MARMARA UNIVERSITY

Term Project of Computer Programming II - CSE1142

“HOOK”

CSE1142: Computer Programming II, Spring 2018

Rıdvan Gülcü / 150117508

Date Submitted: May 20, 2018

1. Problem Definition

The aim of this game is to pull the rope lines at different levels and different positions in each level by clicking on the circle where each rope is connected. When doing this pulling, the lines must not collide with each other. When the lines collide, it informs the user that it has made a mistake and the lines that you have already successfully pulled up are canceled and the level starts again. In order to successfully complete the level, the lines must be pulled up the correct order, that is to say you must click on the circles in correct order.

Firstly, the game starts with a level selection screen. Since no level lock has been opened before, only level 1 is selectable. After you select level 1, you are trying to finish the levels in turn. After the level 5 (last level) has been successfully completed, it returns to the level selection screen. In this case, all the levels are completed, so all of them of locks opened and you can go back to level that you want and play again.

2. Implementation Details

Lines	
+	arcRadius:int
+	hookRadius:int
-	startX:int
-	startY:int
-	endX:int
-	endY:int
-	lineCounter:int
+	Lines (startX:int, startY:int, endX: int, endY: int):void
+	createLine():Line
+	removeLastLine():void

- Line Class extends Group Class and adds the shapes it creates to this Group.
- The coordinates of the line to be created are taken with Constructor.
- Create Line creates a line with the given points.
- removeLastLine method deletes last line added to group.

ThicLine	
-	CircleX:int
-	CircleY:int
+	`ThickLine(startX:int,startY:int, endX:int,endY:int):void`
+	`addHalfCircle(CircleX:int, CircleY:int,position:String):void`
+	`createLine():Line`
+	`createHalfCircle(choice:int,k:int, startAngle:int):void`

- The ThickLine class extends the Lines class and is used to create thick lines in the game.
- The Thick Line object is created in the coordinates given by Constructor.
- In the coordinates given by the Addhalfcircle method, it calls the

CreateHalfcircle method to create and add an arc.

- The create line method is overridden to define the properties of the line.

- The CreateHalfcircle method creates arcs in different directions according to the input. Before creating an arc, it deletes the next part of the arc that it will create and completes the rest of the arc after adding the arc.

ThinLines	
+	ThinLine(startX:int,startY:int, endX:int,endY:int):void
+	addLine(newEndX:int, newEndY:int):void
-	addCorner(choice:int, k:int, start:int, end:int):void
+	createCorner(centerX:int, center:int, startAngle:int, endAngle:int):Arc

- The Thin Line class extends the Line class and is used to create fine lines in the game.

- The Thin Line object is created at the coordinates given by Constructor.

- addLine is used to draw a new line to the new end point, accepting the end of the line as the start. According to the coordinates of the new line, addCorner method is called to add the corner to the end of the old line and add a new line.

- The AddCorner method adds different corners according to the input given, and when doing so, it deletes a part of the line and adds a new arc to it. Adding Arc is done by calling the createCorner method.
- The Create Corner method creates an arc based on the given input.

Holder	
+	Holder(startX:int,startY:int, type:int):void
+	createLine():Line

- The Holder class extends the Line class and is used to create a line drawing place (a growing shrinking place).

- The Constructor is creating the Holder object in the given coordinates and position.

- It overrides the create line method to specify the properties of the line to be created.

Circles	
- -	CenterX:int CenterY:int
+	`Circles(centerX:int, center:int):void`
+	`createCircle():Circle`

- The Circles Class extends the Group Class and adds the shapes it creates to this Group.
- The coordinates of the Circle to be created with the Consructor are being taken.
- The Create Circle method creates a circle in the existing coordinates.

Hole	
+	`Hole(centerX:int, center:int):void`

- The Hole class extends the Circle class and is used to create circles that are clicked in the ball.
- Constructor creates the Hole object in the given coordinates.

Switch	
+	`Switch(int centerX, int centerY, String position):void`
+	`createCircle():Circle`

- The switch class extends the Circle class and is used to create the switches in the game.
- The switch object is created by Constructor in the given coordinates.
- Override the createCircle method to change the properties of the circle to be created.

Levels	
+	Levels():void
+	createSwitch(int centerX, int centerY, String position):Switch
+	Holder createHolder(int startX, int startY, String position):Holder
+	addRectangle(int startX, int startY, int height, int width):void
+	changeLevel(String choice):void
+	createTrueAnimation(Line path, int Time, ThickLine thickLine, Hole hole, Holder holder, ThinLine thinLine):SequentialTransition
+	createFalseAnimation(Line path, ThickLine thickLine, Hole hole, Holder holder): SequentialTransition
+	removeSwitch(Switch swtch): FadeTransition
+	resetCheckProperty():void
+	addNumber(int number):void
+	clickEffect(Hole hole):void

- The Levels class extends the class pane and adds the objects needed to create the leveli to this pane.

