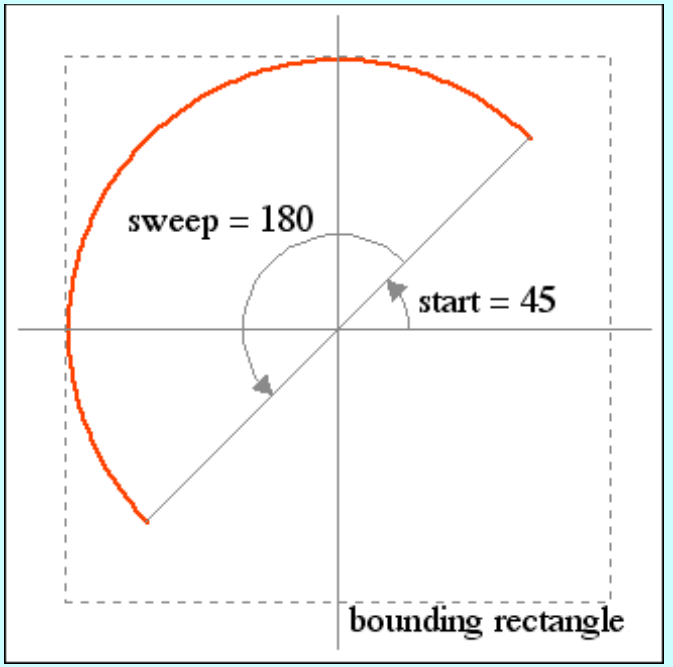
Advanced Animation

Jerry Cain
CS 106AX
October 9, 2023
slides leveraged from those written by Eric Roberts

The `GArc` Class

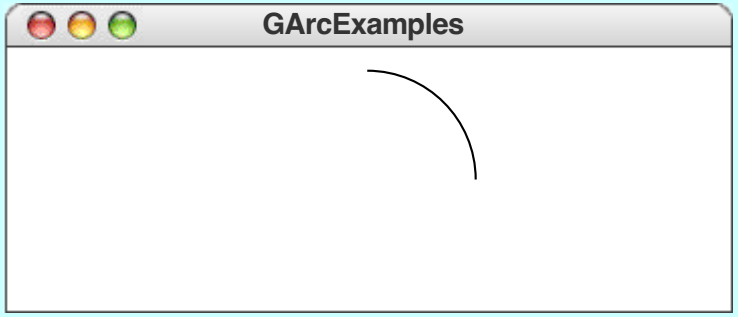
- The **`GArc`** class represents an arc formed by taking a section from the perimeter of an oval.
- Conceptually, the steps necessary to define an arc are:
  - Specify details of bounding rectangle as you would for a full **`GOval`**.
  - Specify the ***start angle***, which is the angle at which the arc begins.
  - Specify the ***sweep angle***, which indicates how far the arc extends.
- The geometry used by the **`GArc`** class is shown in the diagram on the right.
- Angles are measured in degrees starting at the +*x* axis (the 3:00 o'clock position) and increasing counterclockwise.
- Negative values for the *start* and *sweep* angles signify a clockwise direction.



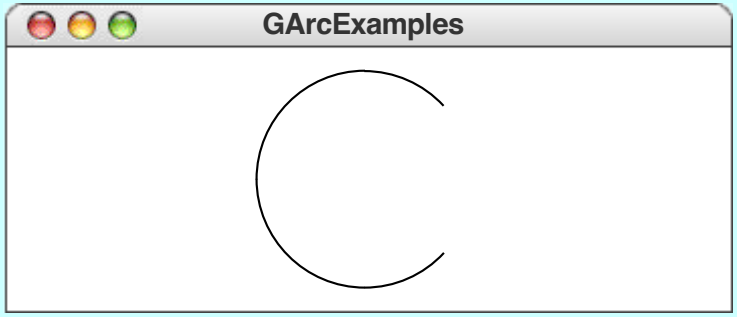
Exercise: `GArc` Geometry

Suppose that the variables **`cx`** and **`cy`** contain the coordinates of the center of the window and that the variable **`d`** is 0.8 times the screen height. Sketch the arcs that result from each of the following code sequences:

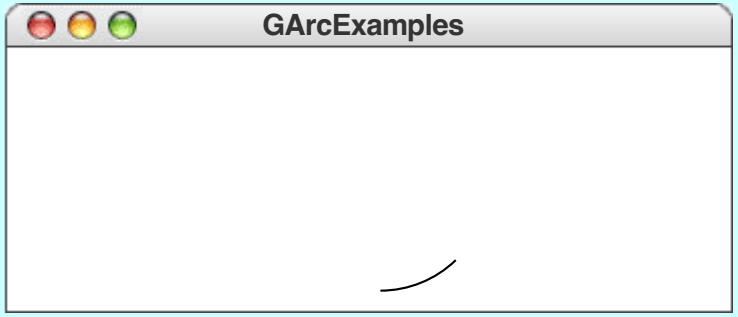
```
let a1 = GArc(d, d, 0, 90);
gw.add(a1, cx - d / 2, cy - d / 2);
```



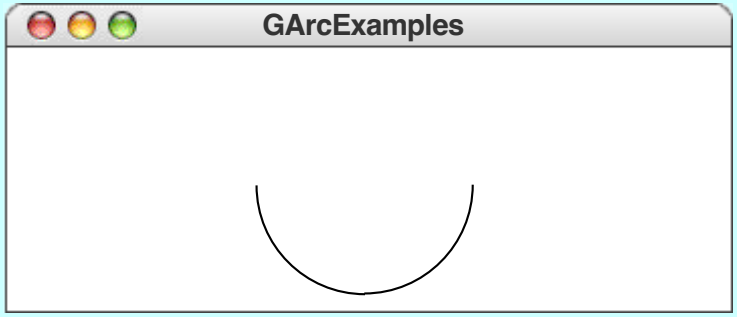

```
let a2 = GArc(d, d, 45, 270);
gw.add(a2, cx - d / 2, cy - d / 2);
```




```
let a3 = GArc(d, d, -90, 45);
gw.add(a3, cx - d / 2, cy - d / 2);
```



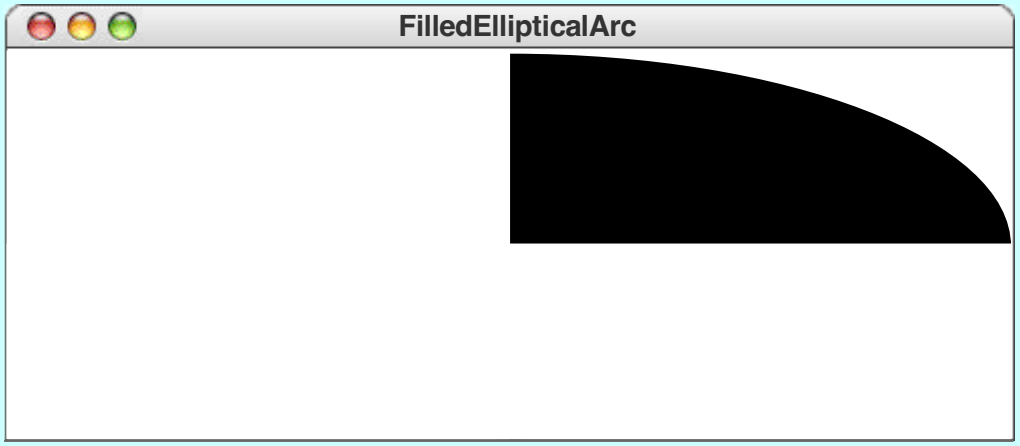

```
let a4 = GArc(d, d, 0, -180);
gw.add(a4, cx - d / 2, cy - d / 2);
```



Filled Arcs

- The **`GArc`** class implements the functions **`setFilled`** and **`setFilledColor`**.
- A filled **`GArc`** is displayed as the pie-shaped wedge formed by the center and the endpoints of the arc, as follows:

```
function FilledEllipticalArc() {
   let gw = GWindow(GWINDOW_WIDTH, GWINDOW_HEIGHT);
   let arc = GArc(0, 0, gw.getWidth(), gw.getHeight(),
                  0, 90);
   arc.setFilled(true);
   gw.add(arc);
}
```



