AZURE

150116504 Zehra Zeynep Pehlivan

150117065 Hümeyra Ceyda Polat

CSE1142: Computer Programming II, Spring 2018

Date Submitted: May 20, 2018

PROBLEM DEFİNİTİON

Main purpose of this game is drawing complex figures and trying to remove all filled-circles without any collision.

The game's arguments are: ball, stick, hook and disconnecter. The main reason for the collision is the stick and the hook. The argument that acts as a button is ball. When the player clicks on the button, the sticks and hooks attached to that ball will disappear. The player must be able to lift the stick and the hook without colliding in order to finish the section and move to the new level. In the first three levels, only balls, sticks and hooks are available. In the fourth and fifth levels, disconnectors is added to the circuit. Disconnector disconnects balls from the remaining parts if the position of the stick inside the connector is on the opposite direction. Dimension of this stick can be changed by clicking on it. If the degree of the stick is 90 degrees it becomes 180 degrees. In a similar manner, if the degree of the stick is 180 degrees it becomes 90 degrees. The player can make a limited number of moves in each level. In the event of a collision or when the maximum number of moves is reached but all the arguments is not removed, current level is restarted. If all the arguments are successfully removed, the player qualifies to pass the new level.

IMPLEMENTATİON DETAİLS

|  |  |
| --- | --- |
| **Level1** | |
| -  -  -  + | secondsPassed: int  buttonClicked: int  arcControl: int  count: int |
| +  +  +  +  +  +  +  +  +  +  +  +  +  +  + | Level1(stage: Stage)  continueloopLine9(Line line): boolean  circleActions(Circle circle): void  continueloopLine1(Line line): boolean  continueloopLine2(Line line): boolean  continueloopArc(Arc arc): boolean  EnlargeLine8(Line line): boolean  ShrinkLine8(Line line, Line line9): boolean  disappearanceButton1(Button button, Circle circle, Line line): boolean  disappearanceButton2(Button button, Circle circle, Line line): boolean  EnlargeLine3(Line line): boolean  ShrinkLine3(Line line, Line line1): boolean  disLine2(Line line): boolean  decrOpacity():void  count():void |

* SecondPassed represents duration of game.
* Count represents the maksimum move that players can able do.
* Timer and Timertask objects are created. Secondspassed variable is incremented by one at run method that created with timertask object.
* ButtonClicked controls, is button clicked or not.
* ArcControl controls number of arc size.
* Button, line, arc and pane objects are created.
* Level1 actions start in drawlevel method. Timer starts at here.
* Circle and eventHandler objects are created.
* Buttons’s setOnAction methods implemented.
* Pane arguments added to pane and entegrated to the scene.
* onSuccess and onFail methods are implemented.
* circleActions method provides button actions when button is clicked.
* continueloopLine1, continueloopLine2, continueloopLine9 methods implement line1, line2 and line9 actions.
* continueloopArc method implements arc actions.
* EnlargeLine3, EnlargeLine8 methods increment line3 and line8 when button is clicked.
* ShrinkLine3, ShrinkLine8 methods decrement line3 and line 8 when collision does not appear.
* disLine2 method implements line2 actions.
* disappearanceButton1 and disappearanceButton2 methods decrements buttons radius if there is no collision when buttons are clicked.
* decrOpacity method decreases pane opacity.
* Count method decrements count when any button is clicked.

|  |  |
| --- | --- |
| **Level2** | |
| -  -  -  -  + | secondsPassed: int  buttonClicked: int  arcControl: int  arcControl2: int  count: int |
| +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  + | Level1(stage: Stage)  continueloopLine6(Line line): boolean  EnlargeLine8(Line line): boolean  ShrinkLine8(Line line, Line line6): boolean  circleActions(Circle circle): void  disappearanceButton3(Button button, Circle circle, Line line): boolean  disappearanceButton4(Button button, Circle circle, Line line): boolean  disappearanceButton1(Button button, Circle circle, Line line): boolean  disappearanceButton2(Button button, Circle circle, Line line1): boolean  decrLine3(Line line): void  decrLine15(Line line): void  EnlargeLine5(Line line): boolean  ShrinkLine5(Line line, Line line1): void  EnlargeLine16(Line line): boolean  ShrinkLine16(Line line, Line line15): boolean  EnlargeLine12(Line line): boolean  ShrinkLine12(Line line, Line line10, Line line11): boolean  continueloopLine10(Line line): boolean  continueloopLine11(Line line): void  continueloopArc3(Arc arc): boolean  decr\_line11(Line line): boolean  continueloopLine1(Line line): boolean  continueloopLine2(Line line): boolean  continueloopArc1(Arc arc): boolean  continueloopArc2(Arc arc): boolean  decrOpacity():void  count():void |