- Constructor creates a Levelin object.

- The addObjects abstracs method is used to add all objects to the panel.

- The switches are created in the coordinates given by the create Switch method and in the first position preference.

- Holder is being created in given coordination and position with Create Holder.

- A rectangle is created in the coordinates given by addRextangle. These rectangles are used to hide behind and give the effect of disappearing when drawing lines.

- The change level method is used to change the level.

- The CreateTrueAnimation method is used to create the animation that will take place when the right click is done.

- The createFalseAnimation method is used to create the path and the animations that will follow the line when true and false clicks are made.

- The removeSwitch method removes the given switch from the level pane.

- The resetCheckProperty abstract method is used to reset check properties to default.

- addNumber methodu is used to add the level number to the panel.

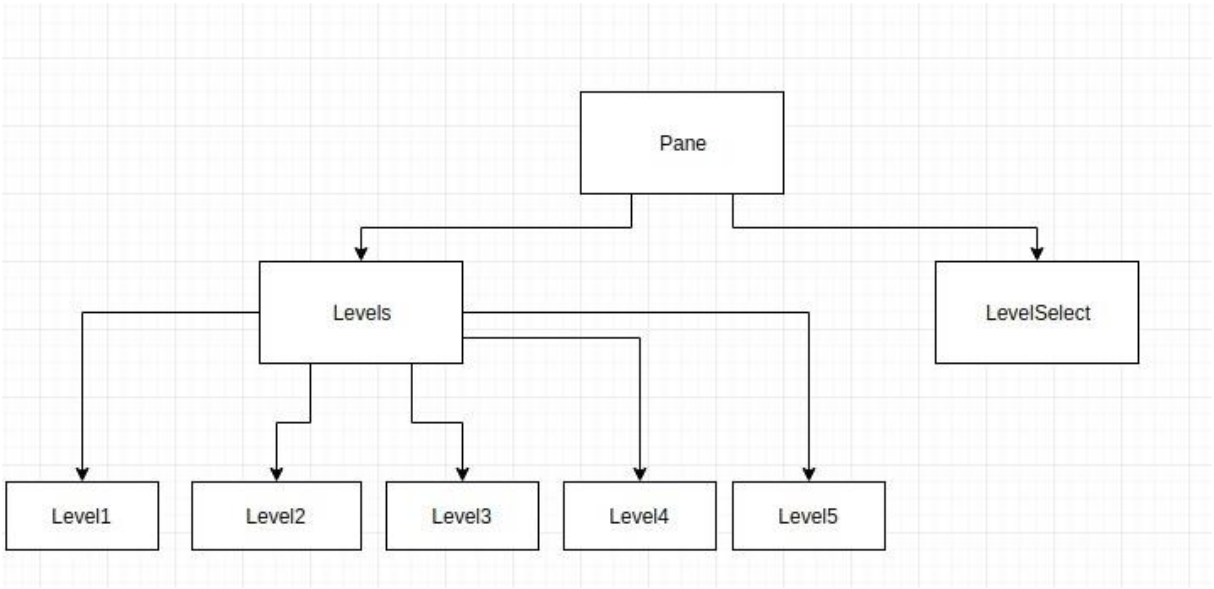
- The clickEffect method is used to give the click effect when it is clicked in the hole.

