

EIA2 Endabgabe: Konzeption

Für welche Plattform ist die Anwendung?

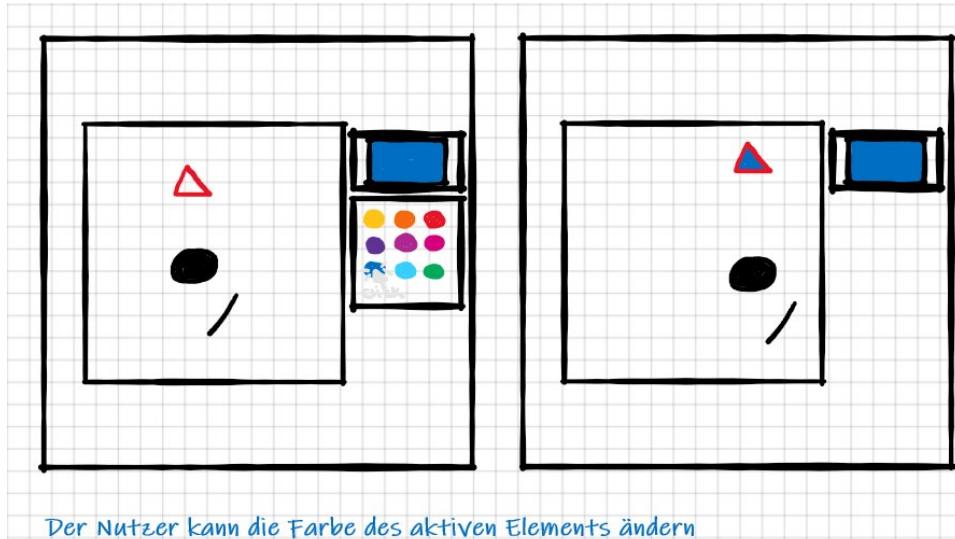
Kriterium	PC/ Laptop	Handy
Wo verwendet der Nutzer das Gerät am häufigsten?	Zu Hause	Unterwegs
Bildschirmgröße	14- 17,3 Zoll 1920 x 1080 Pixel	5 Zoll 640 x 360
Maus vorhanden?	Ja	Nein
Internetverbindung?	Ja	Ja, aber häufig instabil
Hat der Nutzer Zeit, mit der Anwendung zu interagieren?	Die Wahrscheinlichkeit ist höher, dass der Nutzer die Anwendung länger nutzt	Der Nutzer ist unterwegs und hat daher wahrscheinlich weniger Zeit, die Anwendung lange zu nutzen

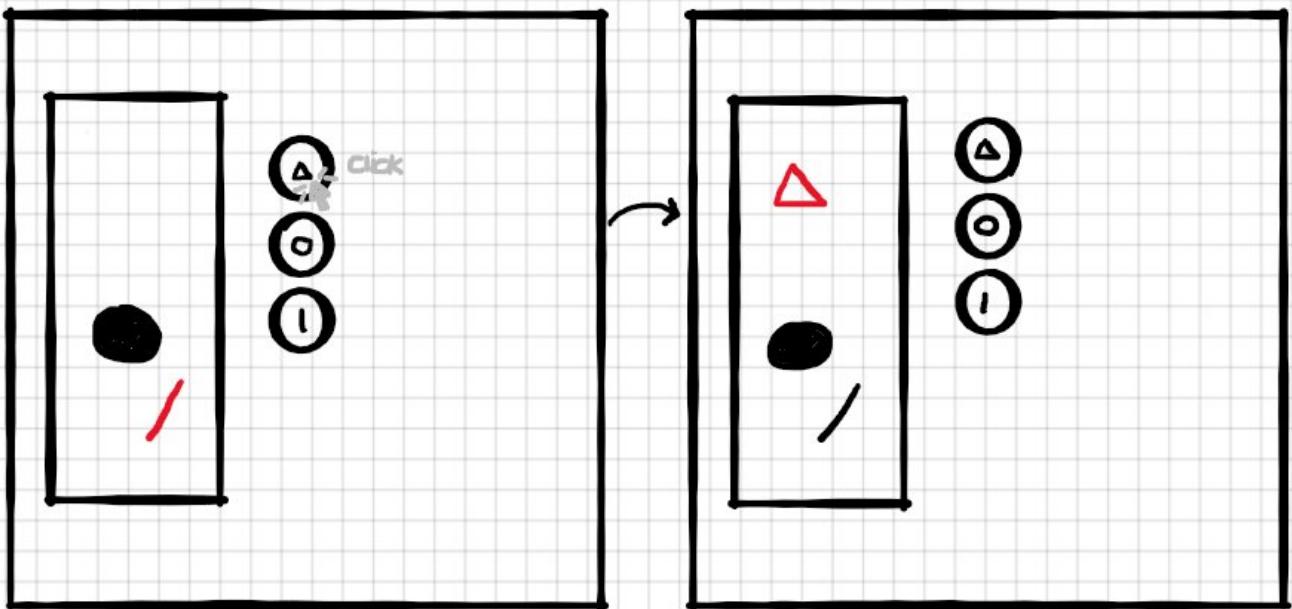
Was sieht der Nutzer während der Interaktion?

An welche Information gelangt der Nutzer noch?

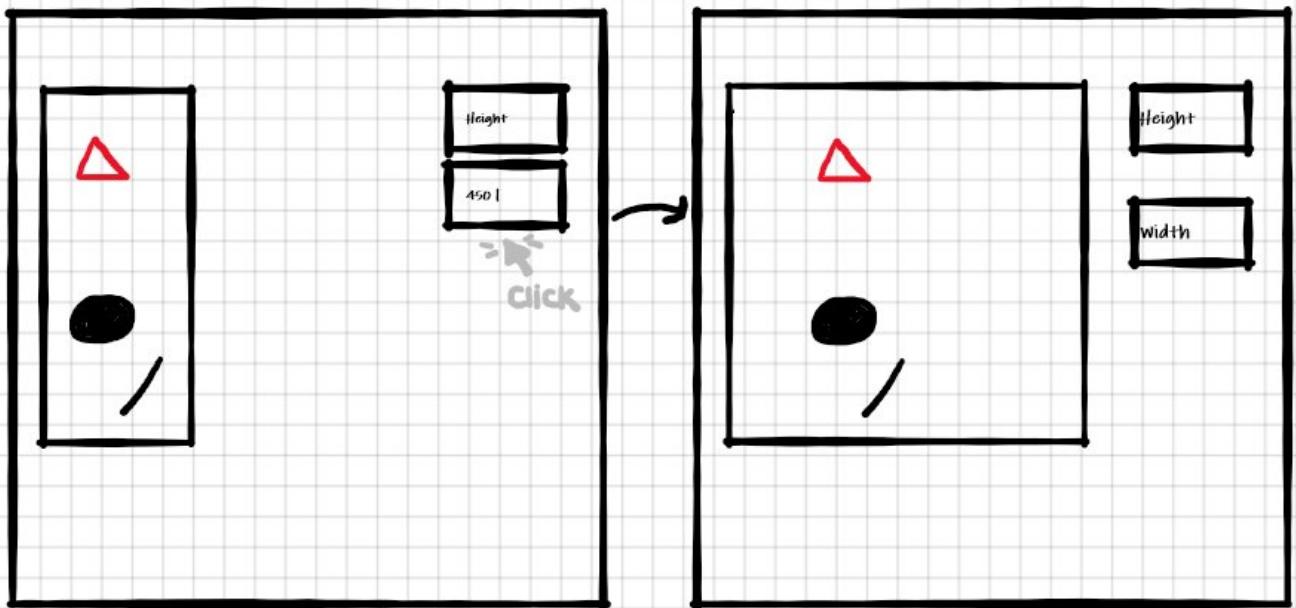
- Das aktive Element wird sichtbar umrandet, sodass der Nutzer immer sieht, an welchem Element er gerade arbeitet
- Mit Alerts wird der Nutzer auf falsche/ fehlende Eingaben hingewiesen. So werden schon einige Fehler vermieden und es wird sichergestellt, dass alle notwendigen Daten/ Werte vorhanden sind
- Es gibt eine Auflistung aller Elemente, die sich gerade auf dem Canvas befinden

Skizzen und Beispielbilder





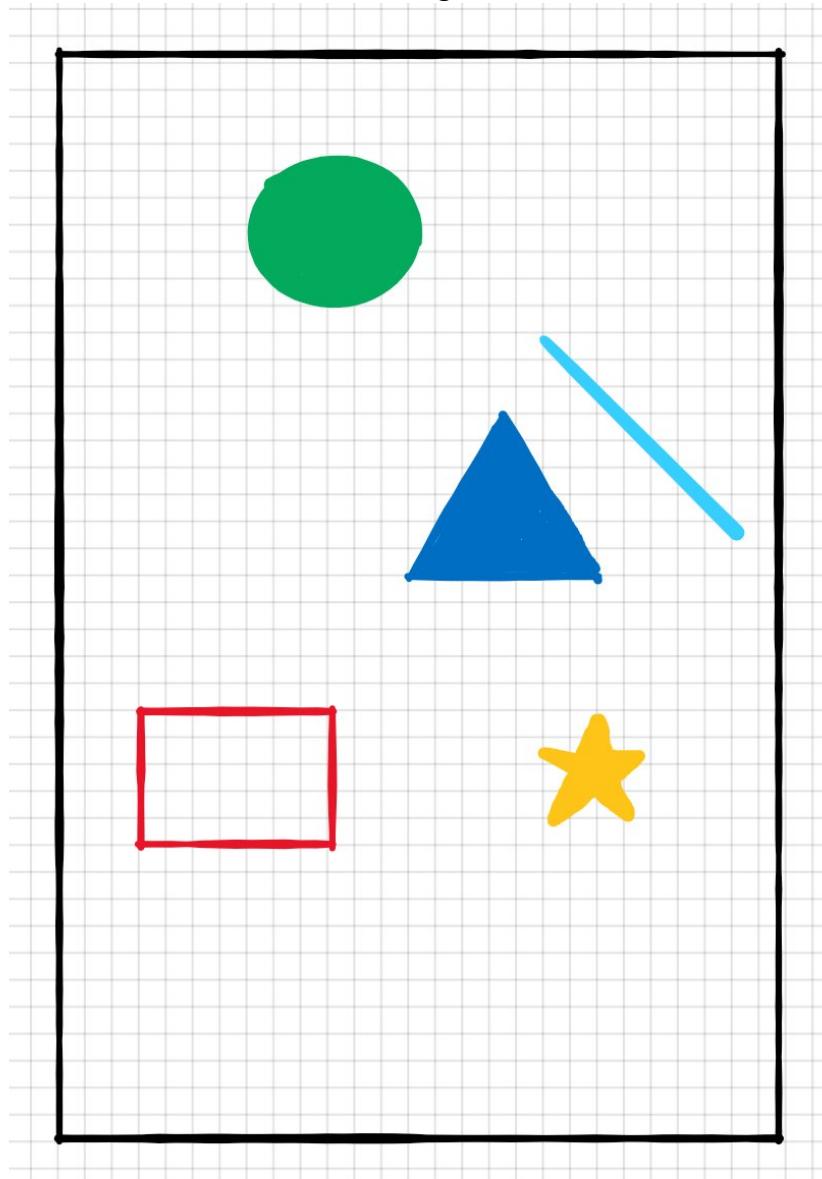
Das aktive Element (= das Element, das bearbeitet werden kann) wird mit einer roten Umrandung hervorgehoben. Neu hinzugefügte Elemente sind automatisch aktiv, aber auch bestehende können ausgewählt und weiter bearbeitet werden.



Der Nutzer kann die Größe und die Farbe des Canvas frei einstellen und jederzeit nach seinen Wünschen anpassen

Beispielbild

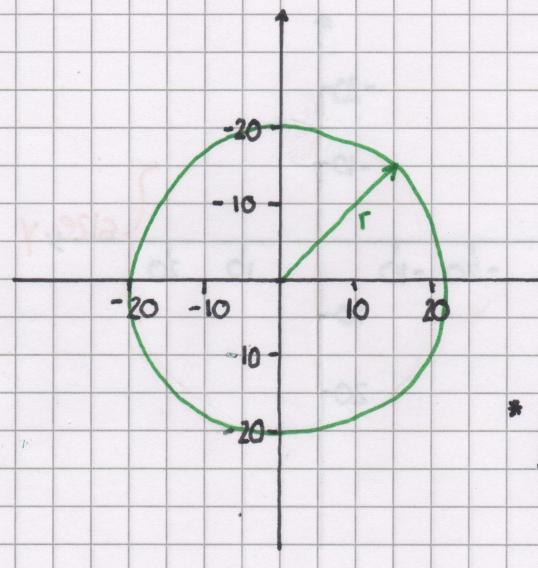
→ Alle Werte wie Position, Größe und Rotation der Figuren können genauso wie die Größe und Farbe des Canvas vom Nutzer frei gewählt werden



Ein kleines Tab-Icon welches das Thema „Zauberbild“ noch einmal aufgreift :)



Kreis

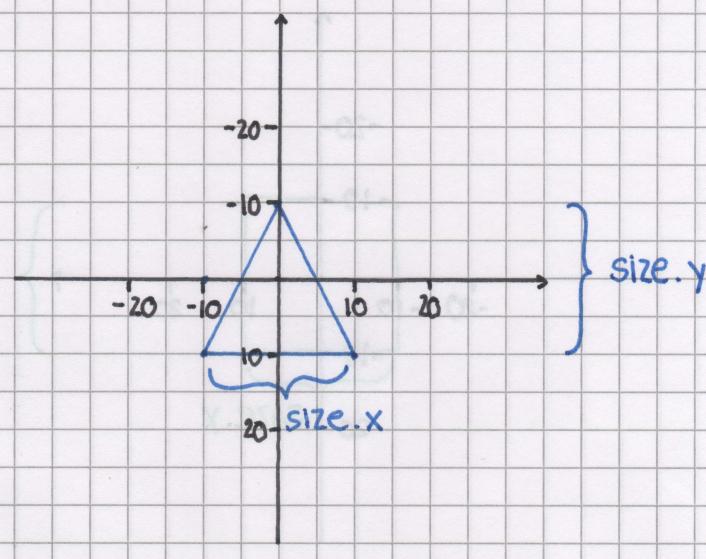


$$r = \text{this.size.x}$$

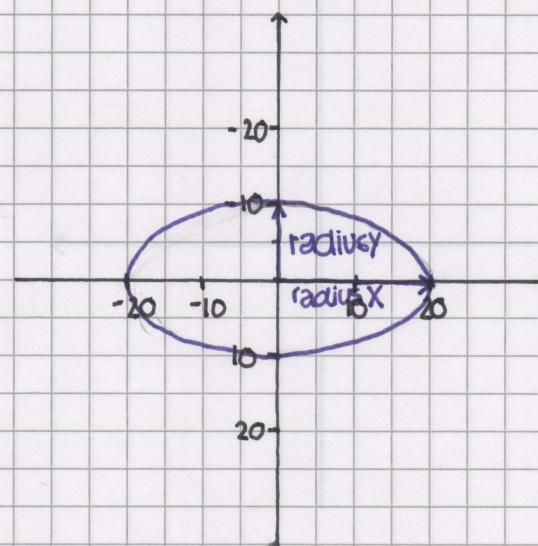
(Beispiel für
this.size.x = 20)

