Understanding the Game Loop

# Overview:

The Update and Draw methods are looped in an XNA Windows Game at 60 times a second by default. It is sometimes hard to understand how this game loop is responsible for everything that occurs in a game because it cycles so quickly. Let’s break the game loop down by examining what happens during just one-sixth (10/60) of a second. Trace the values of RGB during the execution of the following code samples.  
 **Directions:**

1. Trace the values of r, g, and b during the following algorithms and record the values in the charts.
2. Assume the following global (class) declarations: byte r, g, b;

|  |  |  |  |
| --- | --- | --- | --- |
| Seconds | r | g | b |
| 1/60 | 1 | 1 | 1 |
| 2/60 | 2 | 2 | 2 |
| 3/60 | 3 | 3 | 3 |
| 4/60 | 4 | 4 | 4 |
| 5/60 | 0 | 5 | 5 |
| 6/60 | 1 | 6 | 6 |
| 7/60 | 2 | 7 | 7 |
| 8/60 | 3 | 8 | 8 |
| 9/60 | 4 | 9 | 9 |
| 10/60 | 0 | 10 | 10 |

1. Update method:

r++;

g++;

b++;

if (r == 5)

r = 0;

Draw method:

Color aColor = new Color(r, g, b);

graphics.GraphicsDevice.Clear(aColor);

|  |  |  |  |
| --- | --- | --- | --- |
| Seconds | r | g | b |
| 1/60 | 2 | 3 | 4 |
| 2/60 | 4 | 6 | 8 |
| 3/60 | 6 | 9 | 12 |
| 4/60 | 8 | 12 | 16 |
| 5/60 | 10 | 15 | 0 |
| 6/60 | 12 | 18 | 4 |
| 7/60 | 14 | 21 | 8 |
| 8/60 | 16 | 24 | 12 |
| 9/60 | 18 | 27 | 16 |
| 10/60 | 20 | 30 | 20 |

1. Update method:

r = r + 2;

g = g + 3;

b = b + 4;

if (b == g + 5)

b = 0;

Draw method:

Color aColor = new Color(r, g, b);

graphics.GraphicsDevice.Clear(aColor);

|  |  |  |  |
| --- | --- | --- | --- |
| Seconds | r | g | b |
| 1/60 | 10 | 10 | 10 |
| 2/60 | 20 | 20 | 20 |
| 3/60 | 30 | 30 | 30 |
| 4/60 | 40 | 40 | 40 |
| 5/60 | 50 | 50 | 50 |
| 6/60 | 100 | 100 | 100 |
| 7/60 | 110 | 110 | 110 |
| 8/60 | 120 | 120 | 120 |
| 9/60 | 130 | 130 | 130 |
| 10/60 | 140 | 140 | 140 |

1. Write the code that creates the following data values:

Update method:

r = r+10

g = g+10

b = b+10

if(r == 50 && g == 50 && g == 50)

r = r + 50

g = g + 50

b = b + 50

Draw method:

Color aColor = new Color(r, g, b);

graphics.GraphicsDevice.Clear(aColor);

1. Create your own question by writing a code segment or adding data values to the chart. Ask another student to solve it. Compare the answer to your solution.

|  |  |  |  |
| --- | --- | --- | --- |
| Seconds | r | g | b |
| 1/60 | 5 | 5 | 5 |
| 2/60 | 10 | 10 | 10 |
| 3/60 | 15 | 15 | 15 |
| 4/60 | 115 | 35 | 65 |
| 5/60 | 120 | 40 | 70 |
| 6/60 | 125 | 45 | 75 |
| 7/60 | 130 | 50 | 80 |
| 8/60 | 135 | 55 | 85 |
| 9/60 | 140 | 60 | 90 |
| 10/60 | 145 | 65 | 95 |

Update method:

r = r+5

g = g+5

b = b+5

if(r == 20 && g == 20 && g == 20)

r = r + 100

g = g + 20

b = b + 50

Draw method:

Color aColor = new Color(r, g, b);

graphics.GraphicsDevice.Clear(aColor);