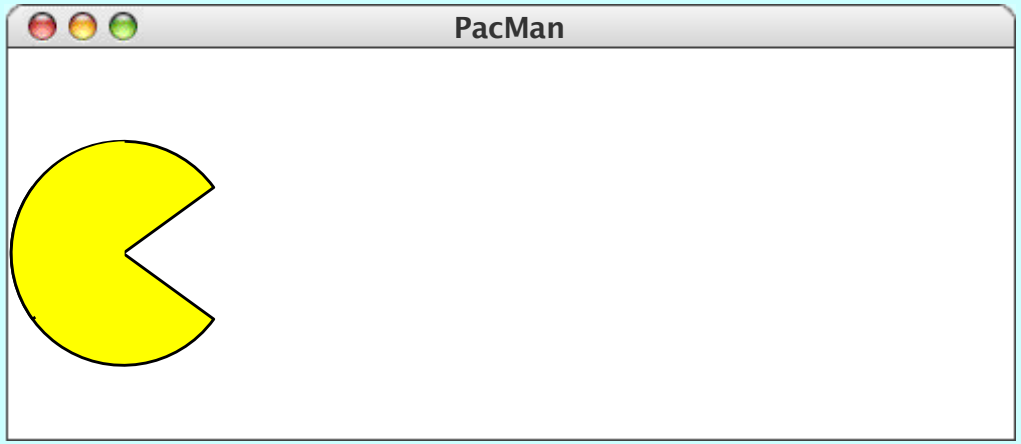
Additional Methods for `GArc`

`setStartAngle(`*start***`)`**	Sets the start angle for the arc
`getStartAngle()`	Returns the start angle for the arc
`setSweepAngle(`*sweep***`)`**	Sets the sweep angle for the arc
`getSweepAngle()`	Returns the sweep angle
`setFrameRectangle(`*x, y, width, height***`)`**	Resets the bounds for the frame

- These methods allow you to animate the appearance of an arc.
- The **`setStartAngle`** and **`setSweepAngle`** methods make it possible to change the starting position and the extent of the arc dynamically.
- The **`setFrameRectangle`** method changes the bounds of the rectangle circumscribing the oval from which the arc is taken.

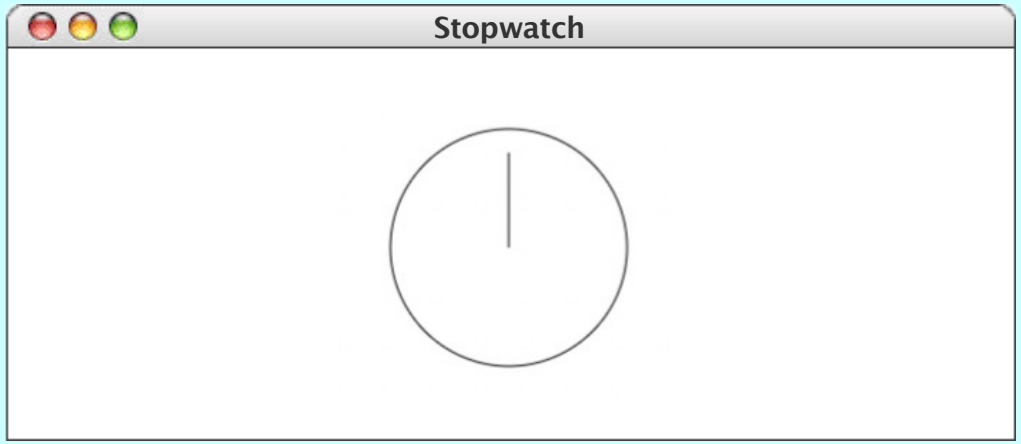
Exercise: PacMan

- Write a program that uses the **`GArc`** class to display a PacMan figure at the left edge of the graphics window.
- Add the necessary timer animation so that PacMan moves to the right edge of the window. As it moves, your program should change the start and sweep angles of the arc so that the mouth appears to open and close.



Exercise: Stopwatch

- Write a simple graphics program that places a stopwatch so that it's centered within a graphics window, as with:



- The stopwatch is stationary until the user clicks the mouse anywhere in the window, at which point the stopwatch's hand rotates at a speed of 360 degrees per minute.
- When the user clicks a second time, the stopwatch stops, but a third click prompts it to continue as if never interrupted.
- Our program requires both mouse and timer events!

The End