# *Symmetries of an Equilateral Triangle*



written by Dorothy Zeiser (zeiser.d@gmc.edu)

## Outline

1. Investigate the symmetries of an equilateral triangle visually.
2. Investigate the composition of two transformations visually.

## Introduction

There are three rotations and three reflections that leave an equilateral triangle occupying the same region in the plane as was occupied by the original triangle. These rigid motions are called the symmetries of the equilateral triangle.

## Define and Plot an Equilateral Triangle

```
> restart:
  with(plots):with(plottools):
Warning, the name changecoords has been redefined
Warning, the assigned name arrow now has a global binding
```

Define the sides of an equilateral triangle.

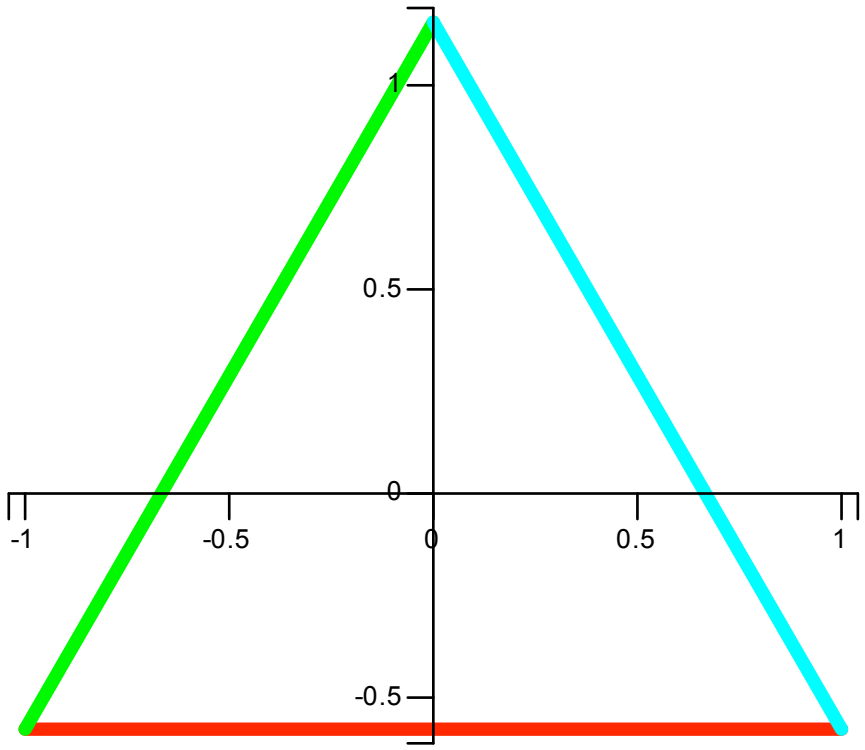
```
> L1 := line([1, -sqrt(3)/3], [0, 2*sqrt(3)/3], color = cyan,
  thickness = 5):
  L2 := line([0, 2*sqrt(3)/3], [-1, -sqrt(3)/3], color = green,
  thickness = 5):
  L3 := line([-1, -sqrt(3)/3], [1, -sqrt(3)/3], color = red,
  thickness = 5):
```

Define tickmarks for displays.

```
> TM := [[-1,-.5,0,.5,1],[-1,-.5,0,.5,1]]:
```

Name the figure so that we can deal with it as an object.

```
> Triangle := display(L1,L2,L3,scaling=constrained,tickmarks=TM):
  Triangle;
```




**Equilateral Triangle with center at the origin and vertical altitude on the y-axis**

>

# Symmetries of an Equilateral Triangle

## Rotations

Define a function to perform rotations..

```
> R := t -> rotate(Triangle,t*Pi/144,[0,0]):
```

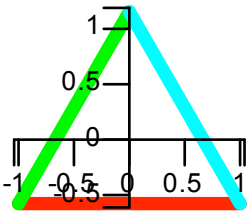
Define lists of rotations for animations below.

```
> R120 := seq(R(t), t=0..96):
  R240 := seq(R(t), t=0..192):
```

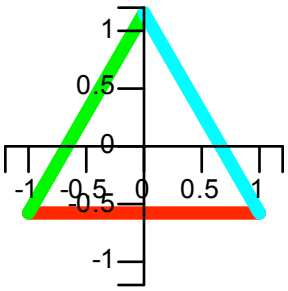
Execute the first column below. Use tab to get to the second column. Execute. Click on the figure. The Context Bar for Animations will appear near the top of the screen. Click on play button to view the animated rotation. Also execute and animate the third column.