* SecondPassed represents duration of game. Stage obejct represents level 2`s stage.
* Count represents the maksimum move that players can able do.
* Timer and TimerTask objects are created. SecondPassed variable is incremented by one at run method that created with timertask object.
* ButtonClicked controls, is button clicked or not. ArcControl controls size of arc length.
* Button, line and pane objects are created.
* Level2 actions starts in drawLevel method.Timer started at here.
* Circle, arc and EventHandler<ActionEvent> objects are created.
* Buttons`s setOnAction methods are implemented.
* Arguments are added to pane and entegrated to scene.
* onSuccess and onFail methods are impelmented.
* circleActions method provides actions to circle when button is clicked.
* continueloopLine6, continueloopLine2, continueloopLine1, continueloopLine11, continueloopLine10 implements line6, line2, line1, line1 and line10`s actions.
* continueloopArc2, continueloopArc1, continueloopArc3 implements arc2, arc1 and arc3`s actions.
* EnlargeLine8, ShrinkLine8, EnlargeLine5, ShrinkLine5, EnlargeLine16, ShrinkLine16, EnlargeLine12, ShrinkLine12 implements line8, line5, line16 and line12 actions.
* disappearanceButton1, disappearanceButton2, disappearanceButton3, disappearanceButton4 methods decrement buttons radius if there is no collision when buttons are clicked.
* decrLine3, decrLine15, decr\_line11 methods implements line3, line 15 and line11`s action.
* decrOpacity method decrements pane`s opacity if there is collision when button is clicked.
* Count method decrements count when any button is clicked.

|  |  |
| --- | --- |
| **Level3** | |
| -  -  -  -  + | secondsPassed: int  buttonClicked: int  arcControl: int  stage: Stage  count: int |
| +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  + | circleActions(Circle circle): void  decrLine12(Line line): void  disLine11(Line line): void  DisArc2(Arc arc):void  EnlargeLine13(Line line): void  ShrinkLine13(Line line, Line line11): boolean  disButton1(Button button, Circle circle, Line line11): boolean  decrLine10\_2(Line line): void  EnlargeLine9\_3(Line line): void  ShrinkLine9\_3(Line line, Line line1): boolean  disButton2(Button button, Circle circle, Line line9): boolean  continueloopLine2(Line line): boolean  DisArc1(Arc arc): boolean  continueloopLine1(Line line): boolean  EnlargeLine3(Line line): boolean  ShrinkLine3(Line line, Line line1): boolean  disButton3(Button button, Circle circle, Line line1): boolean  decrOpacity():void  count():void |

* SecondPassed represents duration of game.
* Count represents the maksimum move that players can able do.
* Timer and Timertask objects are created. Secondspassed variable is incremented by one at run method that created with timertask object.
* ButtonClicked controls, is button clicked or not.
* ArcControl controls number of arc size.
* Stage represents Level3’s stage.
* Button, line, arc and pane objects are created.
* Level3 actions start in drawlevel method. Timer starts at here.
* Circle objects are created.
* Circle and eventHandler objects are created.
* Buttons’s setOnAction methods implemented.
* Pane arguments added to pane and entegrated to the scene.
* onSuccess and onFail methods are implemented.
* circleActions method provides button actions when button is clicked.
* continueloopLine1 and continueloopLine2 methods implement line1 and line2 actions.
* EnlargeLine3, EnlargeLine9\_3 and EnlargeLine13 methods increment line3, line9 and line13 when button is clicked.
* DisArc1 and DisArc2 methods decrement arc1 and arc2 size when collision does not appear.
* decrLine10\_2 disLine11, decrLine12, ShrinkLine3, ShrinkLine9\_3 and ShrinkLine13 methods decrement line2, line3, line9, line10, line11, line12 and line13 when collision does not appear.
* disButton1, disButton2 and disButton3 methods decrements buttons radius if there is no collision when buttons are clicked.
* decrOpacity method decreases pane opacity.
* Count method decrements count when any button is clicked.