* Anmerkung der
Autorin: der Kreis
soll rund sein ü *

Dreieck

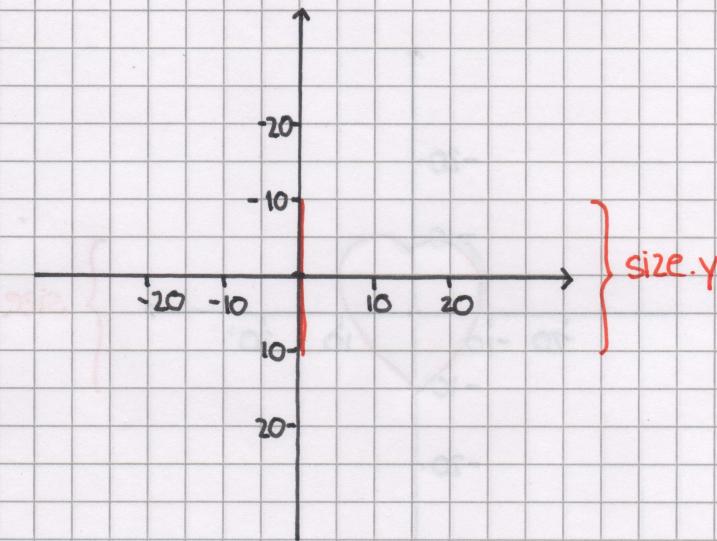


Ellipse

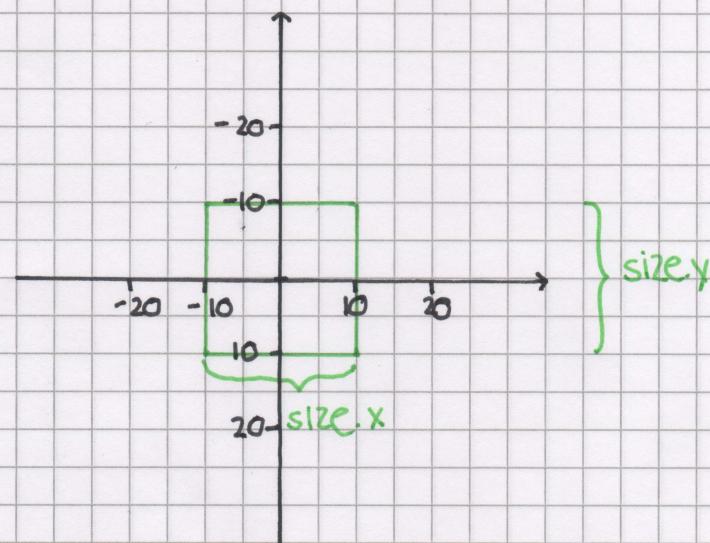


$$\text{radiusX} = \text{size.x}$$
$$\text{radiusY} = \text{size.y}$$

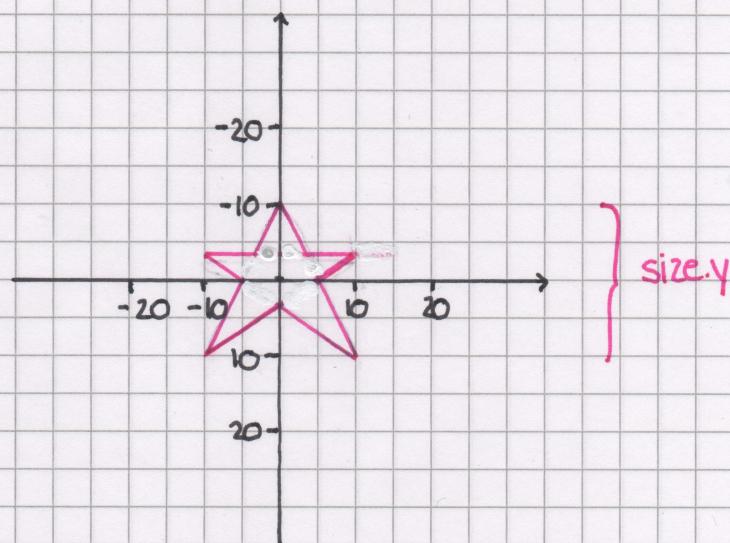
Line



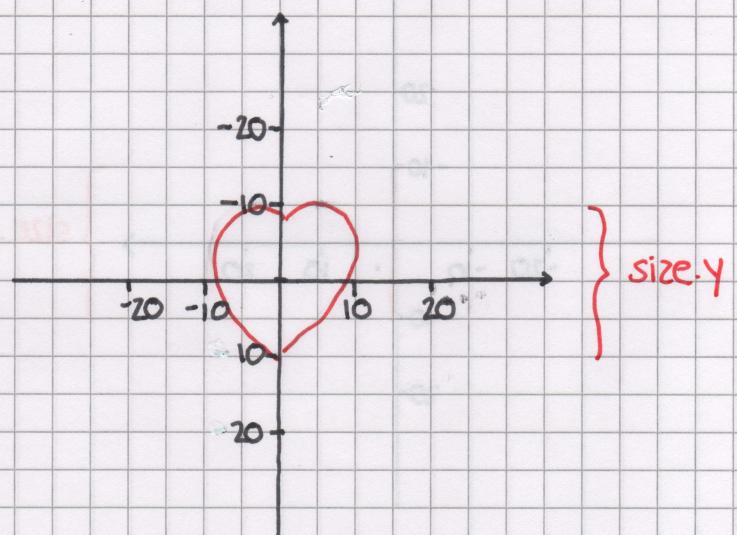
square

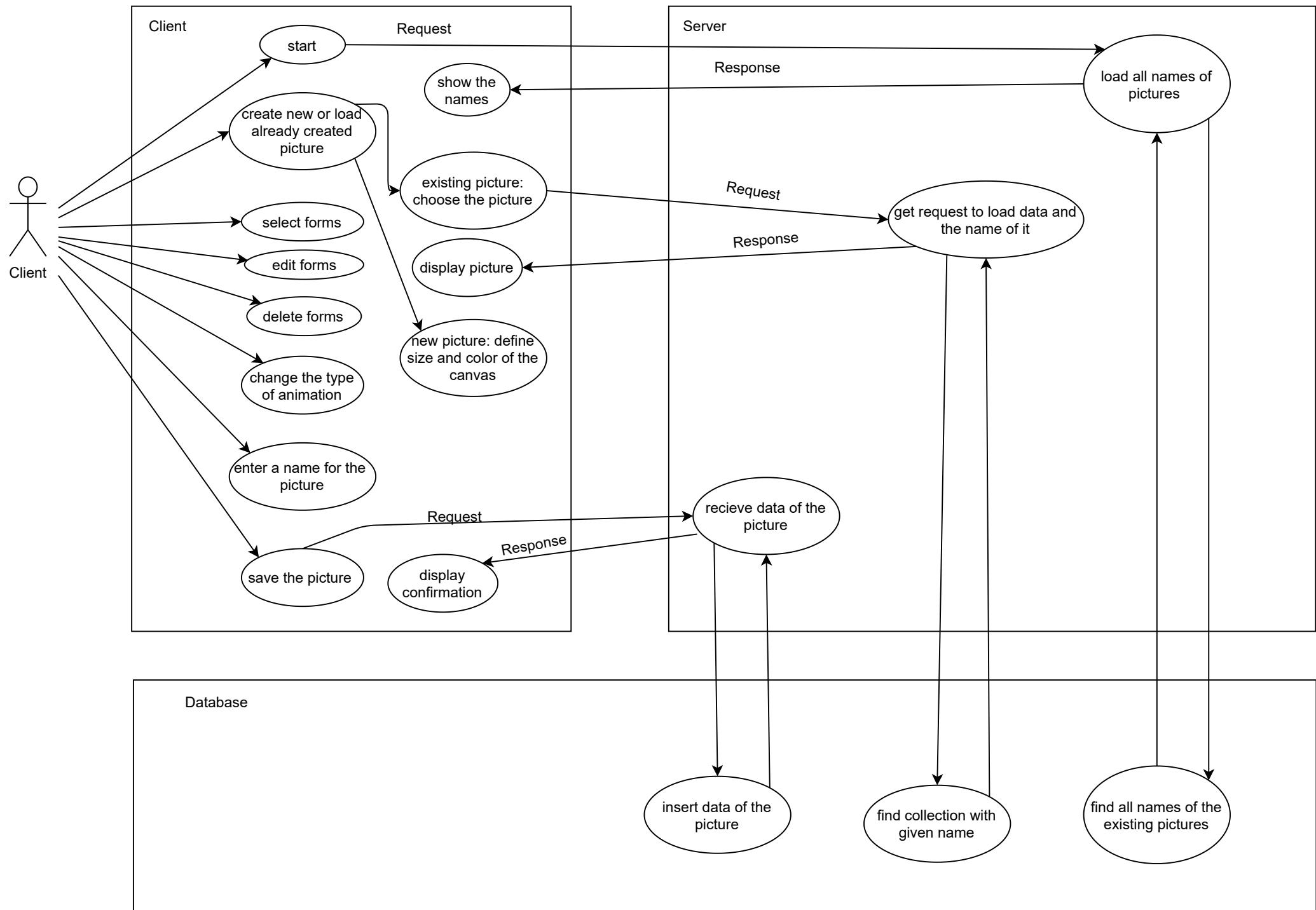


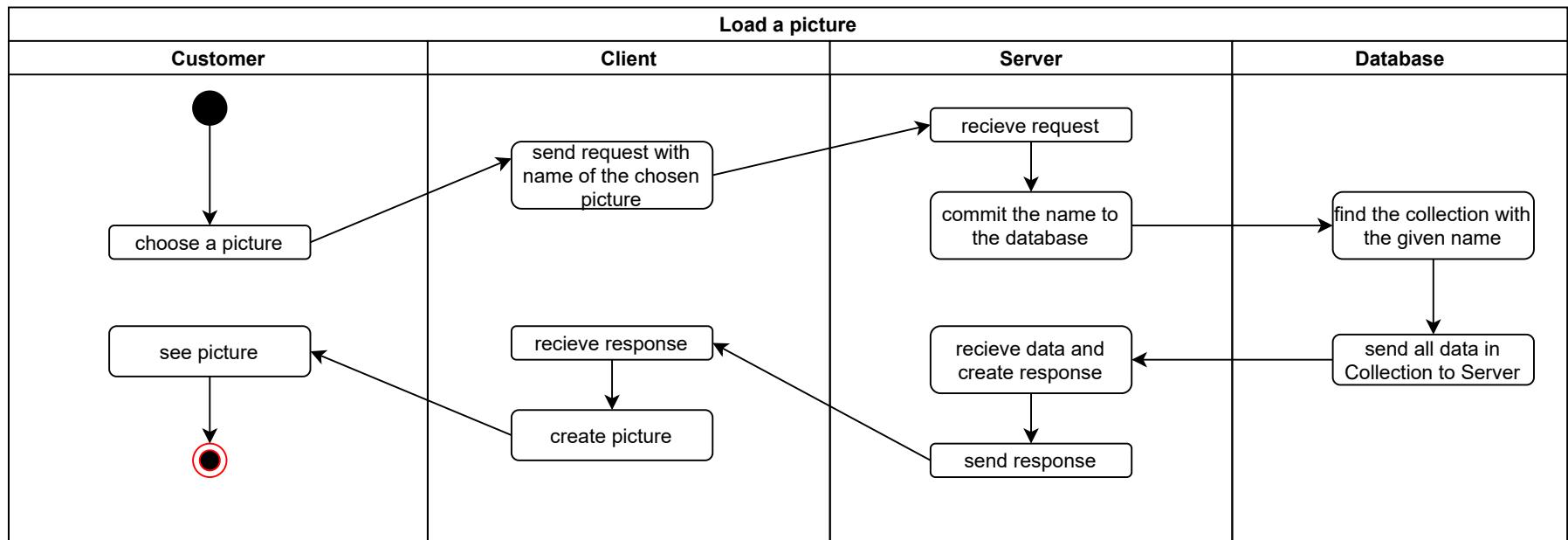
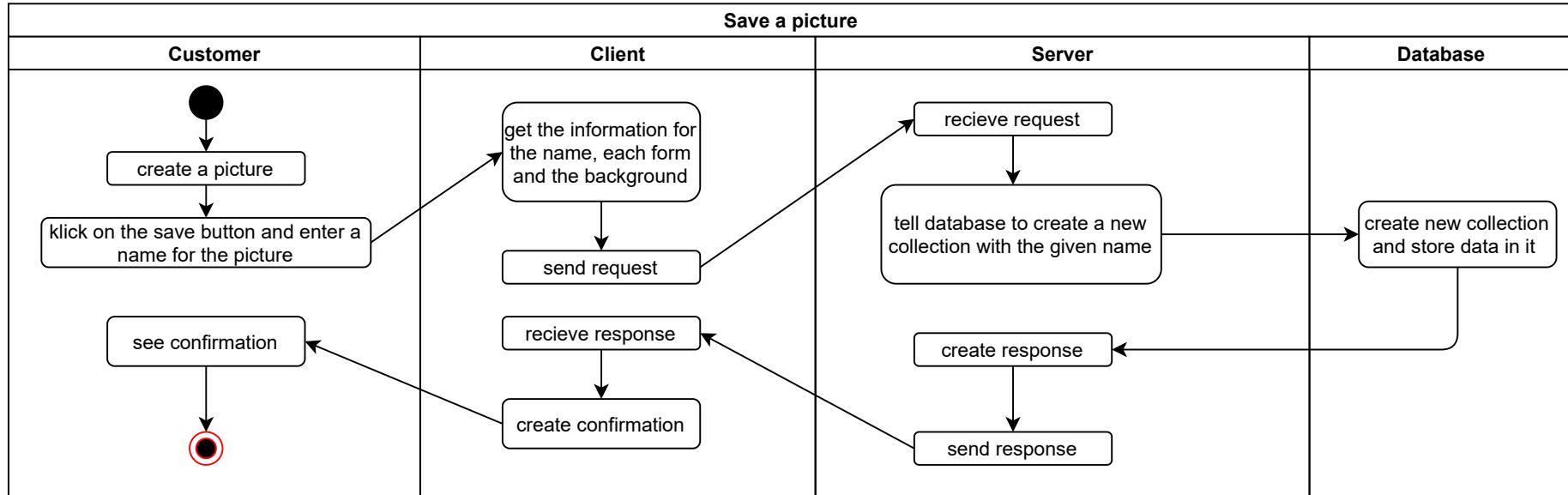
star