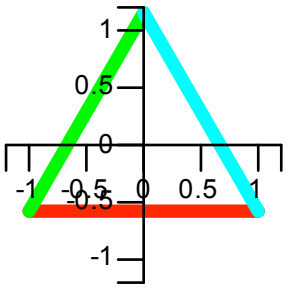
| | | |
|---|---|---|
| `Triangle;` | `display(R120,` | `display(R240,` |

```
tickmarks=TM,
insequence=true);
```




```
tickmarks=TM,
insequence=true);
```




| *Rotation of* $0^o$ | *Rotation of* $120^o$ | *Rotation of* $240^o$ |
|---|---|---|

# Reflections

## *Reflection in the vertical altitude*

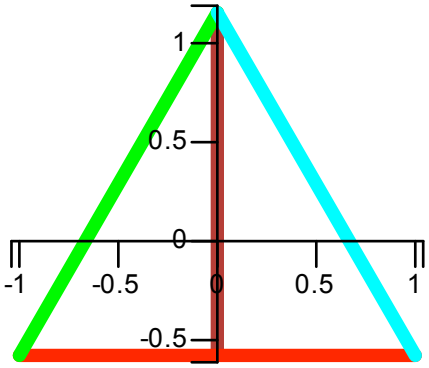
Add axis of reflection to the triangle.

```
> A1 := line([0, -sqrt(3)/3], [0, 2*sqrt(3)/3], color =
  brown, thickness = 5):
  A1Triangle := display(L1,L2,L3,A1,tickmarks=TM,scaling=
  constrained):
```

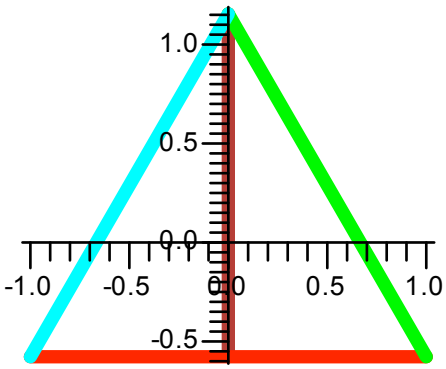
Reflect in the line containing the vertical axis.

```
A1Triangle;
```

```
reflect(A1Triangle,[[0,2*
sqrt(3)/3],[0,-sqrt(3)/3]]
);
```




Triangle in Initial Position




Reflected in Line containing the Vertical Altitude

## *Reflection in altitude from lower left vertex*

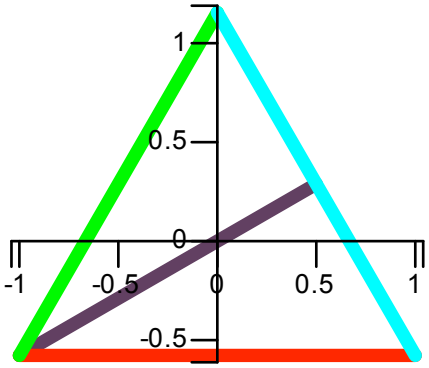
Add axis of reflection to the triangle.

```
> A2 := line([-1,-sqrt(3)/3],[1/2,sqrt(3)/6], color=violet,
  thickness=5):
  A2Triangle := display(L1,L2,L3,A2,tickmarks=TM,scaling=
  constrained):
```

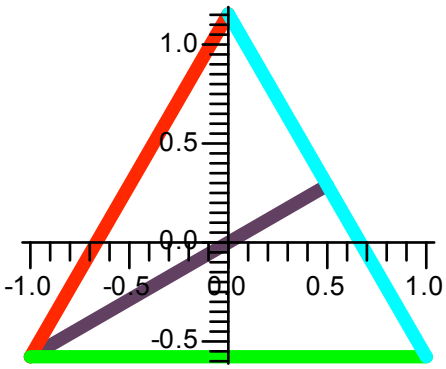
Reflect in altitude from lower left vertex.

```
A2Triangle;
```

```
reflect(A2Triangle,[[-1,-
sqrt(3)/3],[1/2,sqrt(3)/6]
]);
```




