**CSE1142: COMPUTER PROGRAMMING II**

**Spring 2018**

**TERM PROJECT**

**DESIGNING A GAME**

**150117011 - Buse BATMAN**

**150116046 - Fatmanur ÖZDEMİR**

**Date Submitted: May 20, 2018**

**Problem Definition**

In this Project, we designed a game which contains lines, circles, semicircles and disconnectors. Main purpose of this game is clicking to circles in the right order and trying to remove the shapes that connected to that circle. In addition, the level is completed successfully only if all the shapes are removed from the screen.

Basic rules of the game:

-The shapes that the circle connected starts to move if and only if the connection line, which is between circle and disconnector, and disconnector’s line are on the same dimension.

-When the shapes that are connected to circle finished to move and getting shorter, connection line and circle will be removed from the screen.

-If a collision happens, level starts again.

-When all shapes are removed from the screen, game passes to next level.

**Implementation Details**

|  |
| --- |
| **CoverPage** |
| + level1 : Level1 |
| + start(stage:Stage) |

* level1 object created from Level1, passes to Level1 after the button clicked.

In start method:

* We created a Group object, root.
* We set the scene and set the title “HELLO !”.
* We created a button object , added to root and used setOnAction method to pass to level1 when the button clicked.

|  |
| --- |
| **Level1** |
| + a : boolean = false  + level2 : Level2 |
| + start(stage:Stage) |

* We created boolean ‘a’ to check collisions.
* level2 object created from Level2, passes to Level2 after level is completed.

In start method:

* We created 2 circles(circle1, circle2) using Circle. 1 semicircle(sc)-2 connection arcs(arc2, arc3 using Arc, 4 normal lines(line1, line2, line3, line4)-1 vertical line(vLine1)-1 horizontal line(hLine1)-3 thick line(thickLine1, thickLine2, thickLine3) using Line.
* By using setType, setStroke, setStrokeWidth , setStrokeType and setFill methods we indicated the properties of arc.
* By using setStrokeWidth method we indicated the properties of vertical, horizontal and thick lines.
* We created a Group object, root, and added those shapes.
* We set the scene and set the title “Level 1”.
* We added setOnMouseClicked event to circle1. Enlarged the hLine1 with ScaleTransition and shortened the thickLine1 with Animation.When those events are finished, we changed boolean a to true to understand thickLine1 won’t cause a collison. And then removed all the shapes of circle1 with FadeTransition.
* We added setOnMouseClicked event to circle2. We enlarged the vLine1 with ScaleTransition, shortened thickLine2, slided thickLine3 and sc with Animation. Then we checked if a is true. If it is, a collision happened and Level1 started again. If not, animation moved on. And then removed all the shapes of circle2 with FadeTransition.
* After all the shapes are removed from the screen, program passes to level2 with start method.

|  |
| --- |
| **Level2** |
| + a : boolean = false  + b : boolean = false  + c : boolean = false  + level3 : Level3 |
| + start(stage:Stage) |

* We created boolean a, b and c to check collisions.
* level3 object created from Level3, passes to Level3 after level is completed.

In start method:

* We created 4 circles(circle1, circle2, circle3, circle4) using Circle. 3 semicircle(sc1, sc2, sc3)-2 connection arcs(arc1, arc2) using Arc, 6 normal lines(line1, line2, line3, line4, line5, line6)-2 vertical line(vLine1, vLine2)-2 horizontal line(hLine1, hLine2)-7 thick line(thickLine1, thickLine2, thickLine3, thickLine4, thickLine5, thickLine6, thickLine7) using Line.
* By using setType, setStroke, setStrokeWidth , setStrokeType and setFill methods we indicated the properties of arc.
* By using setStrokeWidth method we indicated the properties of vertical, horizontal and thick lines.
* We created a Group object, root, and added those shapes.
* We set the scene and set the title “Level 2”.
* We added setOnMouseClicked event to circle1. Enlarged the vLine1 with ScaleTransition and shortened the thickLine2 with Animation.When those events are finished, we changed boolean ‘a’ to true to understand thickLine2 won’t cause a collison. And then removed all the shapes of circle1 with FadeTransition.
* We added setOnMouseClicked event to circle2. We enlarged the hLine1 with ScaleTransition, shortened thickLine1, slided thickLine3 and sc1, sc2 with Animation. Then we checked if a or c are true. If they are, a collision happened and Level2 started again. If not, animation moved on. And then removed all the shapes of circle2 with FadeTransition.
* We added setOnMouseClicked event to circle3. Enlarged the hLine2 with ScaleTransition and shortened the thickLine6 with Animation.When those events are finished, we changed boolean ‘b’ to true to understand thickLine6 won’t cause a collison. And then removed all the shapes of circle3 with FadeTransition.
* We added setOnMouseClicked event to circle4. We enlarged the hLine1 with ScaleTransition, slided and shortened thickLine5, thickLine7 and sc3 with Animation. Then we checked if ’ b’ is true. If it is, a collision happened and Level2 started again. If not, animation moved on. When those events are finished, we changed boolean ‘c’ to true to understand thickLine5 won’t cause a collison. And then removed all the shapes of circle4 with FadeTransition.
* After all the shapes are removed from the screen, program passes to level3 with start method.