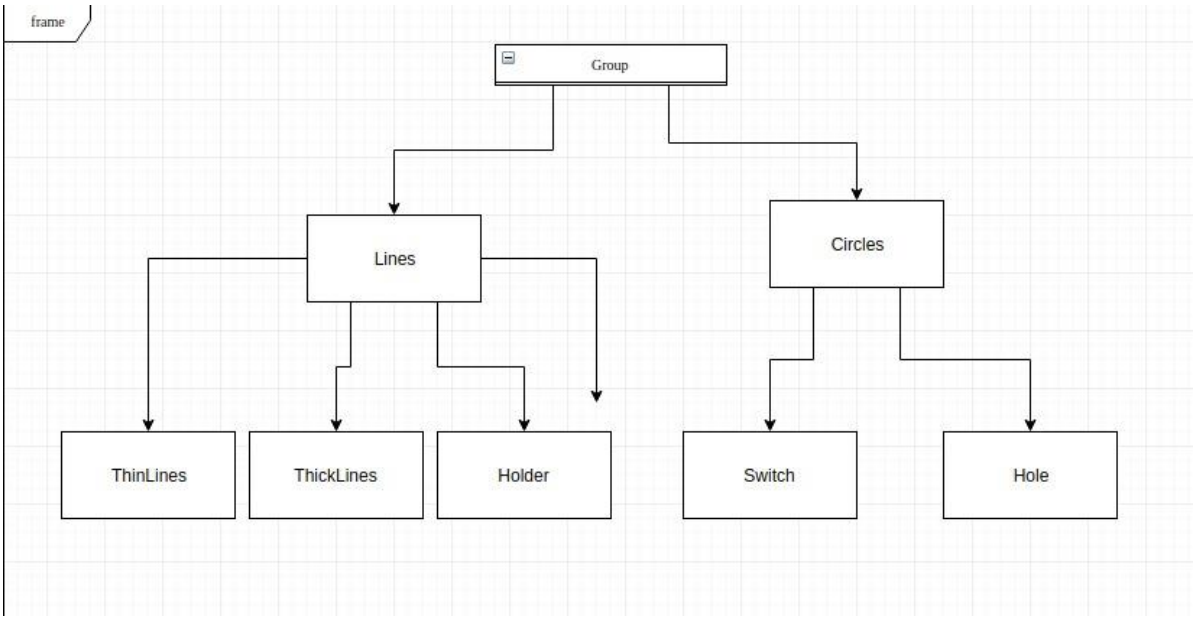
LevelSelect	
-	centerX:int
-	centerY:int
-	maxLevelSize:int
+	circles:Circle[]
+	texts:Text[]
+	`LevelSelect(int totalLevelNumber):void`
+	`addEvent(num:int):void`

- Level Select The Class pane extends the Class and is used to create the Level selection screen.

- In the number given with Constructor, the level selection screen is created by adding the button to the panel.

- Events are added to the buttons created with the addEvent method.



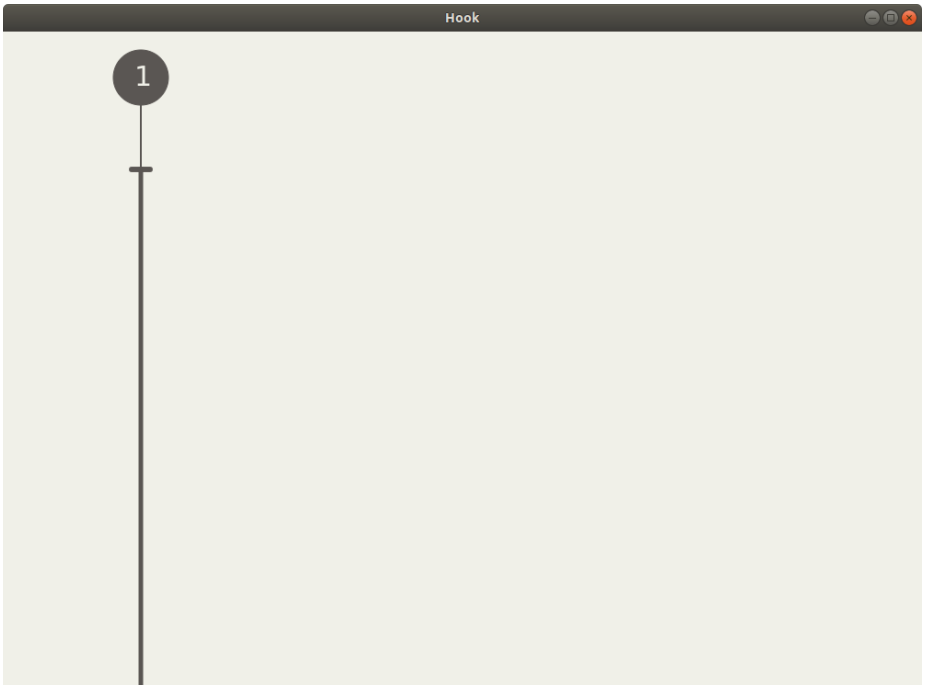


Instead of trying to shorten the length of my thick lines, I put a rectangle in the back of the holder, which will be the same color as the background, and I gave the effect of disappearing by moving the thick line with this path transition. To check for collisions I set a boolean value to default to false for each object, and when I clicked on that object (when it was deleted from the pane) I made this value true. So when I click on an object, it first checks the other objects that this object will hit and creates the animations that it will do.

