been started.  **Postconditions: -**instance variables of the class Maps were initialized  -object colorMap of type HashMap is created to store different color  options for dots.  -arrays are created to store 9 by 9 dot positions and 5 by 5 dot  positions. |

ContractCO4: getNumberofFiles()

|  |
| --- |
| **Operation:** getNumberofFiles()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** None  **Postconditions: -**counted the number of files that store 5 by 5 maps in the map  package.  -instance variable numberOfFiles was incremented. |

ContractCO5: getNumberofFilesNBN()

|  |
| --- |
| **Operation:** getNumberofFilesNBN()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** None  **Postconditions: -**counted the number of files that store 9 by 9 maps in the map  package.  -instance variable numberOfFilesNBN was incremented. |

ContractCO6: storeFBF(int index)

|  |
| --- |
| **Operation:** storeFBF(int index)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** None  **Postconditions: -**used inputStreamReader to get information from text files and  store it in a string array.  -instance variable size was incremented and dot positions was stored  based on information form text files. |

ContractCO6: storeNBN(int indexNBN)

|  |
| --- |
| **Operation:** storeNBN(int indexNBN)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** None  **Postconditions: -**used inputStreamReader to get information from text files and  store it in a string array.  -instance variable size wasincremented and dot positions were stored  based on information form text files. |

Class Name: View

ContractCO1: initBoard()

|  |
| --- |
| **Operation:** initBoard()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game has been started  **Postconditions: -** removed all contents of previous JPanel and adds buttons to JPanel  - initialized entire board along with its GridLayout and buttons. |

ContractCO2: initMenus()

|  |
| --- |
| **Operation:** initMenu()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game has been started  **Postconditions: -** removed all contents of previous JPanel and adds buttons to JPanel.  **-** initialized entire menu along with its GridLayout and buttons. |

ContractCO3:initLevelMenus()

|  |
| --- |
| **Operation:** initLevelMenu()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game has been started  **Postconditions: -**new levelMenu object was created.  -used frame object to set dimensions of levelMenu. |

ContractCO4: initLeveSelection()

|  |
| --- |
| **Operation:** initLevelSelection()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game has been started  **Postconditions: -**board object is created and passed to levelSelection constructor  to initialize levelSelection instance in view.  -all content was removed from JPanel before first post condition. |

Note:initLevelSelection2 has same contract as LevelSelection.

Class Name: Menu

Contract C01: Menu()

|  |
| --- |
| **Operation:** Menu()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game has been started  **Postconditions: -**image object was initialized with as a new image icon containing  the menu's background image. |

Contract CO2: paintComponent(Graphics g)

|  |
| --- |
| **Operation:** paintComponent(Graphics g)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game has been started  **Postconditions: -**image object was used in operation to gather information about the  image and used that to draw the background image.  (association formed) |

Class Name: RefreshListener

Contract CO1: RefreshListener(Board board)

|  |
| --- |
| **Operation:** RefreshListener(Board board)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game has been started  **Postconditions: -**object of class Board was created and was reset to its original state.  (association formed)  -used to implement refresh button's action(association formed with  referesh button) |

Contract CO2: actionPerformed(ActionEvent e)

|  |
| --- |
| **Operation:** actionPerformed(ActionEvent e)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Mouse Clicked  **Postconditions: -**indicated when refresh button was clicked. |

Class Name: Dot

Contract CO1: Dot(int x,int y, Color color)

|  |
| --- |
| **Operation:** Dot(int x,int y, Color color)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** None  **Postconditions: -**initialized instance variables row, columnm and color that  represent the dot's location and color. |

ContractCO2: drawObject(Graphics graphics, int top\_leftX, int top\_leftY, int length, int width )

|  |
| --- |
| **Operation:** drawObject()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** None  **Postconditions: -**used to draw the dot with respect to instance variables. |

Contract CO3: getColor()

|  |
| --- |
| **Operation:** getColor()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** None  **Postconditions: -**got color from instance variable c. |

Class Name: Line

ContractCO1: Line(JPanel panel, int squares, int wholesize, int numPairs)

|  |
| --- |
| **Operation:** Line(JPanel panel, int squares, int wholesize, int numPairs)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Board and Maps are drawn  **Postconditions: -**new line object was created within the board constructor.  Association formed between Line and Board).  -used to set attributes like Wholesize,offset,boxsize,ect. |

