PROGRAMMING

Lecture 10

Hanbat National University
Dept. of Computer Engineering
Changbeom Choi

OUTLINE

Mutability of objects

High order functions

Graphics revisited

Case study: sun animation

MUTABILITY OF OBJECTS

An **object** has its **state** and **actions**.

type	**state**	**actions**
string	characters, length	count, find, strip
Robot	position, orientation. # of beepers carried	move, turn left, drop, pick up, check conditions
Circle	radius, position, fill color, depth	change position, color, size

Objects whose **states** can **never change** are called **immutable** e.g., **strings** and **tuples**. **Objects** whose states can **change** are called **mutable**, e.g., **robots**, **photos**, and **graphic objects**.

An **object** may have more than one name for the same object. In this case, be careful if it is **a mutable** object!

```
sun = Circle(30)
sun.setFillColor("dark orange")
moon = sun
moon.setFillColor("wheat")
print (sun.getFillColor())
```

What will be printed?

HIGH ORDER FUNCTIONS

A **function** is an **object** of **type function**:

```
def f(x):
    return math.sin(x / 3.0 + math.pi/4.0)
print (f) → <function f at 0xb7539a3c>
print (type(f)) → <class 'function'>
```

A **function** itself can be used as an **argument !**

A high order function !!

```
def eval_f(x):
# evaluating f(x) = sin ( x / 3 +π /4)
    return math.sin(x / 3.0 + math.pi/4.0)
def print_table(func, x0, x1, step):
    x = x0
    while x <= x1:
        print (x, func(x))
        x += step
import math
print_table(eval_f,  -math.pi,  3 * math.pi,  math.pi/8)
```

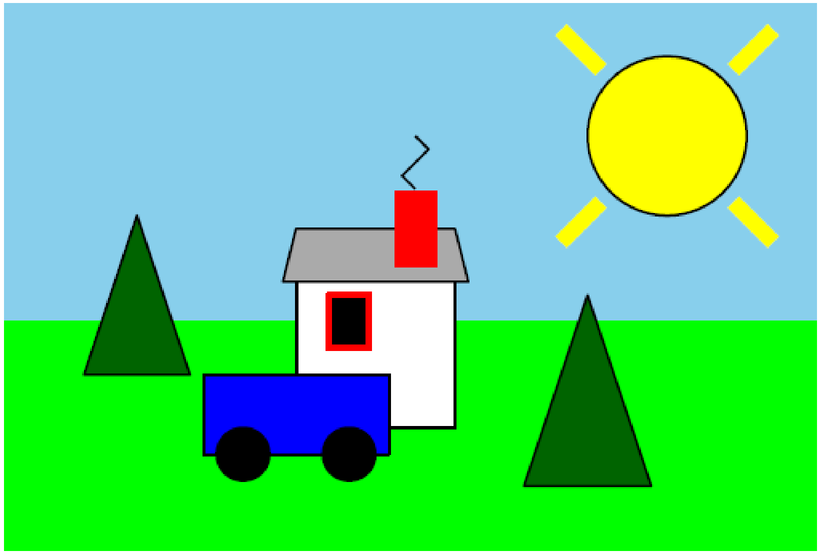
a high order function

-π 3π π/8

GRAPHICS REVISITED

Our task is to draw the following **color picture** which consist of the **sun**, a **house** , two **trees**, and a **car**.