|  |
| --- |
| **Level3** |
| + a : boolean = false  + b : boolean = false  + level4 : Level4 |
| + start(stage:Stage) |

* We created boolean a and b to check collisions.
* level4 object created from Level4, passes to Level4 after the level is completed.

In start method:

* We created 3 circles(circle1, circle2, circle3) using Circle. 2 semicircles(sc1, sc2)- 4 connection arcs(cArc1, cArc2, cArc3, cArc4) using Arc, 5 normal lines(line1, line2, line3, line4, line5)-2 vertical line(vLine1, vLine2)-1 horizontal line(hLine)-5 thick line(thickLine1, thickLine2, thickLine3, thickLine4, thickLine5) using Line.
* By using setType, setStroke, setStrokeWidth , setStrokeType and setFill methods we indicated the properties of arc.
* By using setStrokeWidth method we indicated the properties of vertical, horizontal and thick lines.
* We created a Group object, root, and added those shapes.
* We set the scene and set the title “Level 3”
* We added setOnMouseClicked event to circle1. Enlarged the hLine with ScaleTransition, slided thickLine4 and shortened thickLine5 with Animation. When those events are finished, we changed boolean ‘b’ to true to understand thickLine4 won’t cause a collison. Then we checked if a is false. If it is, a collision happened and Level3 started again. If not we slided and shortened sc2 and thickLine4. Then removed all the shapes of circle1 with FadeTransition.
* We added setOnMouseClicked event to circle2. Enlarged the vLine2 with ScaleTransition, shortened thickLine3 with Animation. After that, we changed boolean ‘a’ to true to understand thickLine3 won’t cause a collison.. Then removed all the shapes of circle2 with FadeTransition
* We added setOnMouseClicked event to circle3. Enlarged the vLine1 with ScaleTransition, slided thickLine2 and shortened thickLine1 with Animatio. Then we checked if b is false. If it is, a collision happened and Level3 started again. If not we slided and shortened sc1 and thickLine2. Then removed all the shapes of circle3 with FadeTransition.
* After all the shapes are removed from the screen, program passes to level4 with start method.

|  |
| --- |
| **Level4** |
| + a : boolean = false  + level5 : Level5 |
| + start(stage:Stage) |

* We created a boolean ‘a’ to check collisions.
* level5 object created from Level5, passes to Level5 after level is completed.

In start method:

* We created 3 circles(circle1, circle2, circle3) using Circle. 1 semicircle(sc)-2 connection arcs(arc2, arc3) using Arc, 6 normal lines(line1, line2, line3, line4, line5, line6)-1 vertical line(vLine1)-1 horizontal line(hLine1)-3 thick line(thickLine1, thickLine2, thickLine3)-1 turnableLine using Line.
* By using setType, setStroke, setStrokeWidth , setStrokeType and setFill methods we indicated the properties of arc.
* By using setStrokeWidth method we indicated the properties of vertical, horizontal and thick lines.
* By using setStroke method we indicated the properties of circle3 and turnableLine.
* We created a Group object, root, and added those shapes.
* We set the scene and set the title “Level 4”.
* We added setOnMouseClicked event to circle1. First we checked if turnableLine is horizontal. If it is, user can click to circle1. If not, user have to change the rotate of turnableLine. When user clicked circle1, we enlarged the hLine1 with ScaleTransition and shortened the thickLine3 with Animation.When those events are finished, we changed boolean ‘a’ to true to understand thickLine3 won’t cause a collison. And then removed all the shapes of circle1 with FadeTransition.
* We added setOnMouseClicked event to circle2. First we checked if turnableLine is vertical. If it is, user can click to circle2. If not, user have to change the rotate of turnableLine. After, we enlarged the vLine1 with ScaleTransition, slided and shortened thickLine1, thickLine2 and sc with Animation. And then we checked if a is true. If it is, a collision happenes and Level4 startes again. If not, animation moved on. And then removed all the shapes of circle2 with FadeTransition.
* We added setOnMouseClicked event to circle3. We changed the rotate of turnableLine depending on click count.
* After all the shapes are removed from the screen, program passes to level5 with start method.