Triangle in Initial Postion




Reflected in Line containing Altitude from Lower Left Vertex

## *Reflection in altitude from lower right vertex*

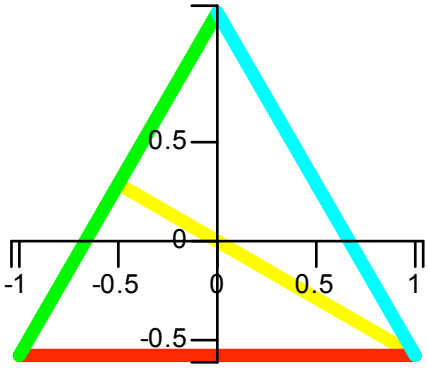
Add axis of reflection to the triangle.

```
> A3 := line([1,-sqrt(3)/3],[-1/2,sqrt(3)/6], color=yellow,
  thickness=5):
  A3Triangle := display(L1,L2,L3,A3,tickmarks=[[-1,-.5,0,.5,
  1],[-1,-.5,0,.5,1]],scaling=constrained):
```

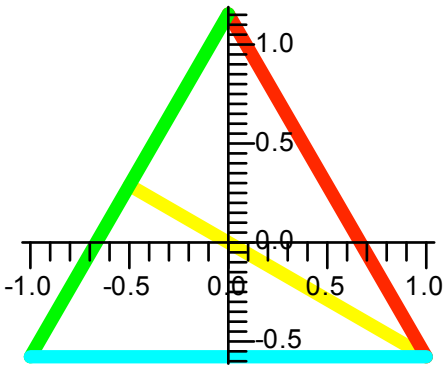
Reflect in altitude from lower right vertex.

```
A3Triangle;
```

```
reflect(A3Triangle,[[1,-
sqrt(3)/3],[-1/2,sqrt(3)
/6]]);
```




Triangle in Initial Postion




Reflected in Line containing Altitude from Lower Right Vertex

## Composition of Transformations

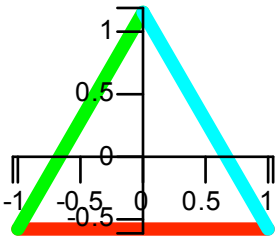
### Example 1

***Reflection in Line containing Altitude from Left Vertex followed by Reflection in the y-axis***

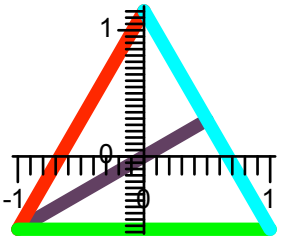
```
Triangle;
```

```
reflect
(A2Triangle,[[-1,
-sqrt(3)/3],[1/2,
sqrt(3)/6]]);
```

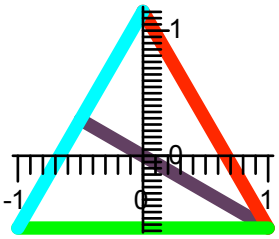
```
reflect(reflect
(A2Triangle,[[-1,
-sqrt(3)/3],[1/2,
sqrt(3)/6]]),[
[0,2*sqrt(3)/3],
[0,-sqrt(3)/3]])
;
```




Triangle in Initial Postion




Reflected in Lower Left Altitude




Reflected in Vertical Altitude

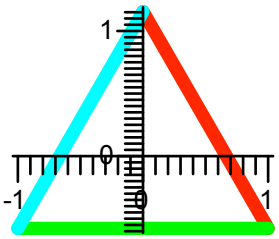
## Example 2

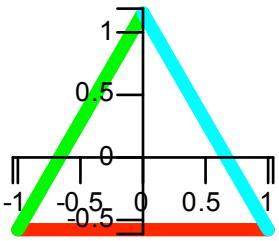
***Rotation of* 240° *followed by Rotation of* 240°**

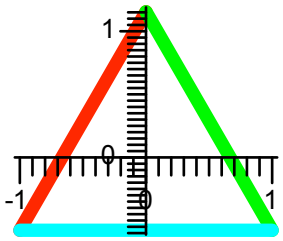
```
Triangle;
```

```
rotate(Triangle,
4*Pi/3,[0,0]);
```

```
rotate(rotate
(Triangle,4*
Pi/3,[0,0]),4*
Pi/3,[0,0]);
```





