

Lecture 10

Chapter 3 Getting Started with Graphics

3.3 Rotating, Scaling, and Flipping

3.4 Cloning

3.3 Rotating, Scaling, and Flipping

By design, all of the Drawable objects can be rotated, scaled, or flipped.

For these operations, the object's reference point is very important (it serves as a point of the shape that stays fixed during the transformation).

Recall that initially

for **Square**, **Circle**, **Rectangle**, or **Text** :

reference point is the **center** of an object,

for **Polygon** and **Path**:

reference point is the location of the **first point**, and

for an **Image**:

reference point is it the **top left corner**.

Rotating

`rotate(angle)` is a method of `Drawable` class

Example:

```
r=Rectangle(100,200,Point(300,350))  
r.setFillcolor('Red')
```

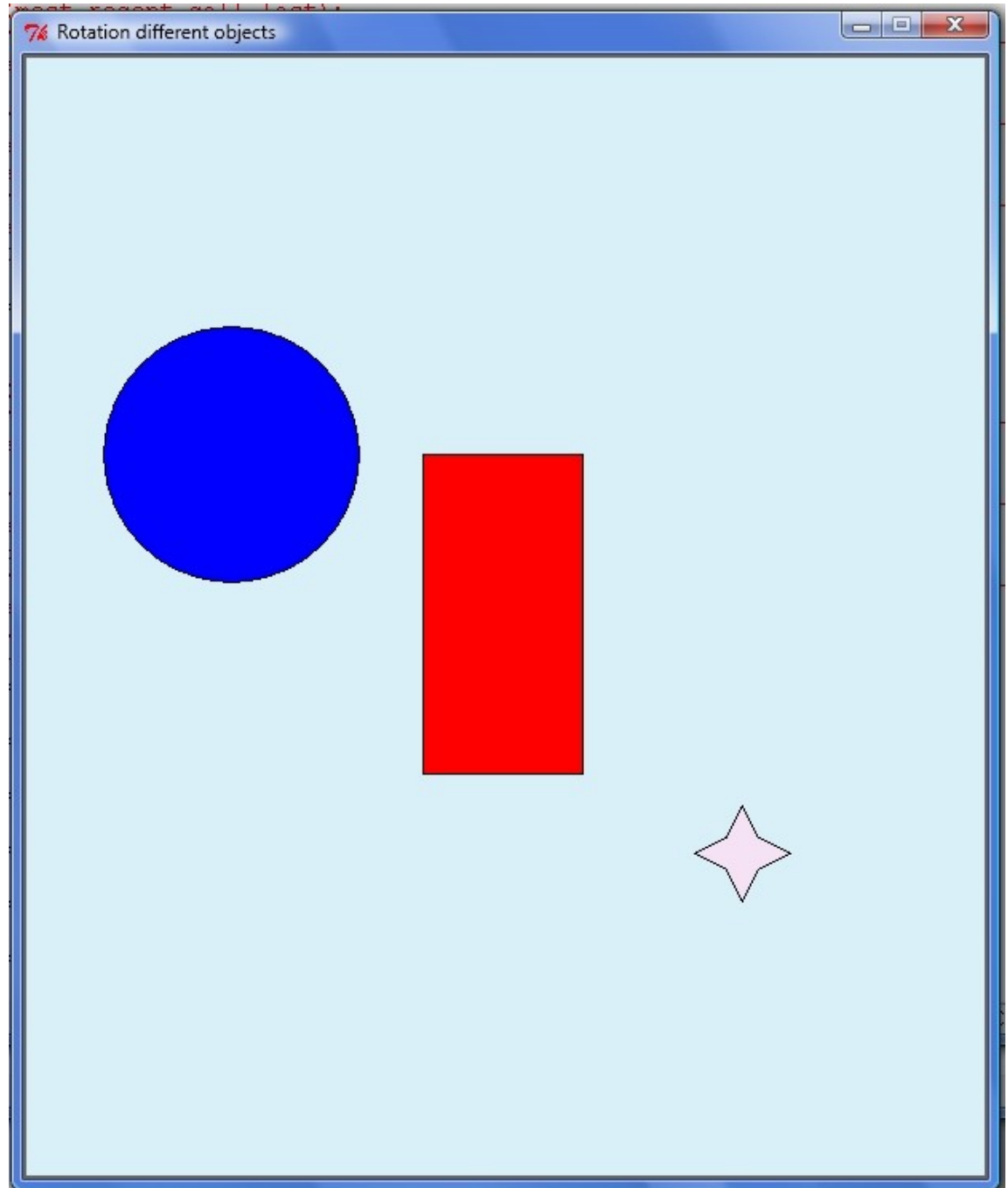
```
c=Circle(80,Point(130,250))  
c.setFillcolor('Blue')
```

```
a,b,c_=Point(10,25),Point(20,20), Point(25,10)  
d,e,f = Point(30,20), Point(40,25), Point(30,30)  
g,h =Point(25,40), Point(20,30)  
p=Polygon(a,b,c_,d,e,f,g,h)  
p.setFillcolor((245,226,245))  
p.move(400,400)
```

```
paper.add(r), paper.add(c), paper.add(p)
```

