

# Translation/Rotation

14.1 & 14.5

# Translate/Rotate

- `translate()` is a clear and simple way to relocate or replicate portions of a drawing
- `rotate()` makes possible some drawing that would be very difficult without it

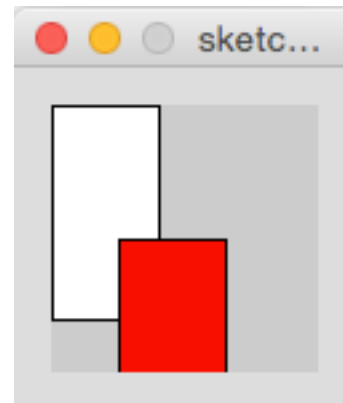
```
// notice both rectangles are "drawn at (0,0)"
```

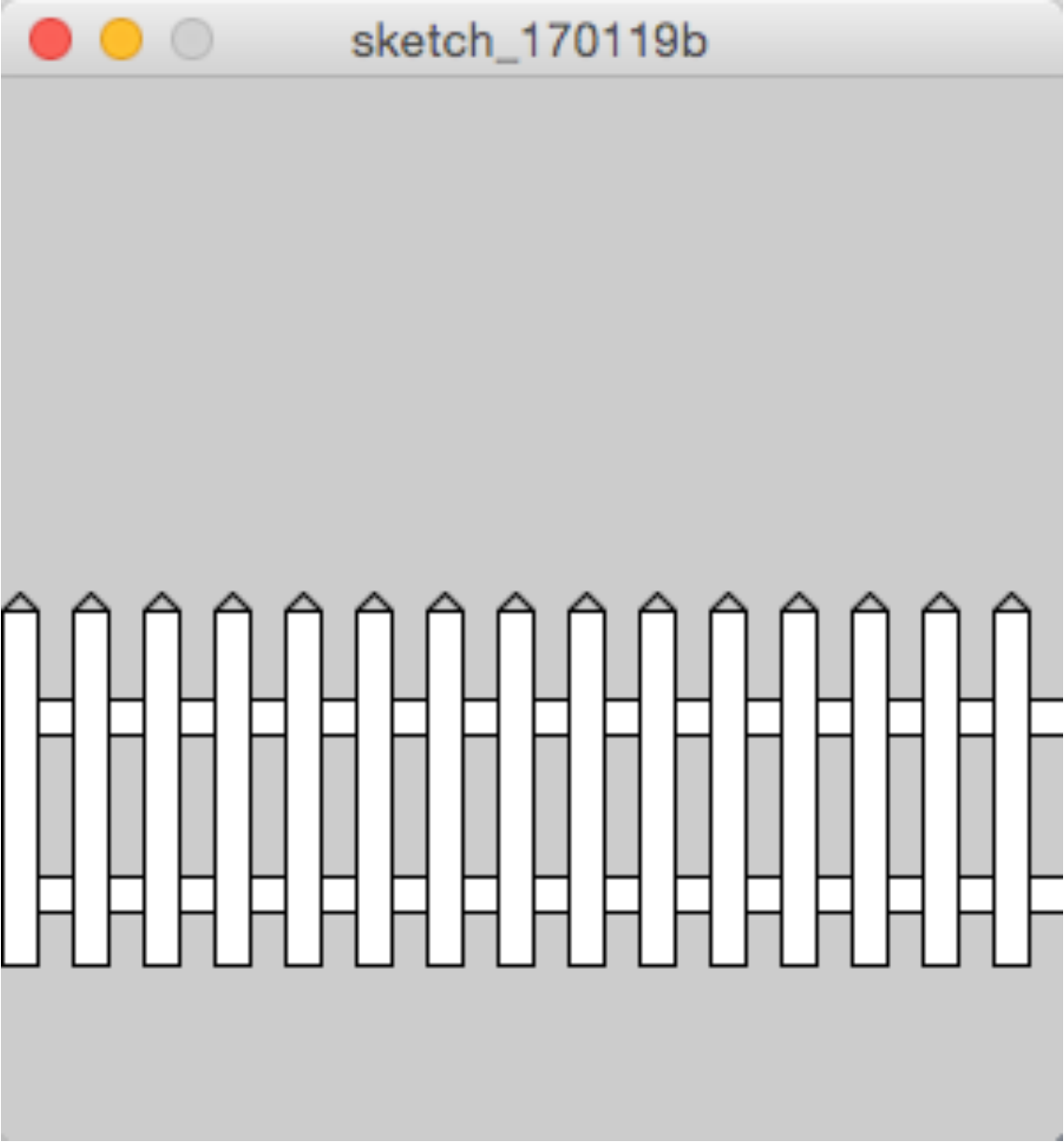
```
rect(0,0,40,80);
```

```
fill(255,0,0);
```

```
translate(25,50);
```

```
rect(0,0,40,80);
```





```
// move to the center of the left edge
translate(0,height/2);
int picketWidth = 10;
int picketHeight = 100;

// draw the top rail, as wide as a single picket
rect(0,picketHeight/4, width, picketWidth);

// draw the bottom rail
rect(0,picketHeight*3/4, width, picketWidth);

// draw one picket
rect(0,0,picketWidth,picketHeight);
line(0,0,picketWidth/2,-picketWidth/2);
line(picketWidth,0, picketWidth/2,-picketWidth/2);

// move over and draw another picket
translate(picketWidth*2, 0);
rect(0,0,picketWidth,picketHeight);
line(0,0,picketWidth/2,-picketWidth/2);
line(picketWidth,0, picketWidth/2,-picketWidth/2);
```

```
// move to the center of the left edge
translate(0,height/2);
int picketWidth = 10;
int picketHeight = 100;
. . .
// move over and draw another picket
translate(picketWidth*2, 0);
