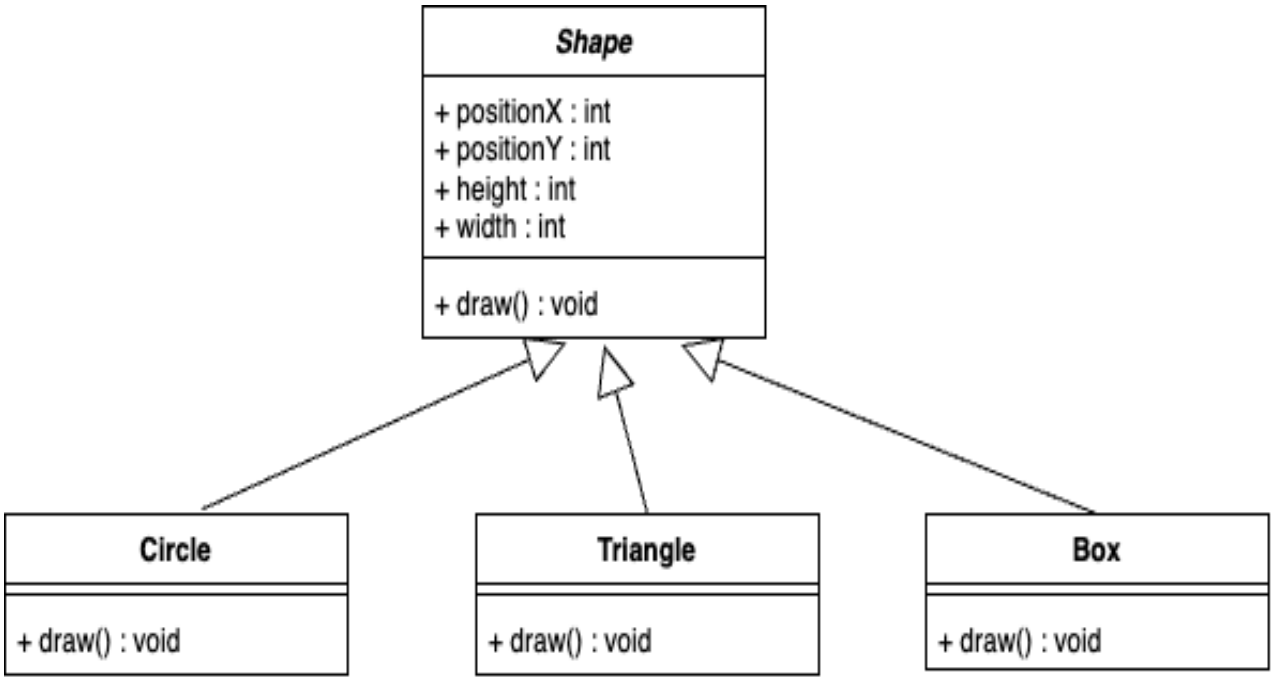
Name - Soham Balkothe
Student ID - 1001500980
Email - svb0980@mav.uta.edu

1. Domain Model Class Diagram



2.Expanded use case for <u>Draw Shape</u>

Actor: User	System: Application
	0. The application displays three buttons and a drawing area.
1. TUCBW the user clicks on one of the three buttons (Circle, Triangle, Box).	*2. The application highlights the button clicked by the user.
3. The user clicks on the drawing area to draw the shape at that position.	*4. The application draws the selected shape and displays the updated drawing area.
5. TUCEW the user sees the shape drawn at the selected location in the drawing area.	

3. Scenario for each non-trivial step

We have two non-trivial steps in the expanded use case - step 2 and step 4

Scenario for step 2

1. The user clicks on one of the three buttons (Circle, Triangle, Box) on CanvasGUI.
2.1. The CanvasGUI highlights the clicked button.
2.2. The CanvasGUI sends the shape type to CanvasController.
2.3. The CanvasController stores the shape type.
3. The user clicks on the drawing area to draw the shape at that position.

Scenario for step 4

3. The user clicks on the drawing area to draw the shape at that position.
4.1. The CanvasGUI captures the position clicked.
4.2. The CanvasGUI sends the captured position to the CanvasController.
4.3. The CanvasController creates a Shape object with the selected shape type and position.
4.4. The CanvasController adds the shape to the CanvasComposite.
4.5. For each shape in CanvasComposite
4.5.1 The CanvasController calls the draw method from the respective shape class(Circle, Box, Triangle).
5. The user sees the shape drawn at the selected location in the Drawing area in CanvasGUI.