ContractCO2: setLineColor(int i,int i)

|  |
| --- |
| **Operation:** setLineColor(int i,int i)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Mouse is being pressed  **Postconditions: -**index of color are retrieved and used.  -array boxColor row and column are set.  -stored current color of the area in the board(association formed)  -used instance variable currentColor to keep track of the colors.  (association formed) |

ContractCO3: setCoordinates(Point p)

|  |
| --- |
| **Operation:** setCoordinates(Point p)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Mouse is pressed  **Postconditions: -**used instance variables boxsize, offset, xPosition,and yPosition to  set the coordinates where the mouse is pressed on the board.  (association formed).  - used instance variables currentI and currentJ to check if a dot is  clicked on the board. (clickedADot instance variable was set to  true if the element in the index currentI and currentJ in array  objectsInSquares was 1) Association with board formed. |

ContractCO4: comparePosition(Point p)

|  |
| --- |
| **Operation:** comparePosition(Point p)  **Cross Reference:** Use Case:Connecting the Dots  **Preconditions:** Line is being drawn on Board  Note: postconditions are for 4 actions present in same operation  (Drawing up,down,left,right)  **Postconditions: -**positions of line were compared using instance variables  such as xPosistion.  -objectsInSquare array was updated to contain a dot and a line if  a matching color dot was found for the line drawn.  -updated the instance variable foundDot to equal true if dot color  matches the corresponding color stored for that square  (Used for signal to mouse action, association formed)  -numDotsFound was incremented to indicate that a dot had been  found that matched the line color. |

ContractCO5:redrawLines(Graphics g)

|  |
| --- |
| **Operation:** redrawLines(Graphics g)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game is being played and Board is initiated  **Postconditions: -**used to redraw lines on board and sets the color that the line  will be. Also checks when to draw horizontal, vertical, to the  right or to the left.(association with Board was formed).  -tested whether linePosition element was -2 which signals that  the line has finished drawing and a new color must be used for  the next line.  -instance variables startinX,startingY,endX,endY are set equal  depending on the change in position of the line and the value of  boxSize. |

ContractCO6:eraseBox(Graphics g)

|  |
| --- |
| **Operation:** eraseBox(Graphics g)  **Cross Reference:** Use Case: Connect the Dots  **Preconditions:** Line is being drawn.  **Postconditions: -**data related to line being drawn was deleted from the linePositon  array and eraseI and eraseJ are set to -2 also board was repainted(). |

ContractCO7:drawHorizontal(Graphics g)

|  |
| --- |
| **Operation:** drawHorizontal(Graphics g)  **Cross Reference:** Use Case: Connect the Dots  **Preconditions:** Line is being drawn.  **Postconditions: -**used to draw line horizontally depending on the current position.  -drawSignal is reset to 2 if boolean movinRight is true. If true,  xPosition is incremented by a factor of boxSize.  (association formed) |

ContractCO7;drawVertical(Graphics g)

|  |
| --- |
| **Operation:** drawHorizontal(Graphics g)  **Cross Reference:** Use Case: Connect the Dots  **Preconditions:** Line is being drawn.  **Postconditions: -**used to draw line vertically depending on the current position.  -drawSignal is reset to 2 if Boolean movinRight is true. If true,  yPosition is incremented by a factor of boxSize.  (association formed) |

Class Name: LevelSelection

ContractCO1: LevelSelection(View view,Board board)

|  |
| --- |
| **Operation:** LevelSelection(View view,Board board)  **Cross Reference:** Use Case: Connect the Dots  **Preconditions:** Game has started  **Postconditions: -**instances of the View and Board classes were initialized through the  value of parameters.  -the background of the JPanel was set to color black, the size of the  JPanel is set, a GridLayout was created for buttons, a border was set  around the Jpanel, the buttons created by operations addButtons()  are added and initialized, and the JPanel is set visible. |

ContractCO2: addButtons()

|  |
| --- |
| **Operation:** addButtons()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** LevelSelection is initialized  **Postconditions: -**array of buttons was filled with operation fillArray();  -added buttons to JPanel with for loop that loops through array of  buttons. |