|  |
| --- |
| **Level5** |
| + a : boolean = false  + b : boolean = false  + c : boolean = false |
| + start(stage:Stage) |

* We created a boolean ‘a’ , ‘b’ and ‘c’ to check collisions.

In start method:

* We created
* 4 circles(circle1, circle2, circle3 and circle4) and 2 empty circles(emptycircle1 and emptycircle2) using Circle. 3 semicircle(sc1 , sc2, sc3)-4 connection arcs(arc1, arc2, arc3, arc4) using Arc, 12 normal lines(line1, line3, line4, line6, line7, line8, line9, line10, line11, line12, line13, line14)-2 vertical lines(vLine1, vLine2)-2 horizontal line(hLine1, hLine2)-3 thick line(thickLine1, thickLine2, thickLine3, thickLine4, thickLine5, thickLine6, thickLine7)-2 turnable line(turnableLine1 and turnableLine2) using Line.
* By using setType, setStroke, setStrokeWidth , setStrokeType and setFill methods we indicated the properties of arcs.
* By using setStrokeWidth method we indicated the properties of vertical, horizontal and thick lines.
* By using setStroke method we indicated the properties of emptycircle1, emptycircle2, turnableLine1 and turnableLine2.
* We created a Group object, root, and added those shapes.
* We set the scene and set the title “Level 5”.
* We added setOnMouseClicked event to circle1. First we checked if turnableLine1 is horizontal. If it is, user can click to circle1. If not, user have to change the rotate of turnableLine1. When user clicked circle1, we enlarged the hLine1 with ScaleTransition, slided thickLine3 and sc2, shortened thickLine5 with Animation. When those events are finished, we changed boolean ‘b’ to true to understand thickLine3 won’t cause a collison. Then we checked if a is false. If it is, a collision happened and Level5 started again. If not we slided and shortened sc2 and thickLine3. Then removed all the shapes of circle1 with FadeTransition.
* We added setOnMouseClicked event to circle2. First we checked if turnableLine2 is horizontal. If it is, user can click to circle2. If not, user have to change the rotate of turnableLine2. When user clicked circle2, we enlarged the hLine2 with ScaleTransition, slided thickLine6 and sc3, shortened thickLine7 with Animation. Then we checked if ’c’ is false. If it is, a collision happened and Level5 started again. If not we slided and shortened sc3 and thickLine6. Then removed all the shapes of circle2 with FadeTransition.
* We added setOnMouseClicked event to circle3. First we checked if turnableLine1 and turnableLine2 are vertical. If it is, user can click to circle3. If not, user have to change the rotate of turnableLine1and turnableLine2. When user clicked circle3, we enlarged the vLine1 with ScaleTransition, slided thickLine2 and sc1, shortened thickLine1 with Animation. When those events are finished, we changed boolean ‘c’ to true to understand thickLine2 won’t cause a collison. Then we checked if ‘b ‘ is false. If it is, a collision happened and Level5 started again. If not we slided and shortened sc1 and thickLine2. Then removed all the shapes of circle3 with FadeTransition.
* We added setOnMouseClicked event to circle4. When user clicked circle4, we enlarged the vLine2 with ScaleTransition, shortened thickLine4 with Animation. When those events are finished, we changed boolean ‘a’ to true to understand thickLine4 won’t cause a collison. Then removed all the shapes of circle3 with FadeTransition.
* We added setOnMouseClicked event to emptycircle1 and emptycircle2. We changed the rotate of turnableLine1 and turnableLine2 depending on click count.
* **Which parts are incomplete in our project ?**

-Semicircles are sliding and fading at the same time, but this event design is not looking nice . We tried so many methods but the final result is the best of what we all tried even we didn’t like.