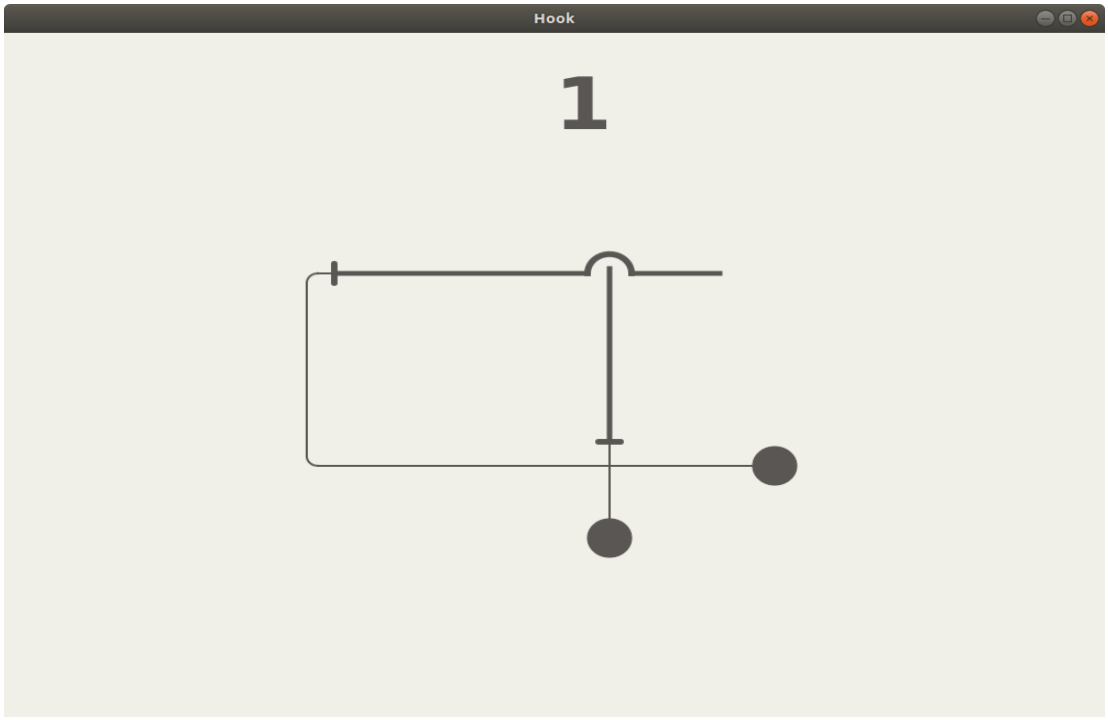
I added an entry page and a level selection screen as an extra in the game. The game does not have any shortcomings and bugs and does the desired tasks successfully.

3. Test Cases

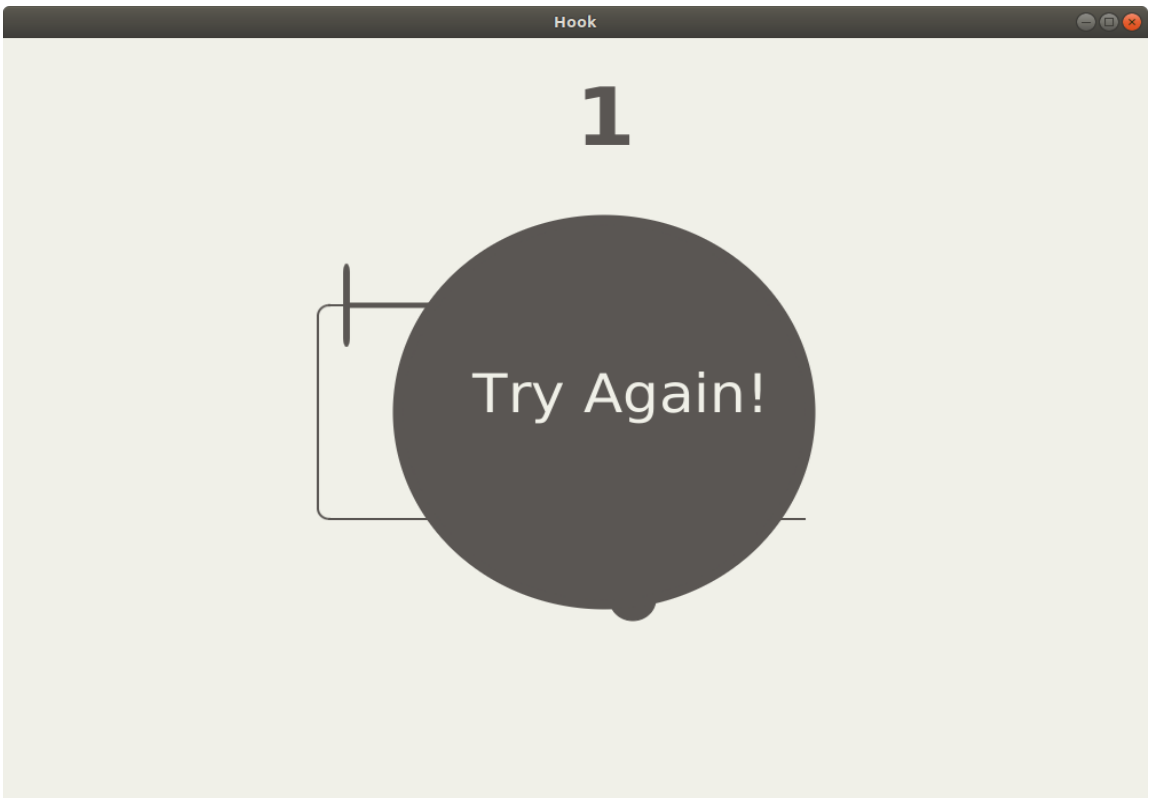
When the game is first opened, the starting page of the game comes up. On this page, the game starts when you click start or close the game when you click exit.