rect(0,0,picketWidth,picketHeight);
line(0,0,picketWidth/2,-picketWidth/2);
line(picketWidth,0, picketWidth/2,-picketWidth/2);

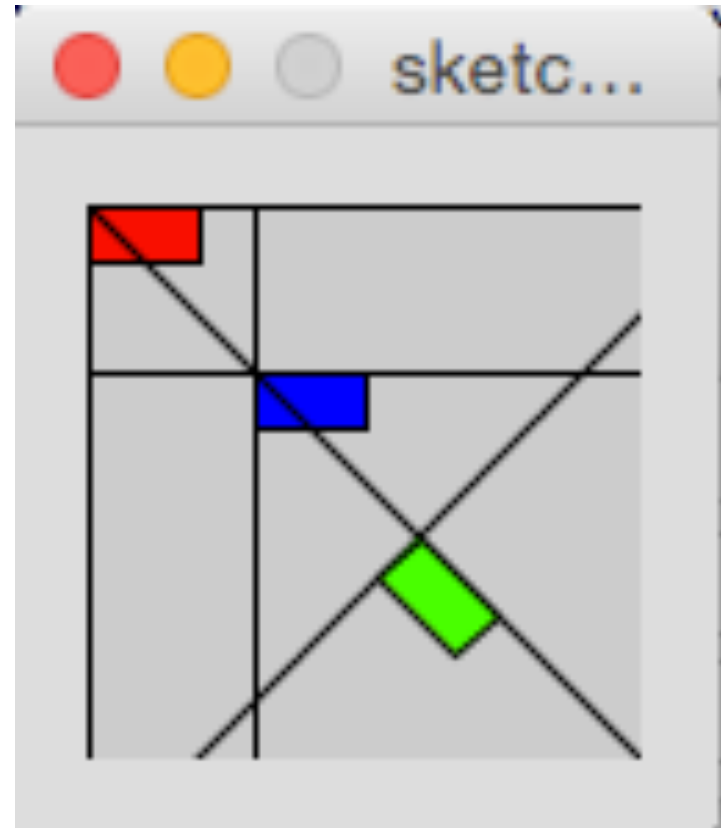
// move over and draw another picket
translate(picketWidth*2, 0);
rect(0,0,picketWidth,picketHeight);
line(0,0,picketWidth/2,-picketWidth/2);
line(picketWidth,0, picketWidth/2,-picketWidth/2);
// repeat as necessary
```

```
// move to the center of the left edge
translate(0,height/2);
int picketWidth = 10;
int picketHeight = 100;
. . .
// looking ahead - doing it with a loop
int picketPosition = 0;
while( picketPosition < width) {
  rect(0,0,picketWidth,picketHeight);
  line(0,0,picketWidth/2,-picketWidth/2);
  line(picketWidth,0, picketWidth/2,-picketWidth/2);
  translate(picketWidth*2, 0);
  picketPosition = picketPosition + picketWidth*2;
}
```

```
fill(255,0,0); // red
line(-100,0,100,0);
line(0,-100,0,100);
rect(0,0,20,10);

translate(30,30);
fill(0,0,255); // blue
line(-100,0,100,0);
line(0,-100,0,100);
rect(0,0,20,10);

translate(30,30);
rotate(PI/4);
fill(0,255,0); // green
line(-100,0,100,0);
line(0,-100,0,100);
rect(0,0,20,10);
```





```
void setup() {  
  size(400,400);  
}  
  
void draw() {  
  translate(width/2, height/2);  
  rotate(mouseX);  
  rect(0, 0, 100, 100);  
}
```