|  |  |
| --- | --- |
| **Level4** | |
| -  -  -  -  + | secondsPassed: int  buttonClicked: int  arcControl: int  stage: Stage  count: int |
| +  +  +  +  +  +  +  +  +  +  +  +  + | circleActions(Circle circle): void  EnlargeLine3(Line line): void  EnlargeLine11(Line line): void  decrLine12(Line line): boolean  ShrinkLine11(Line line, Line line12): boolean  disButton1(Button button, Circle circle, Line line11): boolean  DecrLine10(Line line): boolean  continueloopArc(Arc arc): boolean  continueloopLine8(Line line): boolean  ShrinkLine3(Line line, Line line8): boolean  disButton2(Button button, Circle circle, Line line8): boolean  decrOpacity(): void  count():void |

* SecondPassed represents duration of game. Stage obejct represents level 4`s stage.
* Count represents the maksimum move that players can able do.
* Timer and TimerTask objects are created. SecondPassed variable is incremented by one at run method that created with timertask object.
* ButtonClicked controls, is button clicked or not. ArcControl controls size of arc length.
* Button and pane objects are created.
* Level4 actions starts in drawLevel method.Timer started at here.
* Circle, line, arc and EventHandler<ActionEvent> objects are created.
* Buttons`s setOnAction methods are implemented.
* Arguments are added to pane and entegrated to scene.
* onSuccess and onFail methods are impelmented.
* circleActions method provides actions to circle when button is clicked.
* EnlargeLine3, EnlargeLine11 methods increments size of lines when button is clicked.
* ShrinkLine11, decrLine12, DecrLine10, continueloopLine8, ShrinkLine3, methods decrements size of lines.
* disappearanceButton1 and disappearanceButton2 methods decrement buttons radius if there is no collision when buttons are clicked.
* continueloopArc implements arc actions.
* decrOpacity method decrements pane`s opacity if there is collision when button is clicked.
* Count method decrements count when any button is clicked.

|  |  |
| --- | --- |
|  | Level5 |
| -  -  -  -  + | secondsPassed: int  buttonClicked: int  ArcControl: int  starge : Stage  count: int |
| +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  + | Level5(stage: Sage)  drawLevel(): void  //continueloopLine6(line: Line) : boolean  //EnlargeLine8(line: Line): boolean  //ShrinkLine8(Line line, Line line6): boolean  circleActions(circle: Circle): void  disappearanceButton1(button : Button, circle :Circle, line: Line): boolean  disappearanceButton2(button : Button, circle :Circle, line: Line): boolean  disappearanceButton3(button : Button, circle :Circle, line: Line): boolean  disappearanceButton6(button : Button, circle :Circle, line: Line): boolean  decrLine25(line :Line) : void  decrLine13(line :Line) : void  EnlargeLine9\_24(line :Line): boolean  ShrinkLine9(line :Line, line1: Line): boolean  ShrinkLine24(line :Line, line25: Line): boolean  EnlargeLine14\_19(line :Line): void  ShrinkLine19(line :Line, line11: Line): boolean  ShrinkLine14(line :Line, line12: Line): boolean  DisLine11(line: Line): boolean  DisLine12(line: Line): void  continueloopLine21(line: Line) : boolean  DisArc1(arc: Arc) : boolean  DisArc3(arc: Arc) : boolean  continueloopArc2(Arc arc) : boolean  decrline10(line : Line): void  decrline20(line : Line): void  decrOpacity(): void  count():void |