ContractCO3: fillArray()

|  |
| --- |
| **Operation:** fillArray()  **Cross Reference:** Use Case: Connect the Dots  **Preconditions:** None  **Postconditions: -**array of buttons was filled through a for loop. |

ContractCO4: initButtons()

|  |
| --- |
| **Operation:** initButtons()  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** LevelSelection is initialized  **Postconditions: -**actionPerformed for buttons were initialized through Try-Catch. |

ContractCO5: levelSelect()

|  |
| --- |
| **Operation:** levelSelection()  **Cross Reference:** Use Case: Connect the Dots  **Preconditions:** None  **Postconditions: -**button actionListeners and actionPerfomed were implemented |

Note: Contracts for LevelSelection2 are the same as LevelSelection contracts.

Class Name: LevelMenu

ContractCO1: LevelMenu(View view)

|  |
| --- |
| **Operation:** LevelMenu(View view)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** None  **Postconditions: -**Mouse motion listeners and Mouse listeners were added.  -view object was initialized through parameters. |

ContractCO2: paintComponent(Graphics g)

|  |
| --- |
| **Operation:** paintComponent(Graphics g)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** None  **Postconditions: -**painted the components of levelMenu such as the 5 by 5 and 9 by 9  board selection.(Overriden operations) |

ContractCO3: mouseReleased(MouseEvent e)

|  |
| --- |
| **Operation:** mouseReleased(MouseEvent e)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Mouse has been pressed.  **Postconditions: -**LevelSelection was initialized or LevelSelection2 depending on  user's choice of board size.  -viewLevelSelection displayed LevelSelection screen or  LevelSelection2 depending on user choice. |

ContractCO4: mouseMoved(MouseEvent e)

|  |
| --- |
| **Operation:** mouseMoved(MouseEvent e)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Mouse is being moved  **Postconditions: -**level menu selection changed color when mouse was moved over it. |

Class Name: InstructionsJPanel

ContractCO1:paintComponent(Graphics g)

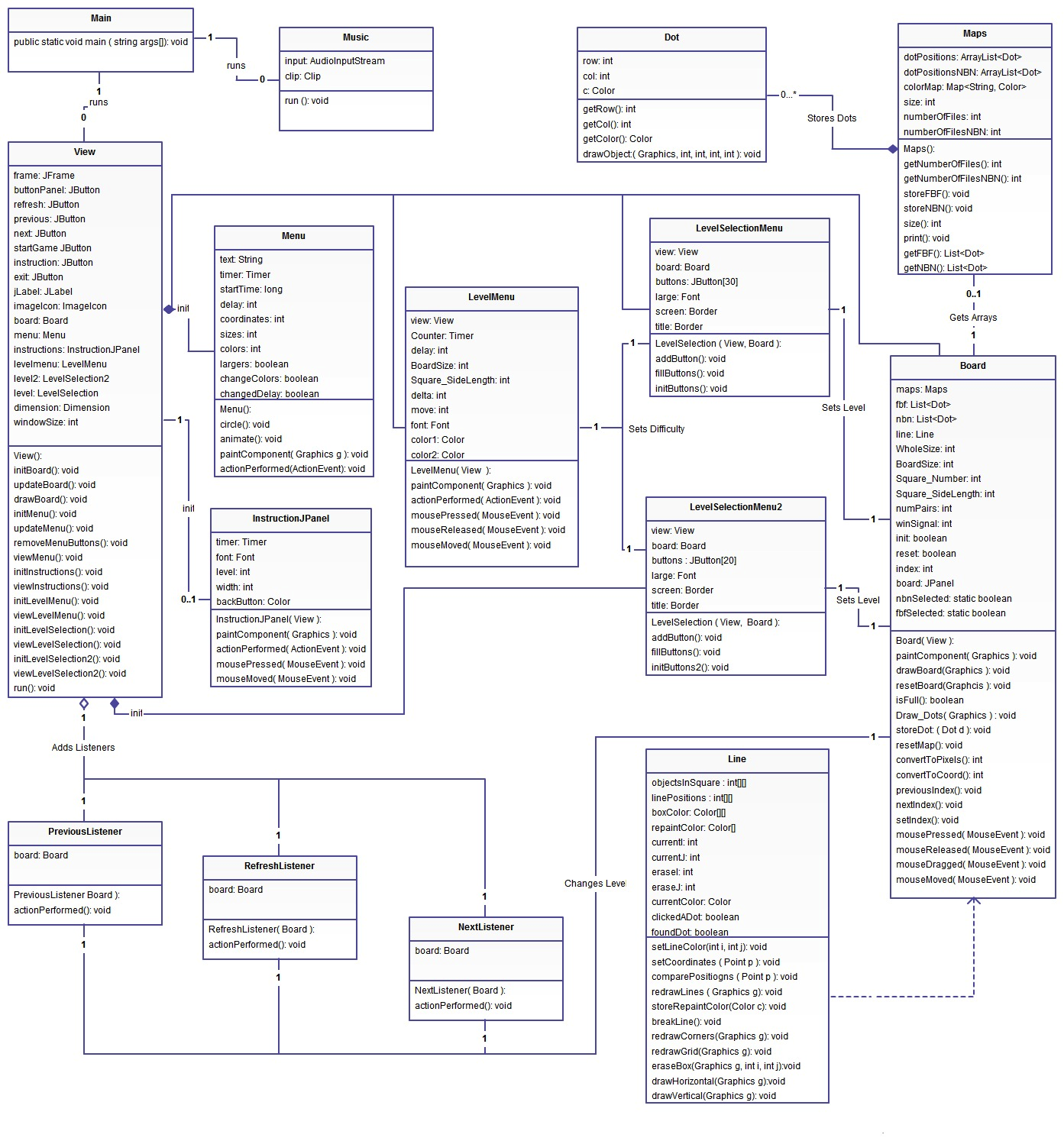
|  |
| --- |
| **Operation:** paintComponent(Graphics g)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game has been started  **Postconditions: -**instructions were painted onto the Jpanel with drawString and  drawRectangle.(association formed) |

