

1. Introduction

About Command Line Vector Graphics Software (Clevis) . Users should be able to create and manipulate vector graphic containing shapes such as line segments, circles, rectangles, and squares using Clevis . Users can operate Clevis through the terminal (CLI mode) or through a GUI viewer.

2. System Requirements

To run the Clevis successfully , please ensure that you have installed the following into your computer:

Download and install Java SE Development Kit 21

<https://www.oracle.com/java/technologies/javase/jdk21-archive-downloads.html>

Download and install IntelliJ IDEA Community Edition 2024.2

<https://www.jetbrains.com/idea/download/other.html>

3. Running Instructions

Running in CLI Mode

Open Application.java

To launch Clevis with logging enabled, use the following command format:

Command enter `java Application.java -html log.html -txt log.txt -gui`

It will contain GUI Mode

A GUI window will appear and update automatically when commands are entered

If you don't want gui

Command enter `java Application.java -html log.html -txt log.txt`

Then it will run the program

[HTML_File_Path]: The path where the HTML log file will be saved (e.g., d:\log.html).

[TXT_File_Path]: The path where the plain text log file will be saved (e.g., d:\log.txt).

All operations you perform will be recorded in these two files.

4. GUI Mode Usage

When Clevis is executed with:

-gui

A graphical display window will appear.

GUI Behavior

Automatically refreshes when the user enters commands

Draws shapes in order of Z-index

Only visualizes (no mouse drawing)

Supports all shape types and updates dynamically

The GUI acts only as a viewer; all commands must still be typed in the console.

5. Notes

Clevis does not save data to disk

All shapes disappear after quitting

Coordinates follow Swing convention (0,0 at top-left corner)

GUI does not accept mouse input

Console commands remain the main control method

6. Command Line Usage

Below are all the commands supported by Clevis:

Shape Creation Commands

1. Create a Rectangle

Command: rectangle n x y w h

Effect: Creates a new rectangle named n.

Parameters:

n: The unique name for the rectangle.

x, y: The coordinates of the top-left corner.

w: The width of the rectangle.

h: The height of the rectangle.

2. Create a Line

Command: line n x1 y1 x2 y2

Effect: Creates a new line segment named n.

Parameters:

n: The unique name for the line.

x1, y1: The coordinates of the first endpoint.

x2, y2: The coordinates of the second endpoint.

3. Create a Circle

Command: circle n x y r

Effect: Creates a new circle named n.

Parameters:

n: The unique name for the circle.

x, y: The coordinates of the center.

r: The radius of the circle.

4. Create a Square

Command: square n x y l

Effect: Creates a new square named n.

Parameters:

n: The unique name for the square.

x, y: The coordinates of the top-left corner.

l: The side length of the square.

Grouping & Deletion Commands

5. Group Shapes

Command: group n n1 n2 ...

Effect: Creates a new group shape named n from existing shapes n1, n2, etc. After grouping, the individual shapes (n1, n2, ...) cannot be accessed directly until the group is ungrouped.

6. Ungroup a Shape

Command: ungroup n

Effect: Ungroups the group shape n into its component shapes. The group n is deleted, and the individual shapes can now be accessed directly again.

7. Delete a Shape

Command: delete n

Effect: Deletes the shape named n. If n is a group, all shapes within the group are also deleted.

Analysis & Transformation Commands

8. Move a Shape

Command: move n dx dy

Effect: Moves the shape named n horizontally by dx and vertically by dy. If n is a group, all shapes within the group are moved.

9. Find Bounding Box

Command: boundingbox n

Effect: Calculates and displays the minimum bounding box for the shape n.

Output Format: x y w h (the coordinates of the top-left corner, width, and height).

10. Check Intersection

Command: intersect n1 n2

Effect: Reports whether the bounding boxes of shapes n1 and n2 intersect.

11. Find Shape at Point

Command: shapeAt x y

Effect: Returns the name of the top-most shape that covers the point (x, y).

Coverage Rule: A shape covers a point if the minimum distance from the point to the shape's outline is less than 0.05. Shapes created more recently are on top of older shapes.

Information Commands

12. List Shape Information

Command: list n

Effect: Displays the basic information about the shape named n.

For simple shapes (rectangle, line, circle, square), it lists their defining parameters (e.g., center and radius for a circle).

For a group, it lists the group's name and the names of all shapes directly contained within it.

13. List All Shapes

Command: listAll

Effect: Lists the basic information for all shapes in the drawing, ordered from top-most (highest Z-index) to bottom-most. Indentation is used to show the hierarchy of groups and their components.

