Computer Science

COMPSCI 230

Assignment TWO

Introduction

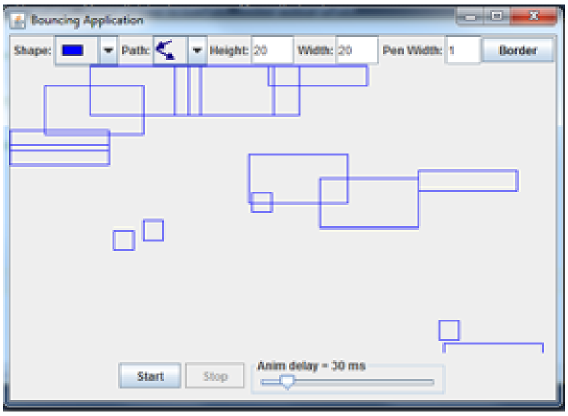
In this programming assignment, you are asked to add extra functions to the bouncing program. The aim of this assignment is to give you experience with oriented programming principles of inheritance and polymorphism.

Due Date

Due:	**4:00 pm Friday 29h April 2016**
Worth:	**5% of the final mark**

Introduction - The Bouncing Program

The application, as given, is a simple bouncing program. Different shapes move around in various paths.



Height: 20
Width: 20
Pen Width: 1

Actions

The user can create a new shape by clicking anywhere within the panel area of the program. The properties of the newly created shape are based on the current values saved in the appropriate UI fields (e.g. height, width etc).

Selecting/deselecting shapes:

A user can select a shape by clicking anywhere on the shape. If a shape is selected, all its handles are shown. The user can change the path types/widths/heights for all selected shapes by changing the current values with the help of the tools provided at the top of the application interface. (But the shape type can't be modified once a shape has been created.)

Clicking on a selected shape will deselect it.

Tools

Shape Combo Box:	The 'Shape' combo box sets the current shape type for new shapes. Clicking in the panel area for Shape Creation will create the selected type of the shape. A rectangle or a circle can be selected in the program.
Path Combo Box:	Users may select one of several moving paths for shapes from the 'Path' combo box. Selecting a new path changes the path of all currently selected shapes. Additionally, the new path becomes the current path for any new shapes that are created.
Width TextField:	Users may change the current width of new shapes and currently selected shapes by entering a valid number in the width text field and pressing "ENTER".
Height TextField:	Users may change the current height of new shapes and currently selected shapes by entering a valid number in the height text field and pressing "ENTER".
Border Colour Button	Users may change the current border colour of new shapes and currently selected shapes by selecting a colour from the colour dialog box and pressing "OK".
Start Button:	Starts the animation.
Stop Button:	Stops the animation.
Animation Slider:	Users may use the animation delay slider to adjust the speed of the animation.
Popup Menu:	The application has a popup menu, which is activated by clicking the right mouse button anywhere in the panel area (on a windows machine). The popup menu contains a menu item called "Clear All" which allows the user to clear all shapes from the program.

What you are to do

Firstly, become familiar with the program supplied. The files included in the program are as follows:

- `A2.java`
- `AnimationPanel.java`
- `MovingShape.java`
- `MovingRectangle.java`

Download all source files from the assignment course page. The design and implementation of the program will be covered in lectures, please refer to the relevant material. It is strongly recommended to start as early as you can and implement the parts you know as soon as they are taught in lectures.

Your assignment is divided into several stages for ease of completion. Please complete the assignment in order of the stages.

Stage 1: Using An `ArrayList` (10 marks)

In this part you are required to modify classes in A2 which enable users to store instances of shapes using an `ArrayList`.

Assessment criteria

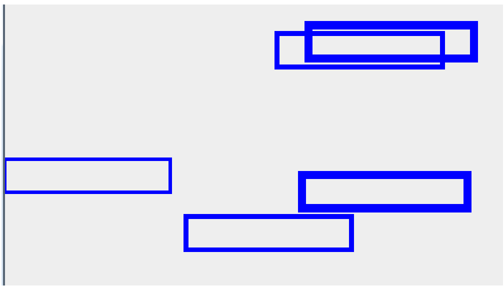
- Users should be able to create a new shape and add it into the `ArrayList`.
- Users should be able to modify path/height/width of all the selected shapes in the `ArrayList`.
- Users should be able to remove all shapes in the `ArrayList`.
- Users should be able to draw all shapes in the `ArrayList`.
- Users should be able to reset all margin sizes of all shapes in the `ArrayList`.

Stage 2: Pen Width (5 marks)

In this part you are required to modify classes in A2 which enable users to change the pen width of all the currently selected shapes and the current pen width that will be used when creating new shapes.

You are required to add a `setCurrentPenWidth`() method into the `AnimationPanel` class in order to set the pen width of all the currently selected shapes and the pen width that will be used when creating new shapes. You are also required to add a `getCurrentPenWidth()` method into the `AnimationPanel` class to return the CURRENT pen width.

You should add the set and get methods to the `MovingShape` class in order to set or get the pen width of shapes. You should also modify the `draw` method of `MovingRectangle` class (and all subclasses) in order to use the pen width attribute stored in the superclass to draw the shape.



Assessment criteria

- Users should be able to change the default pen width.
- Users should be able to change the pen width of all selected shapes.

Stage 3: Adding new Shapes (25%)

The `MovingShape` is an abstract class which contains two abstract methods: draw and contains. You are required to add new subclasses. You may need to implement some or all abstract methods for the new shapes. You may also need to add a private instance field to store a specific property of the new shape. You will need to think carefully on the structure of the inheritance hierarchy.