* **What are the difficulties we have encountered during the implementation?**

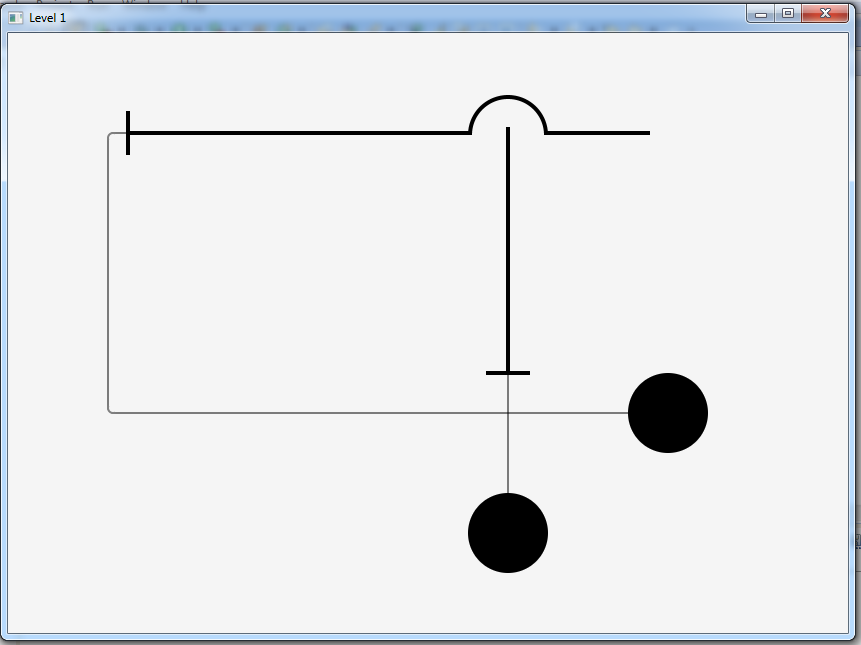
-The hardest part of this game was explaining to program to how to understand a collision happened. And also calculating the coordinates in events was really hard and challenging.

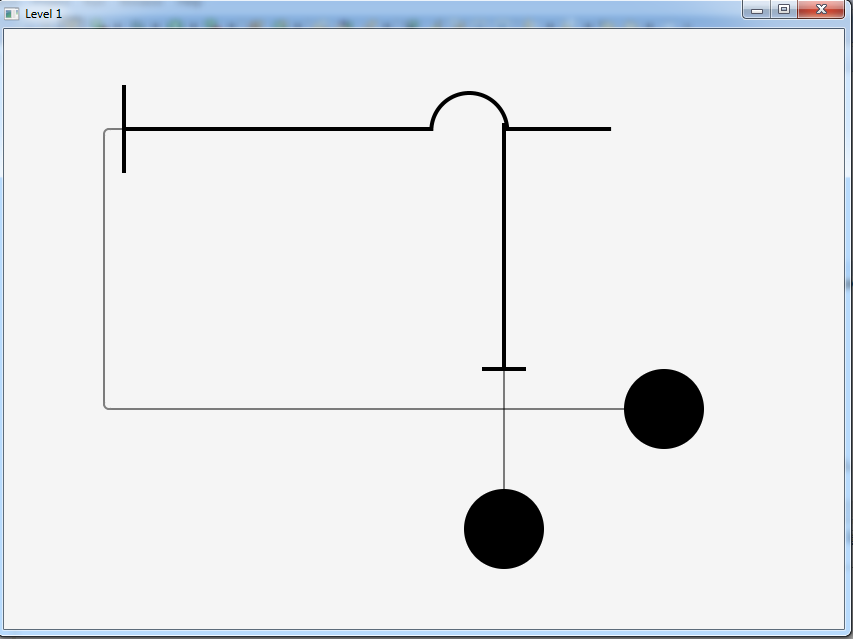
* **What are the additional functionalities of our project added by our team ?**

-While doing graphical design, we used connection arcs to connect two lines to have a better look.