* SecondPassed represents duration of game. Stage obejct represents level 5`s stage.
* Count represents the maksimum move that players can able do.
* Timer and TimerTask objects are created. SecondPassed variable is incremented by one at run method that created with timertask object.
* ButtonClicked controls, is button clicked or not. ArcControl controls size of arc length.
* Button , line and pane objects are created.
* Level5 actions starts in drawLevel method.Timer started at here.
* Circle, arc and EventHandler<ActionEvent> objects are created.
* Buttons`s setOnAction methods are implemented.
* Arguments are added to pane and entegrated to scene.
* onSuccess and onFail methods are impelmented.
* circleActions method provides actions to circle when button is clicked.
* ShrinkLine9, ShrinkLine24, ShrinkLine19, decrLine10, DisLine11, decrLine25, decrLine13, disLine12, ShrinkLine14, decrLine20 and continueloopLine21 methods decrements size of lines.
* EnlargeLine9\_24 and EnlargeLine14\_19 methods increments size of lines when button is clicked.
* DisArc3 , continueloopArc2 and DisArc1 methods implements arc3, arc2 and arc1`s actions.
* disappearanceButton1 , disappearanceButton2, disappearanceButton3 and disappearanceButton6 methods decrement buttons radius if there is no collision when buttons are clicked.
* Button4 and button5 represent connectors.
* decrOpacity method decrements pane`s opacity if there is collision when button is clicked.
* Count method decrements count when any button is clicked.

|  |  |
| --- | --- |
|  | <<interface>>  Level Drawer |
|  |  |
| +  +  +  + | void drawLevel();  void onSuccess();  void onFail();  moveFail():void |

* The drawLevel, onSuccess and onFail abstract methods will be implemented in the classes that implements LevelDrawer interface to provide continuity of game.
* drawLevel method get starts levels.
* moveFail method provides when player reached maximum move number, game starts from level1 if all arguments already removed.

|  |  |
| --- | --- |
| Congratulations | |
| - | starge : Stage |
| + | Congratulations(stage: Stage) |

* Congratulation class implements LevelDraver interface.
* drawLewel method creates StackPane object and text which "CONGRATULATION" and show this in scene.

|  |  |
| --- | --- |
| **Main** | |
|  |  |
| +  + | main(args: String[]): void  start(primaryStage: Stage): void |

* Start method calls Level1 construcer when levelDrawer object created.

Completed Parts of Our Project:

* Actions of balls, sticks, hooks and connecters.
* Connection between levels.

Incompleted Parts of Our Project:

* Level4’s nodes do not disapppear when level is succesfully completed.

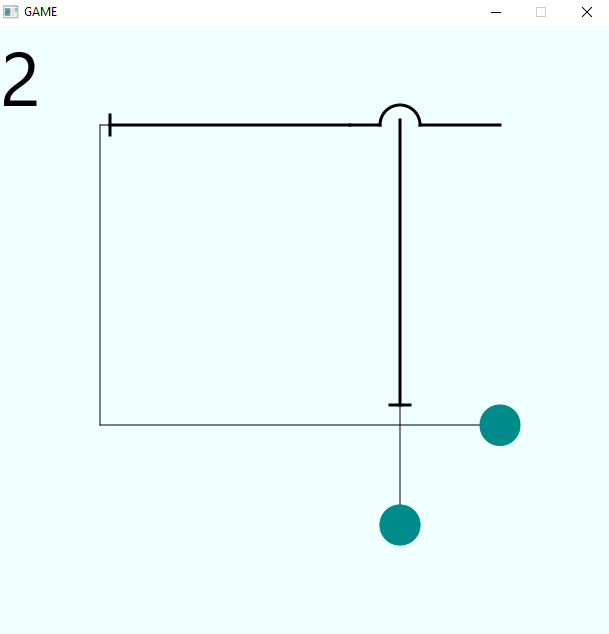
Difficulties we have encountered during the implementation

* Actually lots of things came hard to us in this project. we got mistakes we did not understand like not completely disappearance of the lines and balls.
* The state of fail, provide fade transition and go to next level was hard.
* And set up a timer was hard to us.