`A2` and `AnimationPanel`

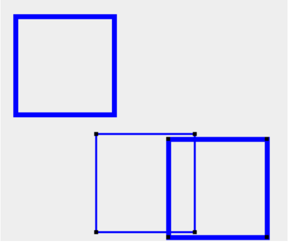
Next, you are required to add a new `ImageIcon` to the '`Shape`' combo box control in the `A2` class for each new type of shape. You are also required to modify the `createNewShape` method in the `AnimationPanel` class which allows users to create each new subclass instance.

Assessment criteria:

- Users should be able to add new shapes to the bounce program using the default border colour, height, width, pen width and path.
- Users should be able to change the width and/or height of the selected shapes.
- Users should be able to change the pen width of the selected shapes.
- Users should be able to change the border colour of the selected shapes.
- Users should be able to change the bouncing path of the selected shapes.

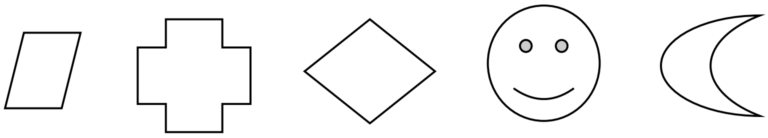
Stage 3A: `MovingSquare` Class (5 marks)

You are required to add a new class to the bouncing program. This class should draw a square based on the current width, height, pen width, border colour and the bouncing path stored in the `AnimationPanel`. Some examples are shown in the following diagram.



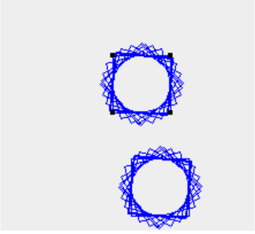
Stage 3B: `MovingPolygon` Class (10 marks)

You are required to add a new class to the bouncing program. This class should draw a polygon based on the current width, height, pen width, border colour and the bouncing path stored in the `AnimationPanel`. Some examples are shown in the following diagram. You can choose any one or all of them.



Stage 3C: `MovingRotatingSquare` Class (10 marks)

You are required to add a new class to the bouncing program. This class should draw a list of rotating squares based on the current width, height, pen width, border colour and the bouncing path stored in the `AnimationPanel`. Some examples are shown in the following diagram.



You may need to use methods from the `AffineTransform` class to rotate squares.

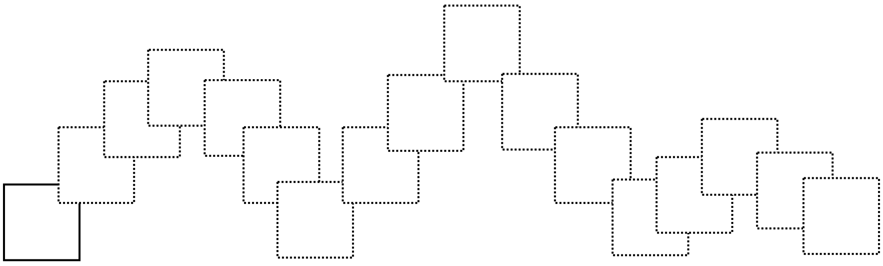
Stage 4: Adding a New Path (10 marks): `JumpingPath` class

In this part, you are required to add a jumping path to the bouncing program.

The `MovingPath` is an abstract inner class which contains an abstract method. You are required to add a new subclass which extends the `MovingPath`. You may need to add a private instance field to store a specific property of the new path. You will need to think carefully on the structure of the inheritance hierarchy.

A2 & MovingShape

Next, you are required to add a new `ImageIcon` to the '`Path`' combo box control in the `A2` class for the new type of path. You are also required to modify the `setPath` method in the `MovingShape` class which allows users to create a new subclass instance. An example is shown as below:



Assessment criteria

- Users should be able to add a new shape which bounce off using the jumping path idea.
- Users should be able to change the bouncing path of the selected shapes to the jumping path.

Submission

You may electronically submit your assignment through the Web Dropbox (https://adb.auckland.ac.nz/) at any time from the first submission date up until the final date. You can make more than one submission. However, every submission that you make replaces your previous submission. Submit ALL your files in every submission. Only your very latest submission will be marked.

Please double check that you have included all the files required to run your program and the A2.txt in the zip file before you submit it. No marks will be awarded if your program does not compile and run.

You are to electronically submit ONE **A2.zip** file containing all the following files:

1. All source files (i.e. new, changed, and unchanged) - Your name, UPI and a comment at the beginning of each file must be included in all your files.
2. All gif files (used as icons in the program)
3. **A2.txt**

What to include inside the A2.txt file

You must include a text file named A2.txt in your submission. This text file must contain the following information:

- Your name, login name and ID number
- How much time did the assignment take overall?
- What areas of the assignment did you find easy?
- What areas of the assignment did you find difficult?
- Which topics of the course did the assignment most help you understand?
- Any other comments you would like to make.

DO NOT SUBMIT SOMEONE ELSE'S WORK:

- The work done on this assignment must be your own work. Think carefully about any problems you come across, and try to solve them yourself before you ask anyone for help.
- Under no circumstances should you take or pay for an electronic copy of someone else's work. This will be penalized heavily.
- Under no circumstances should you give a copy of your work to someone else
- The Computer Science department uses copy detection tools on the files you submit. If you copy from someone else, or allow someone else to copy from you, this copying will be detected and disciplinary action will be taken.
- To ensure you are not identified as cheating you should follow these points:
    - Always do individual assignments by yourself.
    - Never give another person your code.
    - Never put your code in a public place (e.g., forum, your web site).
    - Never leave your computer unattended. You are responsible for the security of your account.
    - Ensure you always remove your USB flash drive from the computer before you log off.