14. Quit the Tool

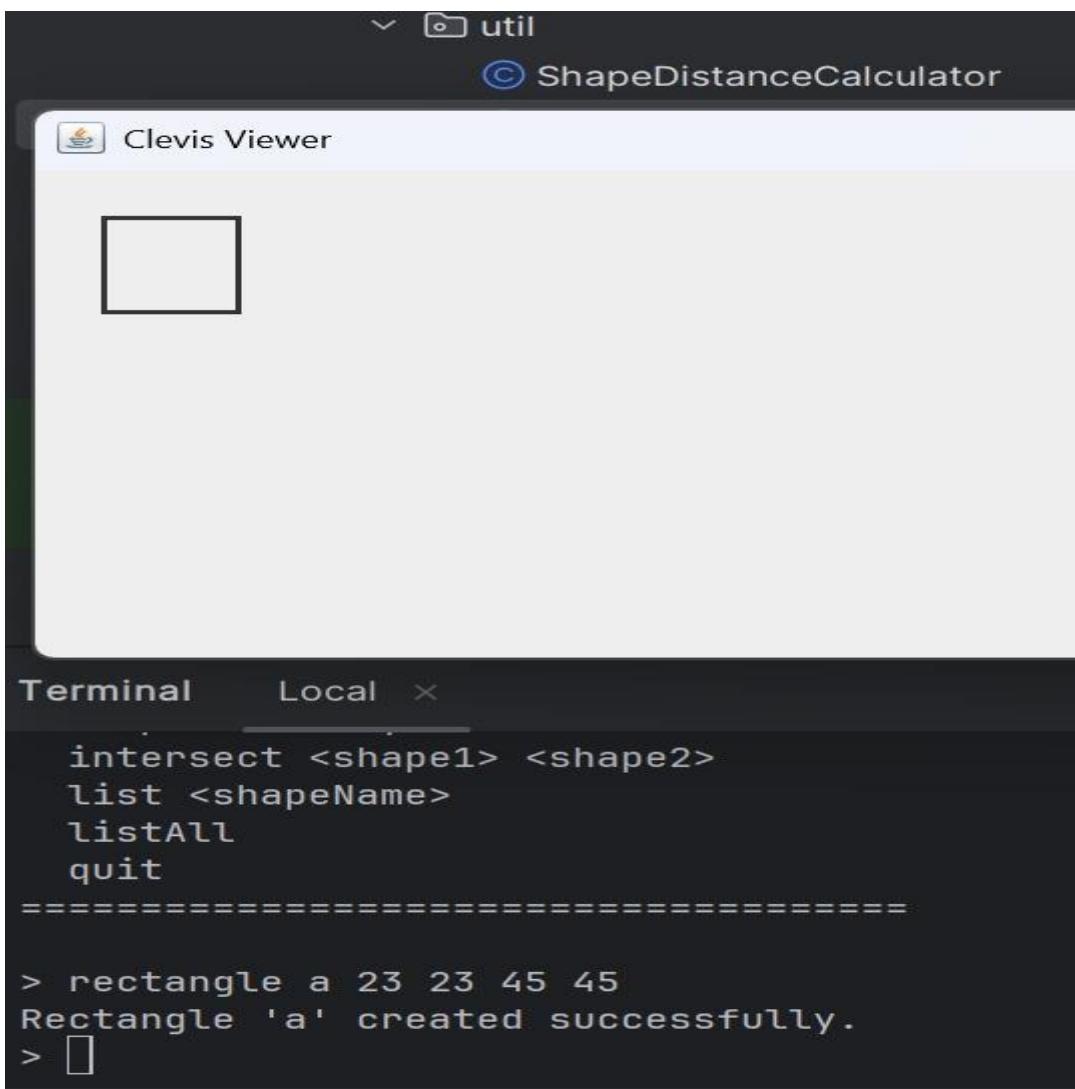
Command: quit

Effect: Terminates the Clevis application.

7. Example Workflow

Enter Rectangle name x y width height

Example: Rectangle a 23 23 45 45



The screenshot shows the Clevis application interface. At the top, there's a menu bar with 'util' and 'ShapeDistanceCalculator'. Below it is a toolbar with a 'Clevis Viewer' icon. The main window contains a drawing area with a single rectangle labeled 'a'. In the bottom right corner of the drawing area, there's a small square icon. At the very bottom of the screen is a terminal window with two tabs: 'Terminal' and 'Local'. The 'Terminal' tab is active, showing the following command history:

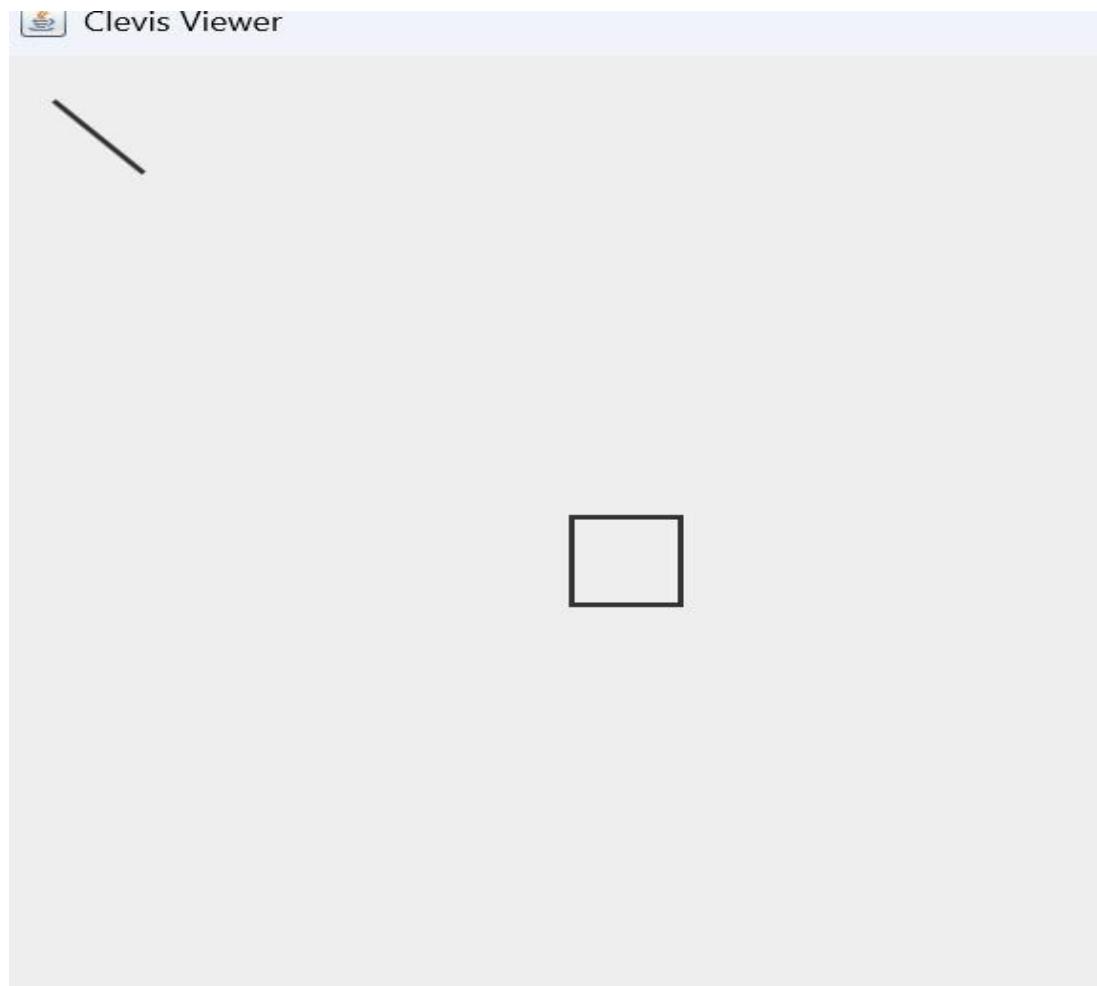
```
Terminal Local ×

intersect <shape1> <shape2>
list <shapeName>
listAll
quit
=====
> rectangle a 23 23 45 45
Rectangle 'a' created successfully.
> █
```

This code will create a rectangle a in the GUI window

Enter line name x1 y1 x2 y2

Example: Line bd 20 20 50 50



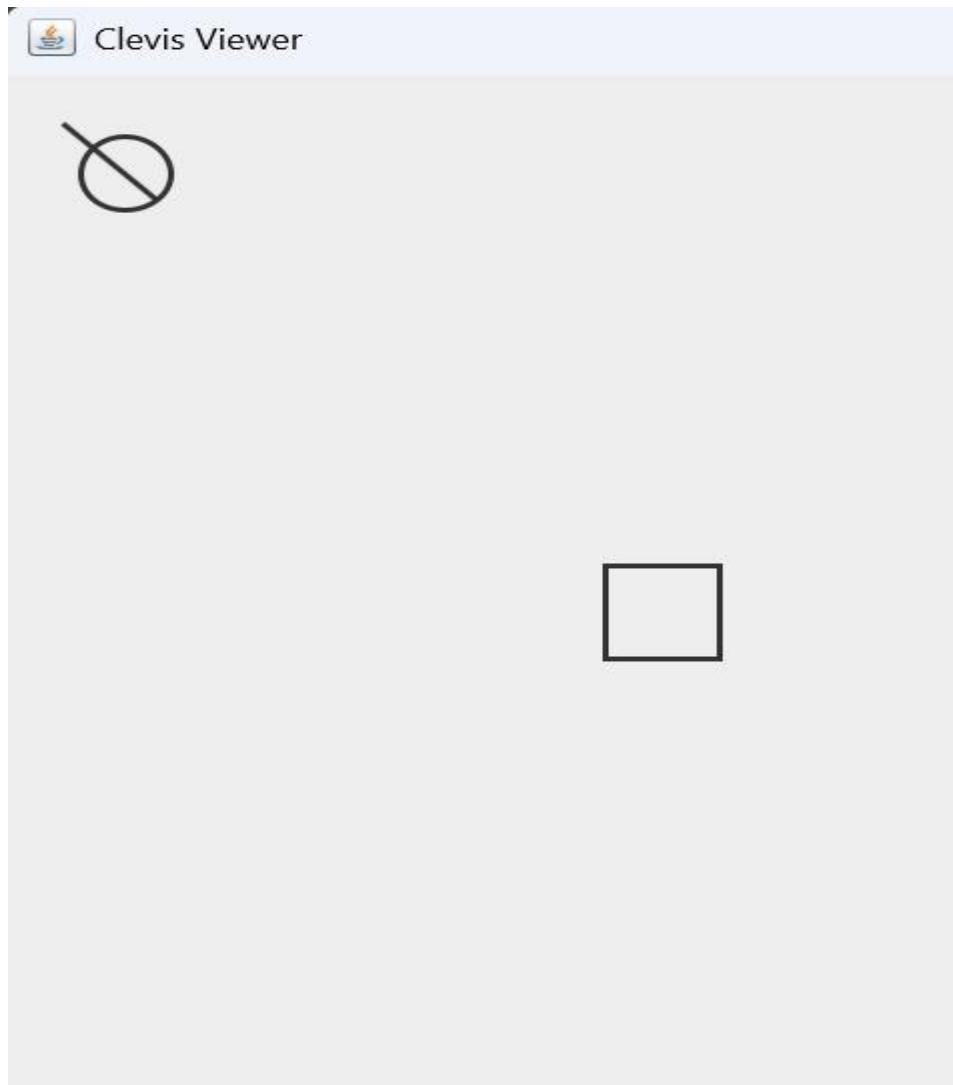
The screenshot shows a window titled "Clevis Viewer". Inside the window, there is a drawing area containing a single black line segment and a single black rectangle. The line segment is oriented diagonally from the top-left towards the bottom-right. The rectangle is a simple square.

```
Rectangle 'dj' created successfully.  
> line bd 20 20 50 50  
Line 'bd' created successfully.  
> []  
> SampleProject > SampleProject > test > hk > edu > |
```