4. Scenario Tables for each scenario

Scenario table for step 2: The application highlights the button clicked by the user.

Table 1

Step	Subject	Subject Action	Other Data/Objects	Object acted upon
1.	The user	Clicks on	button(Circle, Triangle, Box)	CanvasGUI
2.1.	CanvasGUI	highlights	Clicked button	CanvasGUI
2.2.	CanvasGUI	sends	Shape type	CanvasController
2.3.	CanvasController	stores	Shape type	CanvasController
3.	The user	Clicks on	Drawing area	CanvasGUI

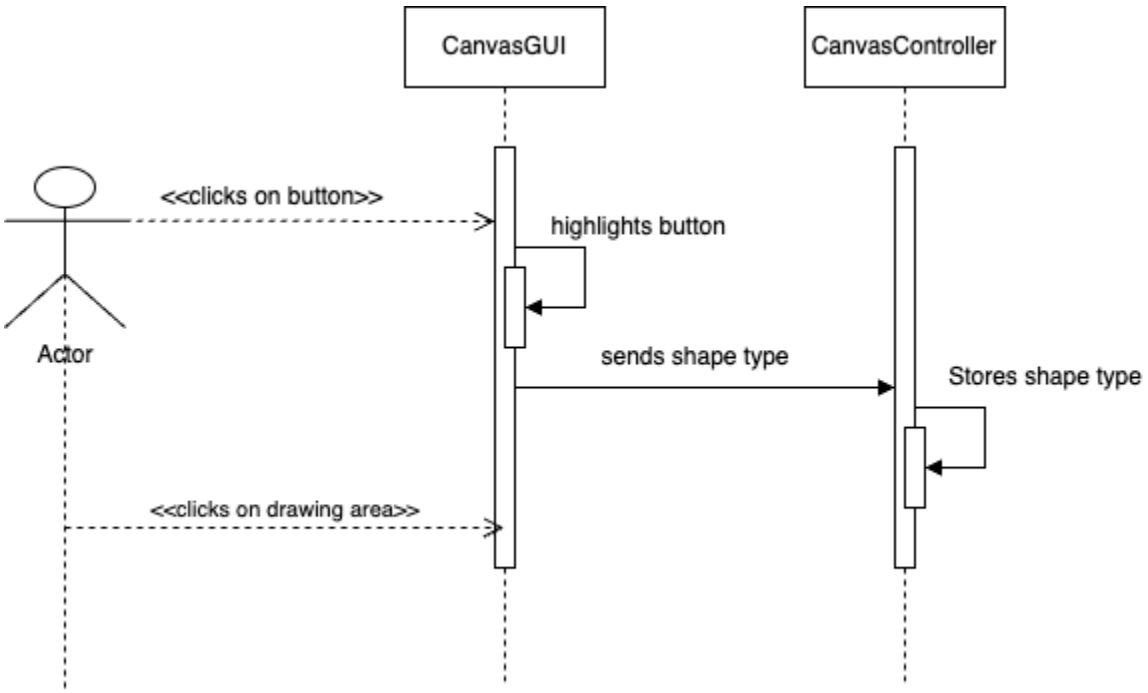
Scenario table for step 4: The application draws the selected shape and displays the updated drawing area.