Reference: http://www.cs1graphics.org/



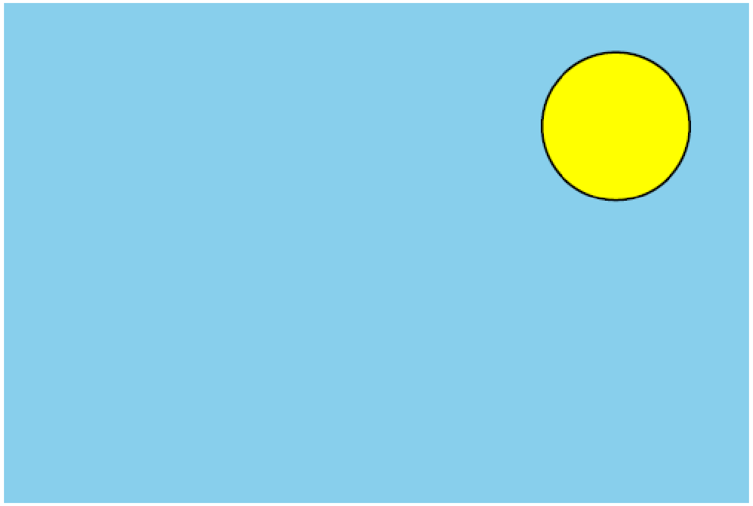
Canvas initialization

```
from cs1graphics import *
paper = Canvas(300, 200, 'skyBlue',
                   'My World')
```

```
Paper = Canvas()
paper.setWidth(300)
paper.setHeight(200)
paper.setBackgroundColor(“skyBlue”)
paper.setTitle(“My World”)
```

Drawing the sun

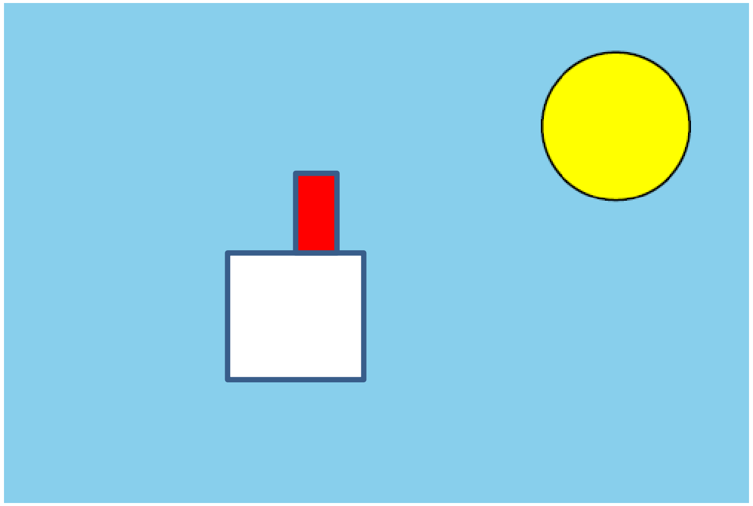
```
sun = Circle()
sun.setRadius(30)
sun.moveTo(250,50)
paper.add(sun)
sun.setFillColor(“yellow”)
```



```
sun = Circle(30, Point(250, 50))
```

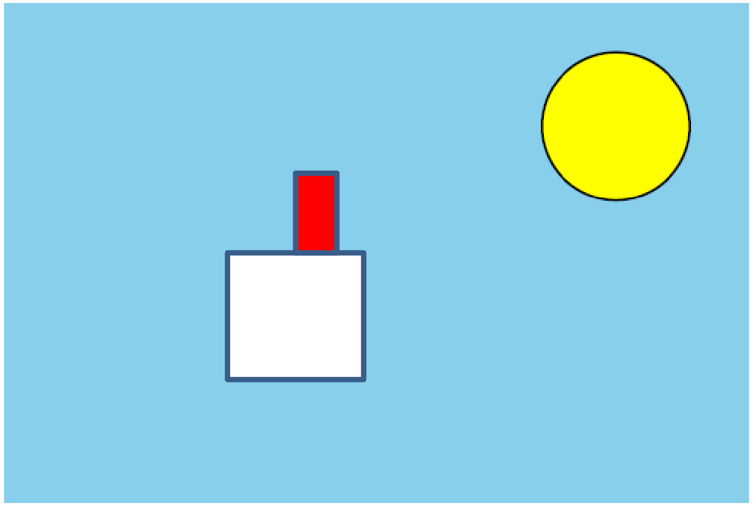
Drawing a house

```
facade = Square(60, Point(140,130))
facade.setFillColor('white')
paper.add(facade)
```