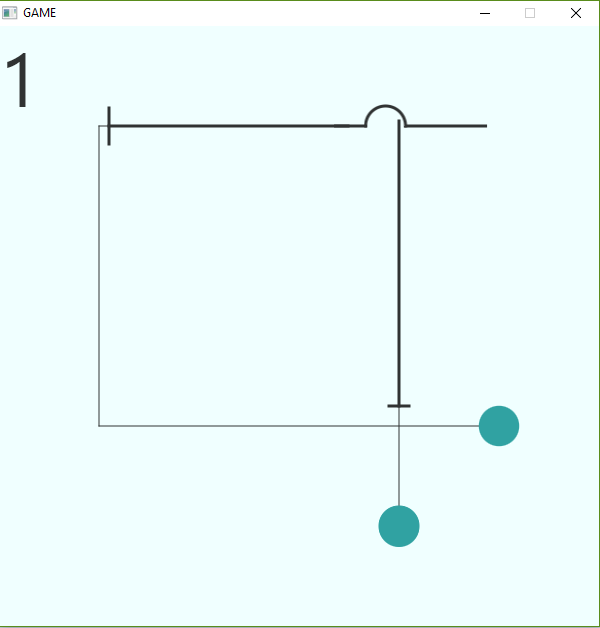
TEST CASES

LEVEL1

Initial state of level 2

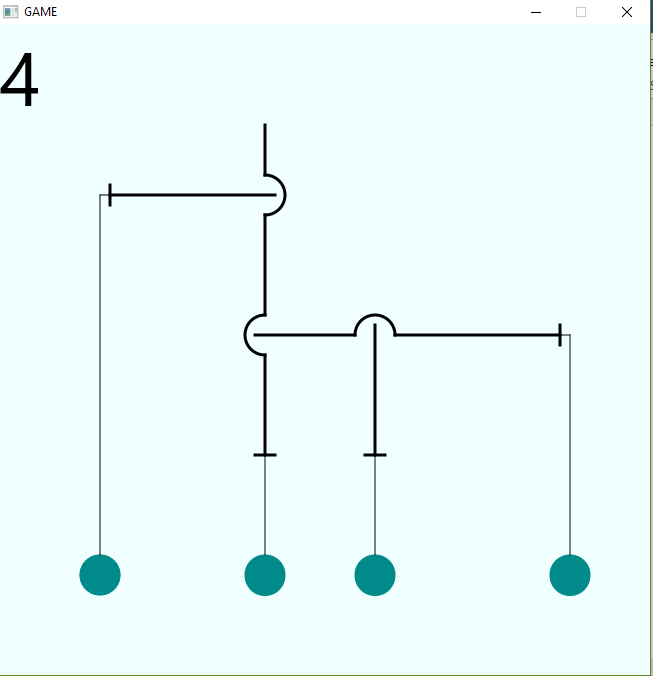


If not clicked button 1 before button 2, collision occcurs.