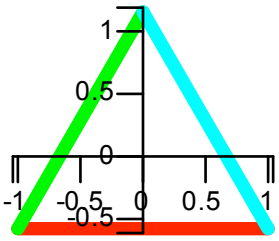




| Triangle in Initial Position | Rotated $240^o$ | Rotated another $240^o$ |
|---|---|---|

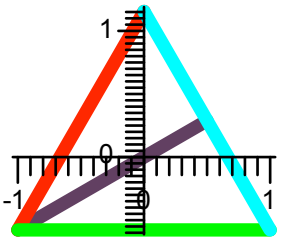
## Example 3

***Reflection in Line containing Altitude from Left Vertex followed by Rotation of*** $120^o$

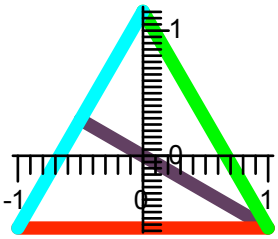
| | | |
|---|---|---|
| `Triangle;` | `reflect (A2Triangle,[[-1, -sqrt(3)/3],[1/2, sqrt(3)/6]]);` | `rotate((reflect (A2Triangle,[[-1, -sqrt(3)/3],[1/2, sqrt(3)/6]])),2* Pi/3,[0,0]);` |




Triangle in Initial Postion




Reflected in Lower Left Altitude




Rotated $120^o$

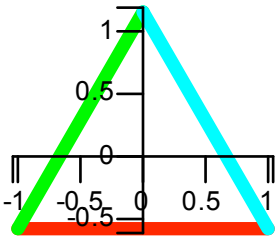
## Example 4

***Rotation of* $120^o$**
***followed by***
***Reflection in Line containing***
***Altitude from Left Vertex***

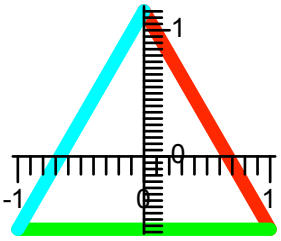
```
Triangle;
```

```
rotate(Triangle,
2*Pi/3,[0,0]);
```

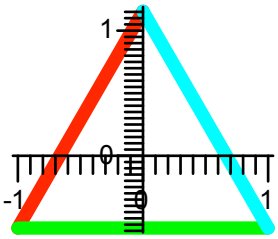
```
reflect(rotate
(Triangle,2*
Pi/3,[0,0]),[[0,
2*sqrt(3)/3],[0,-
sqrt(3)/3]]);
```




Triangle in Initial Postion



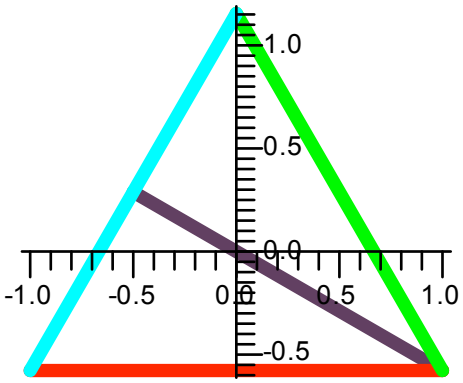

Rotated $120^o$



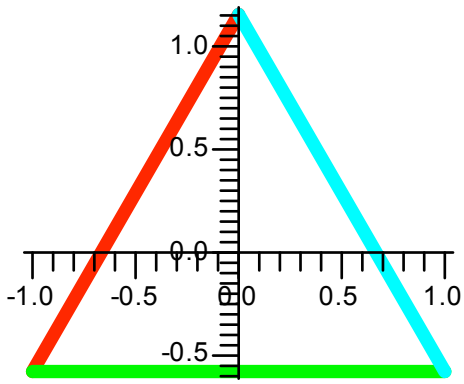

Reflect in Vertical Altitude

## Comparison of the Results of Examples 3 and 4

```
rotate((reflect(A2Triangle,[
[-1,-sqrt(3)/3],[1/2,sqrt
(3)/6]])),2*Pi/3,[0,0]);
```




```
reflect(rotate(Triangle,2*
Pi/3,[0,0]),[[0,2*sqrt(3)/3]
,[0,-sqrt(3)/3]]);
```

| | |
|---|---|
| |  |
| Reflection in the y-axis followed by rotatation of $120^o$ | Rotation of $120^o$ followed by reflection in the y-axis |