Drawing a chimney

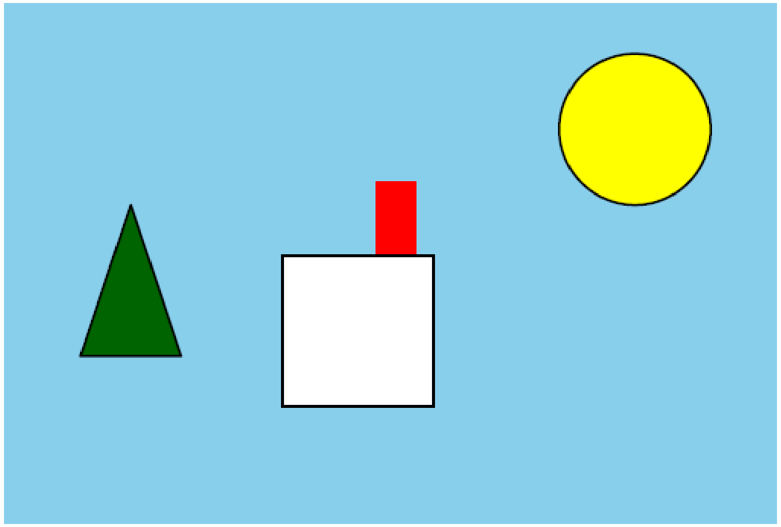
```
chimney = Rectangle(15, 28, Point(155,85))
chimney.setFillColor('red')
paper.add(chimney)
```



DRAWING A TREE

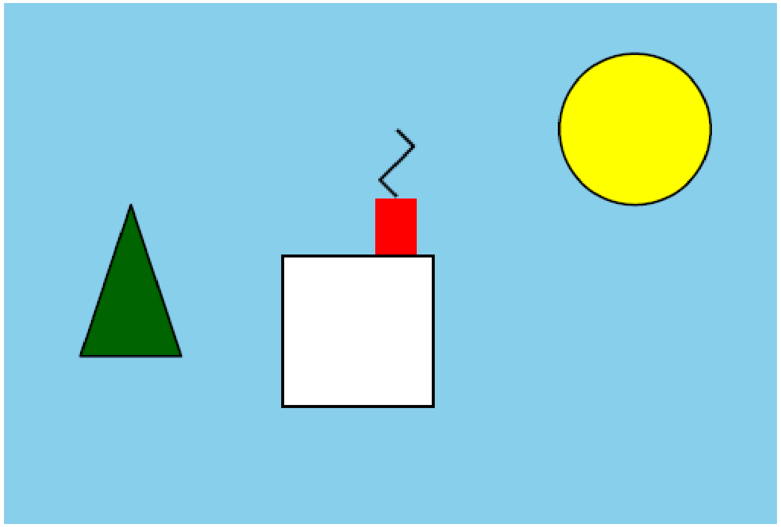
```
tree = Polygon(Point(50,80),Point(30,140),Point(70,140))
tree.setFillColor(“darkGreen”)
paper.add(tree)
```

```
tree = Polygon()
tree.addPoint(Point(50,80))
tree.addPoint(Point(30,140))
tree.addPoint(Point(70,140))
```

