CSE 1142 - COMPUTER PROGRAMMING II
TERM PROJECT

DUE DATE: 20/05/2018 - 23:59 (No extension)

In the term project, you will implement the following game by using JavaFX framework. In this game, there are lines connected to filled-circles. When a filled-circle is clicked on, then the shape connected to it starts to slide towards to the filled-circle.

For example, in Figure 1, there exists a long horizontal line and a short vertical line connected to the filled-circle. If the user clicks on the filled-circle, the connected line will start to slide towards to the filled-circle and the length of the vertical line becomes bigger while the length of horizontal line becomes shorter and shorter gradually. When sliding horizontal line reaches to the vertical line (i.e. connection point), the filled-circle and the part connected to it disappear completely (Figure 3). It should be noted that the thin line connected to the filled circle is irremovable.

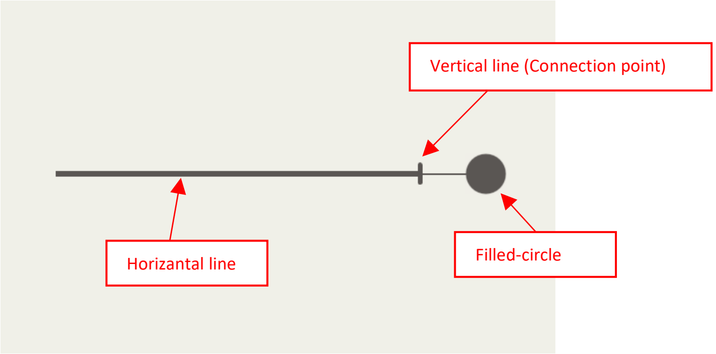


Figure 1. Initial state

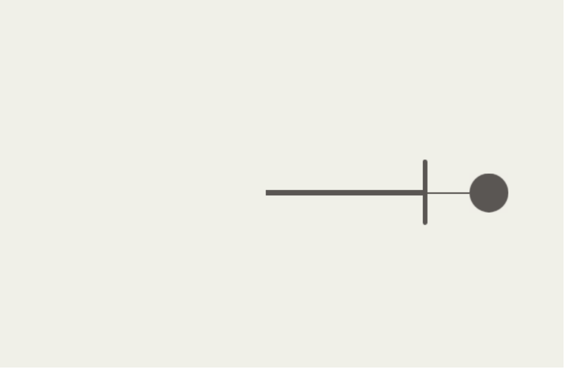


Figure 2. When the filled-circle is clicked on, the connected part starts to slide where the bold horizontal line becomes shorter and the bold vertical line becomes bigger.

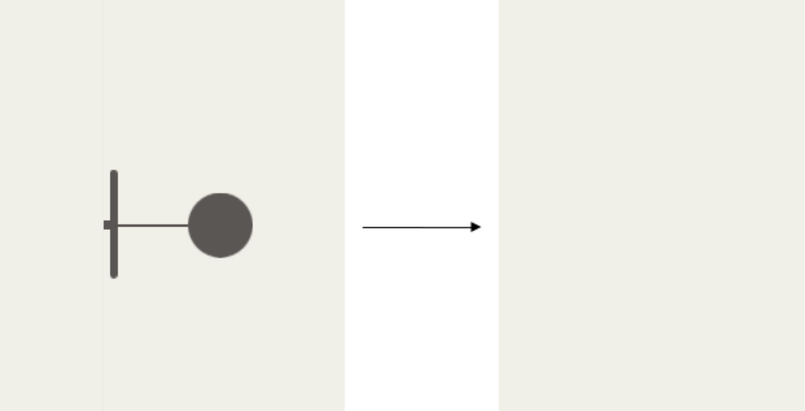


Figure 3. When sliding part reaches to the vertical line (or connection point), sliding part, and filled-circle disappear.

Semi-Circle

Semi-circle is one of the important components of the game (Figure 4). When the user clicks on the filled-circle, size of a semi-circle does not change but it also slides with horizontal line towards to the connection point of the filled-circle (Figure 5). When it reaches to the connection point, all the shapes will disappear (Figure 6).