Rotating

Result:



Rotating

`rotate(angle)` is a method of `Drawable` class (clockwise rotation by `angle` degrees)

Example: next we will add the following loop:

```
for i in range(100):  
    r.rotate(15)  
    c.rotate(15)  
    p.rotate(15)  
    time.sleep(0.25)
```

What can we expect to happen?

See program [rotation-example1.py](#)

We can slightly change program - move the reference point of the circle, see program [rotation-example2.py](#)

! text rotation is not implemented yet

Scaling

- reference point remains fixed, all other points in an object are scaled relative to the reference point

`scale(factor)` is a method of `Drawable` class

Example:

```
r=Rectangle(100,200,Point(300,350))  
r.setFillColor('Red')
```

```
c=Circle(80,Point(130,250))  
c.setFillColor('Blue')  
c.adjustReference(50,10)
```

```
paper.add(r), paper.add(c)
```

```
time.sleep(1)
```

```
r.scale(0.5), c.scale(0.5)
```

```
time.sleep(1)
```

```
r.scale(0.5), c.scale(0.5)
```

What will this code do? See program `scaling-example.py`

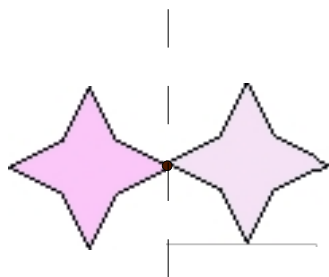
Flipping

- taking a mirror image of an object

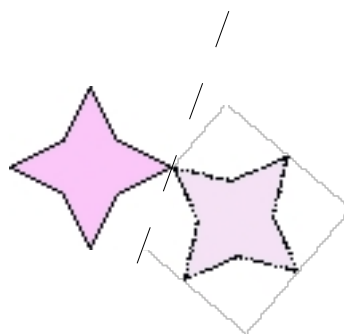
`flip(angle)` is a method of `Drawable` class

- if no argument is given to `flip` method, the flip takes place across a *vertical axis of symmetry*, passing through the reference point.
- parameter `angle` specifies the clockwise rotation of *axis of symmetry* away from vertical.

Example:



`star.flip()`



`star.flip(20)`

Flipping

Example:

```
r=Rectangle(100,200,Point(300,350))
r.setFill('Red')
r.adjustReference(-30,-50)

star=Polygon(a,b,c,d,e,f,g,h)
star.move(400,450)
star.adjustReference(60,0)

r.flip(), star.flip()

time.sleep(2)

r.flip(), star.flip()

time.sleep(3)

r.flip(20), star.flip(20)
```

What can we expect to happen?

See program [flipping-example.py](#)

3.4 Cloning

Drawable types support a convenient `clone` method that returns a **brand new copy**.

The clone has precisely the same settings as the original element, but **is not automatically added to any canvases**.

Example:

```
star=Polygon(a,b,c,d,e,f,g,h)
star.move(400,450)
```

```
star2=star.clone()
star2.move(-100,-50)
star2.scale(2)
paper.add(star2)
```

See program [cloning-example.py](#)

And a more interesting one: [cloning-fun.py](#)

Homework Assignment

HW6 - updated:

Write a program that draws a composite object (i.e. not a square, rectangle, or circle) in the top left corner of the graphics window, then moves this object to the right bottom corner of the window, and then moves it to the right top corner of the window.

During the movement object should change it's color at least 10 times, **flip at least 10 times and rotate at least 10 times**. All movements should be smooth (without jumps).

Afterward in order to notify the user that the program won't do anything else display a text,saying something like "That's it!"

Program must be well-commented.