Table 2

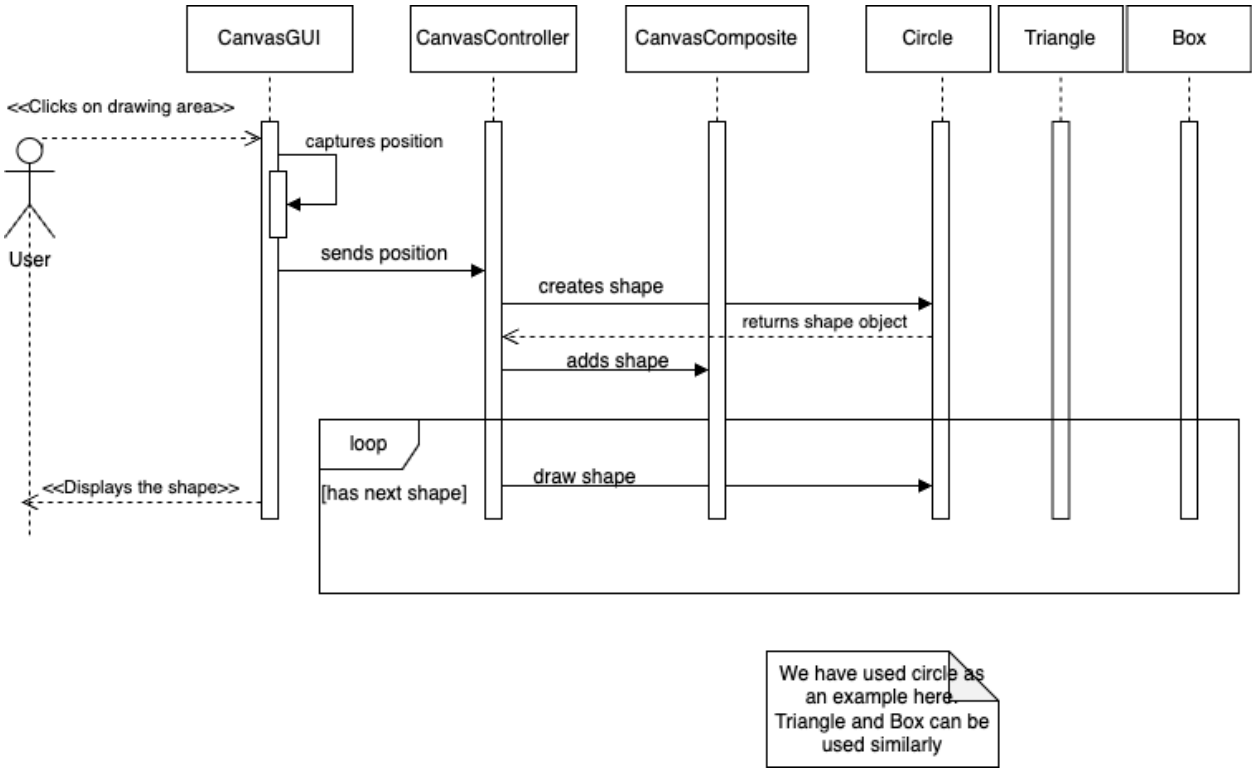
Step	Subject	Subject Action	Other Data/Objects	Object acted upon
3.	The user	Clicks on	Drawing area	CanvasGUI
4.1.	CanvasGUI	captures	Position clicked	CanvasGUI
4.2.	CanvasGUI	sends	Captured position	CanvasController
4.3.	CanvasController	creates	Shape type and position	Shape object
4.4.	CanvasController	adds	Shape object	CanvasComposite
4.5.	For each shape in CanvasComposite			
4.5.1	CanvasController	calls	Draw method	Shape class
5.	User	sees	The shape	CanvasGUI