Class Name: NextListener

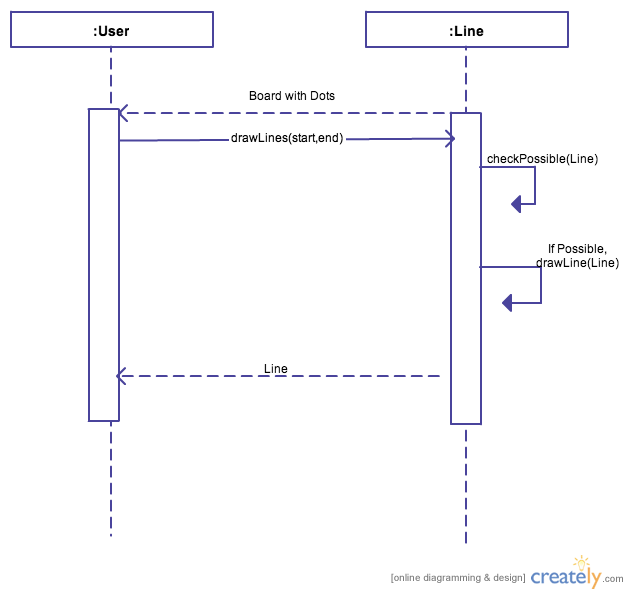
ContractCO1: actionPerformed(ActionEvent e)

|  |
| --- |
| **Operation:** actionPerformed(ActionEvent e)  **Cross Reference:** Use Case: Connecting the Dots  **Preconditions:** Game has been started and Board has been initialized.  **Postconditions: -**object of Board class changed index of map to the next in maps array  in order to change map through next button.  -object of Board was used to access resetMap method to reset the  current map displayed.  -object accessed instance variable to set it to true. Also accessed  repaint method to repaint next map. |

