

Ameer Holmes 2016 Capstone Timeline

<u>Description</u>	<u>Code</u>
<u>April 5.</u> <b>Code Entry</b>  Makeshift RGB cam / Figuring out how to take color output from Camera  <i>RBG Trip Cam</i>	<pre> import processing.video.*; Capture cam; void setup() {     size(1200, 700);     background(178, 212, 223);     String[] cameras = Capture.list();     if (cameras.length == 0) {         println("There are no cameras available for capture.");         exit();     } else {         println("Available cameras:");         for (int i = 0; i &lt; cameras.length; i++) {             println(cameras[i]);         }         cam = new Capture(this, cameras[0]);         cam.start();     } }  void draw() {     cam.updatePixels();     if (cam.available() == true) {         cam.read();     }     image(cam, 0, 0);     cam.loadPixels();     for (int i=0; i&lt;cam.pixels.length; i++)     {         int c=cam.pixels[i];         int r=(c&gt;&gt;16)&amp;0x000000FF;         int g=(c&gt;&gt;8)&amp;0x000000FF;         int b=c&amp;0x000000FF;         if (r&gt;g &amp;&amp; r&gt;b)         {             r=255;             g=0;             b=0;         } else if (g&gt;r &amp;&amp; g&gt;b)         {             r=0;             g=255;             b=0;         } else if (b&gt;r &amp;&amp; b&gt;g)         {             r=0;             g=0;             b=255;         } else         {             r=178;             g=212;             b=223;         }         cam.pixels[i]=(r&lt;&lt;16)   (g&lt;&lt;8)   b;     }     cam.updatePixels(); } </pre>
<u>April 6.</u> <b>Code Entry</b>	<pre> import processing.video.*;  Capture cam; </pre>

Fixing the  
trippy part  
about RGB cam

RGB cam

```
void setup() {
    size(1200, 700);
    background(178, 212, 223);
    String[] cameras = Capture.list();
    if (cameras.length == 0) {
        println("There are no cameras available for capture.");
        exit();
    } else {
        println("Available cameras:");
        for (int i = 0; i < cameras.length; i++) {
            println(cameras[i]);
        }
        cam = new Capture(this, cameras[0]);
        cam.start();
    }
}

void draw() {
    cam.updatePixels();
    if (cam.available() == true) {
        cam.read();
    }
    cam.loadPixels();
    for (int i=0; i<cam.pixels.length; i++)
    {
        int c=cam.pixels[i];
        int r=(c>>16)&0x000000FF;
        int g=(c>>8)&0x000000FF;
        int b=c&0x000000FF;
        if (r>g && r>b)
        {
            r=255;
            g=0;
            b=0;
        } else if (g>r && g>b) {
            r=0;
            g=255;
            b=0;
        } else if (b>r && b>g) {
            r=0;
            g=0;
            b=255;
        } else {
            r=178;
            g=212;
            b=223;
        }
        cam.pixels[i]=(r<<16) | (g<<8) | b;
    }
    image(cam, 0, 0);
}
```

April 6.  
**Code Entry**

Adding Black  
and White  
Detection /  
Moving out of  
RGB cam

Cartoon Cam

```
import processing.video.*;

Capture cam;

void setup() {
    size(1200, 700);
    background(178, 212, 223);
    String[] cameras = Capture.list();
    if (cameras.length == 0) {
        println("There are no cameras available for capture.");
        exit();
    } else {
```

```
println("Available cameras:");
for (int i = 0; i < cameras.length; i++) {
    println(cameras[i]);
}
cam = new Capture(this, cameras[0]);
cam.start();
}
}

void draw() {
cam.updatePixels();
if (cam.available() == true) {
    cam.read();
}
cam.loadPixels();
for (int i=0; i<cam.pixels.length; i++)
{
    int c=cam.pixels[i];
    int r=(c>>16)&0x000000FF;
    int g=(c>>8)&0x000000FF;
    int b=c&0x000000FF;
    if (abs(r-g)<20 && abs(g-b)<20 && abs(r-b)<20) {
        int average = (r+g+b)/3;
        if (average > 225) {
            r = 255;
            g = 255;
            b = 255;
        } else if (average > 185) {
            r = 200;
            g = 200;
            b = 200;
        } else if (average > 125) {
            r = 126;
            g = 126;
            b = 126;
        } else if (average > 50) {
            r = 62;
            g = 62;
            b = 62;
        } else {
            r = 0;
            g = 0;
            b = 0;
        }
    } else if (r>g && r>b)
    {
        r=255;
        g=0;
        b=0;
    } else if (g>r && g>b) {
        r=0;
        g=255;
        b=0;
    } else if (b>r && b>g) {
        r=0;
        g=0;
        b=255;
    } else {
        r=178;
        g=212;
        b=223;
    }
    cam.pixels[i]=(r<<16) | (g<<8) | b;
}
image(cam, 0, 0);
```