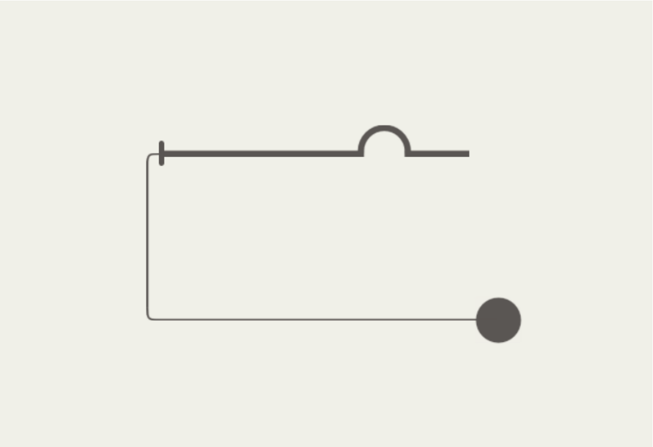


Figure 4. Initial State of a semi-circle

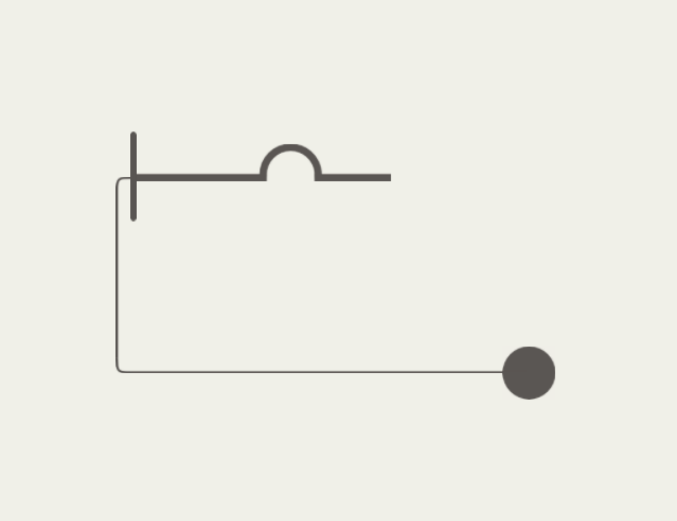


Figure 5. When the filled-circle is clicked on, semi-circle slides toward the connection part.

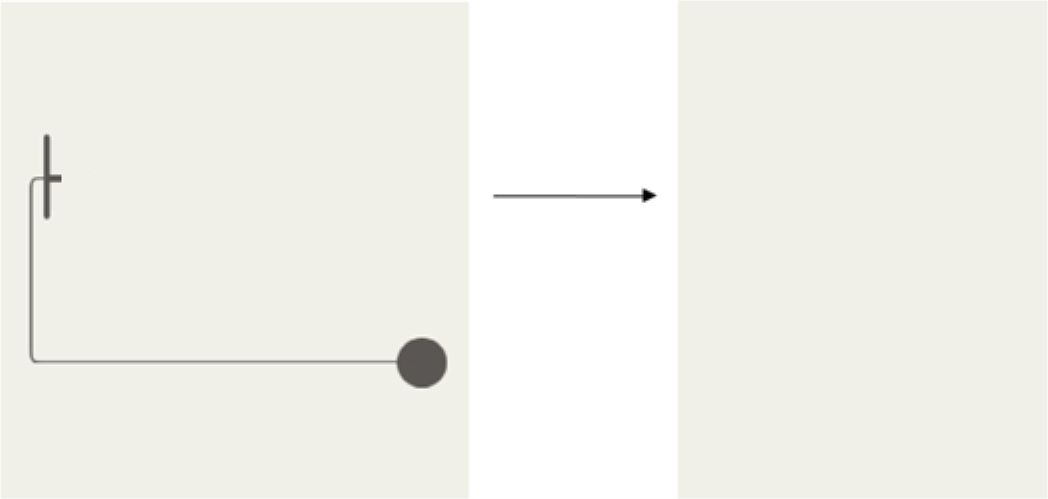


Figure 6. When the sliding part reaches to the connection point, then all the shapes will disappear.

Collision

Main purpose of this game is drawing complex figures and trying to remove all filled-circles without any collision. For this purpose, many filled-circles and their connected parts are drawn in a condition that there exist collisions. User should try to click on filled-circles in a proper order to avoid collision. You can see two collision examples in Figures 7 and 8.