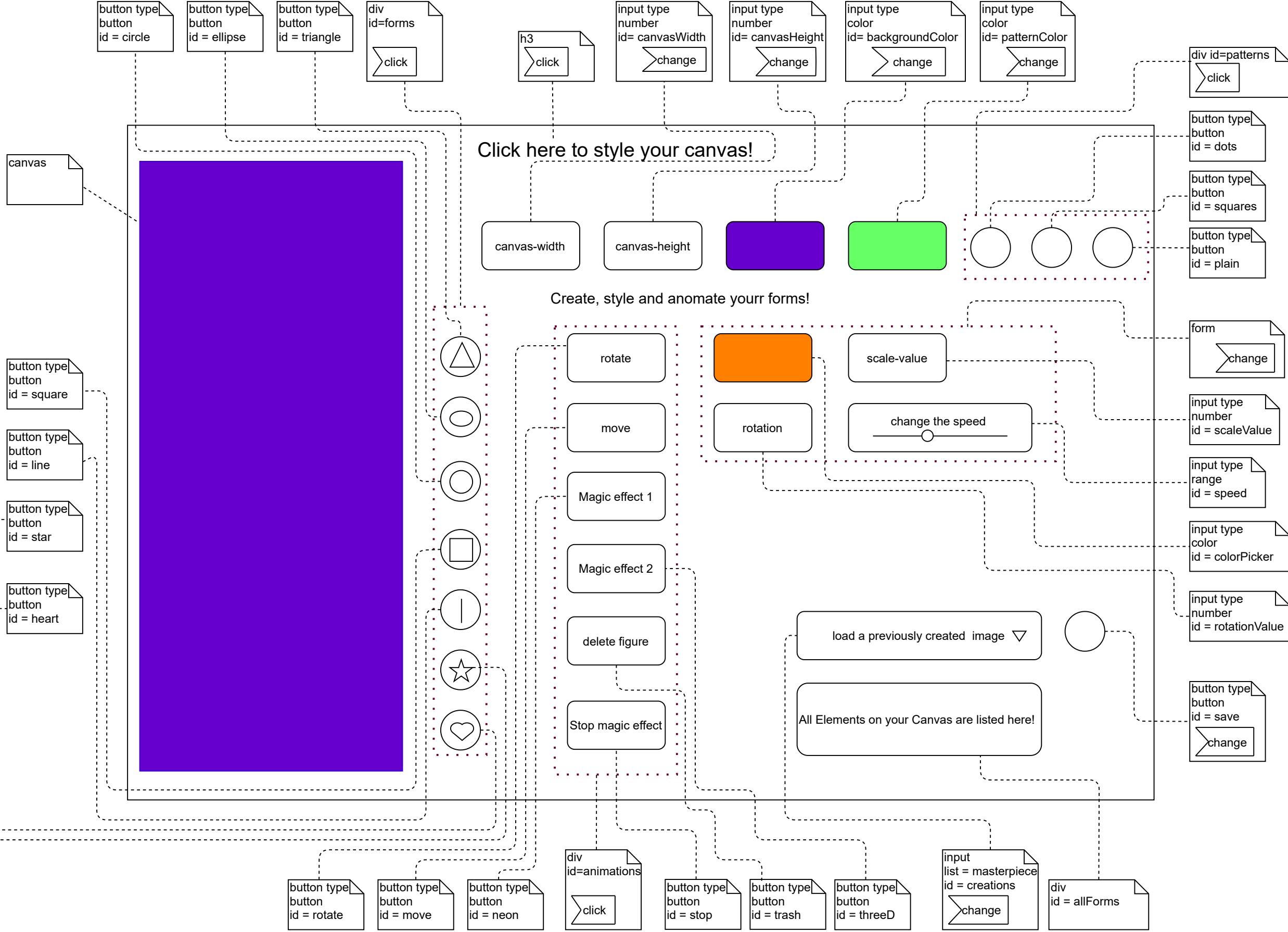


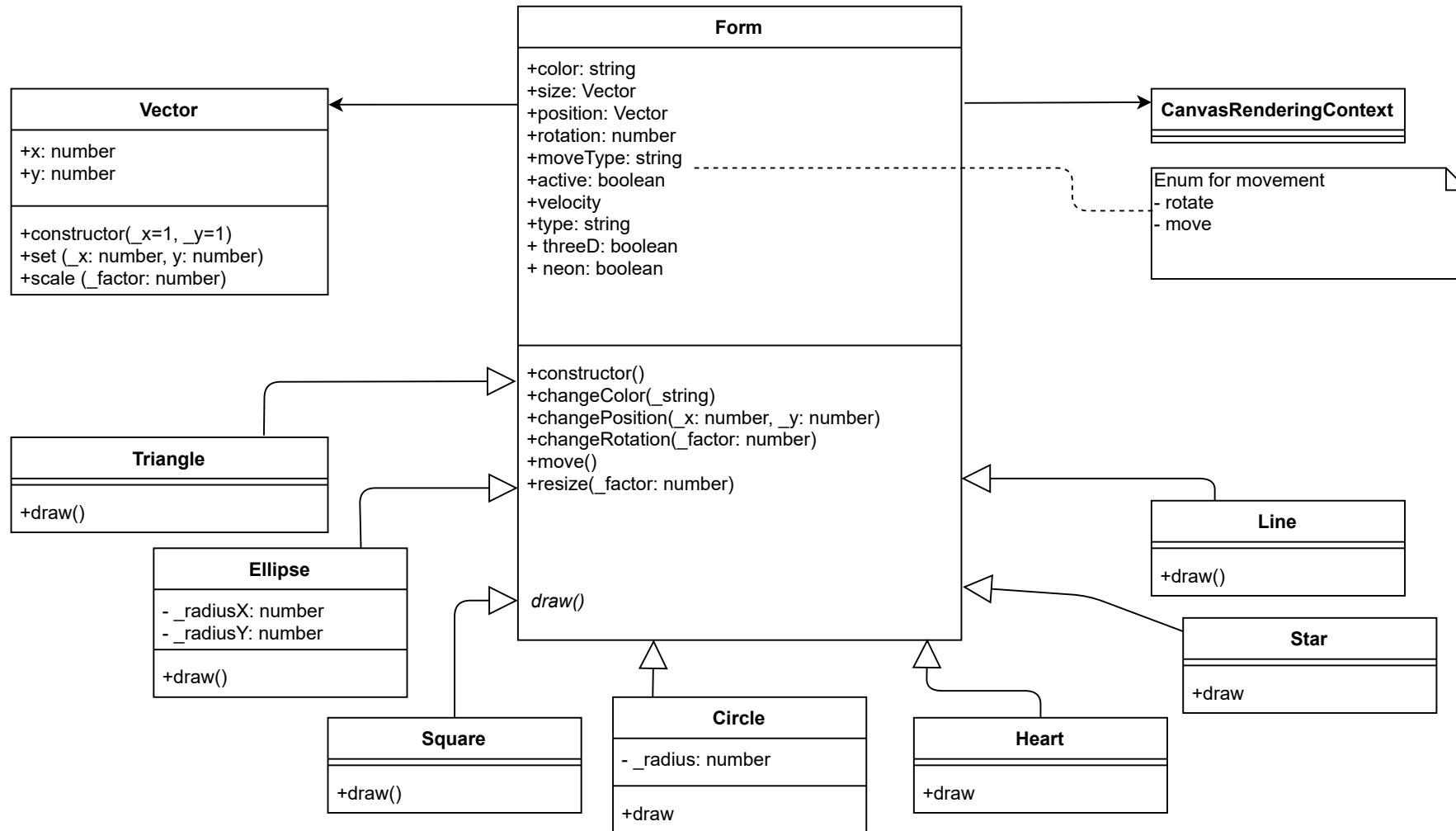
Heart



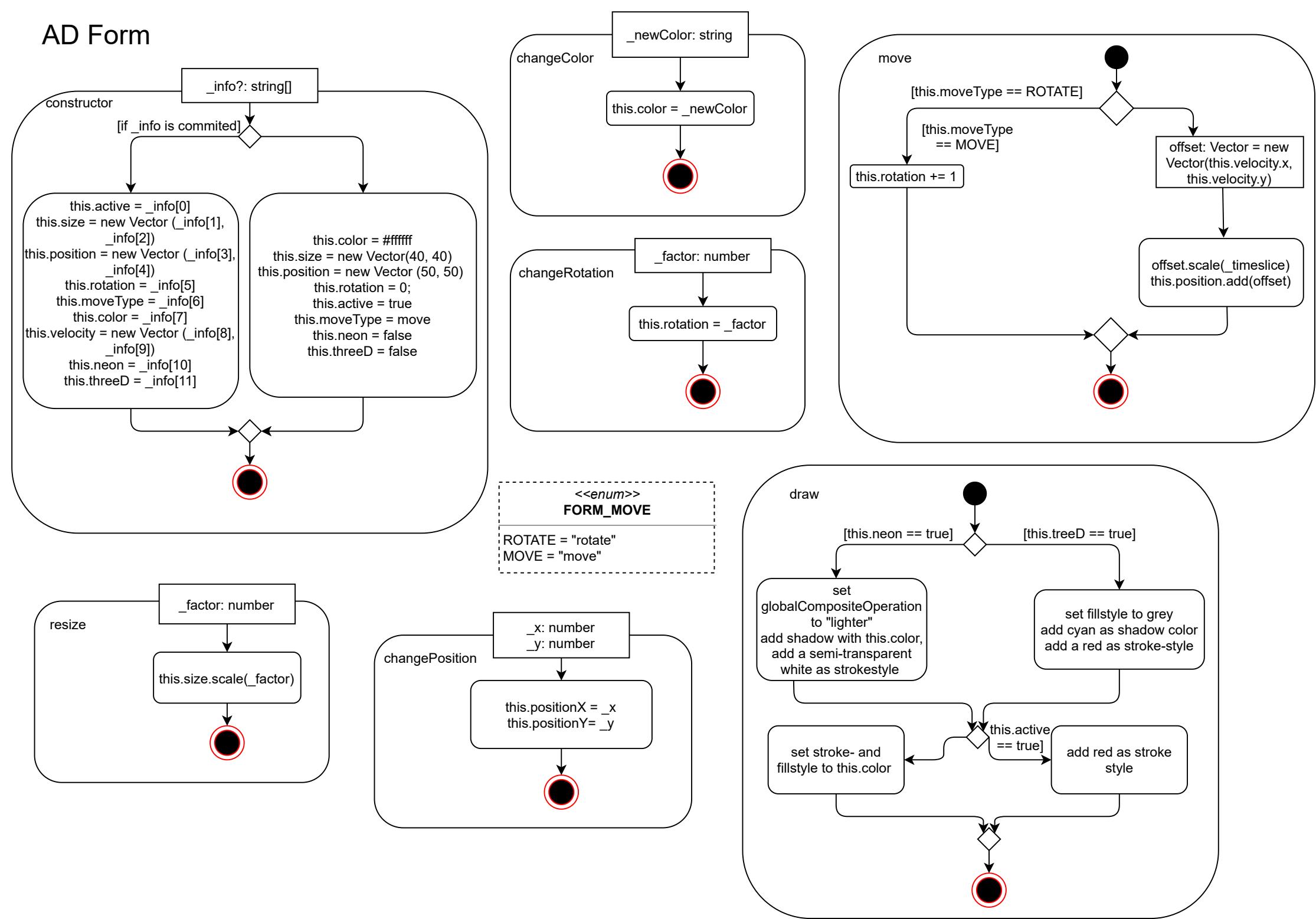




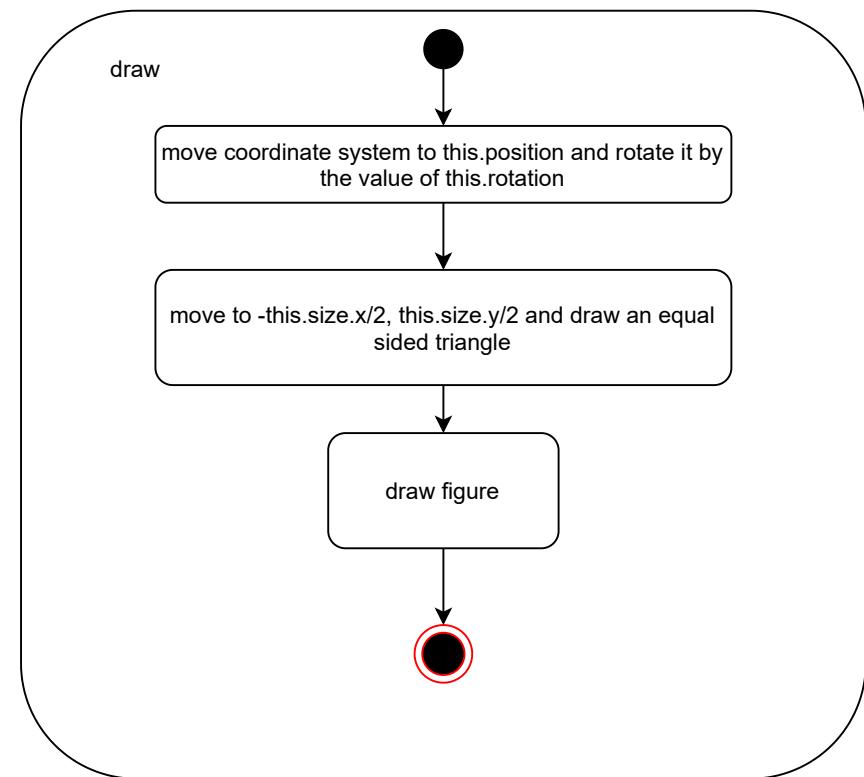
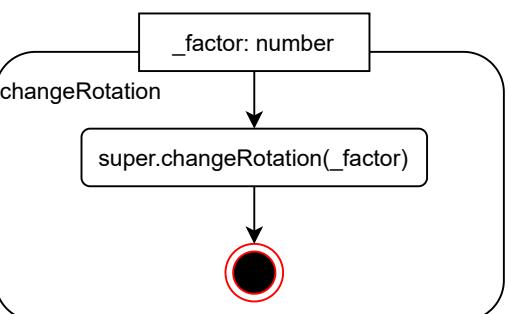
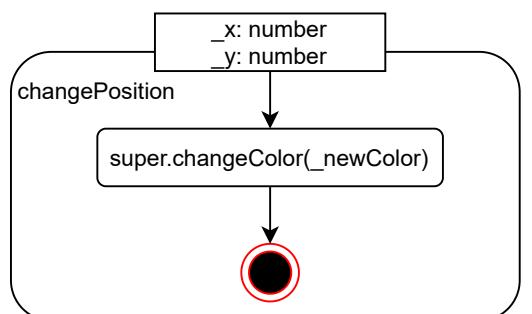
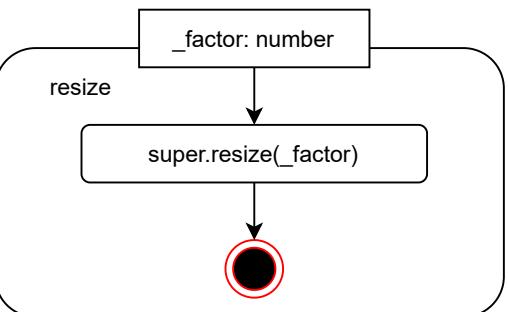
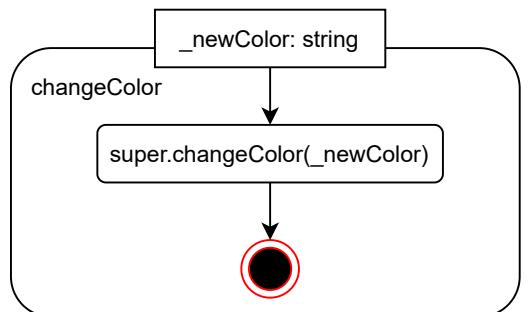
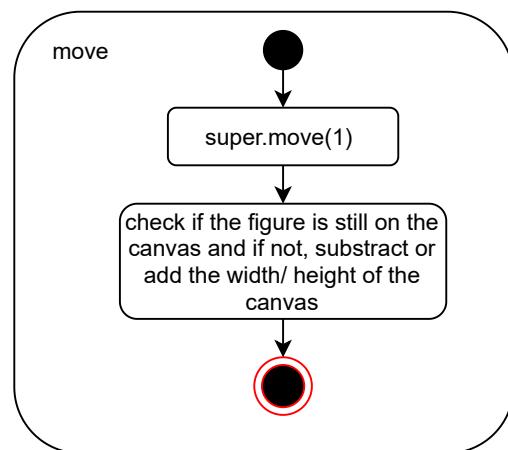
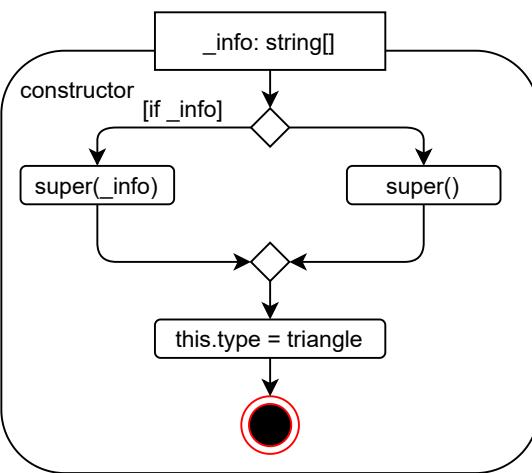