	<pre>         }  <u><b>April 7.</b></u> <b>Code Entry</b>  Messing Around/ Black and White Cam  <i>Green Tint</i> <i>Noir Cam</i>  import processing.video.*; Capture cam; void setup() {     size(1200, 700);     background(178, 212, 223);     String[] cameras = Capture.list();     if (cameras.length == 0) {         println("There are no cameras available for capture.");         exit();     } else {         println("Available cameras:");         for (int i = 0; i &lt; cameras.length; i++) {             println(cameras[i]);         }         cam = new Capture(this, cameras[0]);         cam.start();     } }  void draw() {     tint(255, 255, 204);     cam.updatePixels();     if (cam.available() == true) {         cam.read();     }      cam.loadPixels();     for (int i=0; i&lt;cam.pixels.length; i++)     {         int c=cam.pixels[i];         int r=(c&gt;&gt;16)&amp;0x000000FF;         int g=(c&gt;&gt;8)&amp;0x000000FF;         int b=c&amp;0x000000FF;         {             int average = (r+g+b)/3;             if (average &gt; 255) {                 r = 255;                 g = 255;                 b = 255;             } else if (average &gt; 185) {                 r = 200;                 g = 200;                 b = 200;             } else if (average &gt; 125) {                 r = 126;                 g = 126;                 b = 126;             } else if (average &gt; 50) {                 r = 62;                 g = 62;                 b = 62;             } else {                 r = 0;                 g = 0;                 b = 0;             }         }         cam.pixels[i]=(r&lt;&lt;16)   (g&lt;&lt;8)   b;     }     cam.updatePixels();     image(cam, 0, 0); } </pre>
--	--

April 9.

**Code Entry**

Moving into  
Cartoon cam

*Cartoon Cam*

```
import processing.video.*;

Capture cam;

void setup() {
    size(1200, 700);
    background(178, 212, 223);
    String[] cameras = Capture.list();
    if (cameras.length == 0) {
        println("There are no cameras available for capture.");
        exit();
    } else {
        println("Available cameras:");
        for (int i = 0; i < cameras.length; i++) {
            println(cameras[i]);
        }
        cam = new Capture(this, cameras[0]);
        cam.start();
    }
}

void draw() {

    cam.updatePixels();
    if (cam.available() == true) {
        cam.read();
    }
    cam.loadPixels();
    for (int i=0; i<cam.pixels.length; i++) {

        int c=cam.pixels[i];
        int r=(c>>16)&0x000000FF;
        int g=(c>>8)&0x000000FF;
        int b=c&0x000000FF;

        if (abs(r-g)<20 && abs(g-b)<20 && abs(r-b)<20) {
            int average = (r+g+b)/3;
            if (average > 225) {
                r = 255;
                g = 255;
                b = 255;
            } else if (average > 185) {
                r = 200;
                g = 200;
                b = 200;
            } else if (average > 125) {
                r = 126;
                g = 126;
                b = 126;
            } else if (average > 50) {
                r= (round(r/30))*30;
                g= (round(g/30))*30;
                b= (round(b/30))*30;
            } else {
                r = 0;
                g = 0;
                b = 0;
            }
        } else {
            r= (round(r/60))*60;
            g= (round(g/60))*60;
            b= (round(b/60))*60;
        }
        cam.pixels[i]=(r<<16)|(g<<8)|b;
        //up, up right diagonal, right, down right diagonal, down, down left
    }
}
```