This code will create a line bd in the GUI window

Enter circle name x y radius

Example: Circle dh 40 40 15



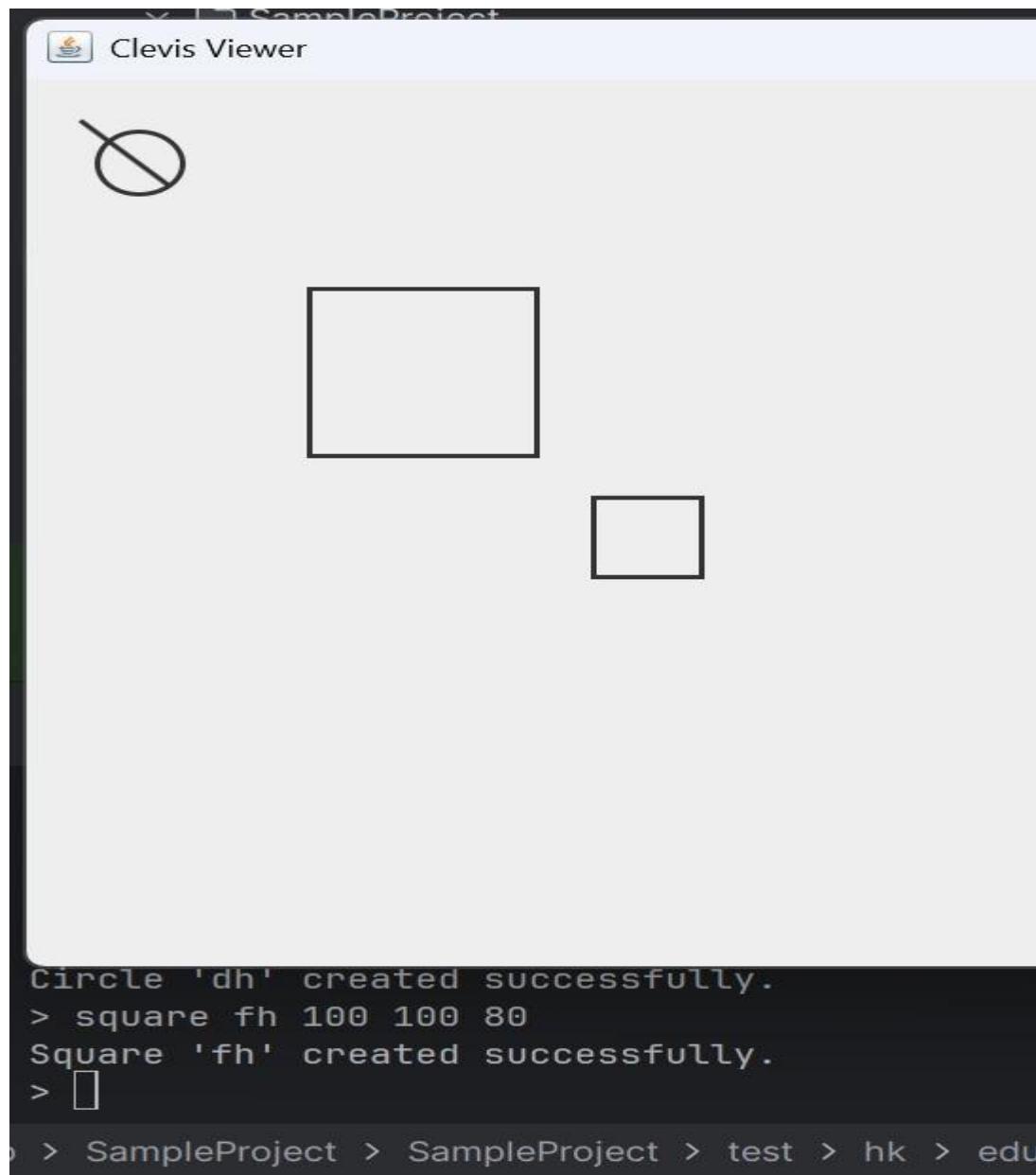
The screenshot shows the Clevis Viewer application window. At the top left is the application icon, which is a stylized flame or oil lamp. To its right is the window title "Clevis Viewer". The main area of the window contains two graphical elements: a circle with a diagonal slash through it in the upper-left quadrant, and a simple black square in the lower-right quadrant. Below the window, a terminal-like interface displays the command-line interaction:

```
Line 'bd' created successfully.  
> circle dh 40 40 15  
Circle 'dh' created successfully.  
> []  
> SampleProject > SampleProject > test > hk
```