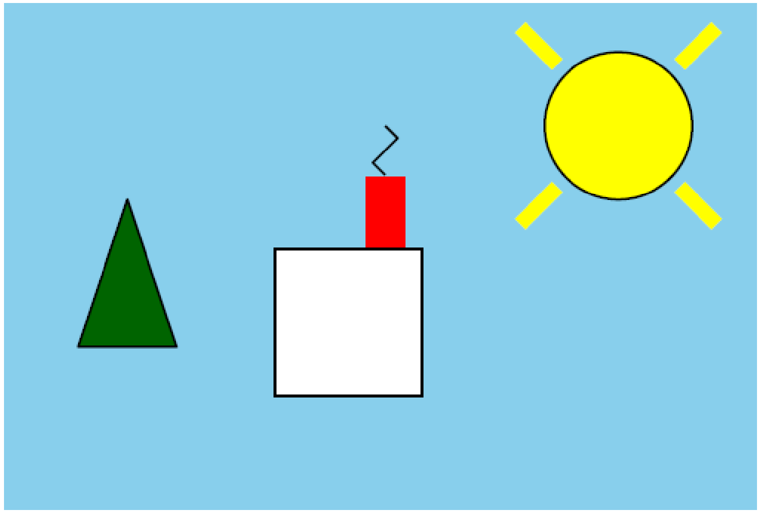


DRAWING SMOKE AND SUN RAYS

```
smoke = Path(Point(155,70), Point(150,65),
        Point(160,55), Point(155,50))
paper.add(smoke)
```



```
sunraySW = Path(Point(225,75), Point(210,90))
sunraySW.setBorderColor('yellow')
sunraySW.setBorderWidth(6)
paper.add(sunraySW)
```



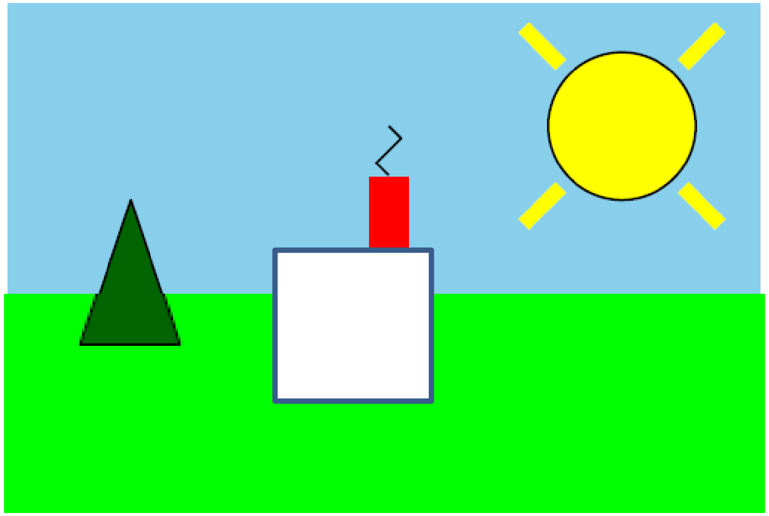
sunraySE,sunrayNW, and sunrayNE can be drawn in a similar manner.

```
sunraySE = Path(Point(275,75), Point(290,90))
sunrayNE = Path(Point(275,25), Point(290,10))
sunrayNW = Path(Point(225,25), Point(210,10))
```

DRAWING GRASS

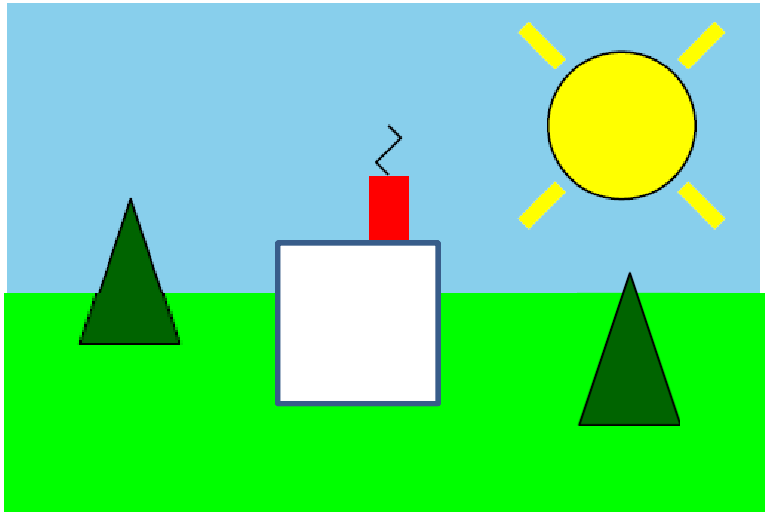
```
grass = Rectangle(300, 80, Point(150,160))
grass.setFillColor('green')
grass.setBorderColor('green')
grass.setDepth(75) # must be behind house and tree
paper.add(grass)
```

What are the depths of the house and the tree?