**Test Cases**

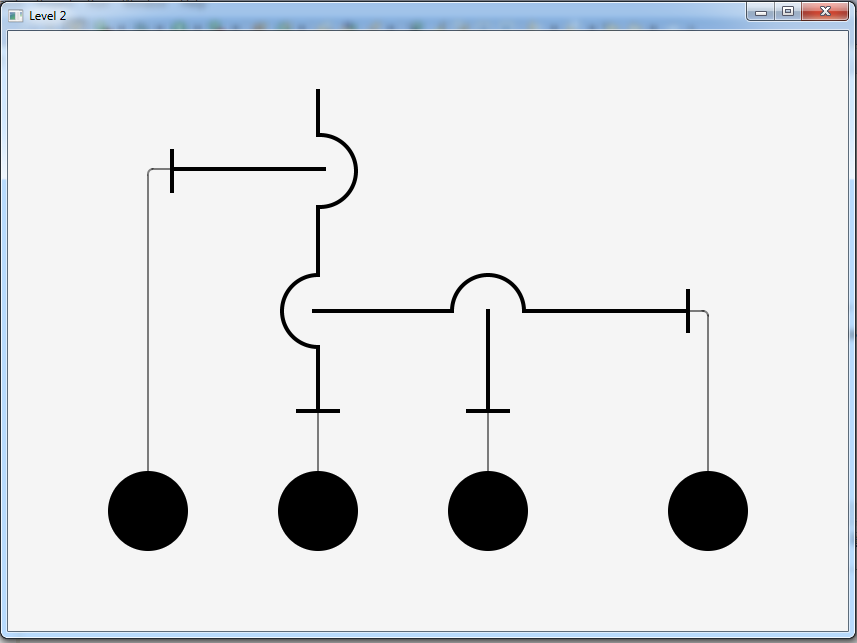
**LEVEL 1:**

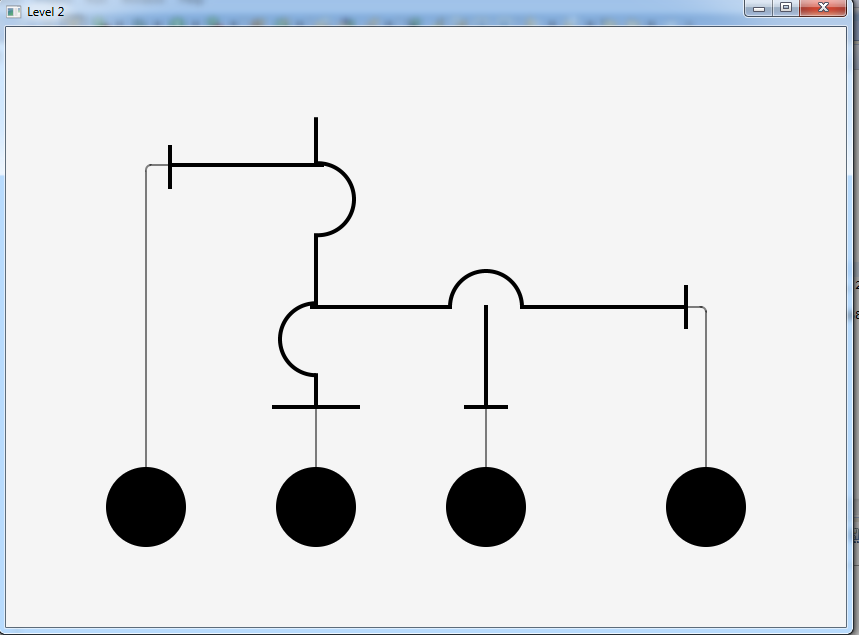




-If thickLine3’s start X coordinate becomes equal to thickLine1’s start X coordinate, collision happens and level1 starts again.