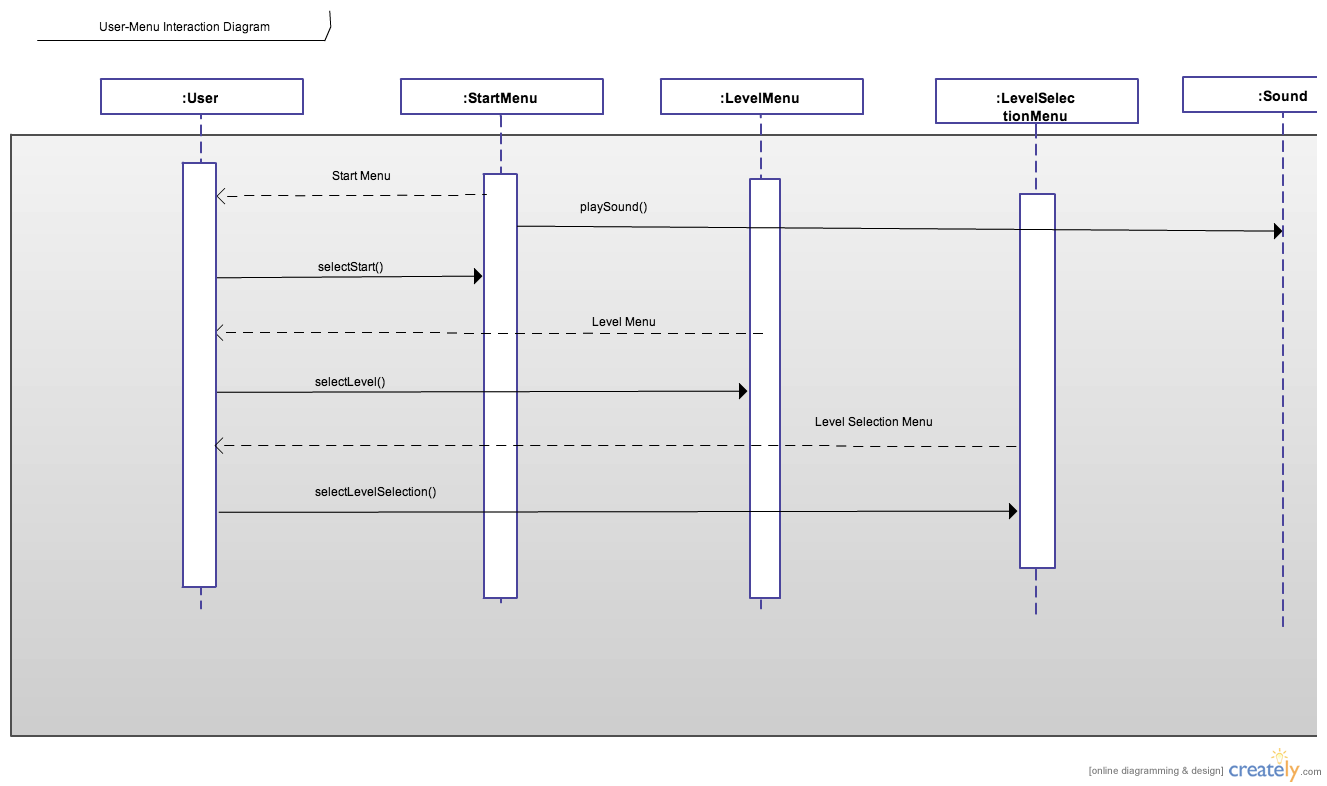
**System Design Artifacts:**  
This is our static class diagram. This shows how our classes relate to each other.

These are our two interaction diagrams. The first one shows the interaction between the user and the line. The second one shows the interaction between the user and the menu.

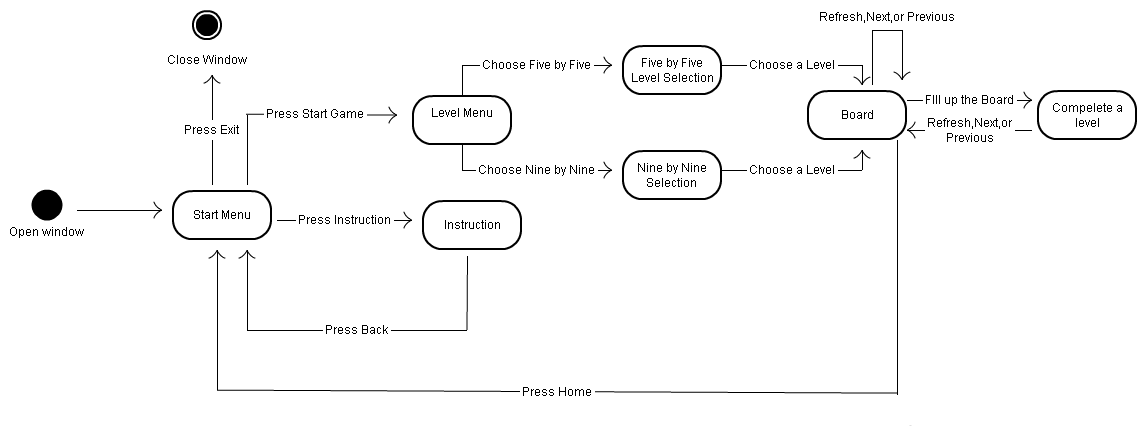
User-Line Interaction Diagram:



User-Menu Interaction Diagram:



This is our state diagram. This shows how the state of our system changes after and during events.

State Diagram:

This is our glossary. This is a reference for the terms used in our program.

**Glossary**

**Classes**:

|  |  |
| --- | --- |
| Main | The main class where the game is started |
| Board | Gets the user input for the number of squares, then loads a specific map from the Maps class |
| Dot | Gets the user input for the number of dots and draws the individual dots based on the color and position assigned in the Maps class |
| Line | Draws lines on the board connecting the dots based on mouse input |
| Maps | Contains preset locations for dot positions and colors that allow the puzzles to be solved |
| LevelMenuMouseListener | Give mouse action for the Level Menu class. |
| RefreshListener | Give the action listener that the board can refresh when the refresh button is activiated |
| LevelMenu | Provide an interface for a level menu |
| Menu | Provide an interface for the start game menu |
| View | Link all the interface together display them accordingly to the user |
| PreviousListener | Action listener for the previous button |
| NextListener | Action listener for the Next button |
| levelSelection | A interface that allow user to fast access the levels |
| Music | Use to play music in the background |
| InstructionJPanel | A instruction interface that shows directions to play the game |

**Methods**:

(View class)

|  |  |
| --- | --- |
| updateBoard | Add action listener to the buttons |
| initMenu | Create button and layout for the menu |
| updateMenu | Add listener for menu button |
| viewMenu | Set menu visible |
| viewInstruction | Set instruction visible |
| viewLevelMenu | Set Level Menu visible |
| viewLevelSelection | Set Level selection menu visble |
| run | Run the menu object |

(PreviousListener Class)

|  |  |
| --- | --- |
| actionPerformed | Go to the previous Index and reset the map |

(NextListener Class)

|  |  |
| --- | --- |
| actionPerformed | Go to the next Index and reset the map |

(levelSelection Class)

|  |  |
| --- | --- |
| addButton | Add Buttons on the level selection menu |
| fillButtons | Give each buttons a label |
| initButtons | Provide an action listener for each Button |

(Music Class)

|  |  |
| --- | --- |
| run | Run the music |

(InstructionJPanel Class)

|  |  |
| --- | --- |
| paintComponent | Paint the direction on the JPanel |
| actionPerformed | Create animation |

(Main Class)

|  |  |
| --- | --- |
| Main | Creates a new board object, then deletes the board and ends the game |

(Board Class)

|  |  |
| --- | --- |
| Clean\_Board | Draws a blank board |
| Draw\_Object | Draws the dots from the desired map onto the board |
| getInt | Helper function that parses string input into integer output |
| paint | Creates a new buffered image and calls the Clean\_Board method |
| view | Creates appropriate JFrames, JLabels, and an ImageIcon |

(Dot Class)

|  |  |
| --- | --- |
| drawObject | Draws the dots based on given parameters |
| getDotNum | Returns the number of dots |
| getCol | Returns the column position of a dot |
| getColor | Returns the color of a dot |
| getRow | Returns the row position of a dot |

(Line Class)

|  |  |
| --- | --- |
| checkSquare | Looks at a particular square and determines whether there is a dot, a line, a dot and a line, or nothing inside the square |
| comparePositions | Takes the last known x and y coordinates of a mouse click and compares them to the new coordinates, then determines which direction to draw the line |
| deleteLine | Finds all the squares with a given line color and deletes the line |
| drawHorizontal | Draws a horizontal line |
| drawObject | Sets the color of the line, then based on the result from comparePositions, calls either drawHorizontal or drawVertical |
| drawVertical | Draws a vertical line |
| setPositions | Based on the mouse click position, sets the x and y coordinates for the line |