5. Informal Sequence Diagrams for each scenario table

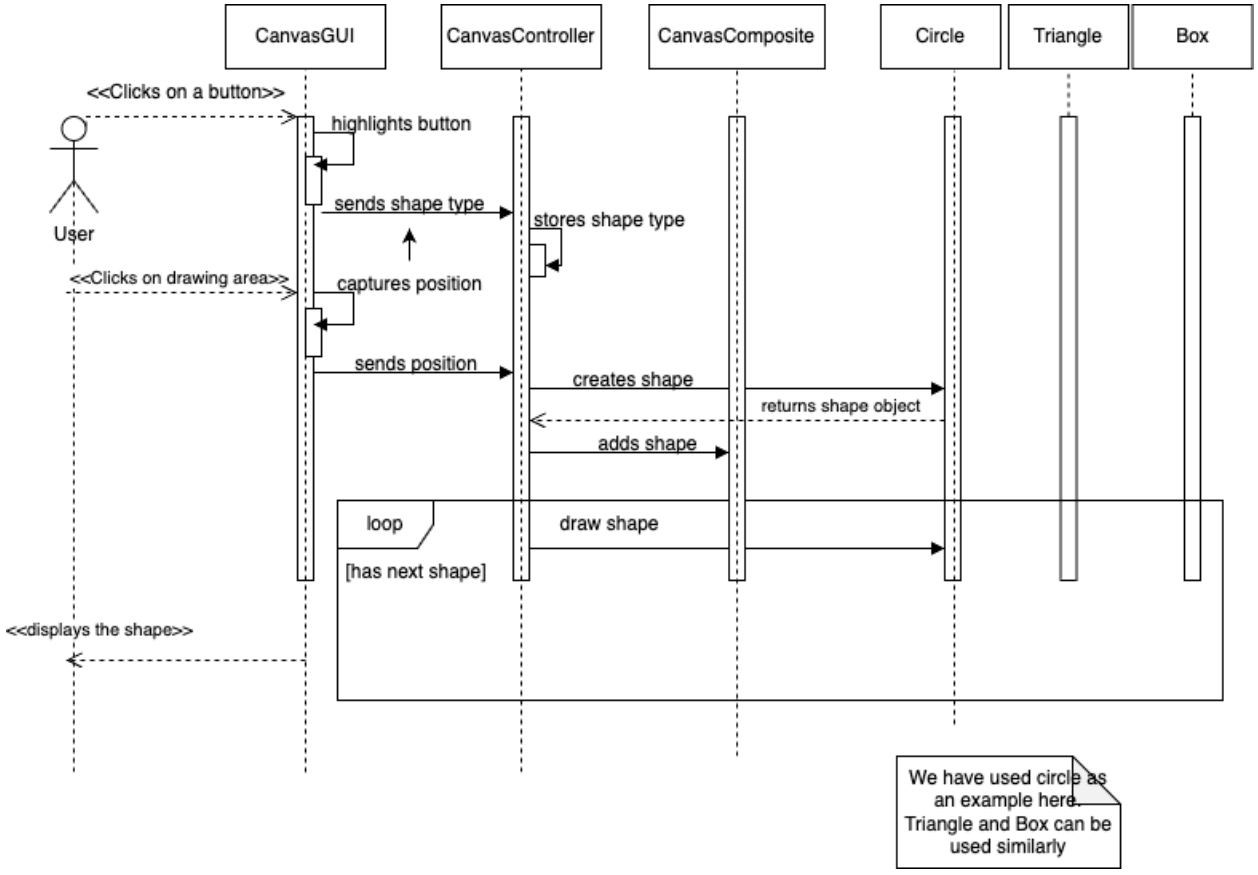
Informal Sequence Diagram for Table 1.



Informal Sequence Diagram for Table 2.