	<pre> diagonal, left, up left diagonal } image(cam, 0, 0); } </pre>
<u>April 30.</u> <b>Code Entry</b>  Color scheme / Trouble - shooting Color Patches  Cartoon Cam	<pre> import processing.video.*;  Capture cam;  void setup() {     size(1280, 720);     background(178, 212, 223);     String[] cameras = Capture.list();     if (cameras.length == 0) {         println("There are no cameras available for capture.");         exit();     } else {         println("Available cameras:");         for (int i = 0; i &lt; cameras.length; i++) {             println(cameras[i]);         }         cam = new Capture(this, cameras[0]);         cam.start();     } }  void draw() {     int moreR = 0;     int moreL = 0;     int moreUp = 0;     int moreD = 0;     int howManyDiff = 0;      cam.updatePixels();     if (cam.available() == true) {         cam.read();     }     cam.loadPixels();     for (int i=0; i&lt;cam.pixels.length; i++) {          int c=cam.pixels[i];         int r=(c&gt;&gt;16)&amp;0x000000FF;         int g=(c&gt;&gt;8)&amp;0x000000FF;         int b=c&amp;0x000000FF;          if (abs(r-g)&lt;20 &amp;&amp; abs(g-b)&lt;20 &amp;&amp; abs(r-b)&lt;20) {             int average = (r+g+b)/3;             if (average &gt; 225) {                 r = 255;                 g = 255;                 b = 255;             } else if (average &gt; 185) {                 r = 200;                 g = 200;                 b = 200;             } else if (average &gt; 125) {                 r = 126;                 g = 126;                 b = 126;             } else if (average &gt; 50) {                 r= ((round(r/60))*60)+20;                 g= ((round(g/40))*40)+10;                 b= ((round(b/30))*30)+20;             } else {                 r= ((round(r/60))*60)+20;                 g= ((round(g/40))*40)+10;                 b= ((round(b/30))*30)+20;             }         }     } } </pre>

```

        g= ((round(g/40))*40)+10;
        b= ((round(b/30))*30)+20;
    }
} else {
    r= ((round(r/60))*60)+20;
    g= ((round(g/40))*40)+10;
    b= ((round(b/30))*30)+20;
}
cam.pixels[i]=(r<<16)|(g<<8)|b;
//up, up right diagonal, right, down right diagonal, down, down left
diagonal, left, up left diagonal
if (i % 1280 == 1 || i % 1280 == 0 || i <= 1280 || i > 1280*719) {
} else {
    if (cam.pixels[i] != cam.pixels[i-1281]) {
        howManyDiff++;
    } else {
        moreL++;
        moreUp++;
    }
    if (cam.pixels[i] != cam.pixels[i-1280]) {
        howManyDiff++;
    } else {
        moreL++;
    }
    if (cam.pixels[i] != cam.pixels[i-1279]) {
        howManyDiff++;
    } else {
        moreL++;
        moreD++;
    }
    if (cam.pixels[i] != cam.pixels[i-1]) {
        howManyDiff++;
    } else {
        moreUp++;
    }
    if (cam.pixels[i] != cam.pixels[i+1]) {
        howManyDiff++;
    } else {
        moreD++;
    }
    if (cam.pixels[i] != cam.pixels[i+1279]) {
        howManyDiff++;
    } else {
        moreR++;
        moreUp++;
    }
    if (cam.pixels[i] != cam.pixels[i+1280]) {
        howManyDiff++;
    } else {
        moreR++;
    }
    if (cam.pixels[i] != cam.pixels[i+1281]) {
        howManyDiff++;
    } else {
        moreR++;
        moreD++;
    }
}
if (howManyDiff > 4) {
    if (moreL == 3 && moreR == 3 && moreUp == 3 && moreD == 3 &&
cam.pixels[i] != cam.pixels[i+1]) {
        cam.pixels[i] = cam.pixels[i+1];
    }
    else if(moreL >1 && moreR >1 && moreUp >1 && cam.pixels[i] !=
cam.pixels[i-1] ) {