This code will create a Circle dh in the GUI window

Enter Square name x y length

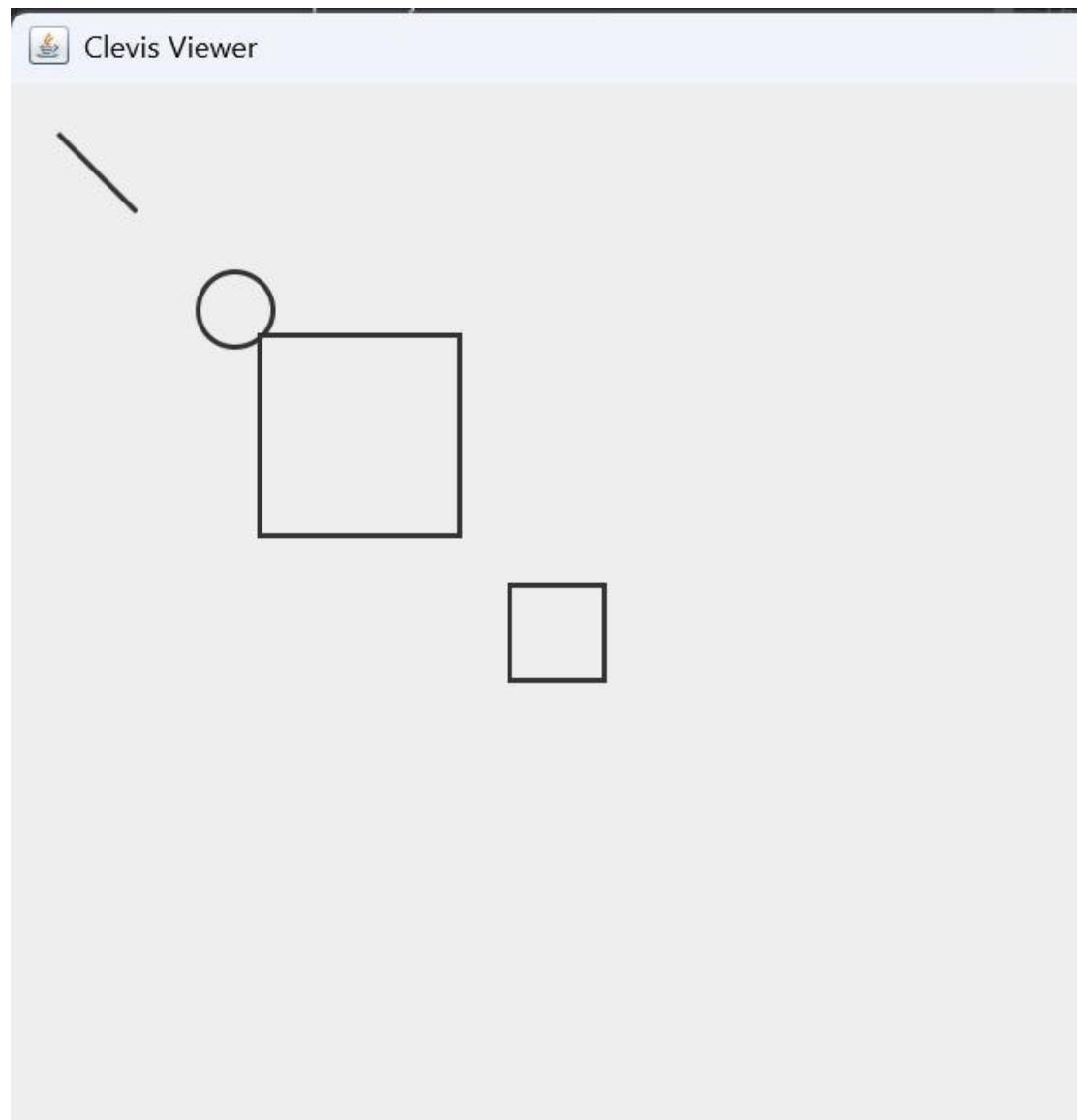
Example: Square fh 100 100 50



This code will create a square fh in the GUI window

Enter move shapename dx dy

Example: Move dh 50 50



The screenshot shows a window titled "Clevis Viewer". Inside the window, there is a light gray canvas area. On the canvas, there is a black-outlined square and a black-outlined circle. A thin black line connects the top-left corner of the square to the center of the circle. Below the canvas, there is a dark gray terminal-like interface displaying the following text:

```
Square 'dh' created successfully.  
> move dh 50 50  
Shape 'dh' moved by (50.00, 50.00).  
> █
```

This code will move the Circle dh 50 50 in the GUI window

Enter Group name shapename1 shapename2

Example: Group fj dh fh bd

Enter ungroup name

Example: Ungroup fj

```
Terminal Local × + ▾
> delete fh
Shape 'fh' deleted successfully.
> ungroup fj
Group 'fj' ungrouped successfully.
> ungroup fj
Error: Shape not found: fj
> ungroup bd
Error: Shape is not a group: bd
> █
> SampleProject > SampleProject > test > hk
```