(Maps Class)

|  |  |
| --- | --- |
| getFBF | Returns the five by five map |
| setFBF | Creates new dots and assigns them specific positions and colors |

（LevelMenuMouseListener class）

|  |  |
| --- | --- |
| LevelMenuMouseListener | Set the view attribute from the parameter |
| getChangeScreen | Return a boolean value to see if the screen is changed or not. |
| mousePressed | Select the level according to the user mouse input and then initialize the board and draw the board. |

(RefreshListener class)

|  |  |
| --- | --- |
| RefreshListener | Initialize the board object according to the parameter |
| actionPerformed | Print out a message that the refresh action is activate and redraw the dots |

(LevelMenu class)

|  |  |
| --- | --- |
| LevelMenu | Constructor that initialize the attributes |
| getBoardSize | Return boardsize |
| convertToCoord | Cover a pixel into a coordinate point |
| paintComponent | Paint the board with designated boardsize |
| main | Currently use to test the class |
| actionPerformed | Repaint for animation |

(Menu Class)

|  |  |
| --- | --- |
| Menu | Construct a new imagecon |
| paintComponent | Paint a background image on the JPanel |
| getImageWidth | Return the image width |
| getImageHeight | Return the Image height |
| main | Currently use to test the class |

**Attributes**:

(Board Class)

|  |  |
| --- | --- |
| BoardSize | Size of the board |
| dot\_num | Number of dots on the board |
| graphics | A Graphics object used to draw the board |
| Square\_Number | Number of squares on the board |
| WholeSize | Size of the entire window |

(Dot Class)

|  |  |
| --- | --- |
| c | Color value of the dot |
| col | Column position of the dot |
| graphics | A Graphics object used to draw the dots |
| row | Row position of the dot |

(Line Class)

|  |  |
| --- | --- |
| objectsInSquare | Array that stores a value corresponding to what objects(if any) are located in a particular square |
| graphics | A Graphics object used to draw the lines |
| xPosition (line) | Starting x coordinate of each line |
| yPosition (line) | Starting y coordinate of each line |

(Maps Class)

|  |  |
| --- | --- |
| dotHolder | ArrayList that stores the dot coordinates and colors |
| fbf,sbs,ebe,nbn | ArrayLists that hold all the maps |

(LeveMenuMouseListener class)

|  |  |
| --- | --- |
| changeScreen | Holds a boolean value |
| view | An view object of the view class |

(RefreshListener class)

|  |  |
| --- | --- |
| board | A board object from the Board class |

(LevelMenu Class)

|  |  |
| --- | --- |
| Counter | A timer object use to create animation |
| delay | Initialize to 500 that use to set the speed of the animation |
| BoardSize | Each subboard size for level choosing |
| Square\_SizeLength | A number that indicate how big is each Square |
| Color | A color array that holds all the color I want to display |
| Coord1 | A random object that we use to generate random coordinate for the animation. |
| numberofsquare | Use to distinguish the different levels |

(Menu class)

|  |  |
| --- | --- |
| text | A string for the title “Connecting the dots” |
| image | And Image object that holds an image to display in the background. |

(view Class)

|  |  |
| --- | --- |
| instructions | A instructionJPanel use to display direction |
| levelmenu | A levelMenu use to display levels |
| level2 | A levelSelectionMenu use to display direction |
| level | A levelSelectionMenu use to display direction |
| changeScreen | A Boolean that indicate if the screen is change or not |
| music | A object that runs the musics |

(PreviousListener Class)

|  |  |
| --- | --- |
| board | A board object |

(nextListener Class)

|  |  |
| --- | --- |
| board | A board object |

(levelSelection Class)

|  |  |
| --- | --- |
| buttons | An array of buttons |
| view | A view object |
| board | A board object |
| large | A font that is different from the default font. |

(Music Class)

|  |  |
| --- | --- |
| input | An AudioInputStream object |
| Clip | A clip object |

(InstructionJPanel Class)

|  |  |
| --- | --- |
| timer | A timer object for animation |
| delay | A int value set to 30 for animation |
| font | A font that is bigger for the instruction |
| Font2 | A font that is smaller for the demonstration. |