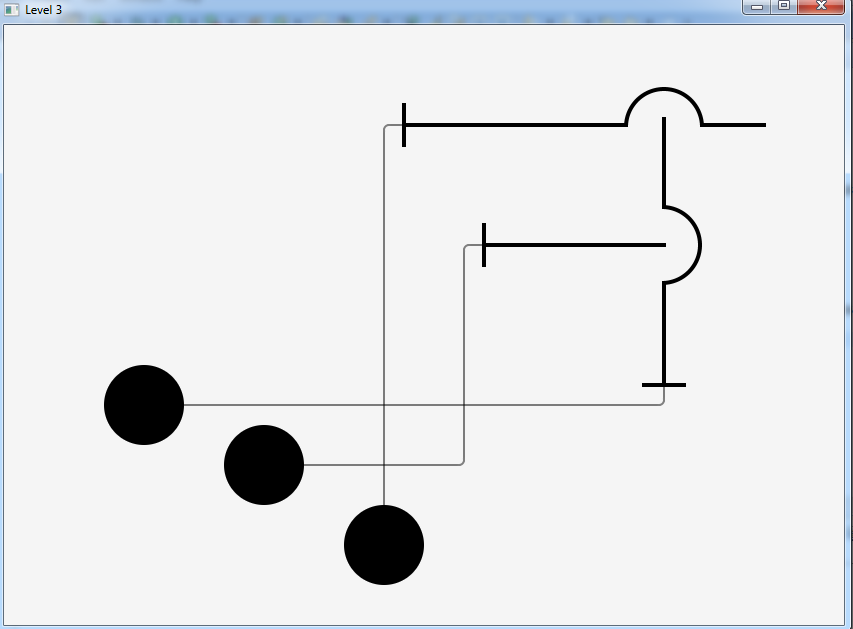
**LEVEL 2:**

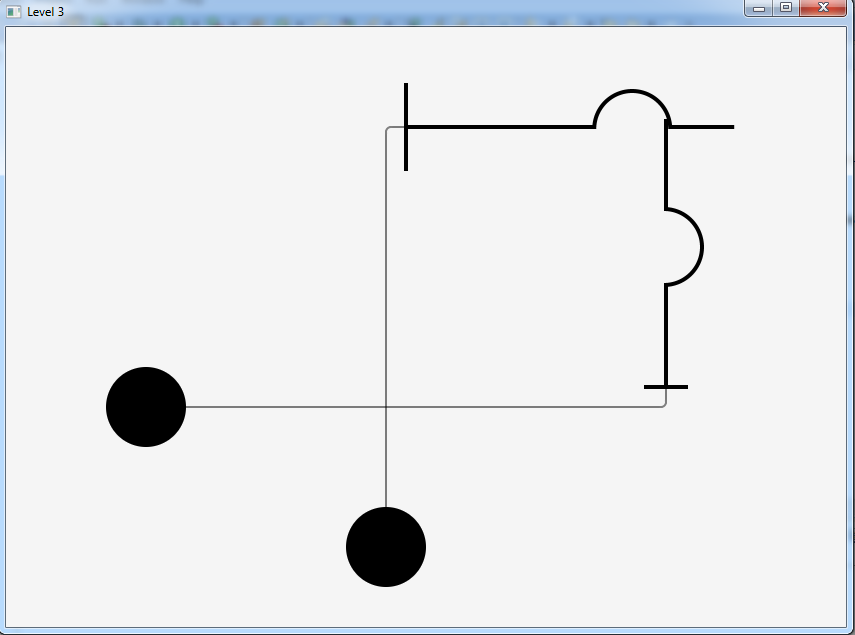




-If thickLine1’s end Y coordinate becomes equal to thickLine2’s end Y coordinate or thickLine3’s end Y coordinate becomes equal to thickLine5’s start Y, collision happens and level2 starts again.