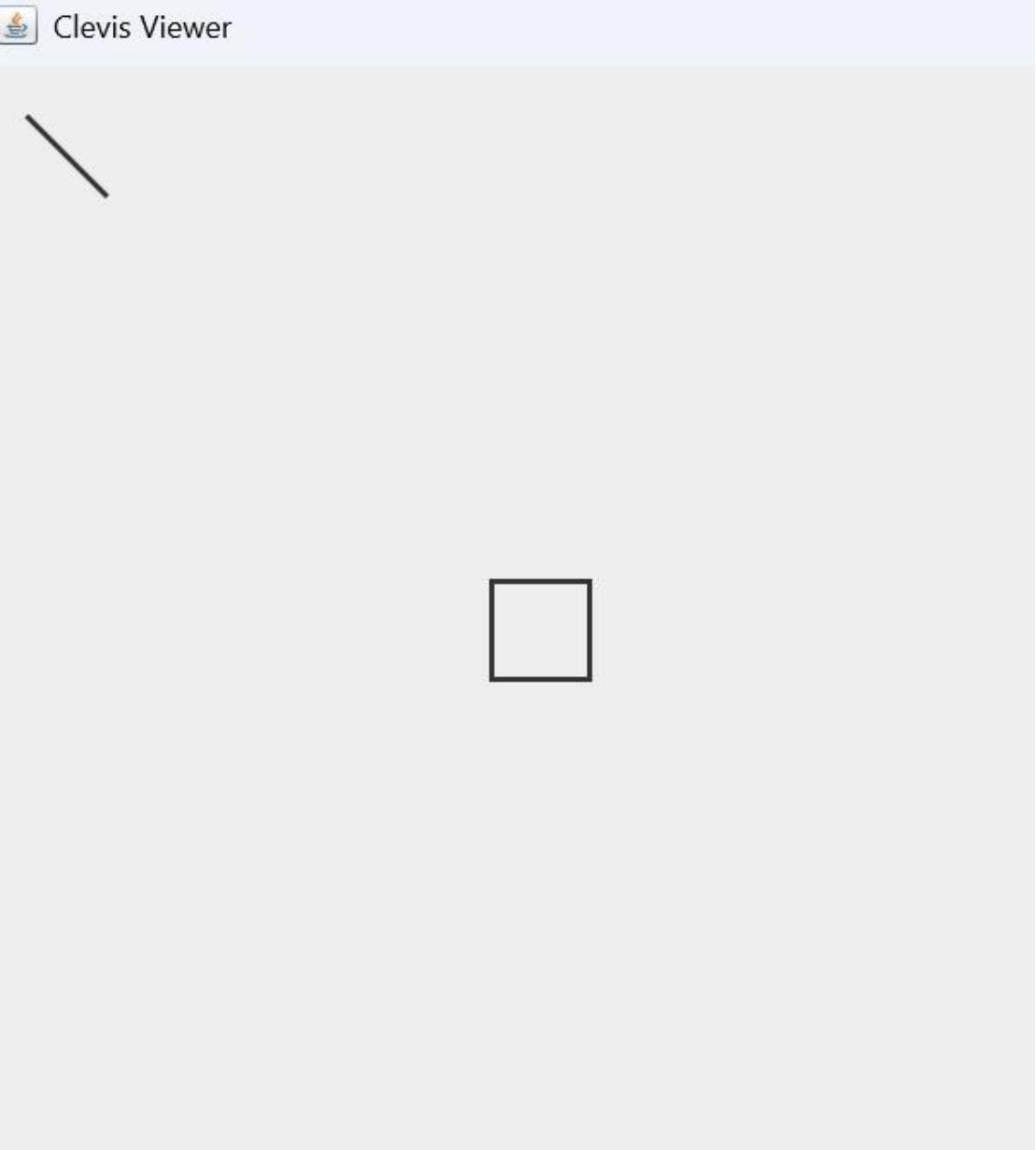
There are two code here

Group will make the shape names which you enter here become to a group

Ungroup will cancel the group you created before

Enter Delete name

Example: Delete fh, dh



```
> delete dh
Shape 'dh' deleted successfully.
> delete fh
Shape 'fh' deleted successfully.
> []
> SampleProject > SampleProject > test > hk > edu > pol
```

This code will delete the shape you created before

Enter bounding name

Example: Boundingbox bd

```
Terminal Local × + ▾  
> ungroup bd  
Error: Shape is not a group: bd  
> boundingbox fj  
Error: Shape not found: fj  
> boundingbox bd  
20.00 20.00 30.00 30.00  
> shapeAt 20 20  
Shape at (20.0, 20.0): bd  
>  
↳ > SampleProject > SampleProject > test > hk
```

This code will show the two side coordinates of the shape name you enter

Enter shapeAt xy

Example: shapeAt 20 20

```
Terminal Local × + ▾  
> boundingbox fj  
Error: Shape not found: fj  
> boundingbox bd  
20.00 20.00 30.00 30.00  
> shapeAt 20 20  
Shape at (20.0, 20.0): bd  
> circle 20 40 50  
Error: Usage: circle <name> <cx> <cy> <radius>  
> █
```

This code will show the shape which on your enter coordinates

Enter intersect shapename1 shapename2

Example: Intersect hd bd

```
Terminal Local × + ▾  
> circle 20 40 50  
Error: Usage: circle <name> <cx> <cy> <radius>  
> circle hd 20 20 40  
Circle 'hd' created successfully.  
> intersect hd bd  
Shapes 'hd' and 'bd' intersect.  
> list hd  
hd circle 20.00 20.00 40.00  
>  
> SampleProject > SampleProject > test > hk > edu >
```

This code will check your enter shape name are they Intersect

Enter List name

Example: list hd

Exampe: listAll

```
Terminal Local × + ▾  
> intersect hd bd  
Shapes 'hd' and 'bd' intersect.  
> list hd  
hd circle 20.00 20.00 40.00  
> listAll  
hd circle 20.00 20.00 40.00  
bd line 20.00 20.00 50.00 50.00  
dj rectangle 200.00 200.00 38.00 38.00  
> █  
> SampleProject > SampleProject > test > hk >
```