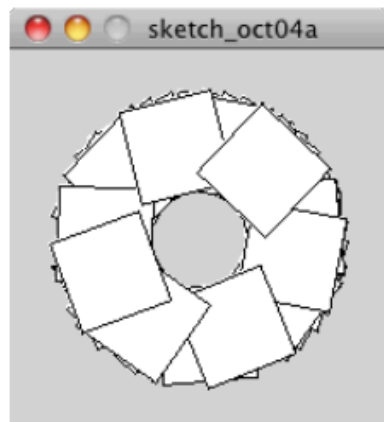
A



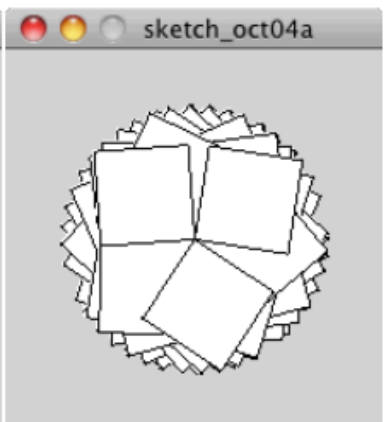
B



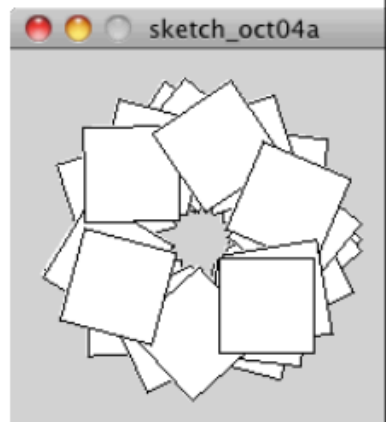
C



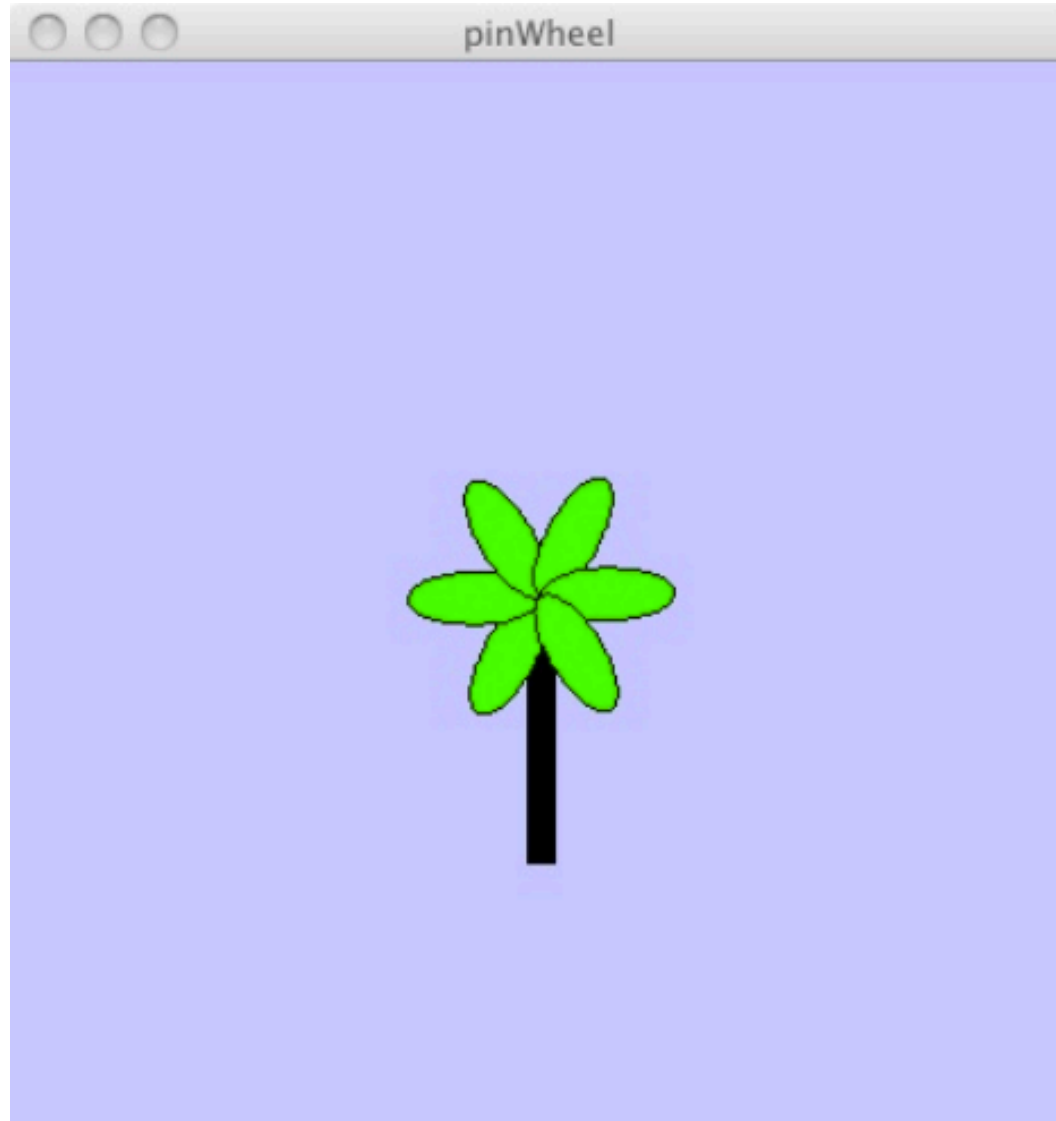
D



E



Write a program that draws this



```
void setup() {  
  size(200,200);  