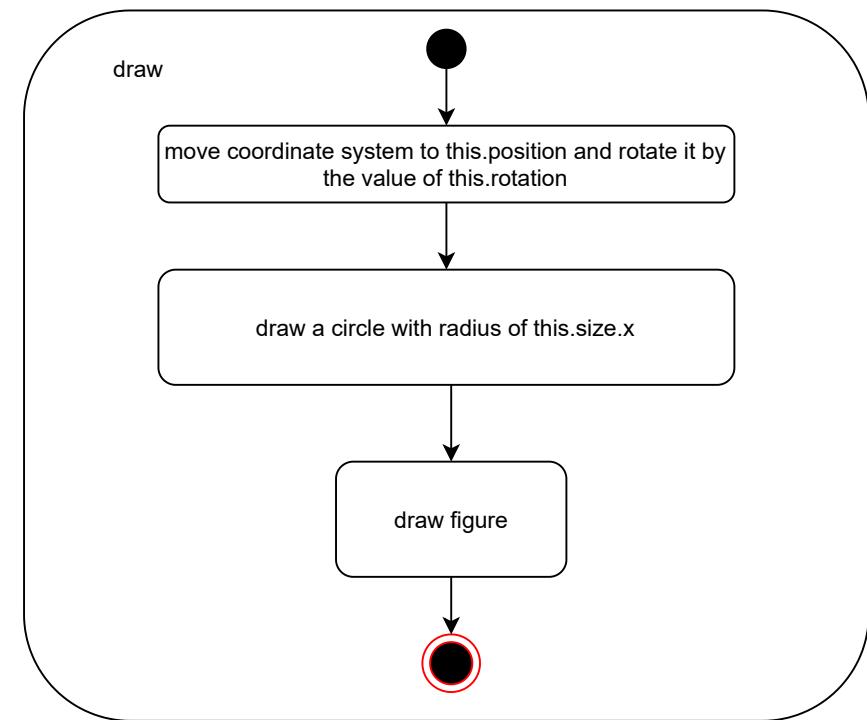
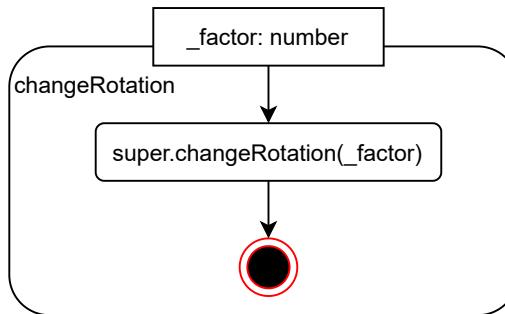
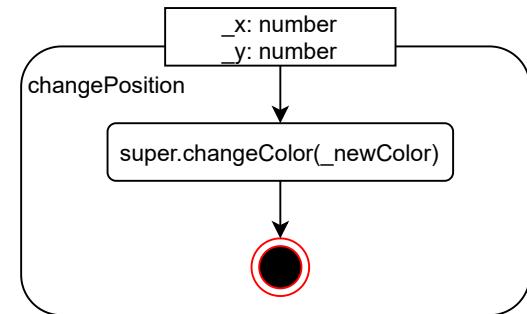
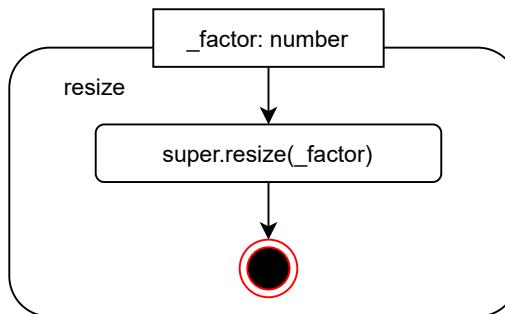
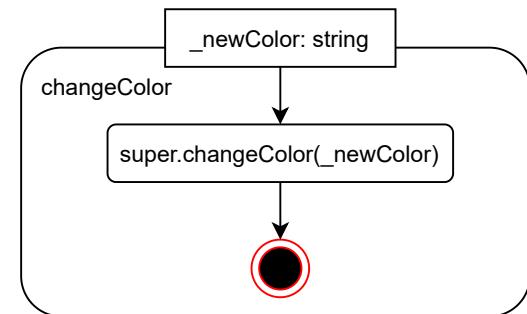
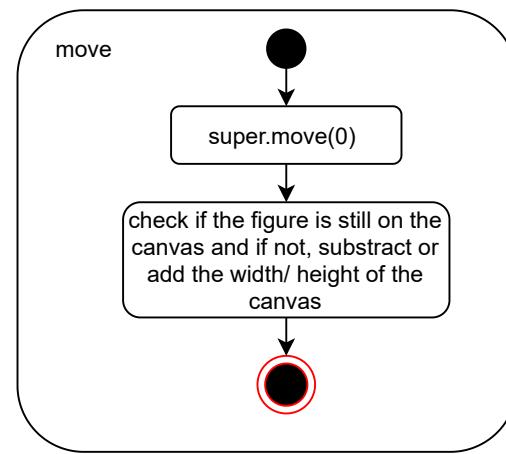
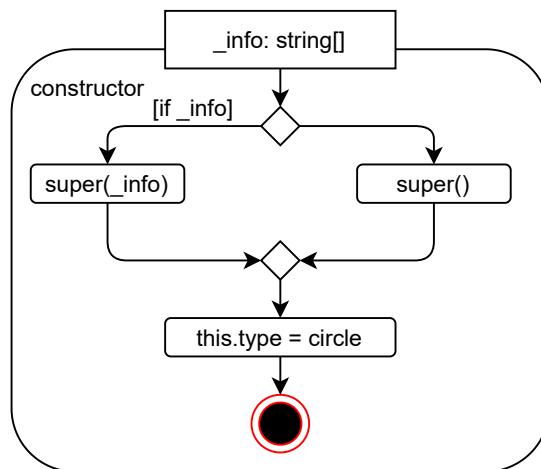
AD Form



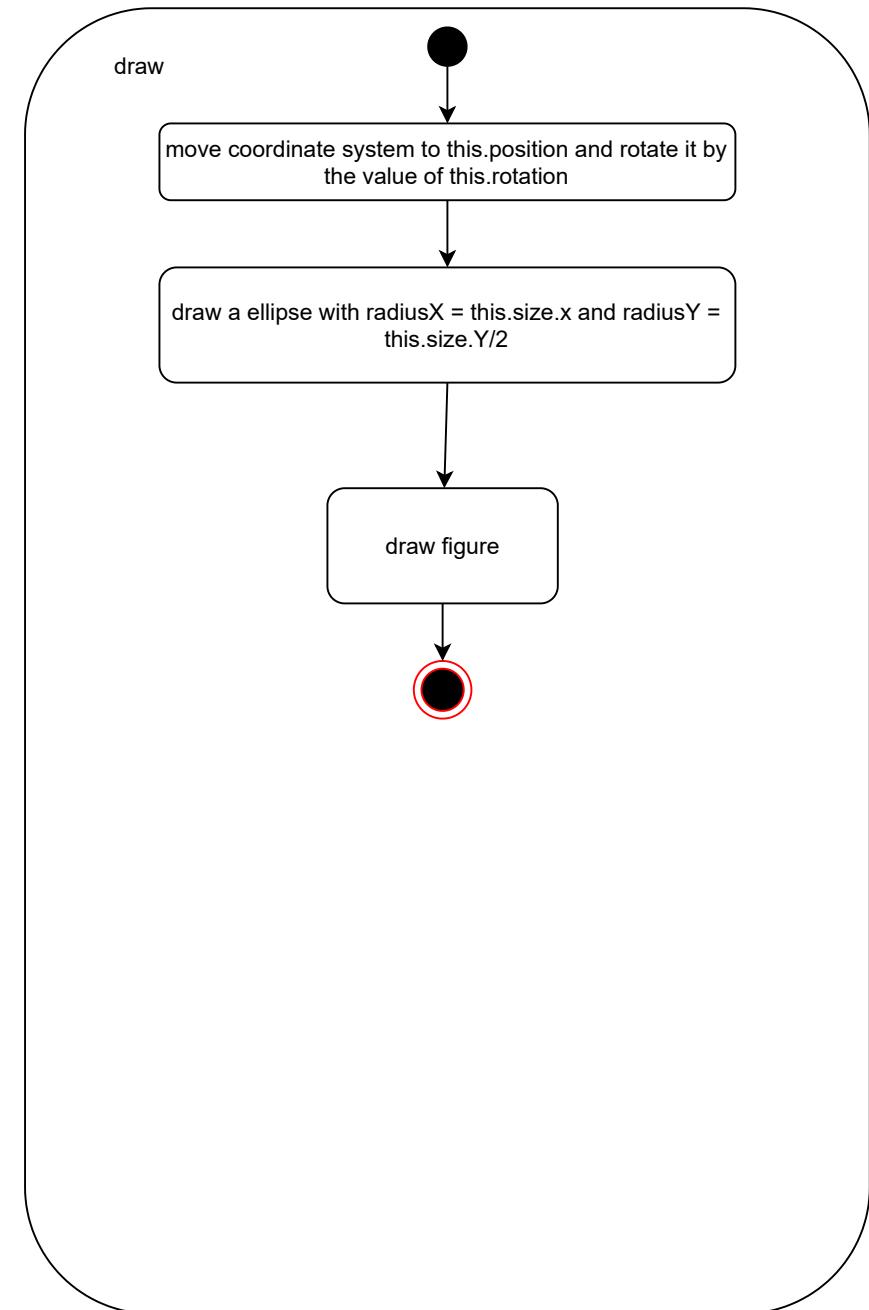
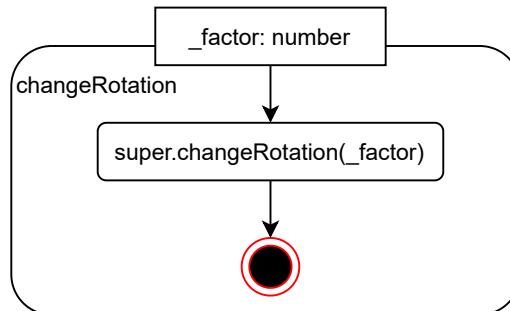
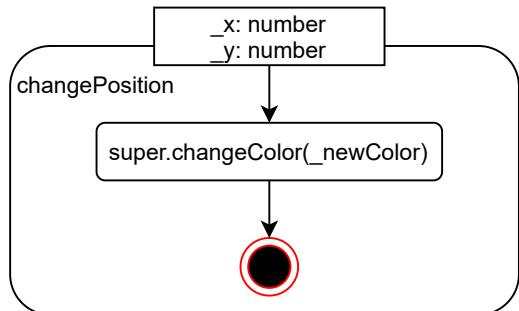
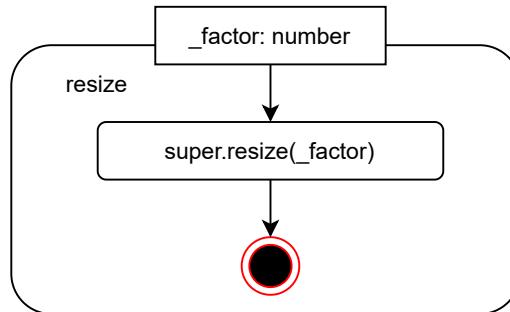
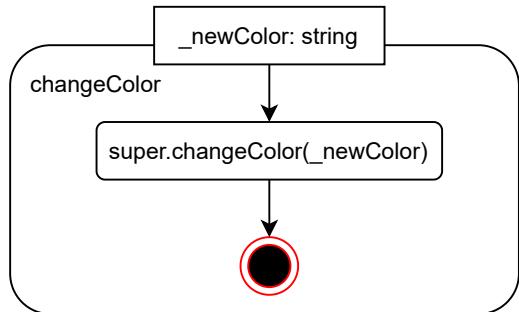
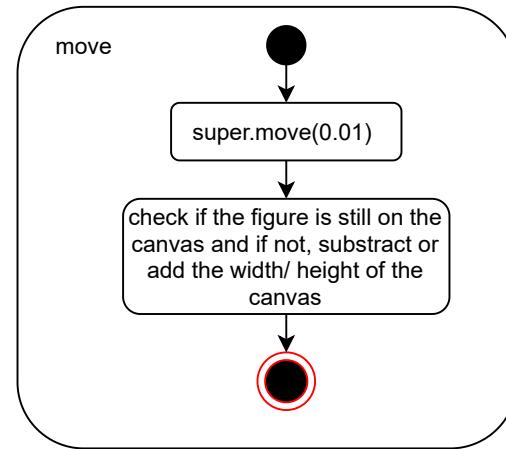
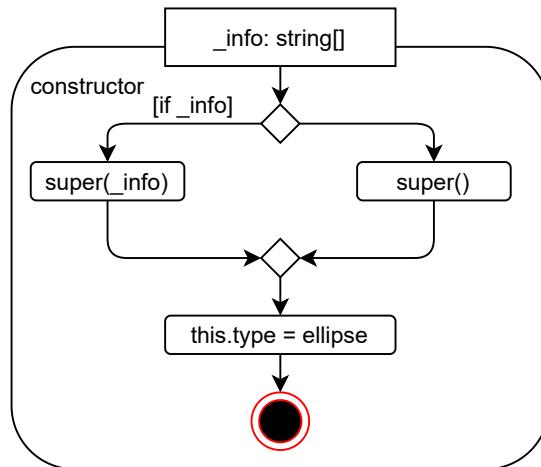
AD Triangle



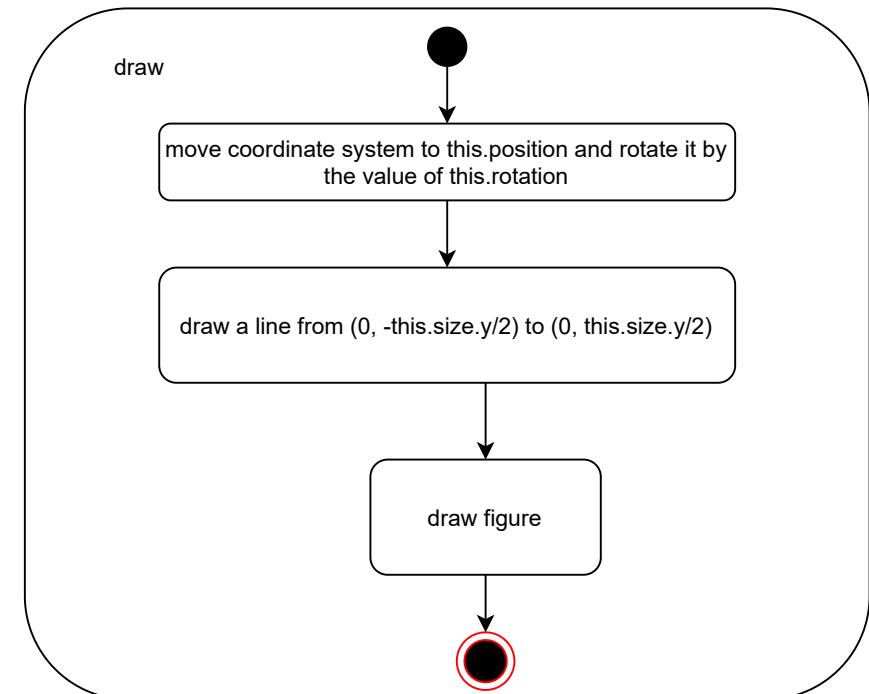
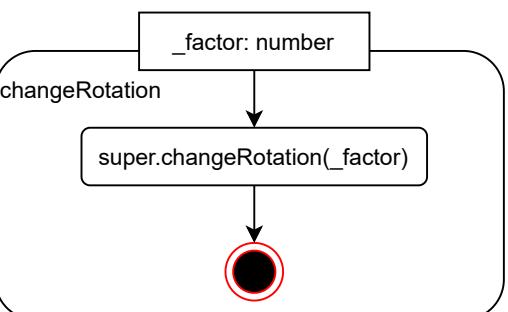
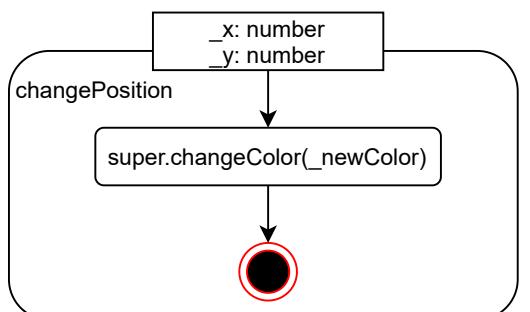
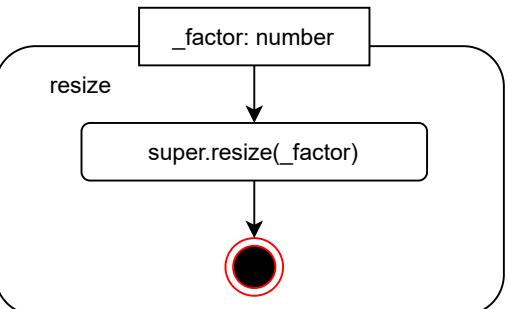
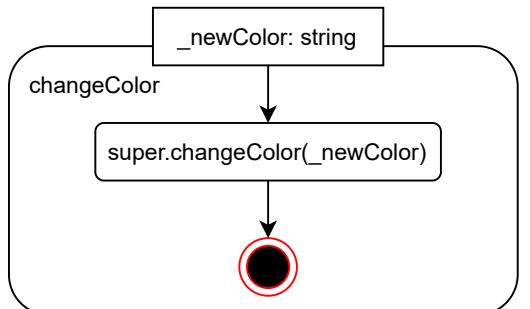
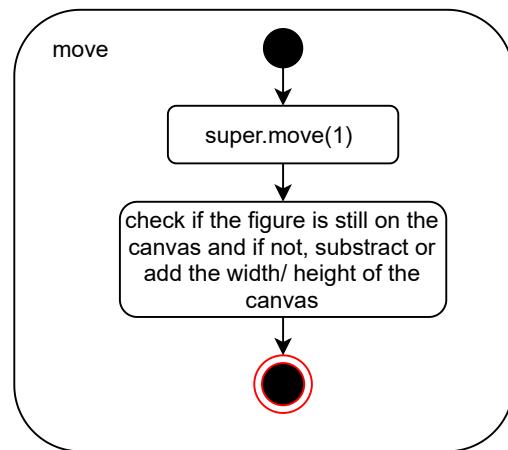
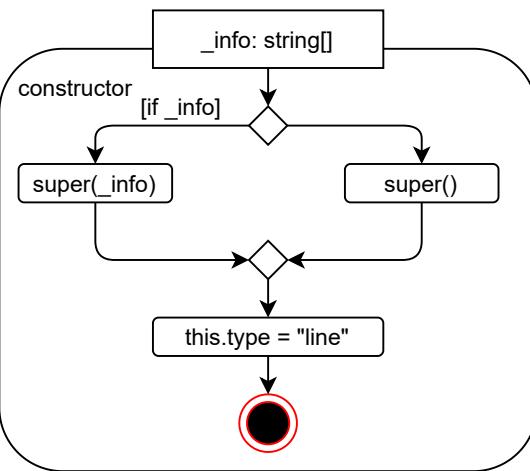
AD Circle