There are two code here

List name will only show one shape information

Listall will show all the shape information which do not be deleted

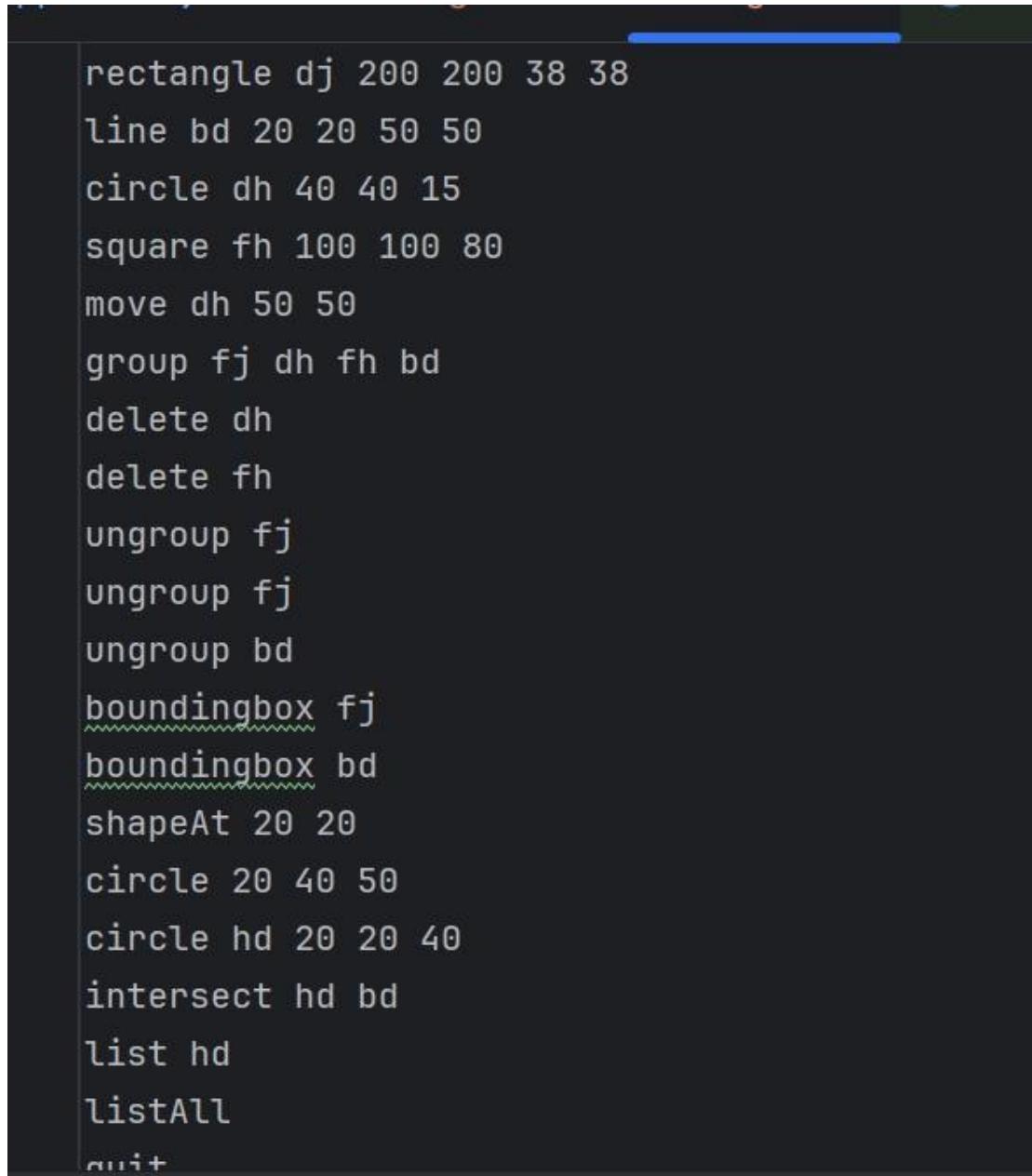
Enter quit

Example: quit

```
Terminal Local × + ✓
dj rectangle 200.00 200.00 38.00 38.00
> quit
Saving logs and exiting...
TXT log saved to: log.txt
HTML log saved to: log.html
Thank you for using Clevis. Goodbye!
TXT log saved to: log.txt
HTML log saved to: log.html
(base) PS C:\Users\lsu22\Downloads\git\oop\SampleProject\SampleProject\src\hk\edu\polyu\comp\comp2021\clevis>
> SampleProject > SampleProject > test > hk > edu > polyu > comp > comp2021 > clevis > 🌱 ClevisTest > 🍎 testList
```

Terminates the execution of Clevis.