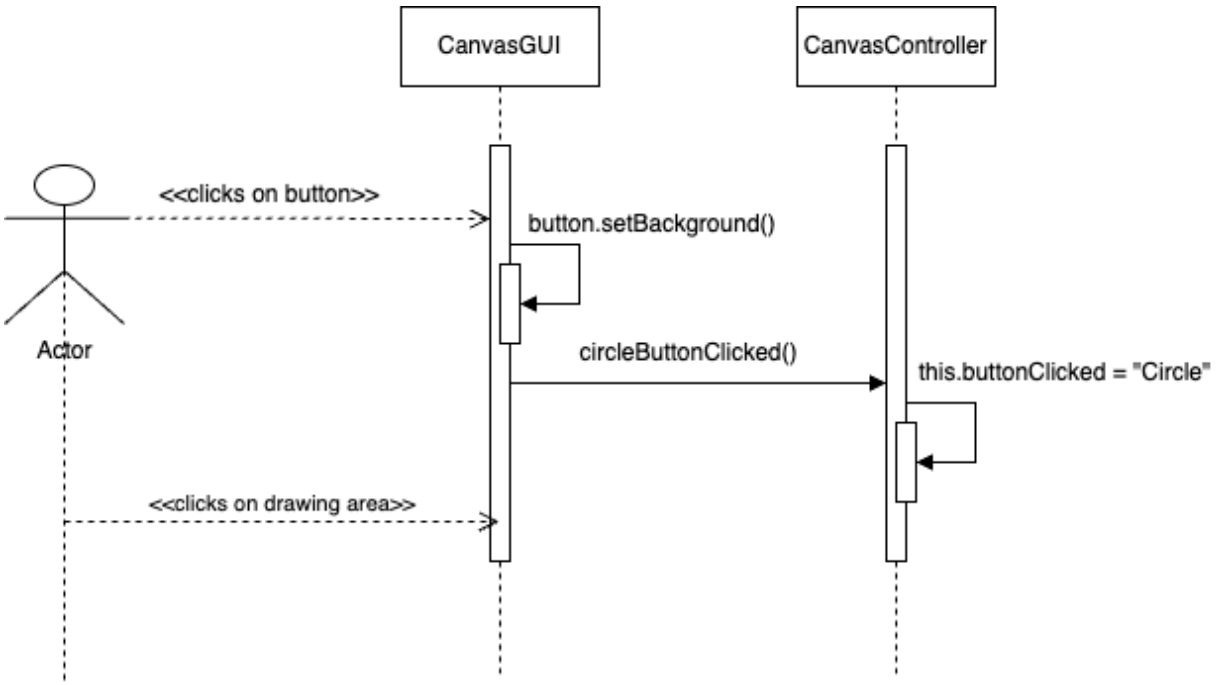


Resulting Informal Sequence Diagram

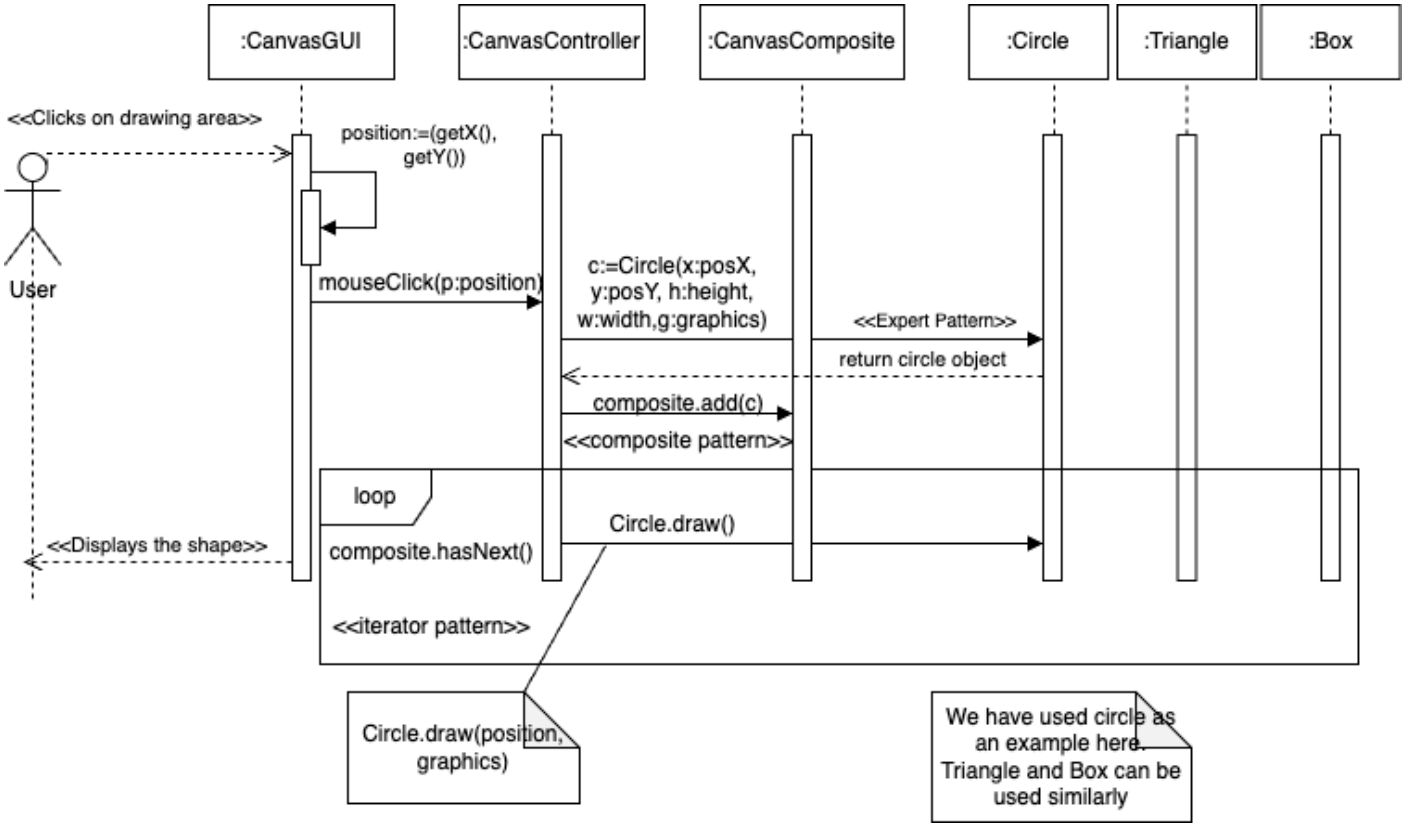