```

	<pre>         cam.pixels[i] = cam.pixels[i-1];     }     else if(moreL &gt;1 &amp;&amp; moreR &gt;1 &amp;&amp; moreD &gt;1 &amp;&amp; cam.pixels[i] != cam.pixels[i+1] ){         cam.pixels[i] = cam.pixels[i+1];     }     else if(moreL &gt;1 &amp;&amp; moreUp &gt;1 &amp;&amp; moreD &gt;1 &amp;&amp; cam.pixels[i] != cam.pixels[i+1] ){         cam.pixels[i] = cam.pixels[i+1];     }     else if(moreR &gt;1 &amp;&amp; moreUp &gt;1 &amp;&amp; moreD &gt;1 &amp;&amp; cam.pixels[i] != cam.pixels[i+1] ){         cam.pixels[i] = cam.pixels[i+1];     } } }  image(cam, 0, 0); } </pre>
<u>May 6.</u> <b>Code Entry</b>  Trouble - shooting Color Patches / Trippy Cam accident  <i>Trippy Cam v.1</i>	<pre> import processing.video.*;  Capture cam;  void setup() {      size(1280, 720);     background(178, 212, 223);     String[] cameras = Capture.list();     if (cameras.length == 0) {         println("There are no cameras available for capture.");         exit();     } else {         println("Available cameras:");         for (int i = 0; i &lt; cameras.length; i++) {             println(cameras[i]);         }         cam = new Capture(this, cameras[0]);         cam.start();     } }  void draw() {     int r = 0;     int g = 0;     int b = 0;      cam.updatePixels();     if (cam.available() == true) {         cam.read();     }     cam.loadPixels();     for (int i = 0; i &lt; cam.pixels.length; i++) {         int c = cam.pixels[i];         r = (c&gt;&gt;16)&amp;0x000000FF;         g = (c&gt;&gt;8)&amp;0x000000FF;         b = c&amp;0x000000FF;          if (abs(r-g) &lt; 20 &amp;&amp; abs(g-b) &lt; 20 &amp;&amp; abs(r-b) &lt; 20) {             int average = (r+g+b)/3;             if (average &gt; 225) {                 r = 255;                 g = 255;                 b = 255;             } else if (average &gt; 185) { </pre>

```

        r = 200;
        g = 200;
        b = 200;
    } else if (average > 125) {
        r = 126;
        g = 126;
        b = 126;
    } else if (average > 50) {
        r= ((round(r/60))*60)+20;
        g= ((round(g/40))*40)+10;
        b= ((round(b/30))*30)+20;
    } else {
        r= ((round(r/60))*60)+20;
        g= ((round(g/40))*40)+10;
        b= ((round(b/30))*30)+20;
    }
} else {
    r= ((round(r/60))*60)+20;
    g= ((round(g/40))*40)+10;
    b= ((round(b/30))*30)+20;
}
cam.pixels[i]=(r<<16) | (g<<8) | b;
}

for (int i = 0; i < cam.pixels.length; i++) {
    if (i % width == (width-1) /*right border*/ || i % width == 0 /*left border*/
    || i < width /*top row*/ || i > width*(height - 1) /*bottom row*/)
    {
    } else {
        int uL = cam.pixels[i-(width+1)]; /*up left RGB*/
        int rUL=(uL>>16)&0x000000FF;
        int gUL=(uL>>8)&0x000000FF;
        int bUL=uL&0x000000FF;

        int u = cam.pixels[i-width]; /*up RGB*/
        int rU=(u>>16)&0x000000FF;
        int gU=(u>>8)&0x000000FF;
        int bU=u&0x000000FF;

        int uR = cam.pixels[i-(width-1)]; /*up right RGB*/
        int rUR=(uR>>16)&0x000000FF;
        int gUR=(uR>>8)&0x000000FF;
        int bUR=uR&0x000000FF;

        int l = cam.pixels[i-1]; /*left RGB*/
        int rL=(l>>16)&0x000000FF;
        int gL=(l>>8)&0x000000FF;
        int bL=l&0x000000FF;

        int ri = cam.pixels[i+1]; /*right RGB*/
        int rRi=(ri>>16)&0x000000FF;
        int gRi=(ri>>8)&0x000000FF;
        int bRi=ri&0x000000FF;

        int dL = cam.pixels[i+(width-1)]; /*down left RGB*/
        int rDL=(dL>>16)&0x000000FF;
        int gDL=(dL>>8)&0x000000FF;
        int bDL=dL&0x000000FF;

        int d = cam.pixels[i+width]; /*down RGB*/
        int rD=(d>>16)&0x000000FF;
        int gD=(d>>8)&0x000000FF;
        int bD=d&0x000000FF;

        int dR = cam.pixels[i+width]; /*down right RGB*/
    }
}