Record



```
rectangle dj 200 200 38 38
line bd 20 20 50 50
circle dh 40 40 15
square fh 100 100 80
move dh 50 50
group fj dh fh bd
delete dh
delete fh
ungroup fj
ungroup fj
ungroup bd
boundingbox fj
boundingbox bd
shapeAt 20 20
circle 20 40 50
circle hd 20 20 40
intersect hd bd
list hd
listAll
exit+
```

It will record the commands

8. Error Handling

Clevis provides user-friendly messages when errors occur:

Possible error situations:

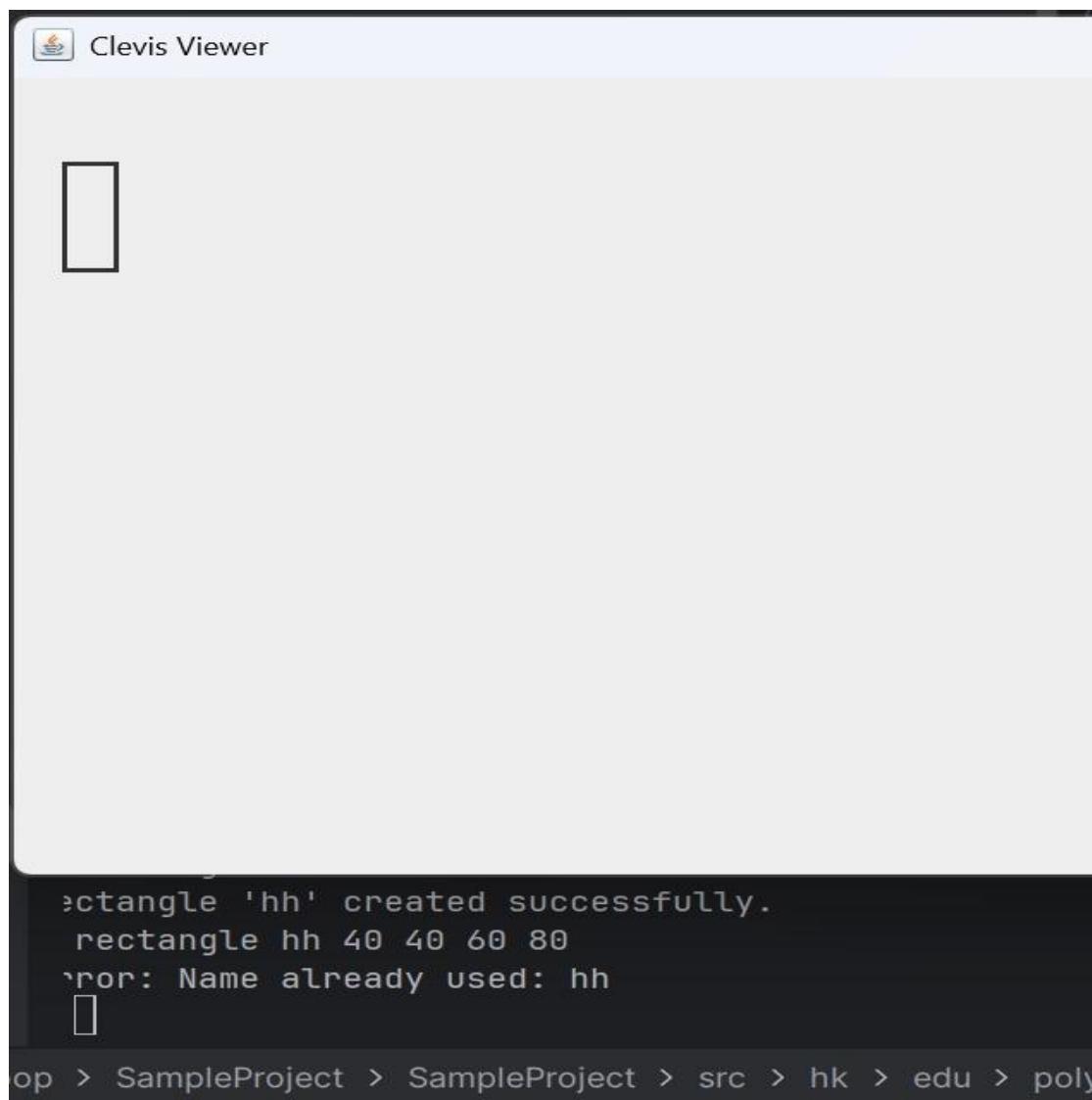
- 1.Duplicate shape name
- 2.Invalid numeric values (e.g., negative width and height)
- 3.Shape without name

Sample messages:

Error: name already used

Example: User enters a shape name which already use

Name hh have been used before



The screenshot shows a window titled "Clevvis Viewer" with a toolbar icon. Inside, there is a single black rectangle labeled "hh". Below the viewer is a terminal window displaying the following text:

```
rectangle 'hh' created successfully.
rectangle hh 40 40 60 80
error: Name already used: hh
[]
```

At the bottom of the terminal window, the path "op > SampleProject > SampleProject > src > hk > edu > poly" is visible.

User should change another name for the shape

Error: radius/width and height must be positive

Example: User enters a negative radius/width and height

-45 radius

```
> rectangle a 23 23 45 45
Rectangle 'a' created successfully.
> rectangle c 23 23 -34 34
Error: width and height must be positive!!
> circle c 30 30 -45
Error: radius must be positive!!
> █
```

Some value such as radius, width and height must be positive. User should change them

Error: no name

Example: User forgot to create a name for the shape

```
Terminal Local × + ▾
> boundingbox fj
Error: Shape not found: fj
> boundingbox bd
20.00 20.00 30.00 30.00
> shapeAt 20 20
Shape at (20.0, 20.0): bd
> circle 20 40 50
Error: Usage: circle <name> <cx> <cy> <radius>
> █
```

User need to create a name for the shape

just enter the commands again this time don't forget the name