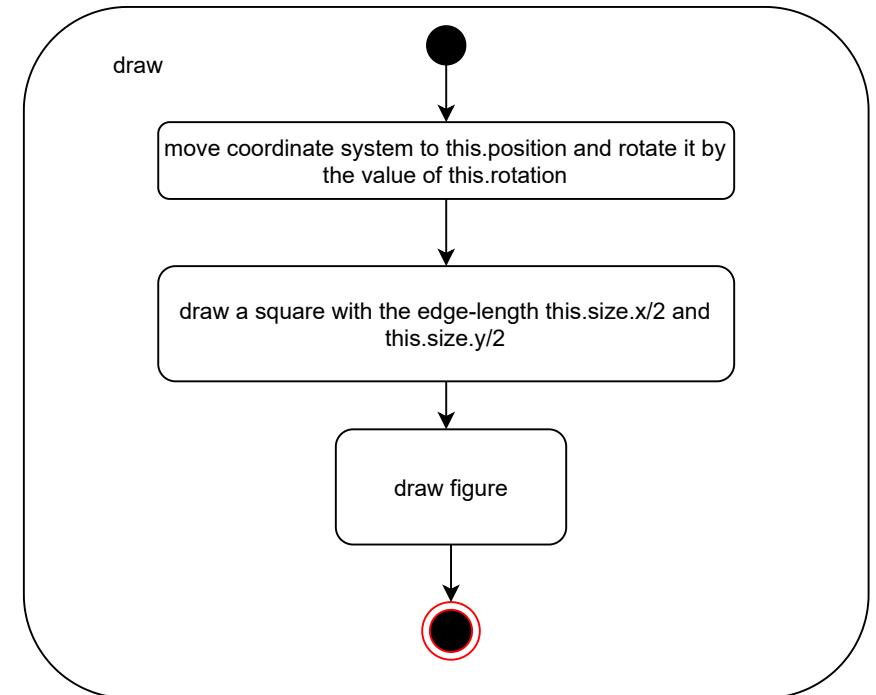
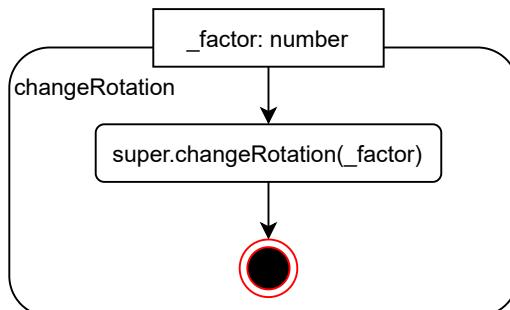
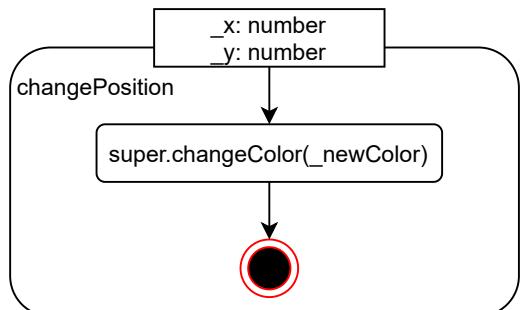
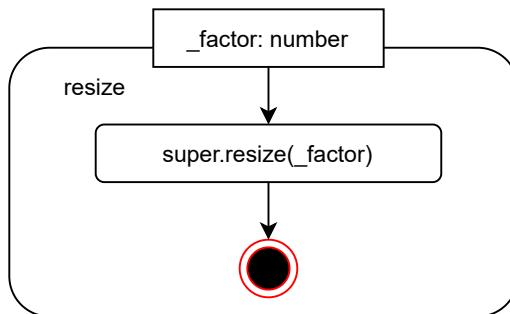
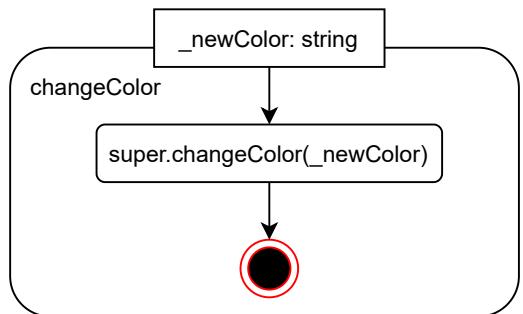
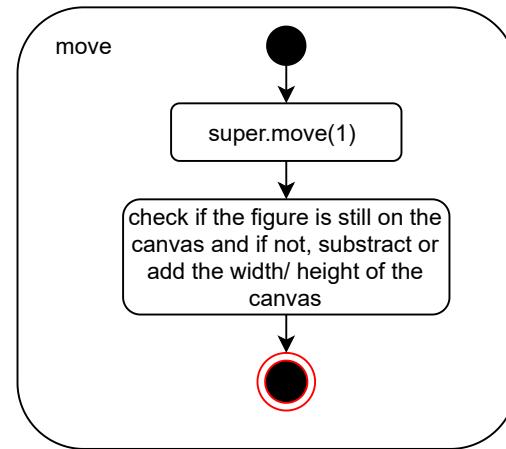
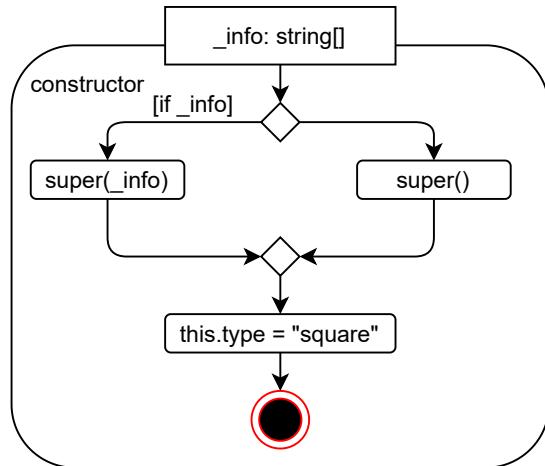
AD Ellipse



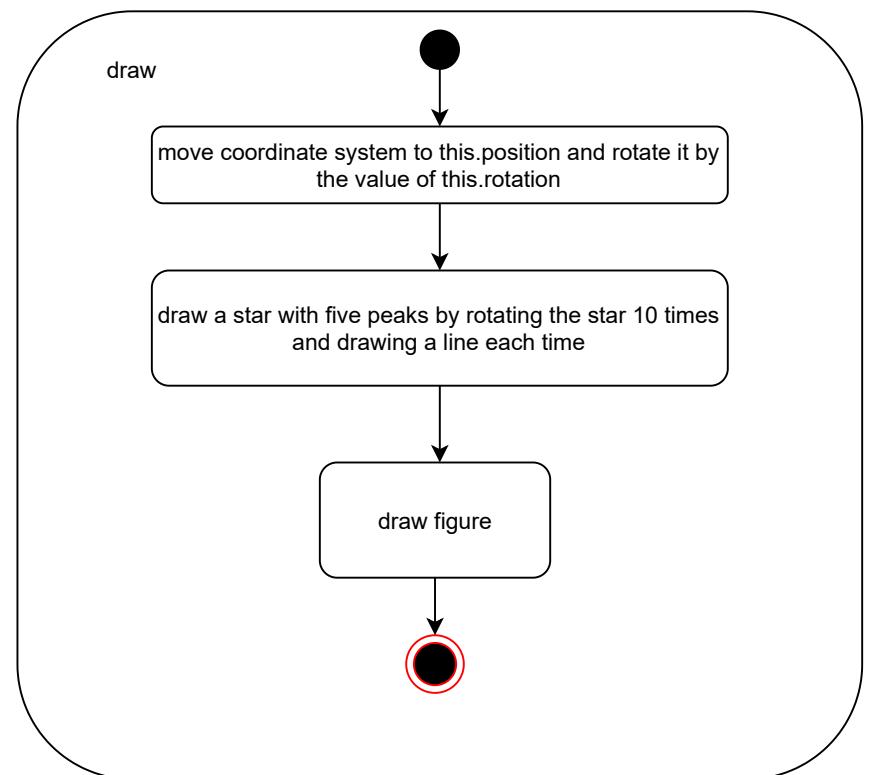
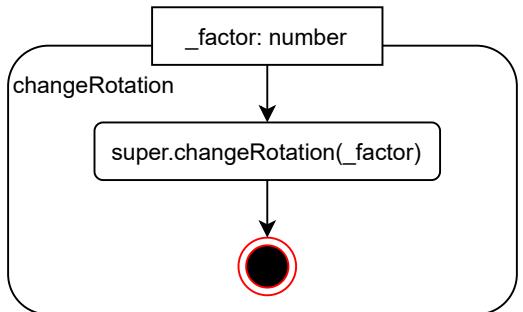
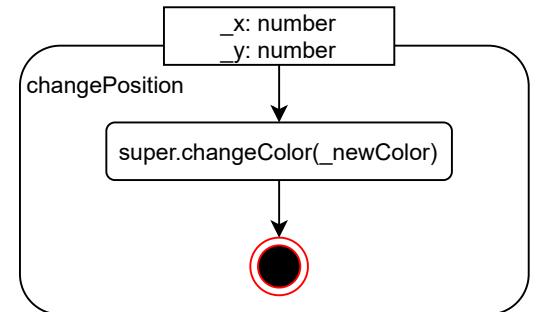
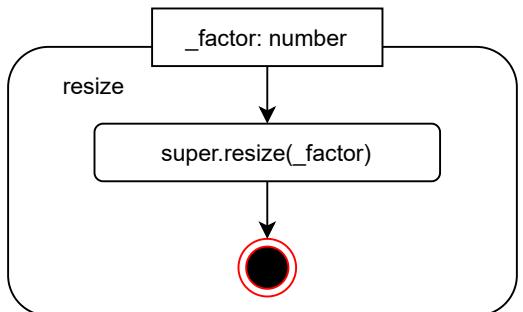
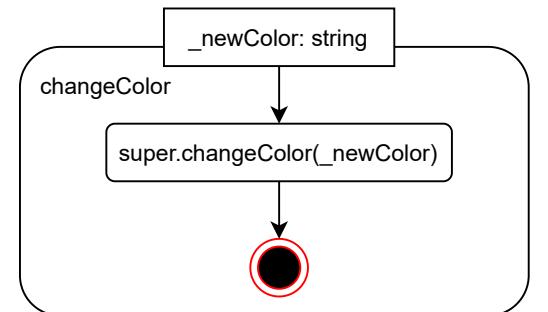
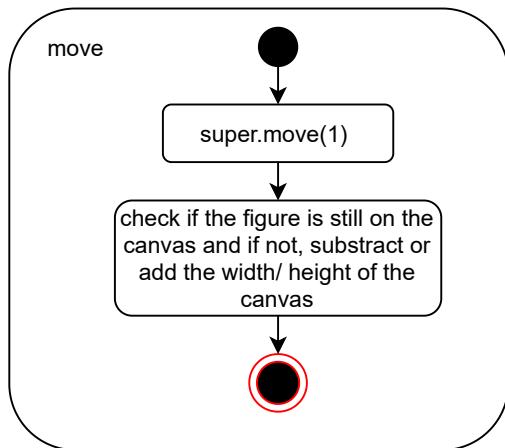
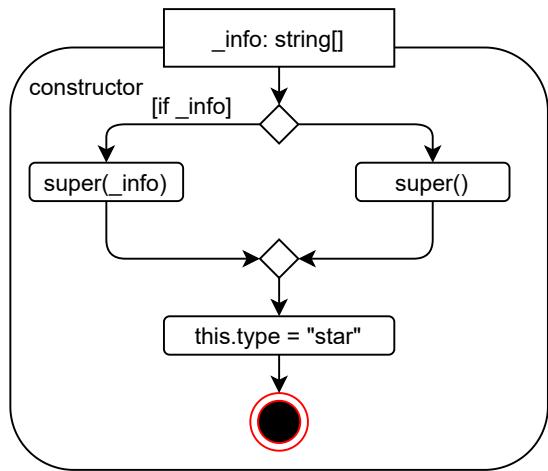
AD Line