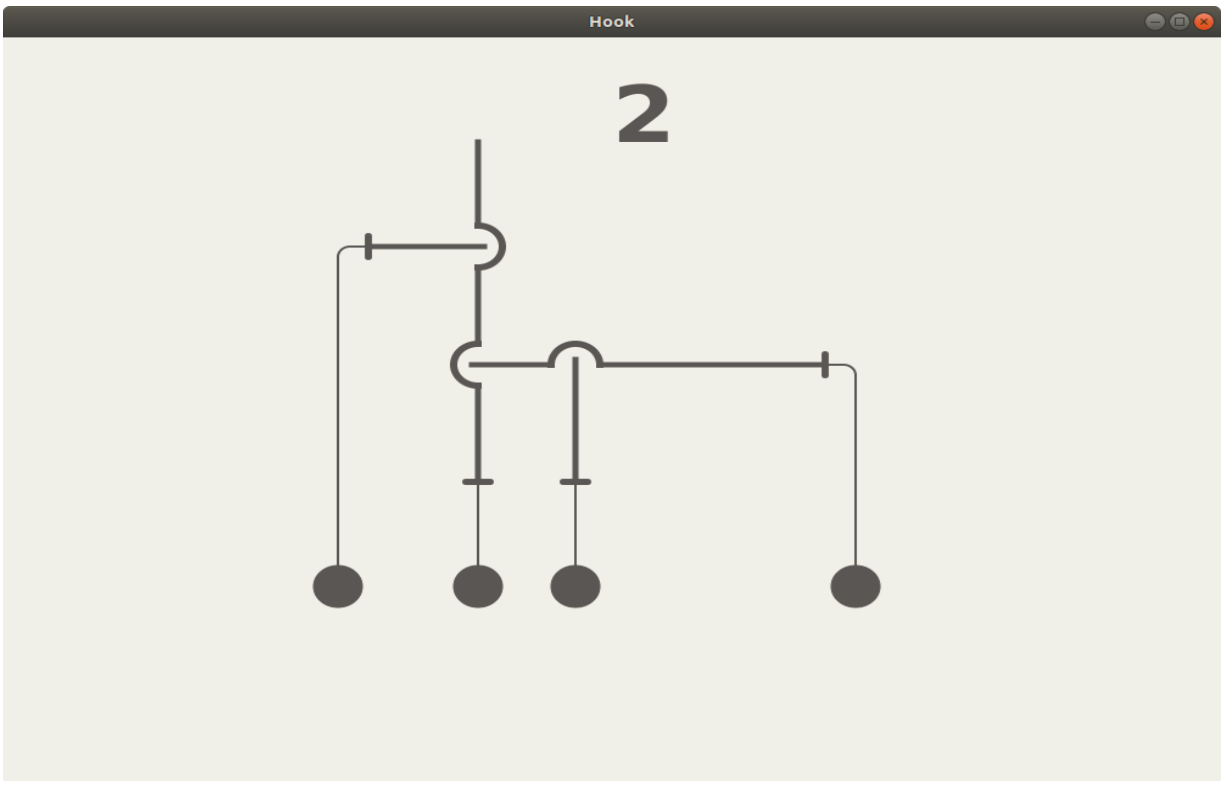
After pressing the start button, this level selection screen comes up. Only the 1st level was not locked, so only the 1st level is visible.



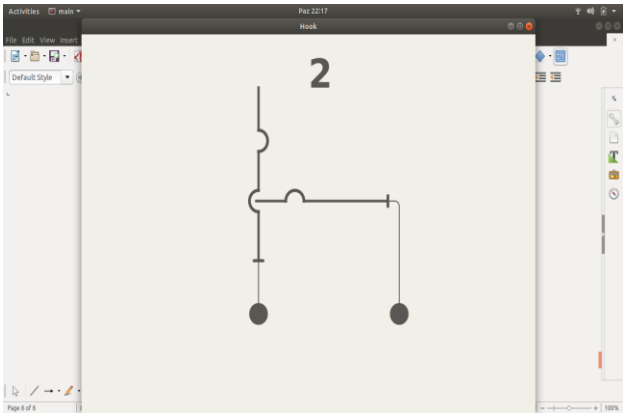
When I click on level 1, level 1 is turned on like this and my game stays on this screen until the 1st level is passed.

