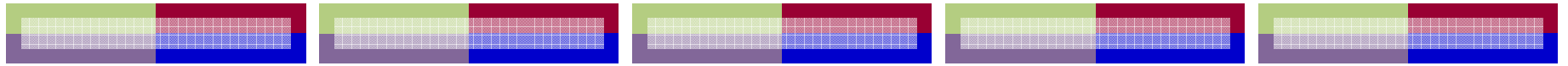


CS101

Problem Solving and Object-Oriented Programming

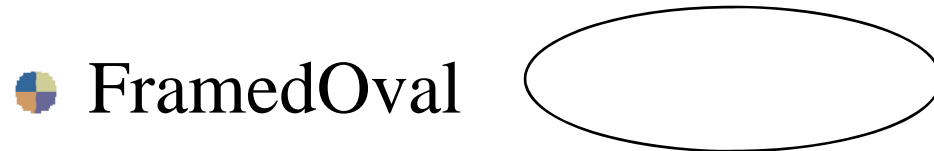
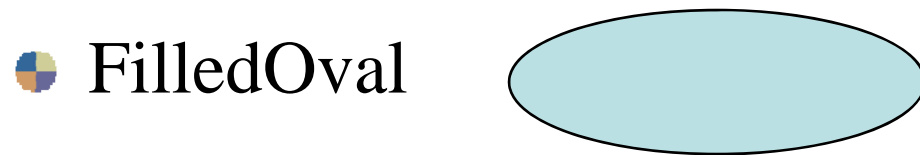
L11: Graphical Objects; Naming and Modifying Objects



Creating Graphical Objects

- FilledRect, FramedRect, FilledOval, FramedOval

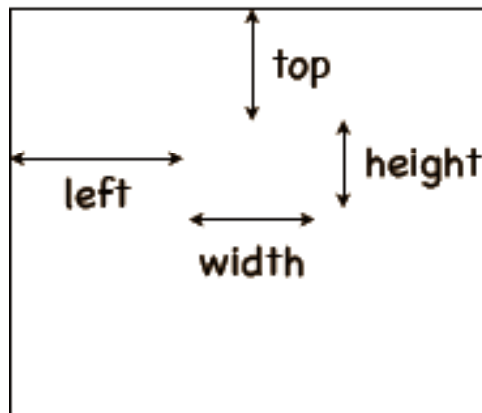
- Specify left, top, width, height, canvas
- `new FilledRect (50, 200, 300, 100, canvas);`





Creating Graphical Objects

- FilledRect, FramedRect, FilledOval, FramedOval
 - Specify **left**, **top**, **width**, **height**, canvas
 - new FilledRect (**50**, **200**, **150**, **100**, canvas);

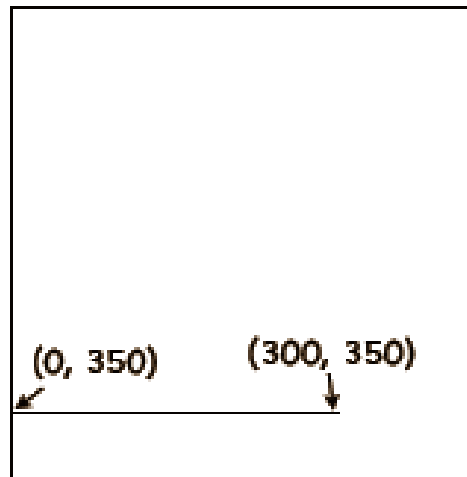




Creating Graphical Objects

● Line

- Specify 2 endpoints as x,y values, canvas
- new Line (0, 350, 300, 350, canvas);