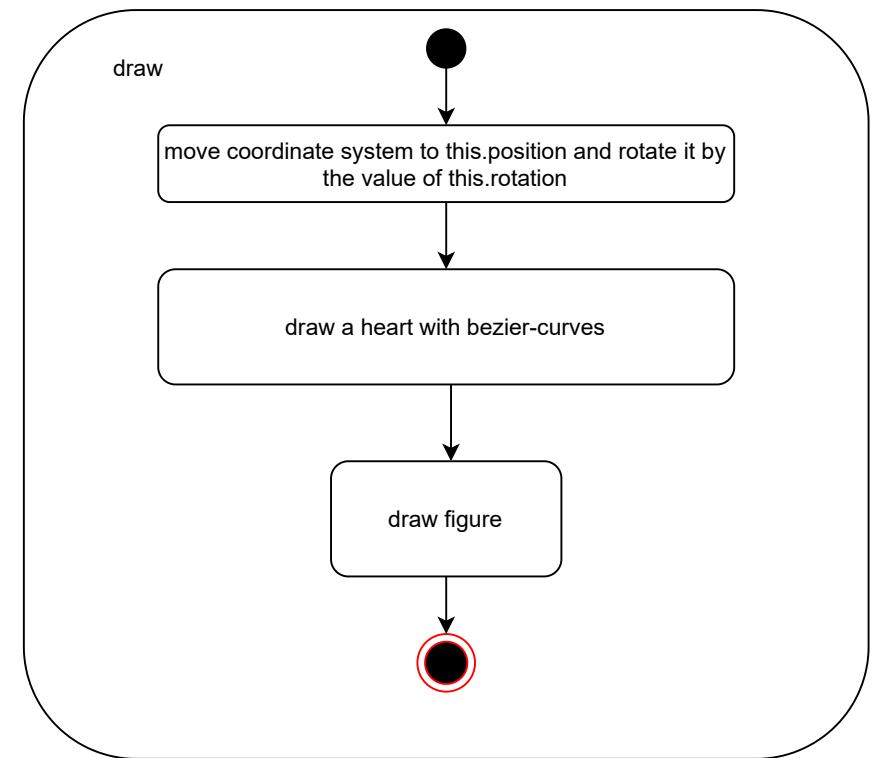
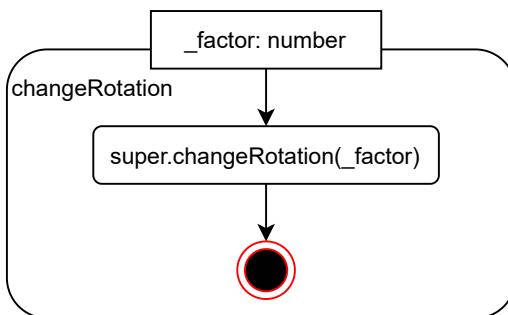
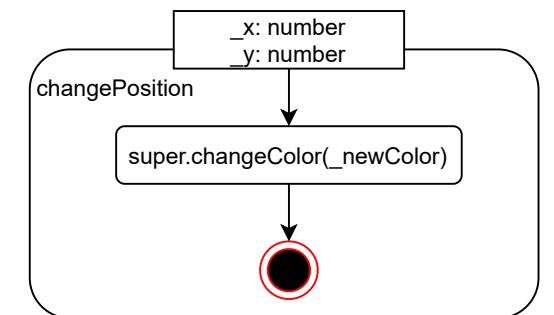
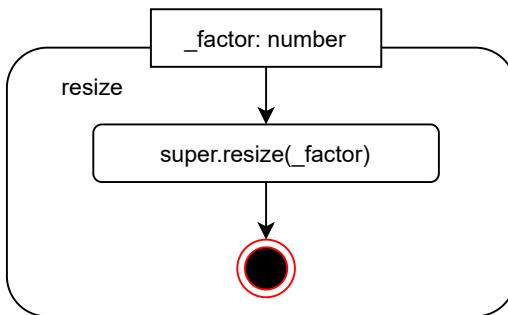
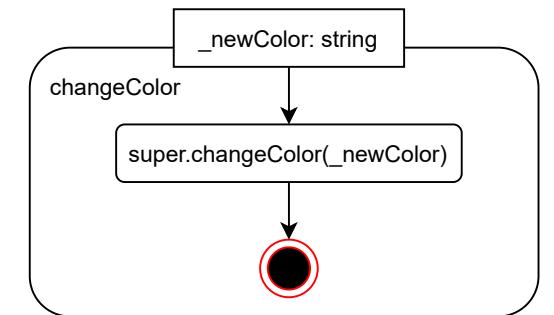
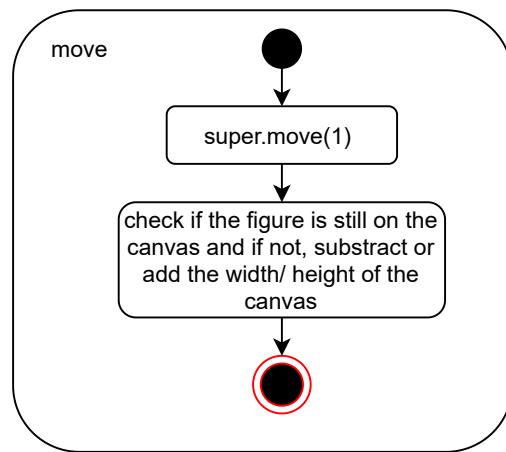
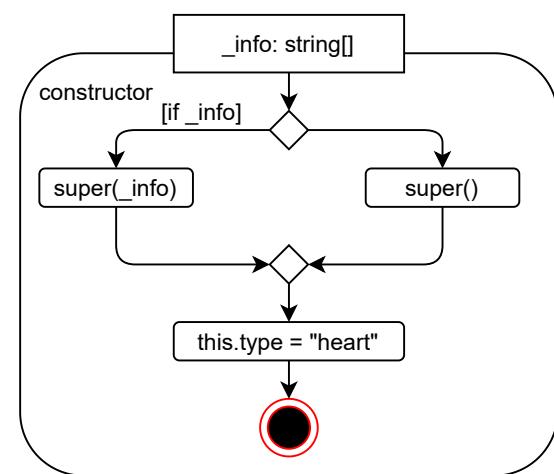
AD Square



AD Star



AD Heart



AD Main

```
let canvas: HTMLCanvasElement
let crc2: CanvasRenderingContext2D
figures: Form[]
backgroundColor: HTMLInputElement
background: string
creations: HTMLInputElement
backgroundImage: ImageData
backgroundPattern: string = "plain"
canvasWidth: HTMLInputElement
canvasHeight: HTMLInputElement
backgroundColorWrapper: HTMLElement
patterns: HTMLDivElement
patternColor: HTMLInputElement
patternColorWrapper: HTMLDivElement
forms: HTMLDivElement
animations: HTMLDivElement
form: HTMLFormElement
h3: HTMLHeadingElement
save: HTMLButtonElement
allForms: HTMLDivElement
```

```
add load-listener to window
```

```
> click on div id= animations
```

```
setAnimation
```

```
> click on button id= save
```

```
getName
```

```
> click on div id= patterns
```

```
createPattern
```

```
> change in form
```

```
handleFormInput
```

```
> click on canvas
```

```
handleClick
```

```
> change in input id= backgroundColor
```

```
createBackground
```

```
> change in input id= canvasHeight
```

```
setCanvasHeight
```

```
> change in input id= canvasWidth
```

```
setCanvasWidth
```

```
> click on div id= forms
```

```
createElement
```

```
> change in input id= patternColor
```

```
createBackground
```

```
> click on h3
```

```
toggleCanvasPrpoerty
```

```
> change on input id=creations
```

```
loadPicture
```

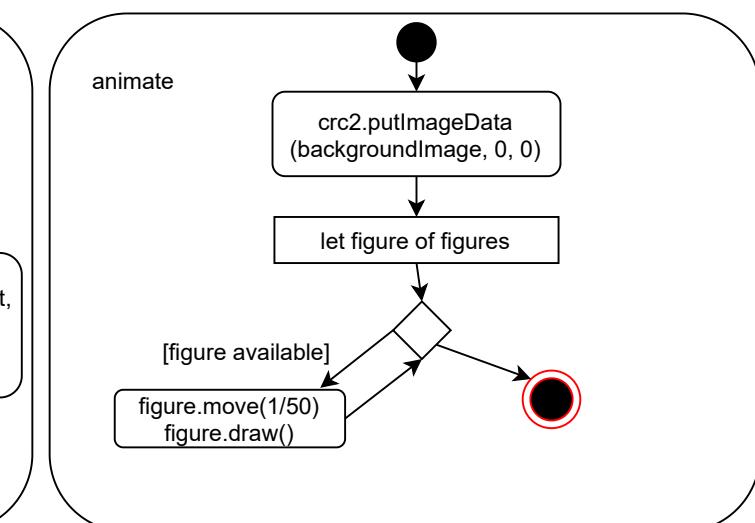
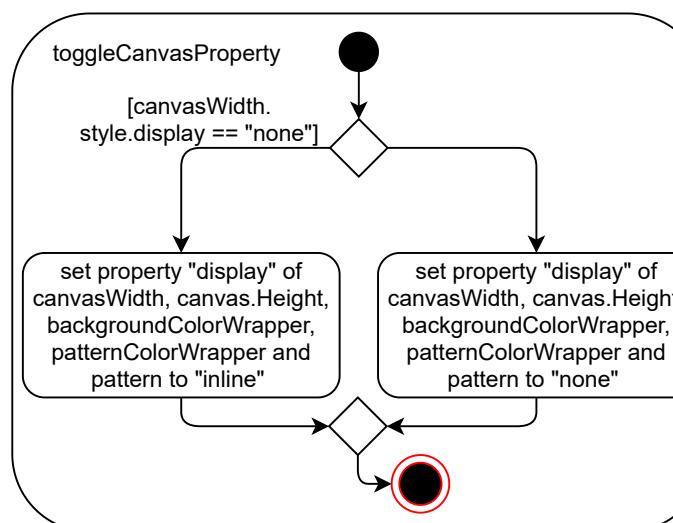
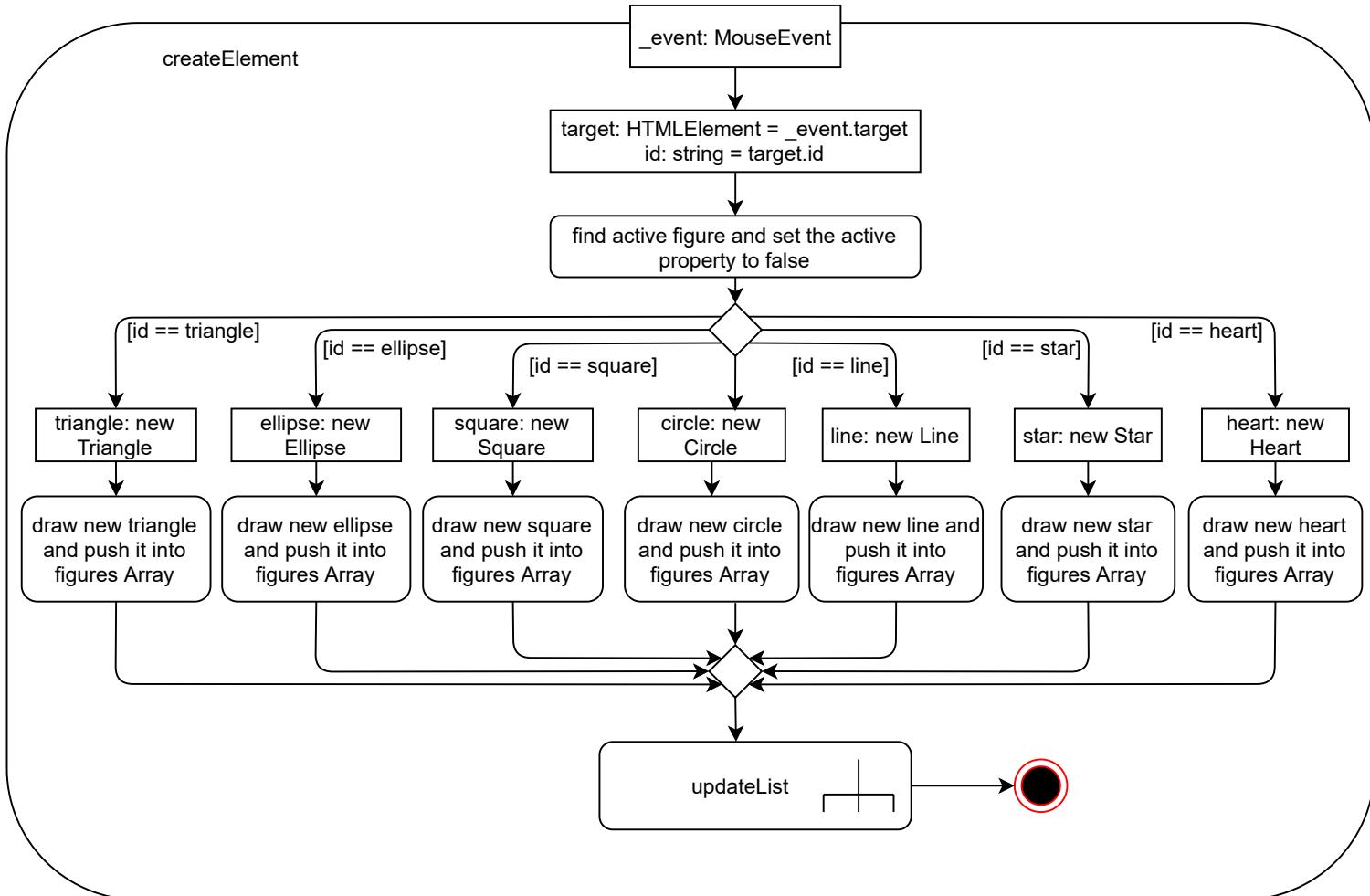
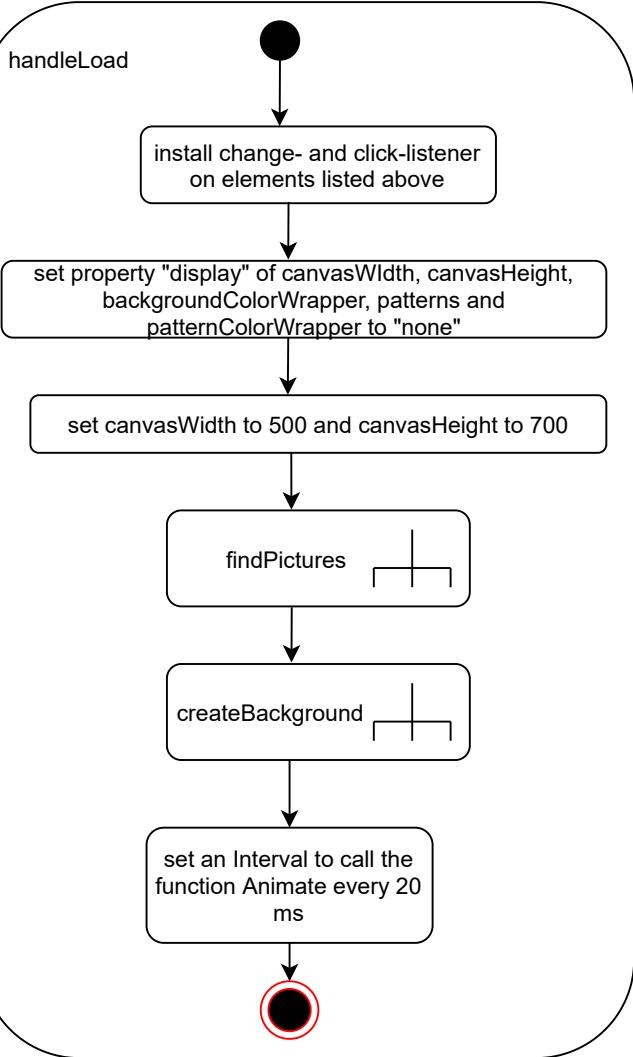
```
animate
```



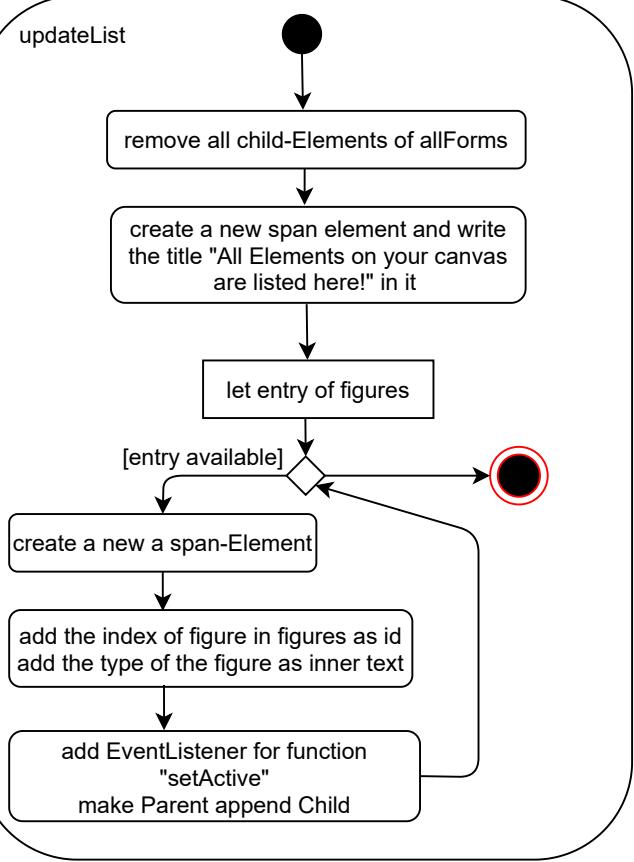
```
every  
20ms
```

```
click on child of all Forms
```

```
setActive
```



updateList



setCanvasWidth

get value from input-field with id = canvasWidth

canvas.width = value from input-field

createBackground



setCanvasHeight

get value from input-field with id = canvasHeight

canvas.height = value from input-field

createBackground



createPattern

target = _event.target
id = target.id

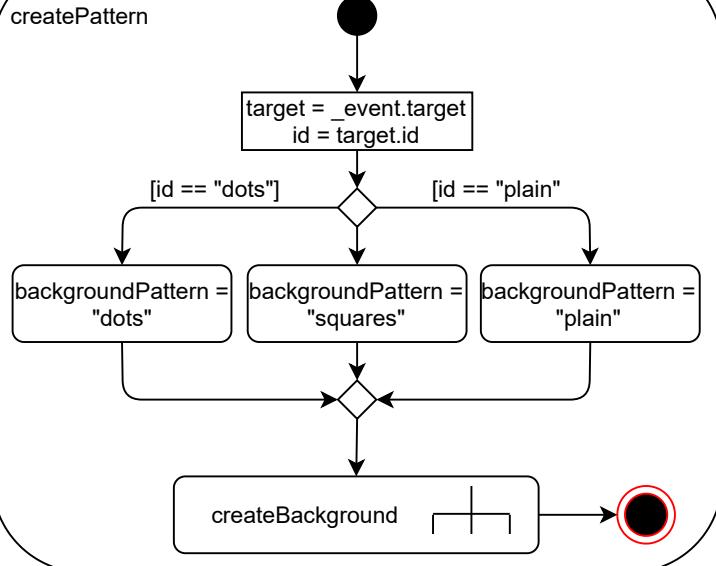
[id == "dots"] [id == "plain"]