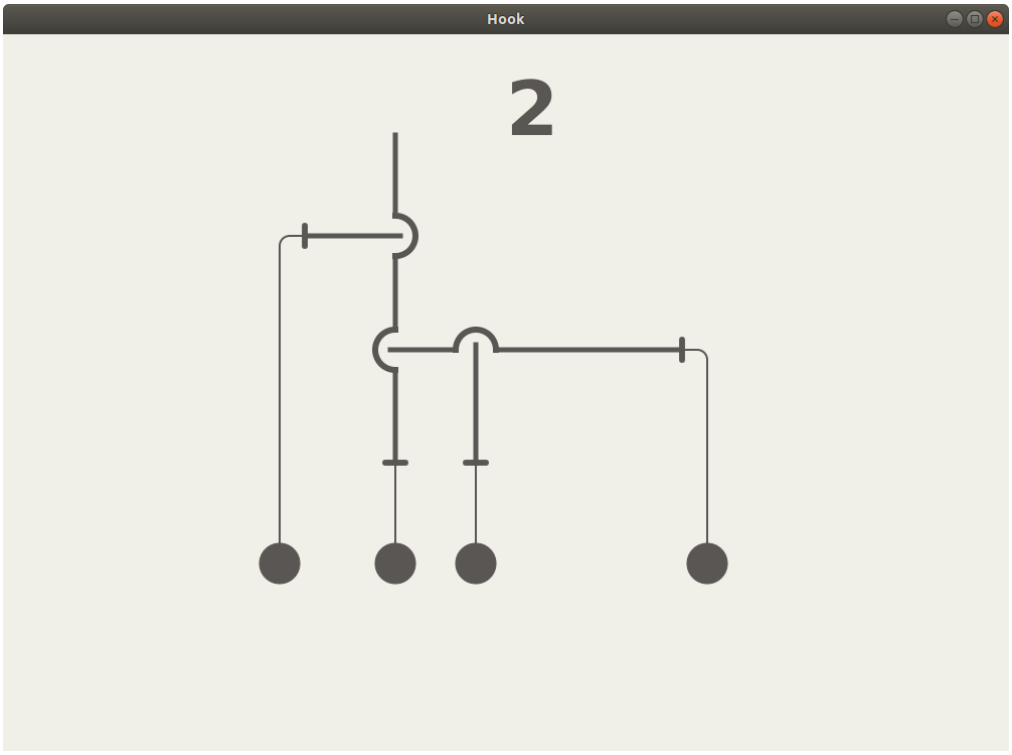


If I try to draw the wrong line by clicking on the wrong circle at first, the game will fail in this way and the level will start again.