  rectMode(CENTER);  
}  
void draw() {  
  translate(width/2, height/2);  
  rotate(mouseX);  
  translate(50,0);  
  rotate(PI/4);  
  rect(0, 0, 50, 50);  
}
```

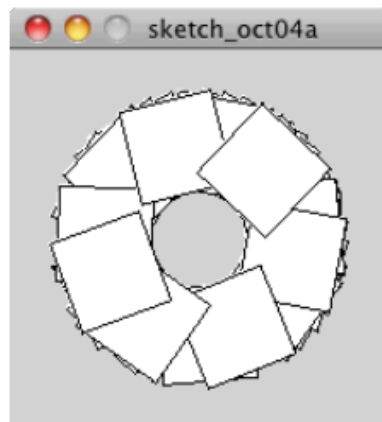
A



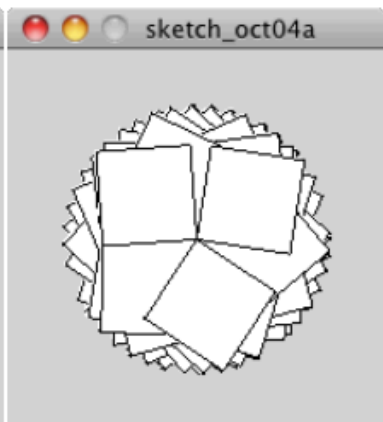
B



C



D



E