6. Design Sequence Diagrams for each informal sequence diagram

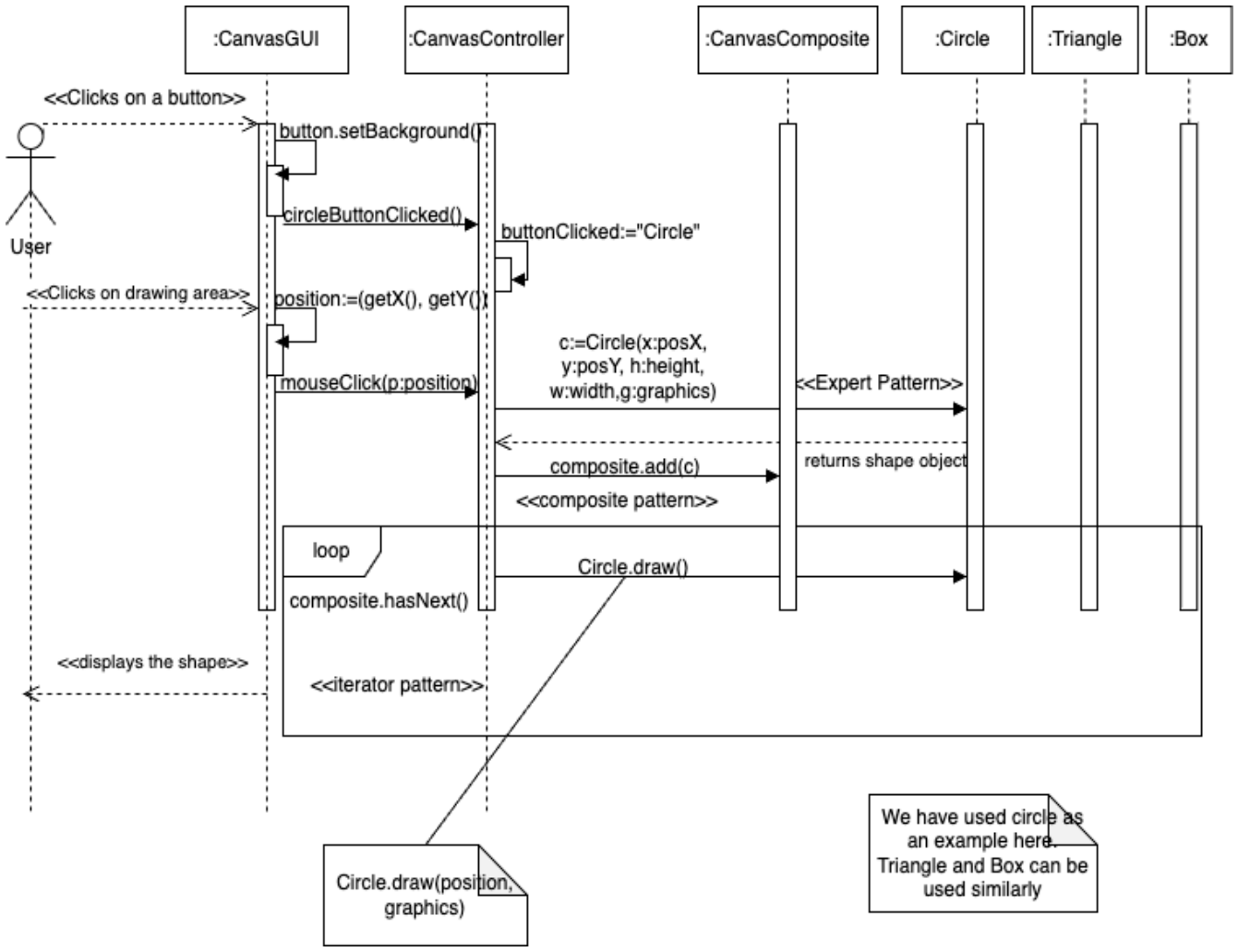
Design Sequence Diagram for Informal Sequence Diagram 1.