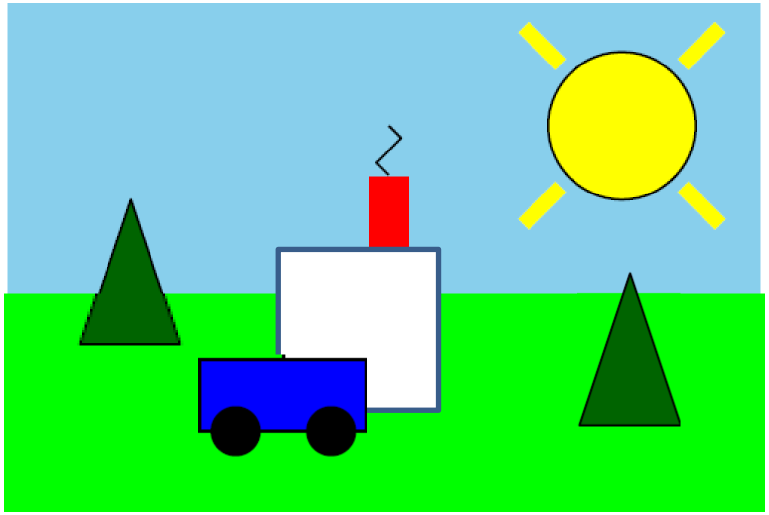
CLONING A TREE

```
otherTree = tree.clone( )
otherTree.moveTo(170,30)
otherTree.scale(1.2)
paper.add(otherTree)
```



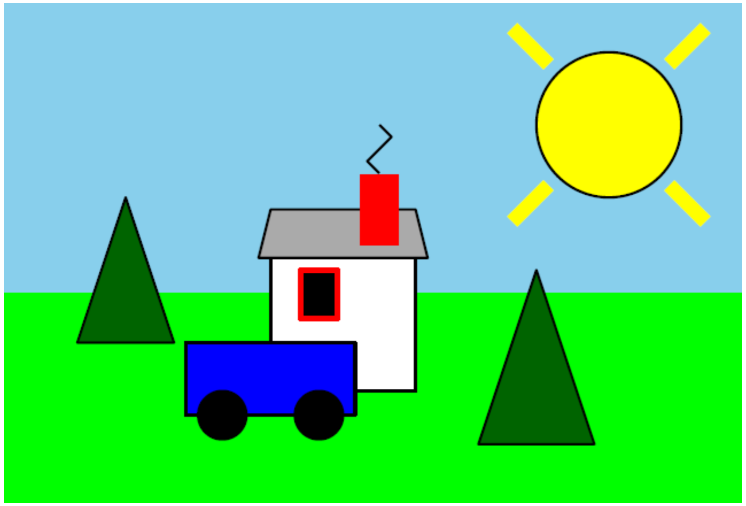
Adding a car

Refer to your previous lecture



PROBLEM 1: PICTURE COMPLETION

Add a **roof** and a **window** to complete the following picture and make the car move across the grass yard. You should also implement the functions that draw the components of the picture, such as the sun, the house, the car, and trees, respectively. A function may include one or more other functions.