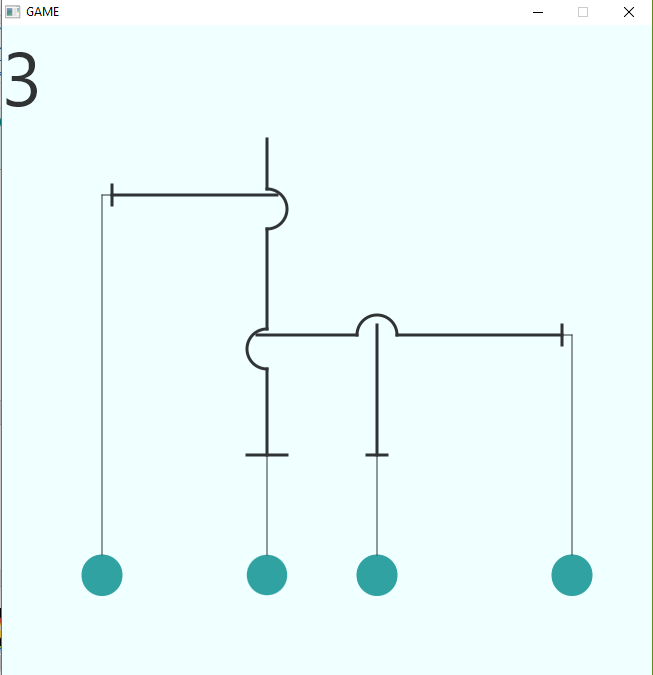


LEVEL2

Initial state of level 2

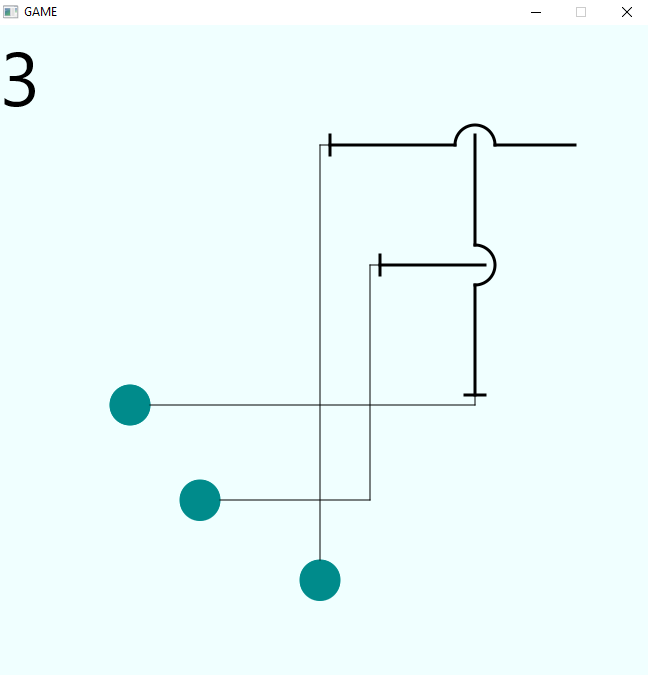


If not clicked button 1 or button 4 before button 2; button 3 before button 4 collision occcurs.