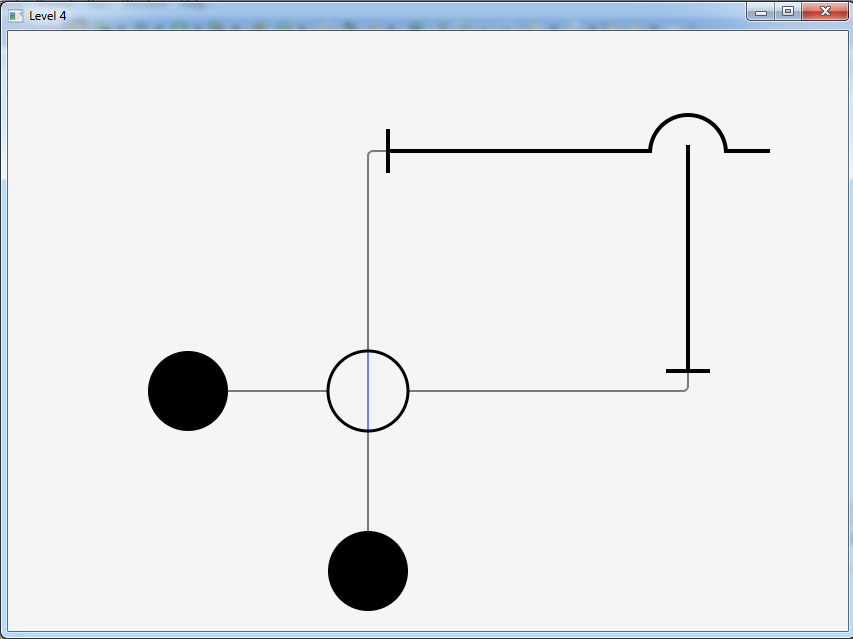
**LEVEL3 :**



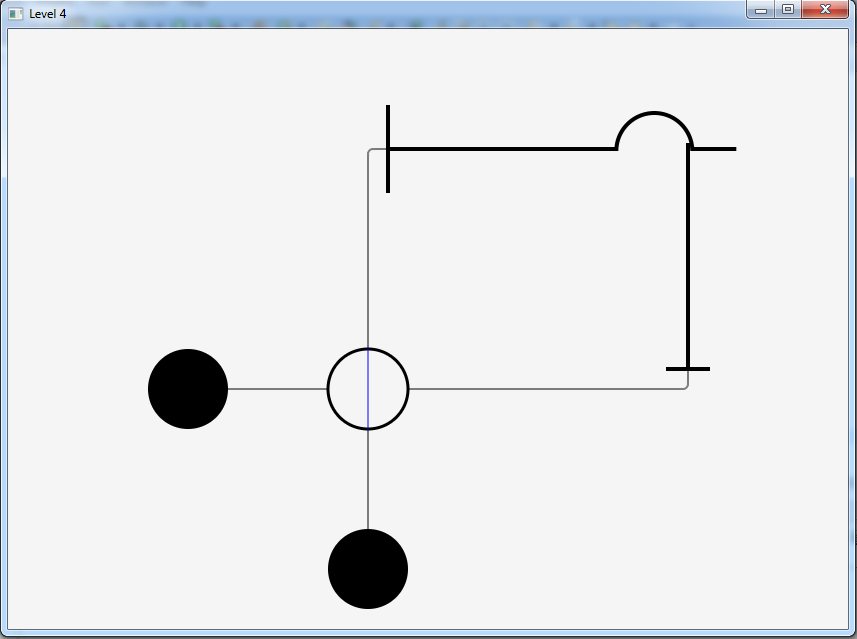


-If thickLine2’s startX coordinate becomes equal to thickLine4’s start X coordinate, collision happens and level3 starts again.

**LEVEL 4:**

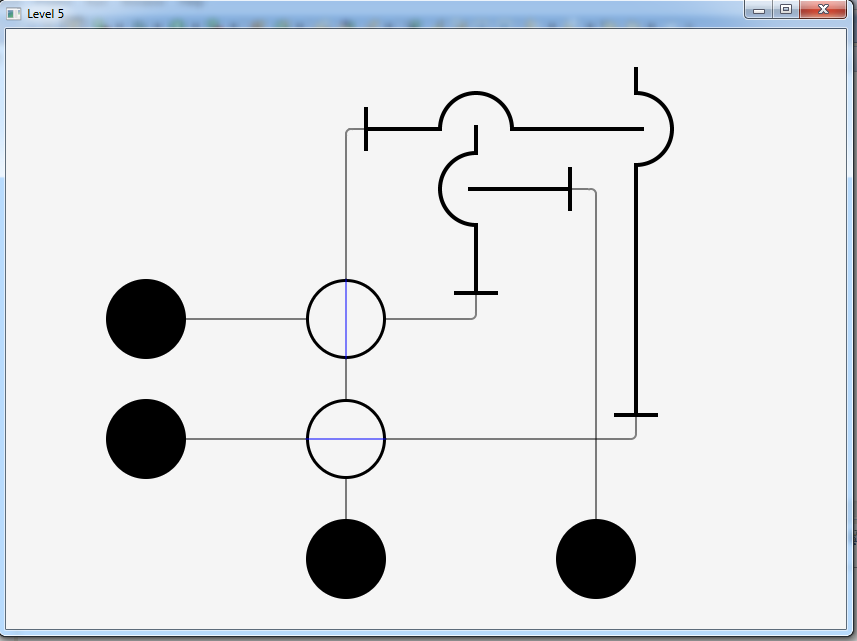


-In order to click circle1 or circle2, first user has to click to circle that contains blue turnable line.