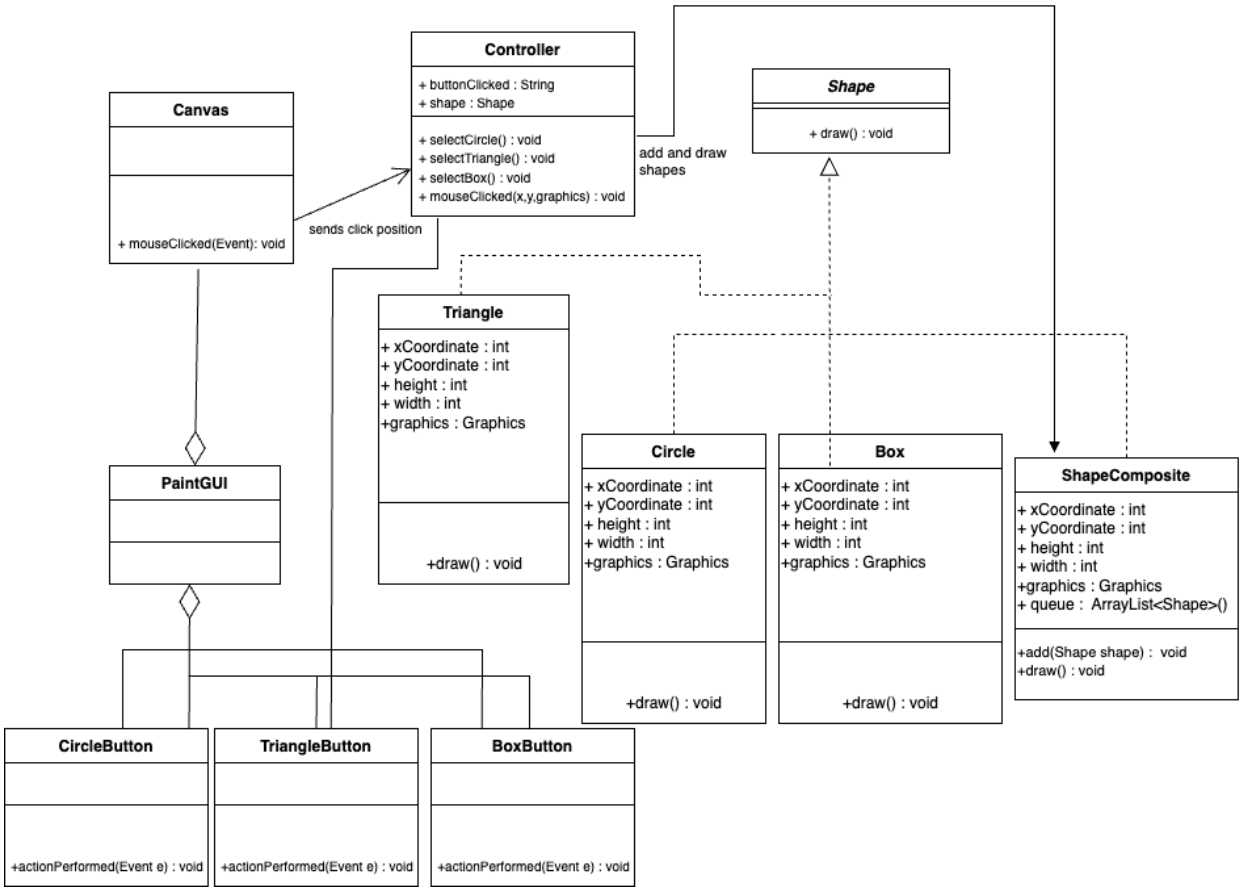
Design Sequence Diagram for Informal Sequence Diagram 2.



Resulting Design Sequence Diagram



7. Design Class Diagram



8. Application implemented using java in implementation folder.