To illustrate it, assume that the initial state of the game is given in Figure 7(a) and the user clicks on the filled-circle B first. In that case, the semi-circle connected to the filled-circle B collides with the edge of the filled-circle A. Conversely, if the user clicks on the filled-circle A first, no collision occurs and the part is completed successfully. Your program should handle the collision cases successfully.

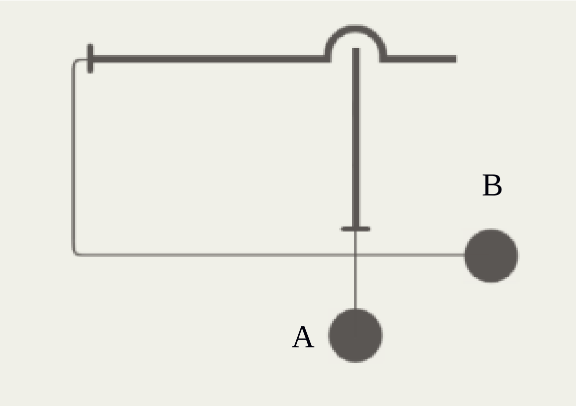


Figure 7(a) Initial State

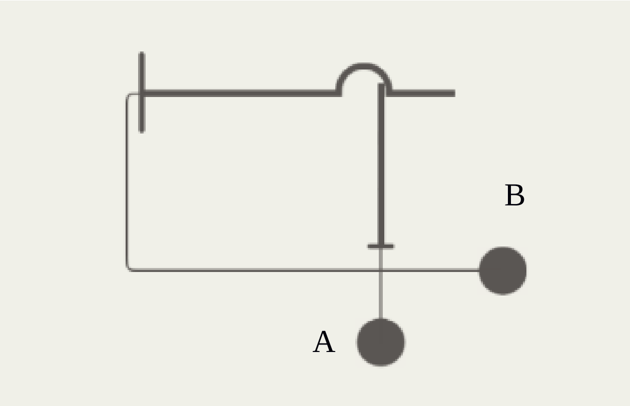


Figure 7(b) Collision

Another scenario is given in Figure 8. In this case, when the user clicks on the filled-circle A first, the semi-circle collides with the edge of the filled-circle B (Figure 8(b)). Conversely, if the user clicks on the filled-circle B first, no collision occurs and the part is completed successfully.

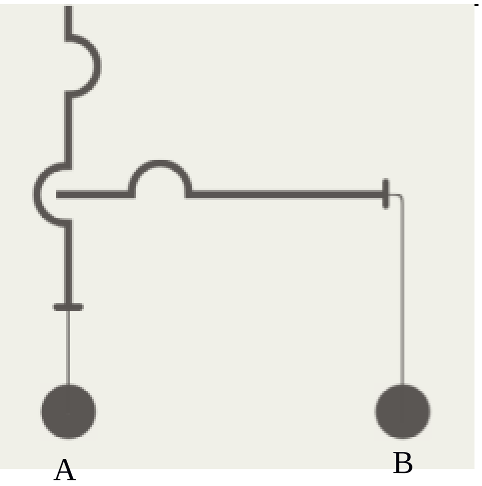


Figure 8(a) Initial State

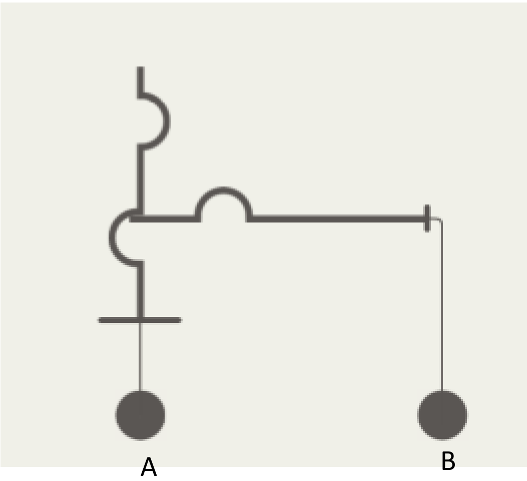


Figure 8(b) Collision

Disconnector

Disconnector disconnects filled-circles from the remaining parts if the position of the purple-colored line inside the connector is on the opposite direction.