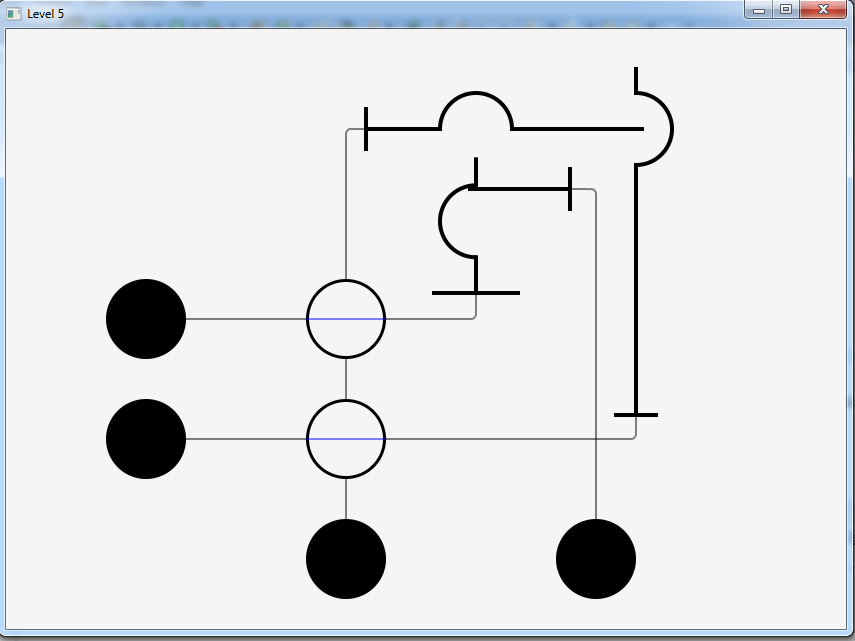


-If thickLine2’s startX coordinate becomes equal to thickLine3’s start X coordinate, collision happens and level4 starts again.

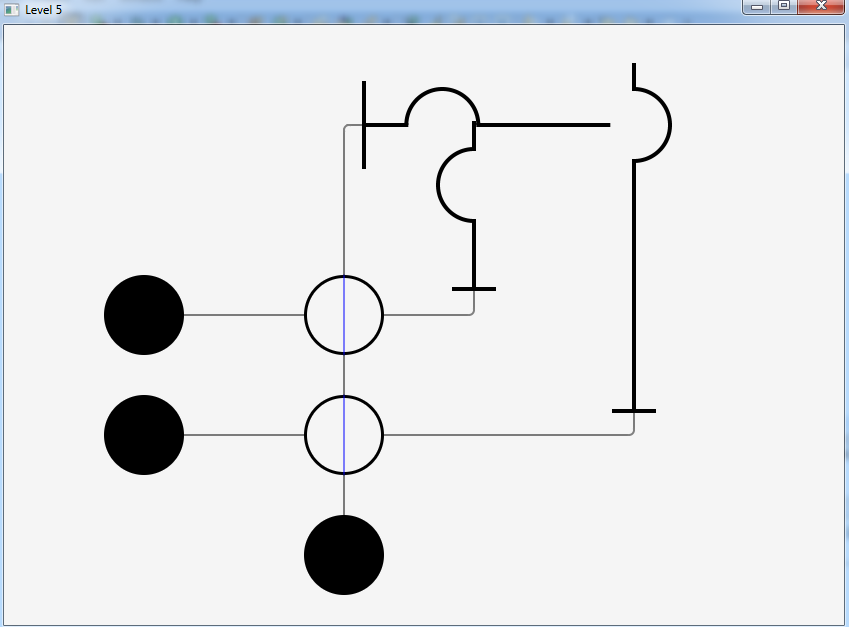
**LEVEL 5:**



-In order to click circle1, circle2 or circle3 first user has to click to empty circles that contains blue turnable lines.



-If thickLine3’s end Y coordinate becomes equal to thickLine4’s start Y coordinate, collision happens and level5 starts again.



-If thickLine2’s start X coordinate becomes equal to thickLine3’s start X coordinate, collision happens and level5 starts again.