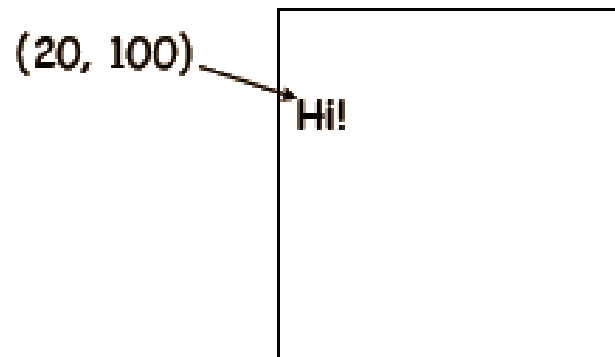




Creating Graphical Objects

● Text

- Specify String, left, top, canvas
- `new Text ("Hi!", 20, 100, canvas);`

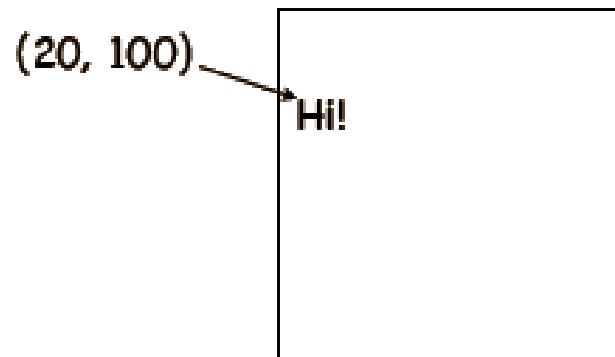


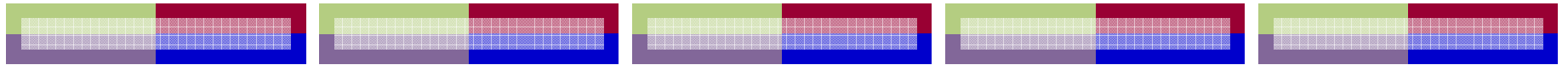


Creating Graphical Objects

● Text

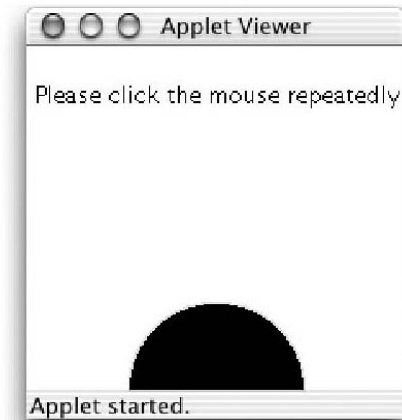
- Specify String, left, top, canvas
- `new Text ("Hi!", 20, 100, canvas);`





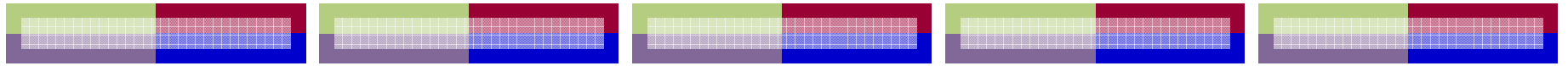
A Sample Program

A rudimentary solar simulation:



We want a click of the mouse in the window to make the 'sun' rise





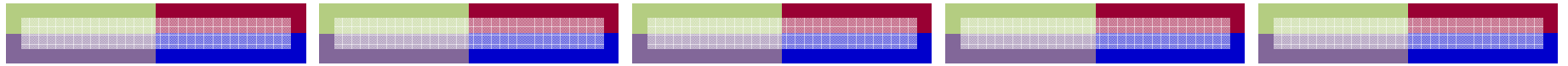
Moving an Oval

- Mutator methods: change an object that's been created

```
move ( 0, -5 );
```

- But how does the computer know what to move?
Give the object a name!





Some Building Blocks

```
private FilledOval sun;
```

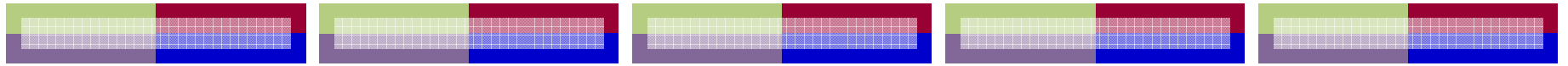
-
-
-

```
sun = new FilledOval( 50, 150, 100, 100, canvas );
```

-
-
-

```
sun.move( 0, -5 );
```





Giving Names

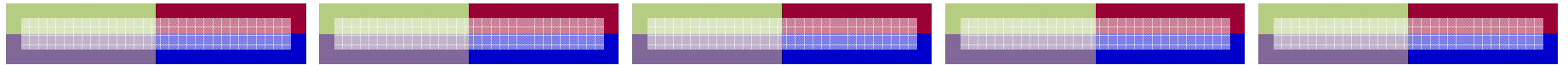
- Declare all instance variables

```
private FilledOval sun;
```

- Appropriate Names

- Start with letters
- Case sensitive
- Letters, digits, underscores
- Not a word already in Java