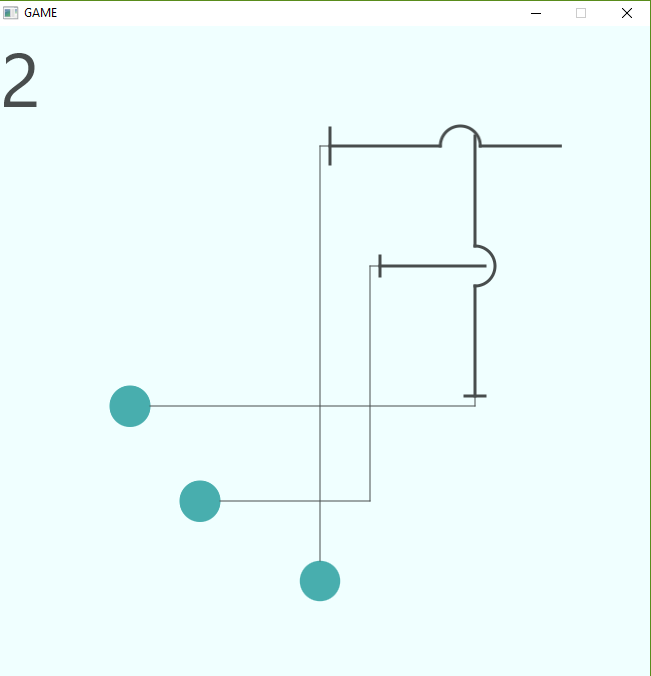


LEVEL 3

Initial state of level 3

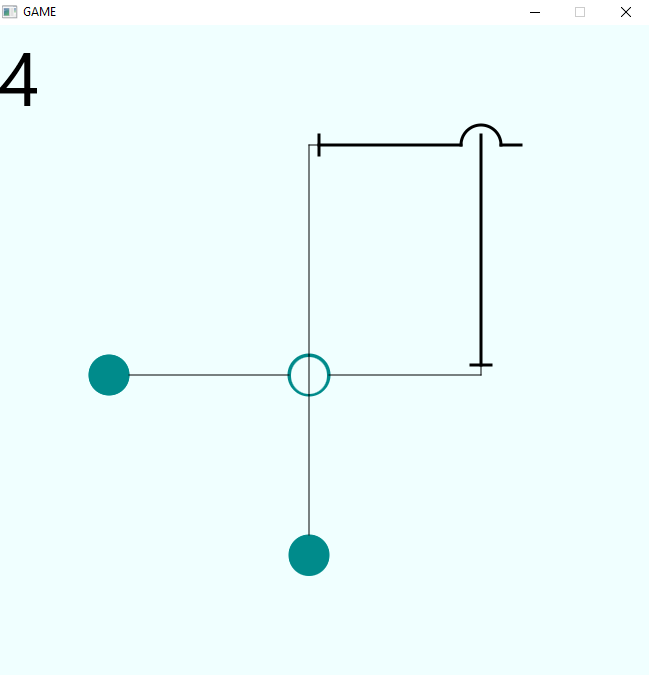


If not clicked button 2 before button 1 or button 1 before button 3 collision occcurs.

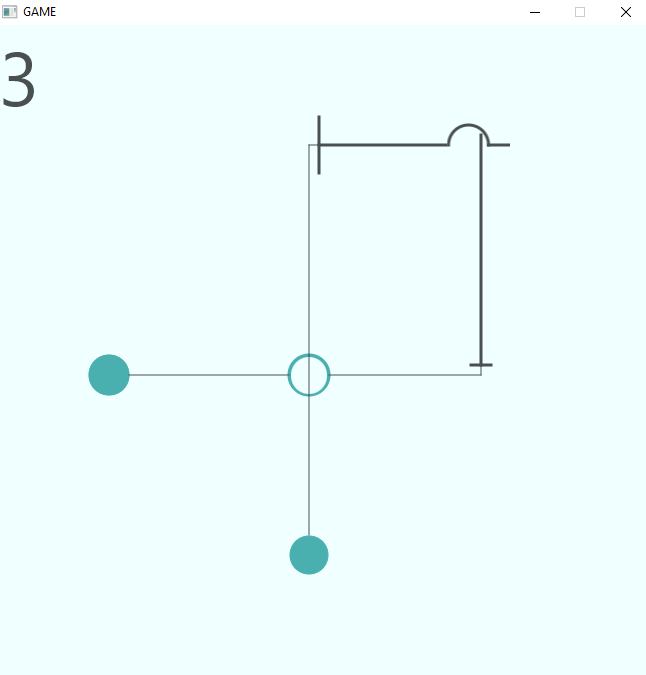


LEVEL 4

Initial state of level 4

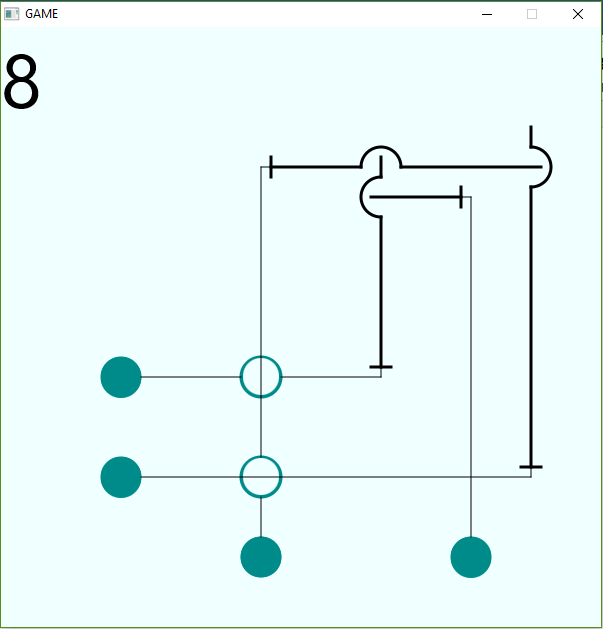


If not clicked button 2 before button 1, collision occcurs.



LEVEL 5

Initial state of level 5



If not clicked button 6 before button 2;,button 1 before button 6 collision; button 3 before button 1 , collision occcurs.