- Dimension of this line can be changed by clicking on it. If the degree of the line is 90 degrees it becomes 180 degrees. In a similar manner, if the degree of the line is 180 degrees it becomes 90 degrees.
- More than one filled-circle can be connected to the same connector and this connects the filled-circles to their edges having the same direction as the connector line.

For example, if the initial state is Figure 9(a), the edge of the filled-circle cannot move even if the user clicks on the filled-circle. Because, the line inside the connector is on the opposite dimension of the edge. If the user clicks on the connector, the dimension of the line changes. Therefore, the edge of the filled-circle can slide if the filled-circle is clicked Figure 9(b).

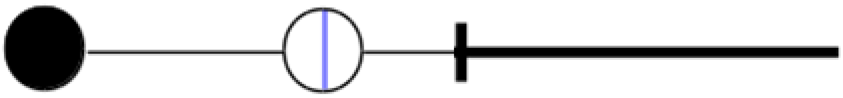


Figure 9(a)

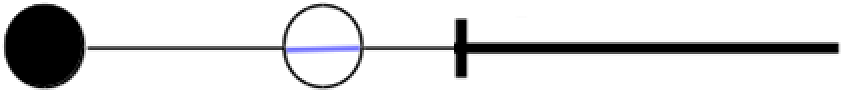


Figure 9(b)

Your Tasks

1. In the next section, you will see the levels of the game. You are expected to implement all levels.

2. If there are collision probabilities in a level your drawing should be done accordingly. In other words, you cannot eliminate any collision probability on your own by drawing some shapes smaller/bigger/on a more distant point etc. to eliminate/not to handle these cases. Additionally, size of filled-circles, semi-circles, connectors, lines and expansion rate of lines should be identical for each level.

3. Your code should be able to interpret whether any of the filled-circles is clicked.

- If it is so, it should start to slide the connected part of that filled-circle.
- The user can also click on other filled-circles one after the other. In this case, your code should slide all connected parts of all clicked filled-circles immediately.

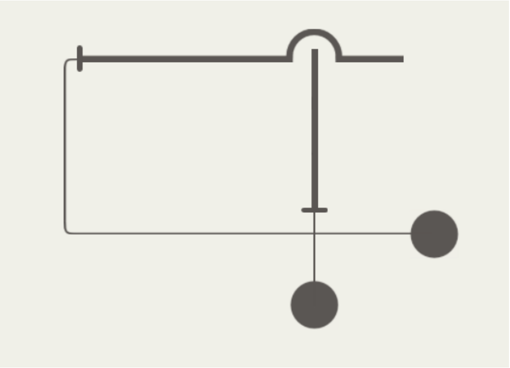
4. Your code should detect collisions. Please do not forget that there can be more than one collision at the same time because there can be many shapes sliding simultaneously.

5. If any collision is detected, then sliding should stop and you should warn the user about the situation and restart **the corresponding** level again.

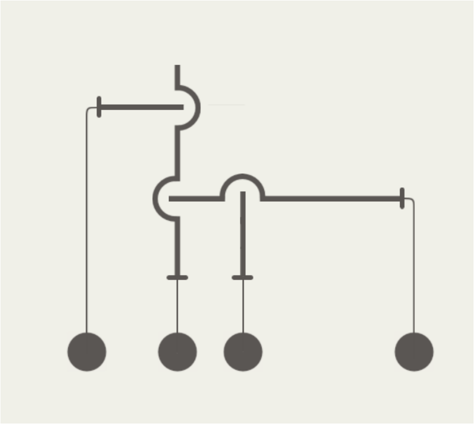
6. If the user does not cause any collision while playing, then your code should detect whether all of the filled-circles are removed or not. If there is no collision and all of the filled-circles are removed successfully, then you should start the next level.