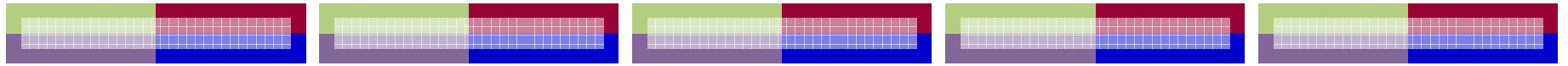
Comments for Clarity

- In our line

```
private FilledOval sun; //sun is the name
```

- `sun is the name` is a comment



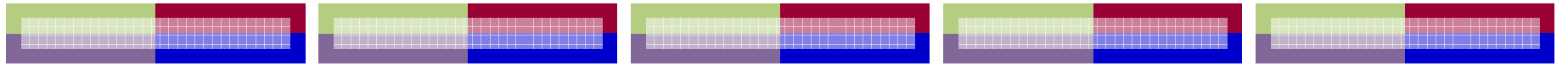


Mutator Methods in Context

- We want to move the sun when the user clicks the mouse:

```
public void onMouseClick( Location point){  
    sun.move( 0, -5 );  
}
```





More Mutator Methods

```
private void showInstructions() {  
    //Display of instructions  
}
```

```
    instructions[i] = new Text (...);  
}
```

```
    instructions[i].hide();  
}
```

```
instructions.show();  
}
```





```
public class RisingSun extends WindowController {
```

```
    private FilledOval sun; // Circle that represents the sun
```

```
    private Text instructions; //Display of instructions
```

```
    public void begin() { //Place the sun and brief instructions on screen
```

```
        sun = new FilledOval( 50, 150, 100, 100, canvas );
```

```
        instructions = new Text("Please click the mouse ",  
                                20, 20, canvas );
```

```
    }
```

```
    //Move the sun up each click
```

```
    public void onMouseClick( Location point ) {
```

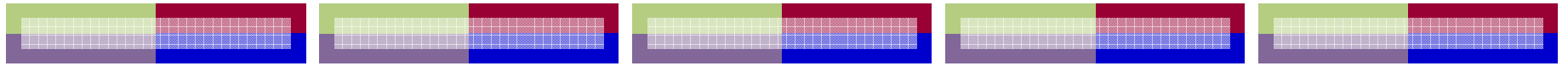
```
        sun.move( 0, -5 );
```

```
        instructions.hide();
```

```
    }
```

```
}
```





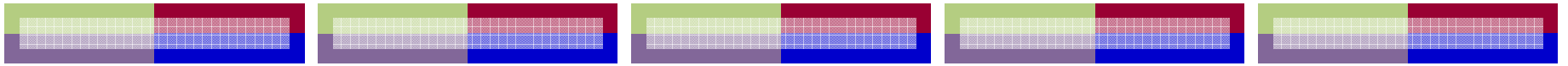
More Classes

• Classes so far: Line, FilledOval, Text

• Can also have nongraphical classes!

- Color
- Location





Colors

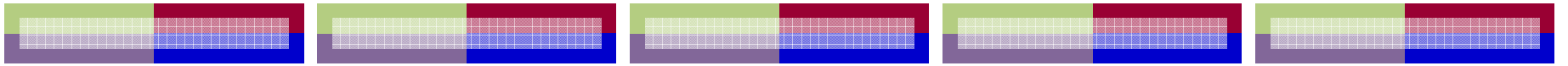
```
private • or purple;  
•  
•
```

```
purple = • w Color (255, 0, 255);  
•  
•
```

```
sun.setC • r( Color.YELLOW ); //makes sun yellow  
•  
•
```

```
instructions.setColor( purple ); //makes instr purple
```





Locations

```
private • cation initialPosition;  
•  
•
```

```
initial • tion = new Location( 50, 150 );  
•  
•
```