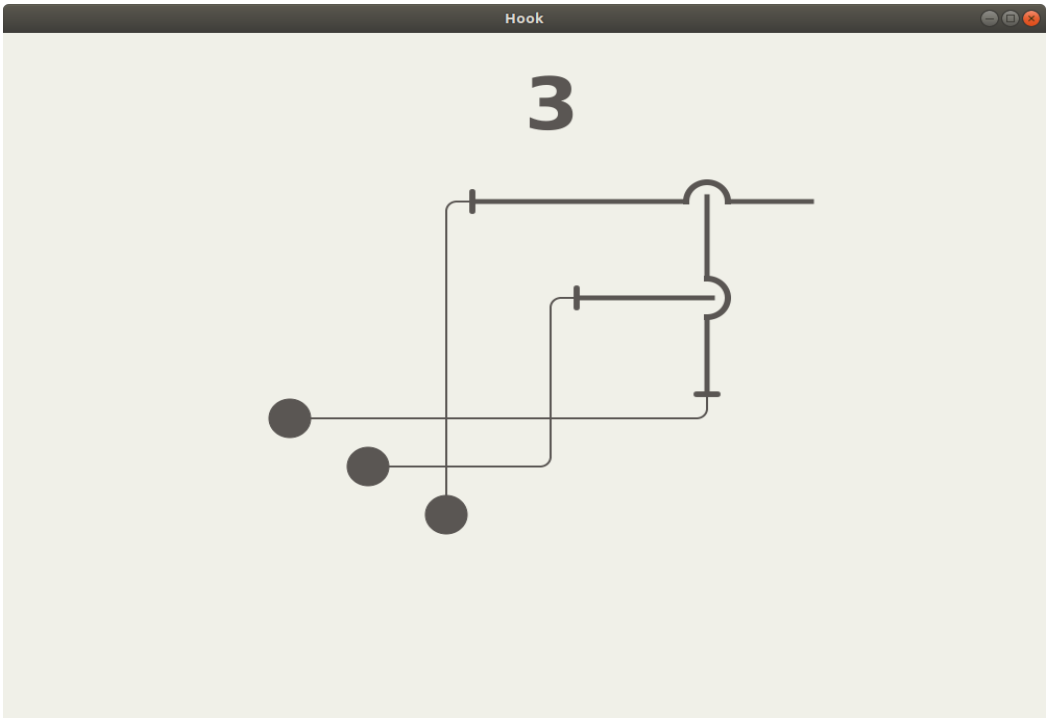


If I can successfully shoot two objects, the level is completed and the game automatically switches to 2nd level.

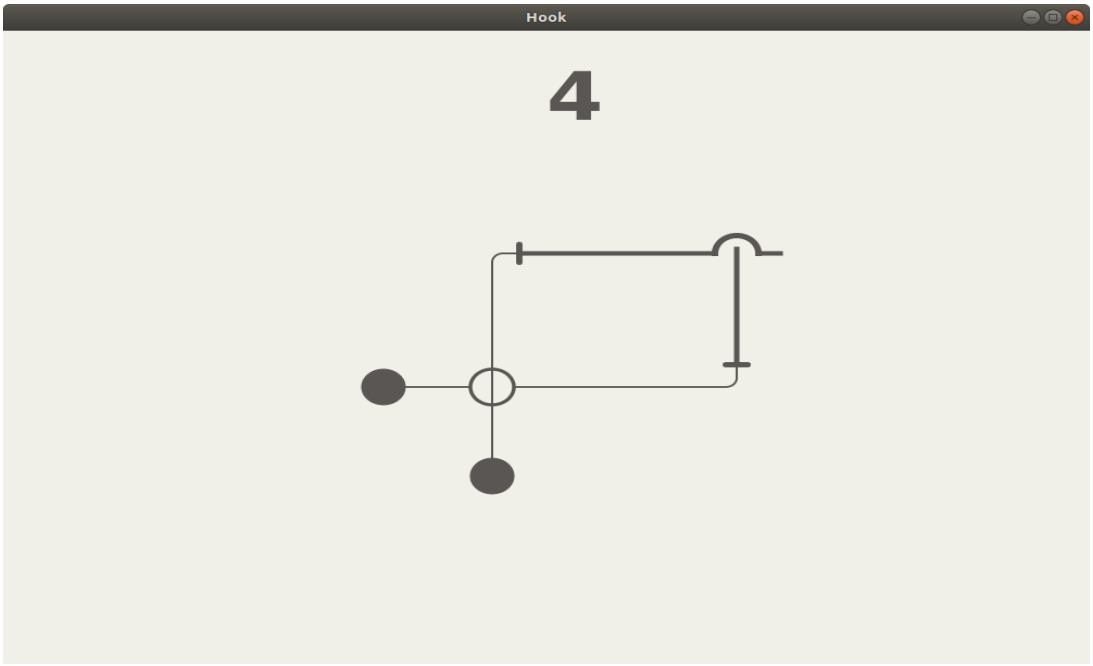




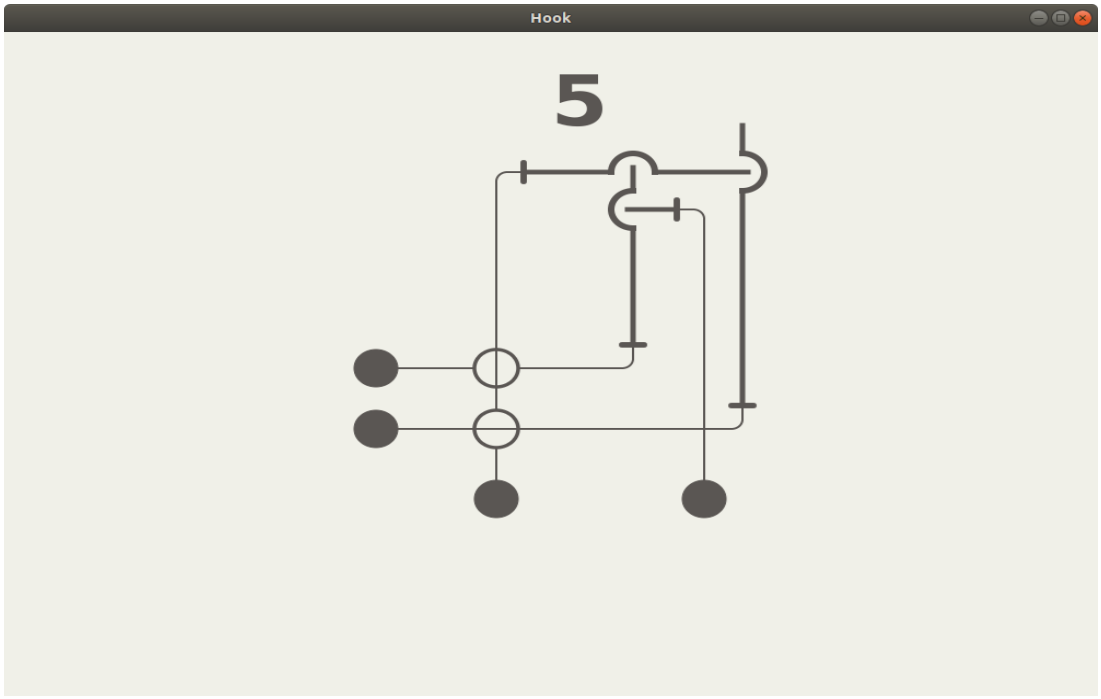
Even if I make a mistake after taking 2 bodies successfully like this picture, the level goes back to what it was from the beginning.