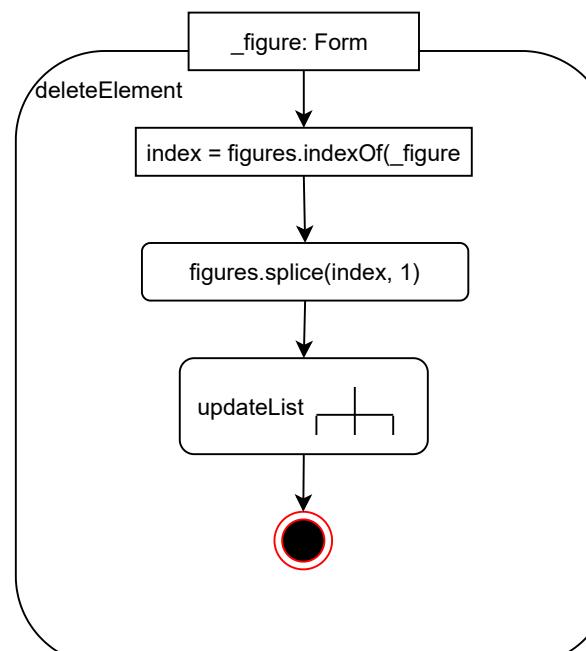
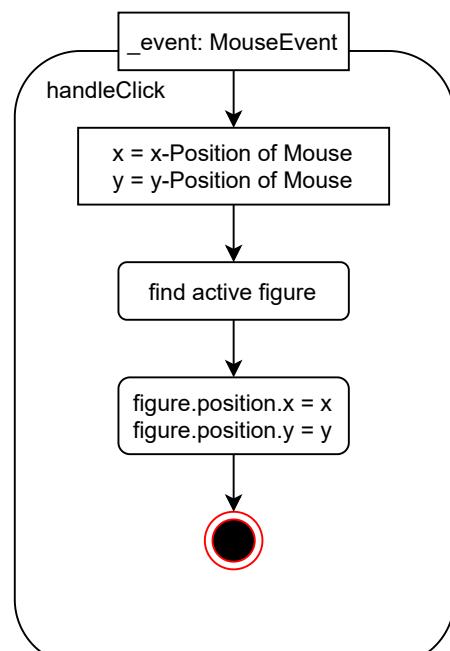
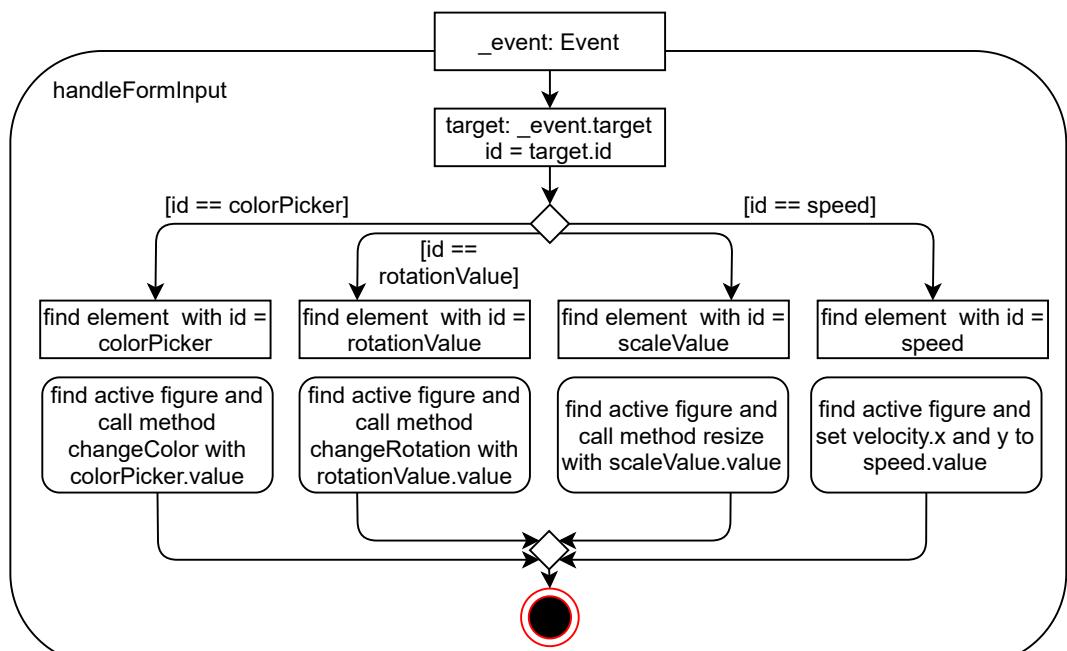
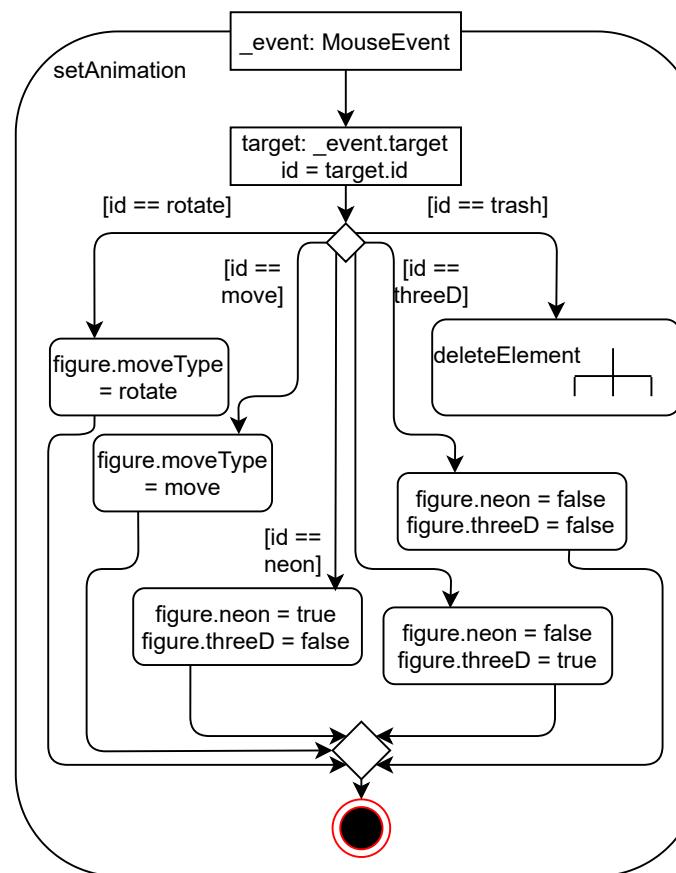
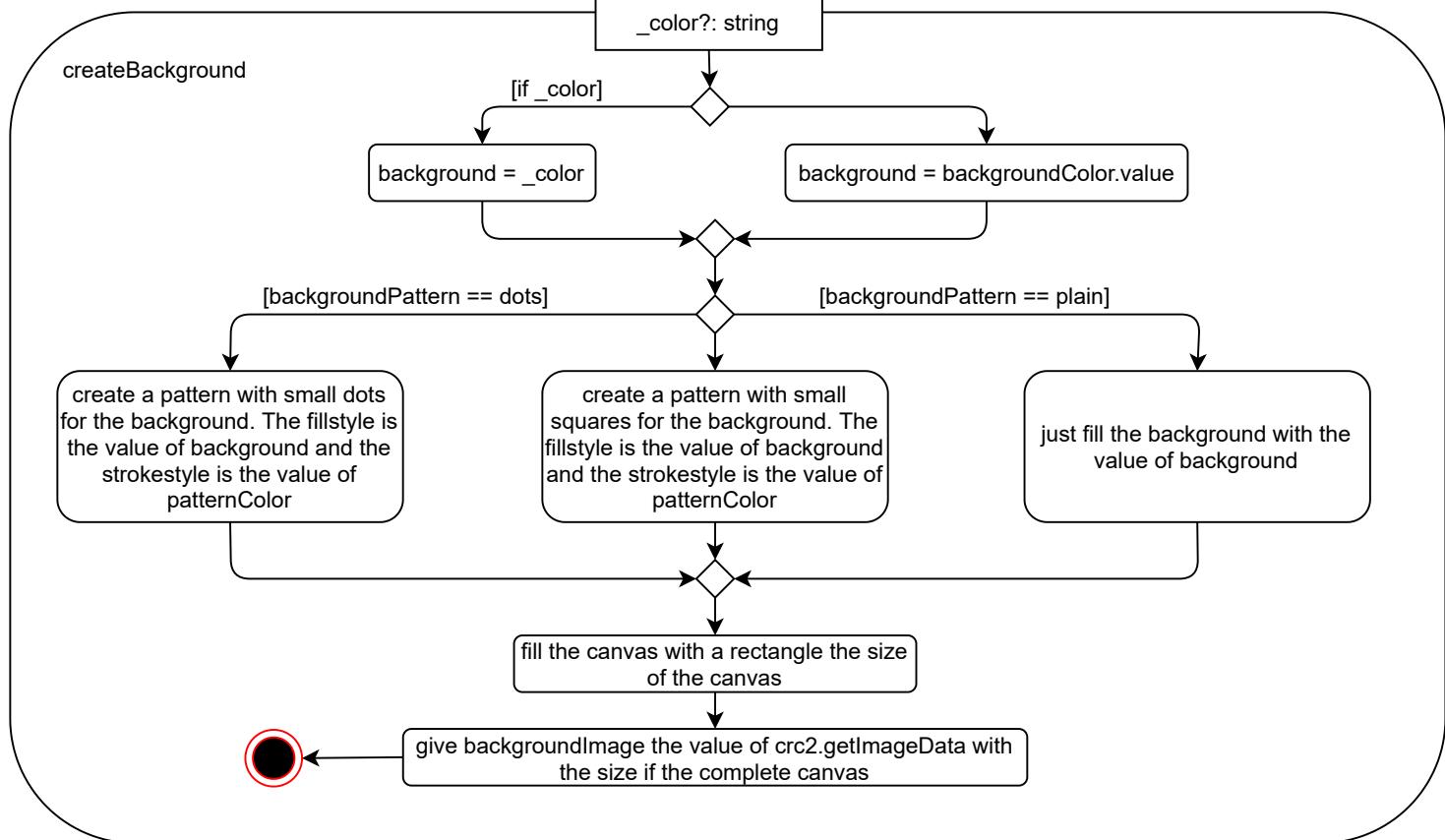
backgroundPattern = "dots"

backgroundPattern = "squares"

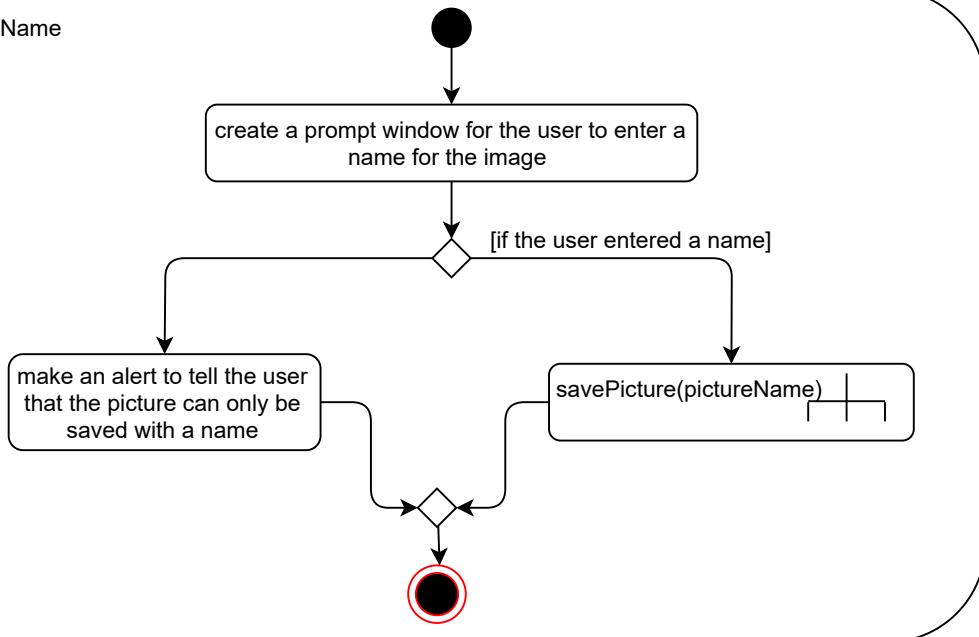
backgroundPattern = "plain"