CASE STUDY: SUN ANIMATION

According to an old story, the sun rises from an east mount and moves west during the day. After a long journey in the sky, it goes down over a west mount to enter into the earth and gets back to the east mount during the night so that it rises again from the mount in the next morning. This old story can be animated using **trigonometric functions**, **sin** and **cos**.

```
def animate_sunrise(sun):
    w = canvas.getWidth()
    h = canvas.getHeight()
    r = sun.getRadius()
    x0 = w / 2.0
    y0 = h + r
    max_x  = w / 2.0 - r
    max_y  = h
    for angle in range(361):
        rad = (math.pi/180.0) * angle
        x = x0 - max_x * math.cos(rad)
        y = y0 - max_y * math.sin(rad)
        sun.moveTo(x, y)
```

```
for angle in range(361):
    rad = (math.pi/180.0) * angle
    x = x0 – max_x * math.cos(rad)
    y = y0 – max_y * math.sin(rad)
    sun.moveTo(x, y)
```

$0 \leq angle \leq \pi/2$

x is increasing(west)

y is decreasing(up)

$\pi/2 \leq angle \leq \pi$

x is increasing(west)

y is increasing(down)

$\pi \leq angle \leq 3\pi/2$

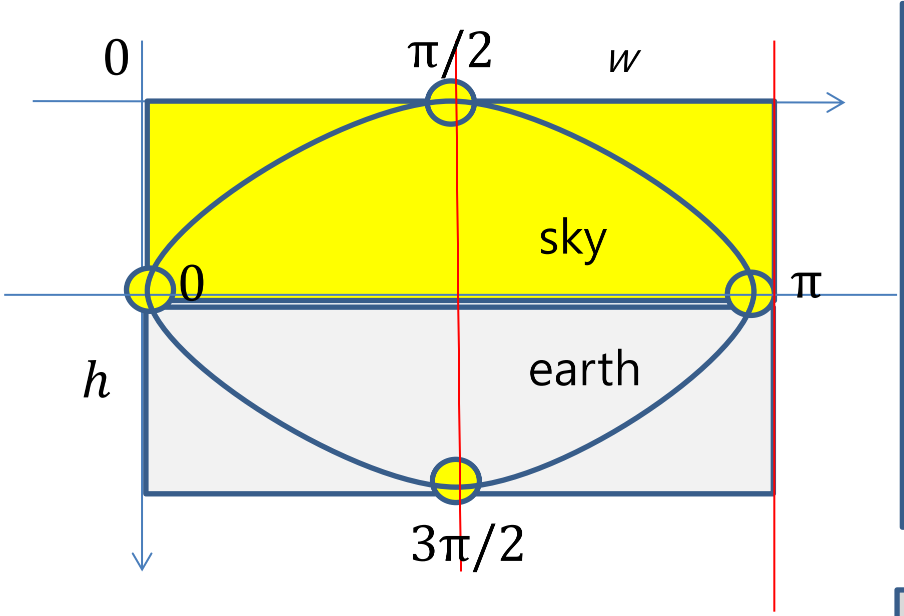
x is decreasing(east)

y is increasing(down)

$3\pi/2 \leq angle \leq 2\pi$

x is decreasing(east)

y is decreasing(up)



$0 \le angle \le \pi/2$

x is increasing(west)

y is decreasing(up)

$\pi/2 \le angle \le \pi$

x is increasing(west)

y is increasing(down)

day

$\pi \le angle \le 3\pi/2$

x is decreasing(east)

y is increasing(down)

$3\pi/ \le angle \le 2\pi$

x is decreasing(east)

y is incereasing(up)

night

COLOR INTERPOLATION

PROBLEM 2: SUN ANIMATION

The color of sky changes from time to time. For example,

In the early morning, the sky is dark grey. and in the mid-day, it becomes blue in a clear day. So does the color of the sun. Your task is to complete the sun animation program taught in this lecture through color interpolation of the sky and sun. You may choose you own colors for the sky and sun. Please try to implement your program using functions.

Color interpolation

1.Get a color name. Useful information on color names is available in the following web site:

http://cloford.com/resources/colours/500col.htm

(500+ Named Colors with rgb and hex values)

2. Use the following function to convert a color name to its equivalent (r, g, b) value:

Color(color name).getColorValue()

3. Perform color interpolation as you learned in the previous lecture.