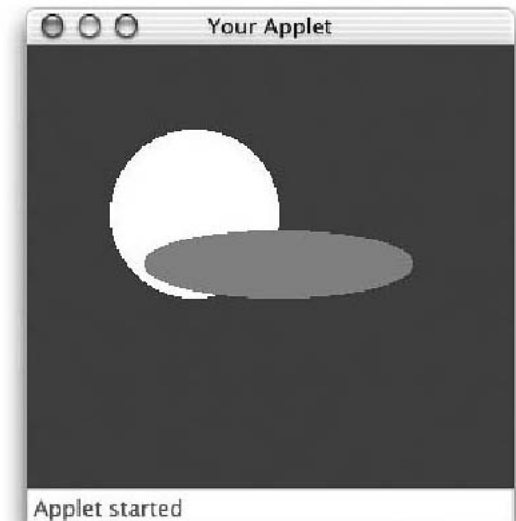
```
sun.moveTo ( initialPosition );
```

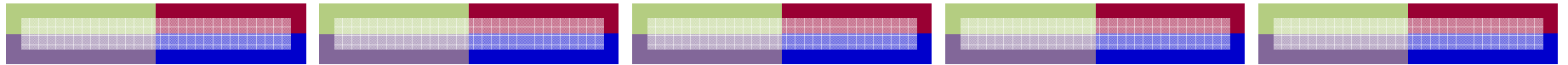




Layering the Canvas

- Create two overlapping FilledOvals. Which one's on top?
 - Answer: The most recently constructed
- How do we change the order?
 - Answer: Mutator Methods
 - `sendBackward()`
 - `sendForward()`
 - `sendToBack()`
 - `sendToFront()`





Using a Mouse Point

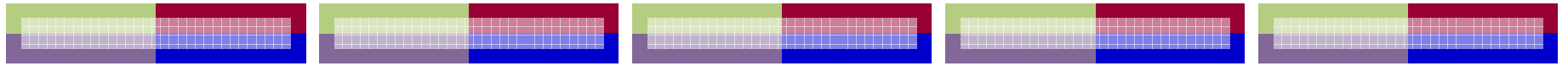
- Recall `onMousePress(Location point);`
 - We can use the Location!

● Consider:

```
public void onMousePress ( Location point ) {  
    new Text ( "Pressed", point, canvas );  
}
```

Displays "Pressed" wherever the mouse is clicked





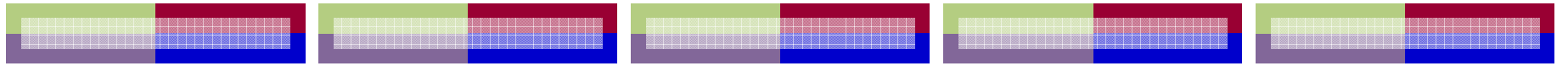
Using Multiple Points

```
private Location firstPoint;

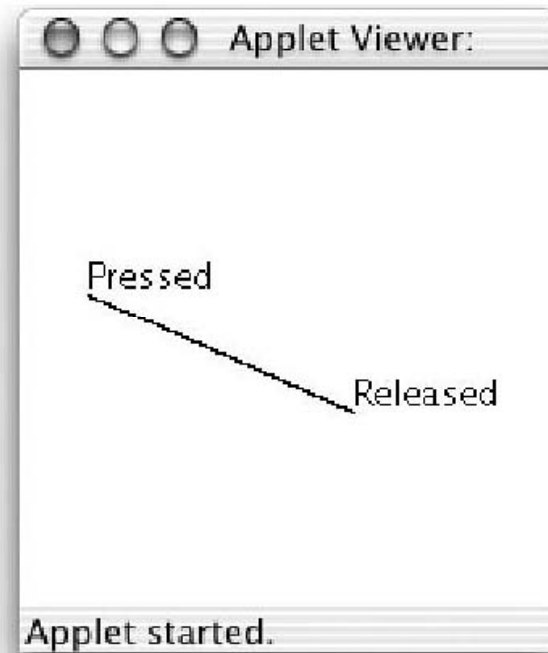
public void onMousePress( Location pressPt ){
    new Text("Pressed", pressPt, canvas );
    firstPoint = pressPt;
}

public void onMouseRelease ( Location releasePt){
    new Text("Released", releasePt, canvas );
    new Line( firstPoint, releasePt, canvas );
}
```



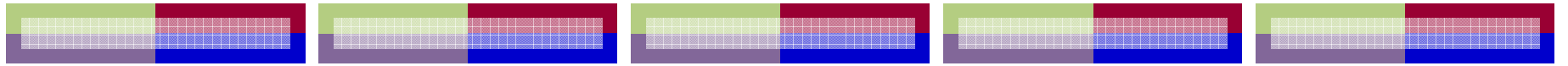


Multiple Point Picture



- Can play connect the dots!





Review

- Graphical and nongraphical objects
- Names and mutator methods
- Layering the canvas