createBackground





getName



setActive

_event: MouseEvent

find active figure and set property active to false

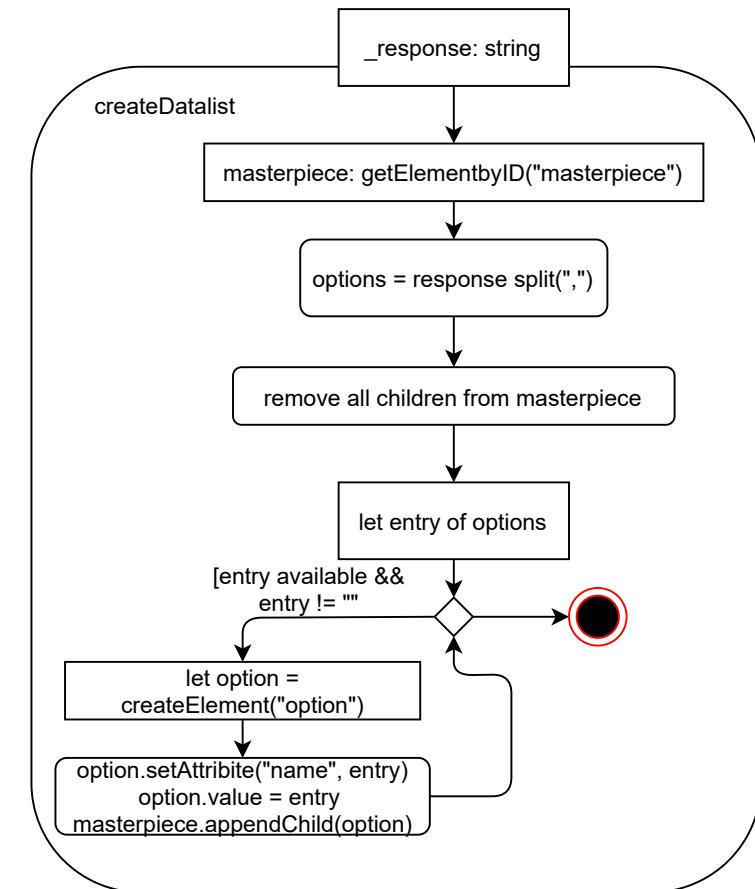
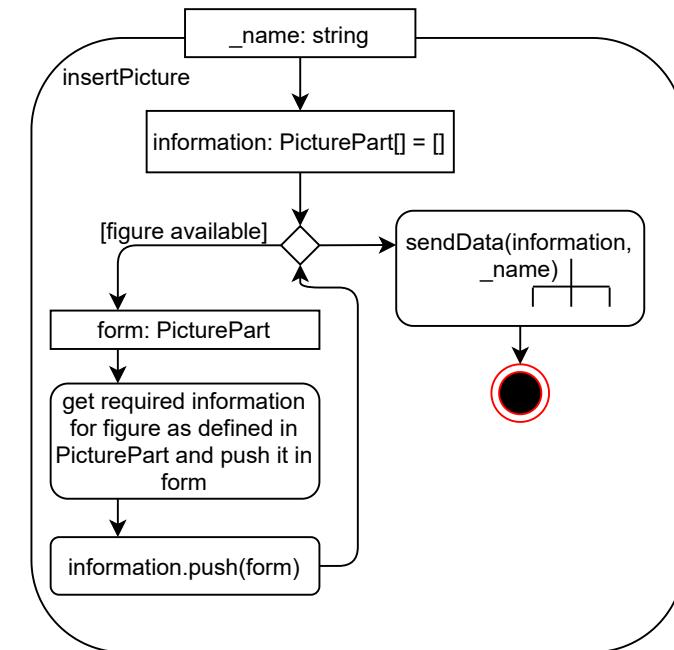
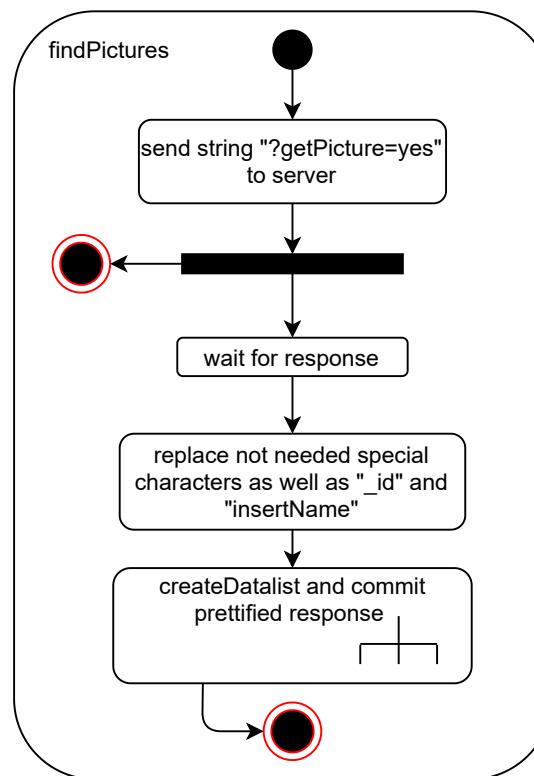
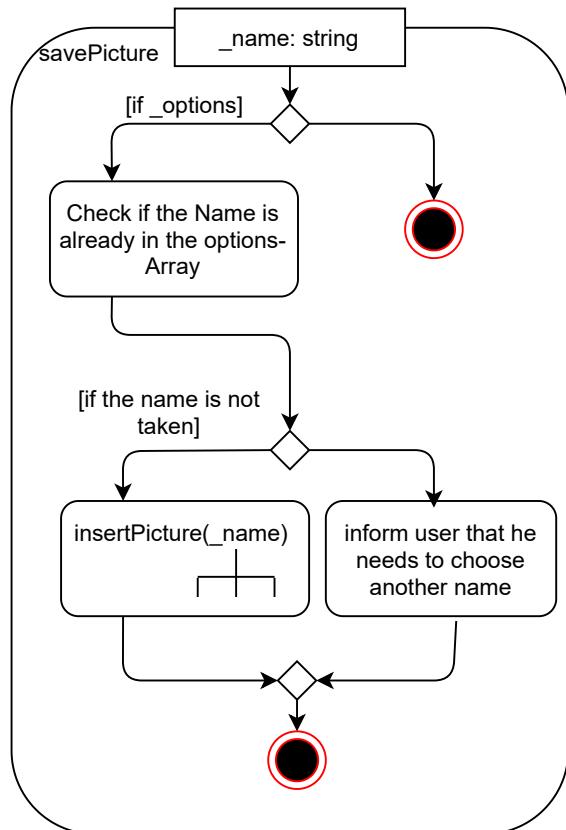
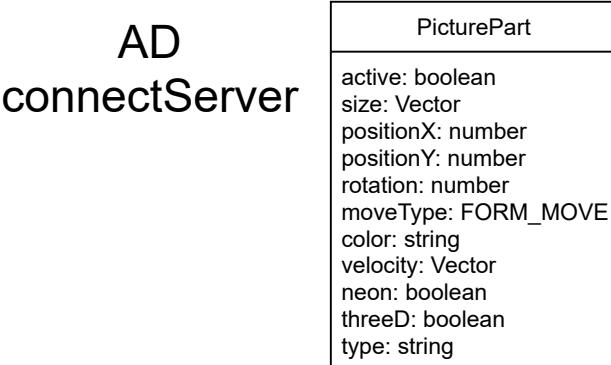
target = _event.target

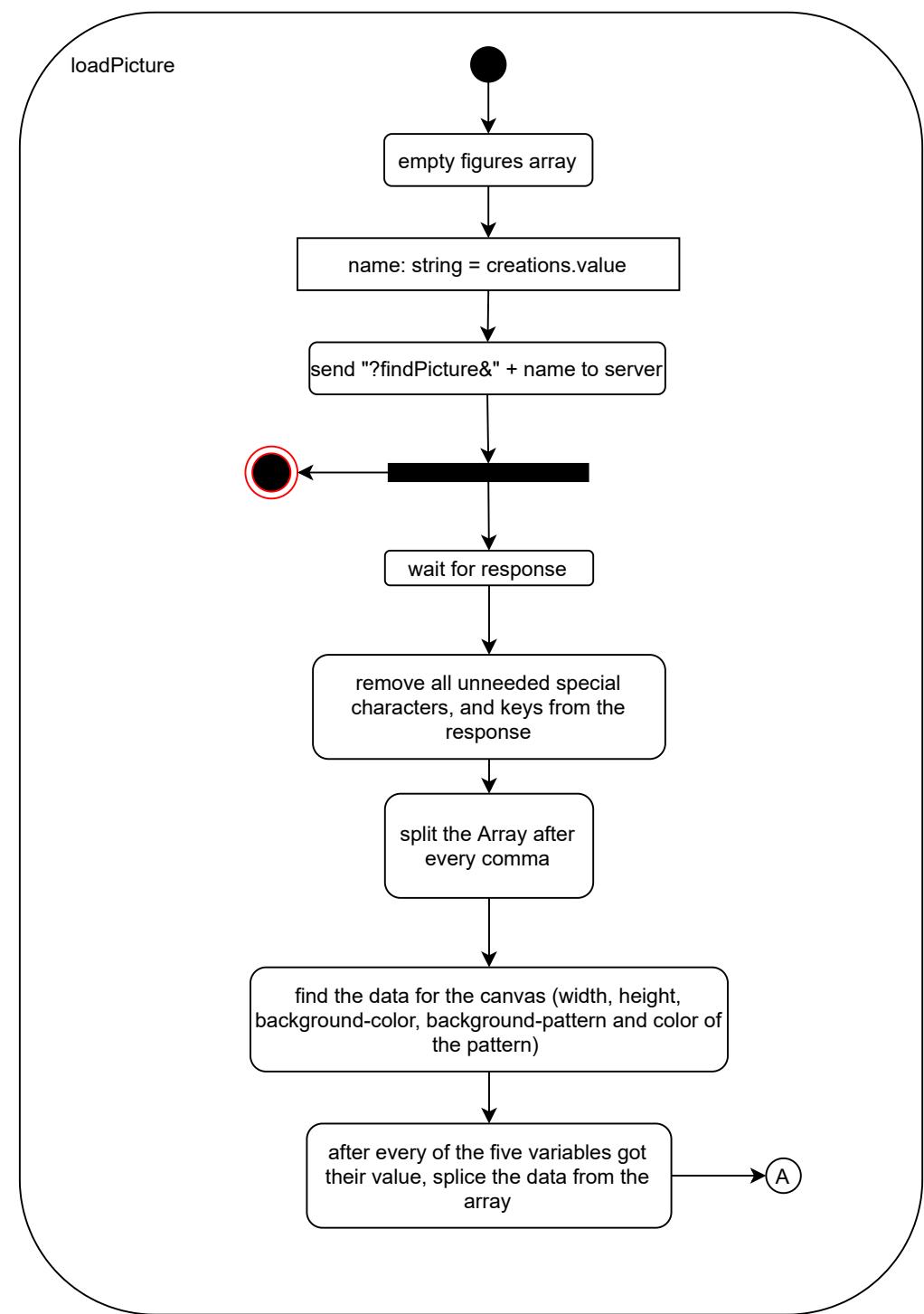
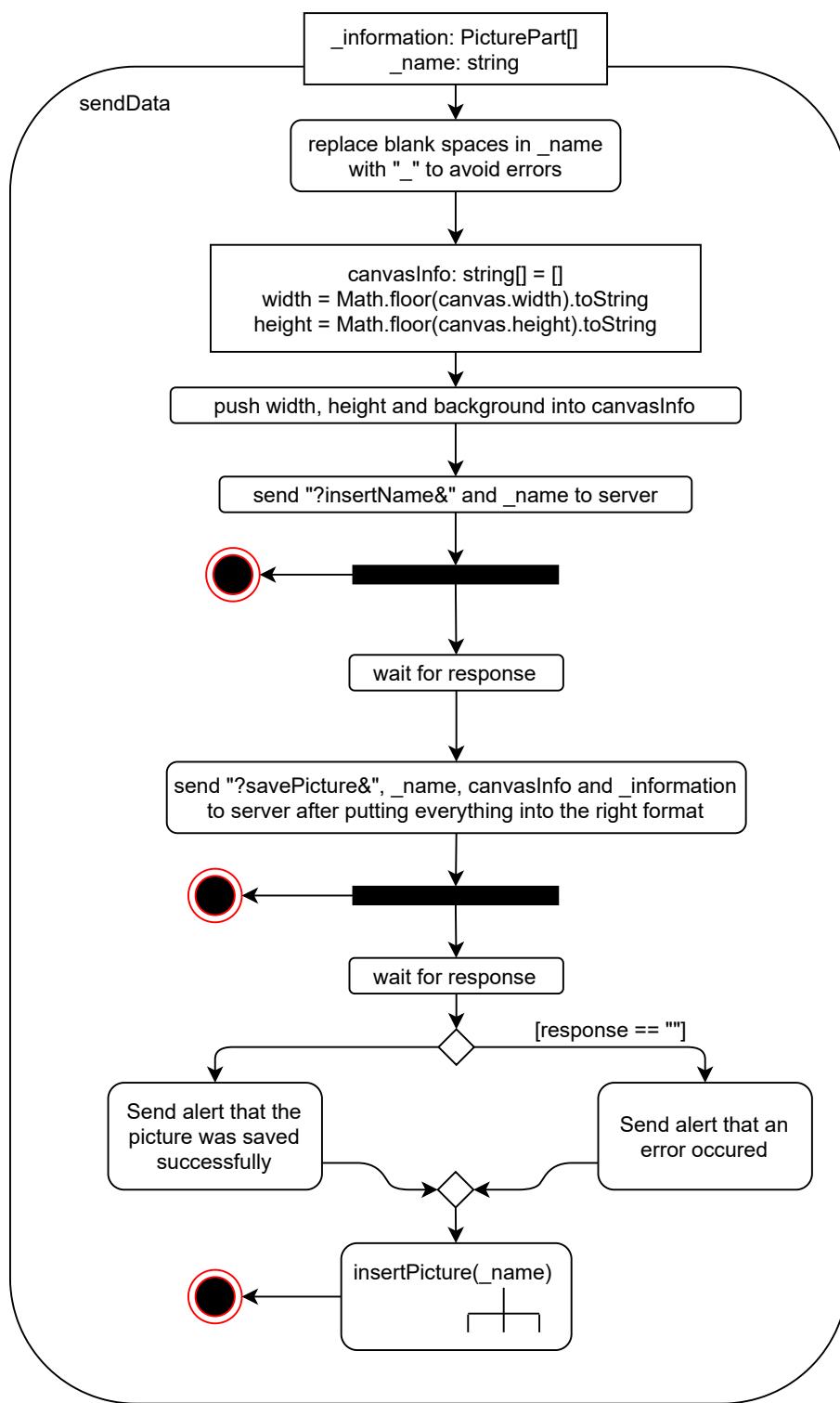
num = id-string to number

figures[num].active = true

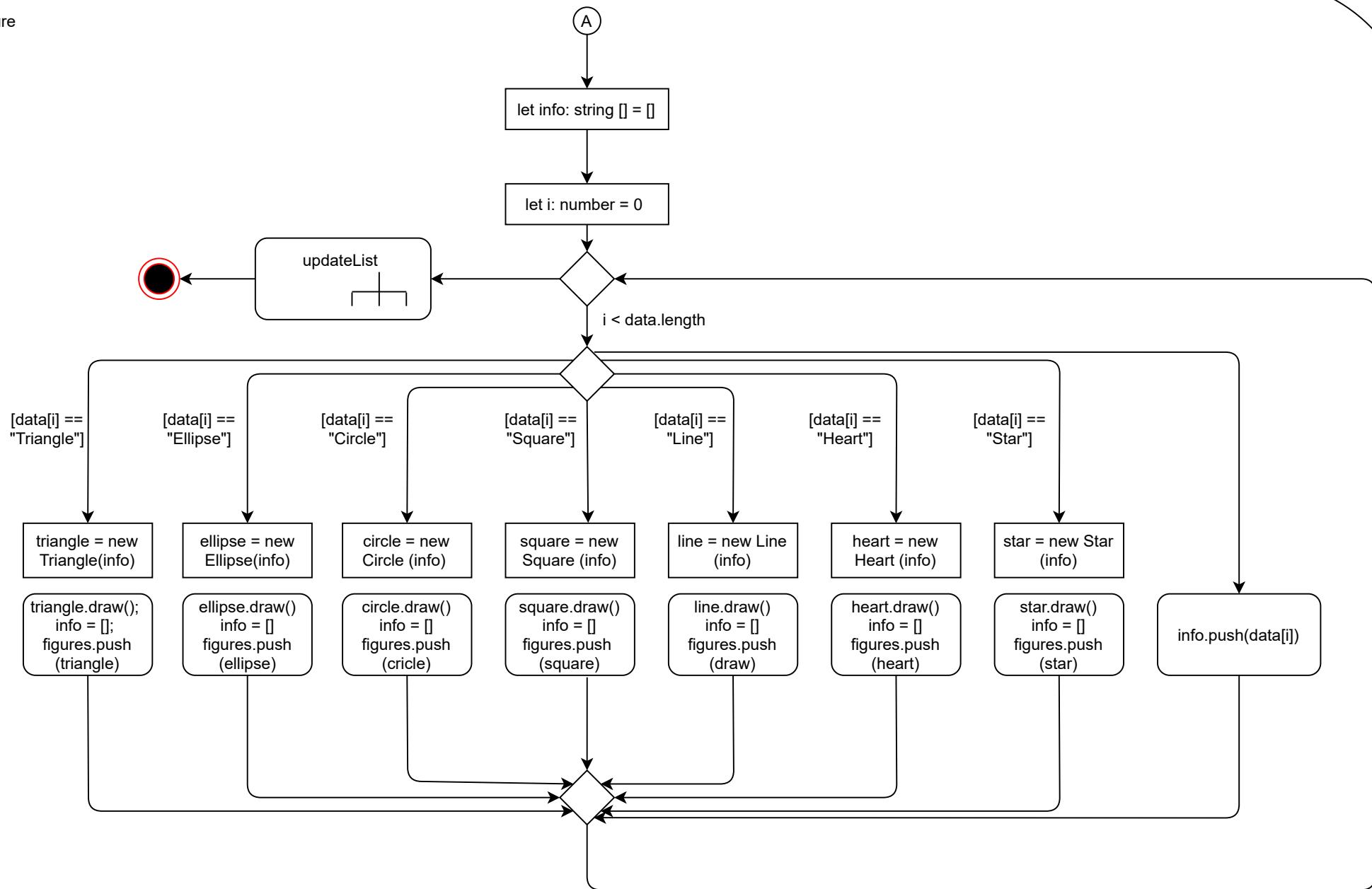


AD connectServer





loadPicture



AD Server