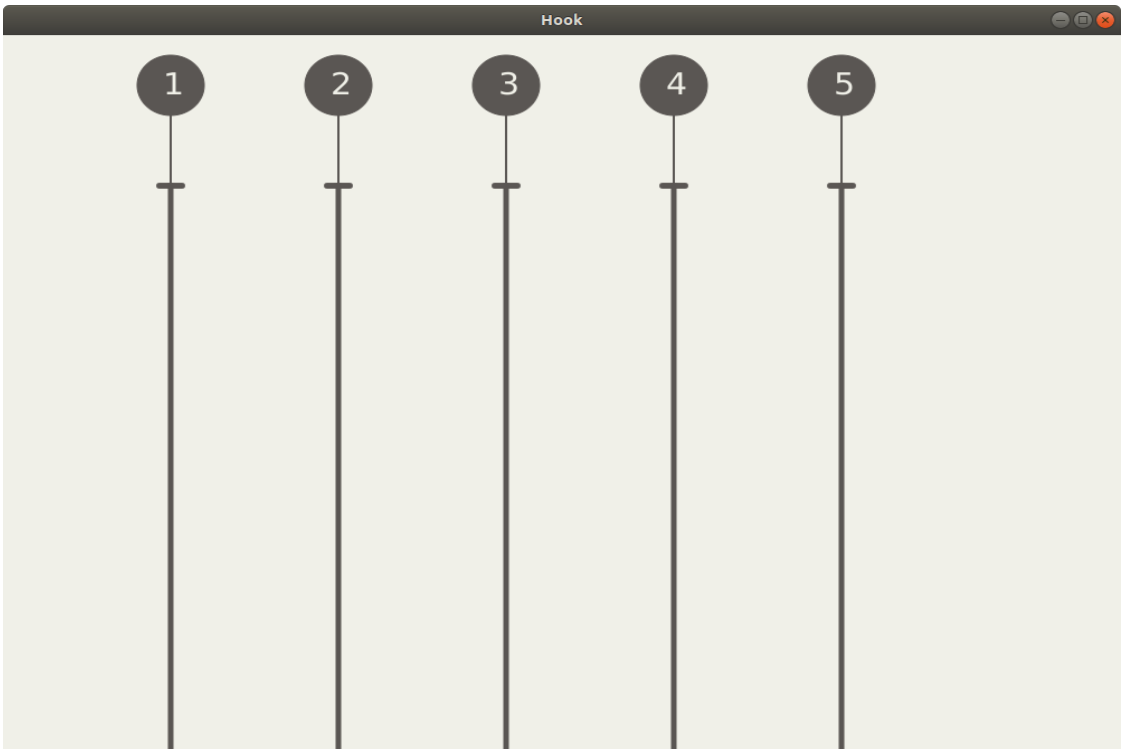
After successfully completing all objects in level 2, level 3 is automatically passed.



After successfully completing all the objects in level 3, the level 4 automatically switches to level 4. In this position, the object does not move because the key is closed when the left object is clicked, but when the key is in the correct position, the object can move.



When the key is in the correct position and the objects are successfully completed, it automatically switches to the 5th level. In this case there are 2 keys and 2 keys to which an object is connected. This object also controls 2 keys to move, and only if they are both doing their movements.



After completing correctly on the 5th level objects, the game returns to the level selection screen and all the planes are unlocked. From this screen you can choose the level you want and play again.