LEVELS

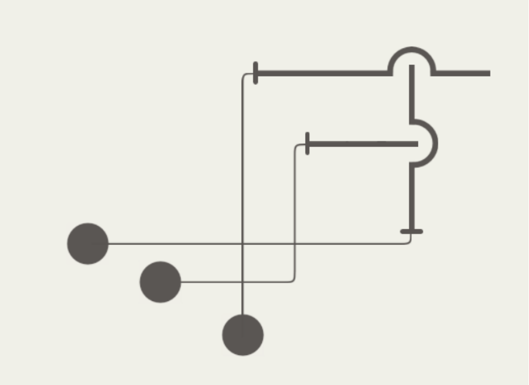
Level1



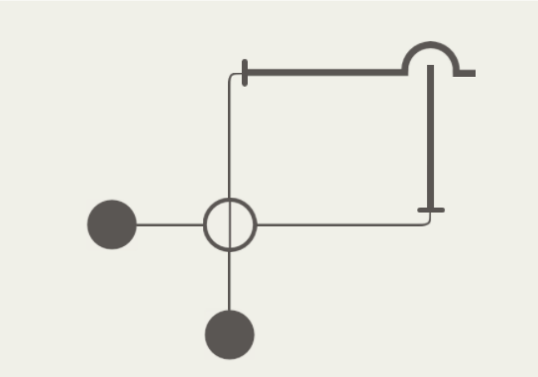
Level2



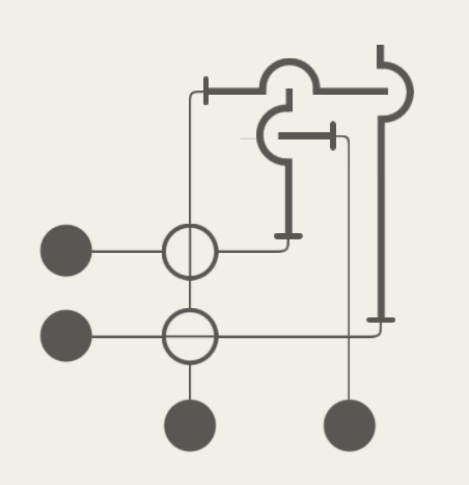
Level3



Level4



Level5



Submission Instructions

The final due date for the project is 20/05/2018 23:59. However, you should implement the GUI part of your project (without events) and submit it through the Canvas before 25/04/2018 23:59.

You are supposed to work in groups of 2 people.

Please zip and submit your files using filename Student1Number_Student2Number_Project.zip (ex: 150713852_150713098_Project.zip) to Canvas system (under Assignments tab).
Your zip file should contain the following files:

1. The commented source code of your project.
2. A 5-10 pages long report that contains UML diagram of your project, implementation details, and screenshots.

Demo Sessions

You will have demo sessions in the week of 21-25/05/2018. The exact time and date will be announced later. You should demonstrate what you have done in 20 minutes. You should also answer some questions about your implementation. Each group member will be assessed separately.

Details about Project Report:

- Your project reports are to be typed with normal sizes (Ex: Times New Roman 12pt.).
- Your report must have a cover page with the following information:
  Title
  Project Name
  Authors (IDs, Names, and Surnames)
  CSE1142: Computer Programming II, Spring 2018
  Date Submitted: May 20, 2018
- Firstly, you should provide a section named as “Problem Definition” and briefly describe the problem or the game in your project in 1-2 paragraphs by your own words.
- Then you should add a section named as “Implementation Details” and provide the UML diagrams of your project. In this section, you should also describe how you design and implement the project in more detail.
  - Additionally, you may provide information about
    - which parts are complete/incomplete in your project
    - what are the difficulties you have encountered during the implementation
    - what are the additional functionalities of your project added by your team
- Then, you should add a section named as “Test Cases” and this section should contain the results of your testing phase. You should provide the screenshots of your project execution for the given test cases and add explanations about them (Filling this part only with screenshots is not a feasible solution.)
- These are the minimum requirements for your project report. You can add more things into it.

- Be careful to have correct spelling and proper English grammar.
- The most important part about writing a project report is using your own words without copying-pasting from the Internet or the project document. Please show your own work.

Notes:

1. Write a comment at the beginning of each program to explain the purpose of the program.
2. Write your names and student IDs as a comment.
3. Include detailed comments to explain your actions. **Since this is the term project lack of comments will have an effect on your final grade.**
4. Select meaningful names for your variables and class names.
5. In case of any forms of cheating or copying, both giver and receiver are equally culpable and suffer equal penalties. You should submit your own work
   All types of plagiarism will result in FF grade from the course.
6. No late submission will be accepted.

Grading:

- First GUI Submission → %20
- Report → %10
- Final Project → %70