```

	<pre> int rDR=(dR&gt;&gt;16)&amp;0x000000FF; int gDR=(dR&gt;&gt;8)&amp;0x000000FF; int bDR=dR&amp;0x000000FF;  if (cam.pixels[i] != d) {     r = rD;     g = gD;     b = bD; } cam.pixels[i]=(r&lt;&lt;16)   (g&lt;&lt;8)   b; }  image(cam, 0, 0); } </pre>
<u>May 7.</u> <b>Code Entry</b>  Trouble - shooting / Adjusting color scheme  <i>Cartoon Cam</i>	<pre> import processing.video.*;  Capture cam;  void setup() {      size(1280, 720);     background(178, 212, 223);     String[] cameras = Capture.list();     if (cameras.length == 0) {         println("There are no cameras available for capture.");         exit();     } else {         println("Available cameras:");         for (int i = 0; i &lt; cameras.length; i++) {             println(cameras[i]);         }         cam = new Capture(this, cameras[0]);         cam.start();     } }  void draw() {     int r = 0;     int g = 0;     int b = 0;      cam.updatePixels();     if (cam.available() == true) {         cam.read();     }     cam.loadPixels();     for (int i = 0; i &lt; cam.pixels.length; i++) {         int c = cam.pixels[i];         r = (c&gt;&gt;16)&amp;0x000000FF;         g = (c&gt;&gt;8)&amp;0x000000FF;         b = c&amp;0x000000FF;         int average = (r+g+b)/3;         if (abs(r-g) &lt; 20 &amp;&amp; abs(g-b) &lt; 20 &amp;&amp; abs(r-b) &lt; 20) {              if (average &gt; 225) {                 r = 255;                 g = 255;                 b = 255;             } else if (average &gt; 185) {                 r = 200;                 g = 200;                 b = 200;             } else if (average &gt; 125) { </pre>

```

        r = 126;
        g = 126;
        b = 126;
    } else if (average > 50) {
        r = ((round(r/60))*60)+20;
        g = ((round(g/40))*40)+10;
        b = ((round(b/30))*30)+20;
    } else {
        r = ((round(r/60))*60)+20;
        g = ((round(g/40))*40)+10;
        b = ((round(b/30))*30)+20;
    }
} else if (abs(r-g) < 100 && abs(g-b) < 100 && abs(r-b) < 100 && average
> 185 ) {
    r = 200;
    g = 200;
    b = 200;
} else {
    r = ((round(r/60))*60)+60;
    g = ((round(g/60))*60)+40;
    b = ((round(b/60))*60)+35;
}
cam.pixels[i]=(r<<16)|(g<<8)|b;
}

for (int i = 0; i < cam.pixels.length; i++) {
    if (i % width == (width-1) /*right border*/ || i % width == 0 /*left
border*/ || i < width /*top row*/ || i > width*(height - 1) /*bottom row*/)
    {
    } else {
        int c = cam.pixels[i];
        r = (c>>16)&0x000000FF;
        g = (c>>8)&0x000000FF;
        b = c&0x000000FF;

        int uL = cam.pixels[i-(width+1)]; /*up left RGB*/
        int rUL=(uL>>16)&0x000000FF;
        int gUL=(uL>>8)&0x000000FF;
        int bUL=uL&0x000000FF;

        int u = cam.pixels[i-width]; /*up RGB*/
        int rU=(u>>16)&0x000000FF;
        int gU=(u>>8)&0x000000FF;
        int bU=u&0x000000FF;

        int uR = cam.pixels[i-(width-1)]; /*up right RGB*/
        int rUR=(uR>>16)&0x000000FF;
        int gUR=(uR>>8)&0x000000FF;
        int bUR=uR&0x000000FF;

        int l = cam.pixels[i-1]; /*left RGB*/
        int rL=(l>>16)&0x000000FF;
        int gL=(l>>8)&0x000000FF;
        int bL=l&0x000000FF;

        int ri = cam.pixels[i+1]; /*right RGB*/
        int rRi=(ri>>16)&0x000000FF;
        int gRi=(ri>>8)&0x000000FF;
        int bRi=ri&0x000000FF;

        int dL = cam.pixels[i+(width-1)]; /*down left RGB*/
        int rDL=(dL>>16)&0x000000FF;
        int gDL=(dL>>8)&0x000000FF;
        int bDL=dL&0x000000FF;
    }
}

```

```

        int d = cam.pixels[i+width]; /*down RGB*/
        int rD=(d>>16)&0x000000FF;
        int gD=(d>>8)&0x000000FF;
        int bD=d&0x000000FF;

        int dR = cam.pixels[i+width]; /*down right RGB*/
        int rDR=(dR>>16)&0x000000FF;
        int gDR=(dR>>8)&0x000000FF;
        int bDR=dR&0x000000FF;

        if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d == dL
        && dL == l && l == cam.pixels[i]) {
            } else if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d
        == dL && dL == l && l != cam.pixels[i]) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else if (uL == u && u == uR && uR == l && ri == rD && rD == d && d
        == dL && dL != l && l != cam.pixels[i] && ri != cam.pixels[i]) {
            if (random(10) < 5) {
                r = rRi;
                b = bRi;
                g = gRi;
            } else {
                r = rL;
                b = bL;
                g = gL;
            }
        } else if (cam.pixels[i] != uL && cam.pixels[i] != u && cam.pixels[i]
        != uR && cam.pixels[i] != ri && cam.pixels[i] != rD && cam.pixels[i] != d &&
        cam.pixels[i] != dL && cam.pixels[i] != l) {
            if (random(10) < 5) {
                r = rRi;
                b = bRi;
            } else {
                r = rL;
                b = bL;
                g = gL;
            }
        }
        cam.pixels[i]=(r<<16) | (g<<8) | b;
    }
}

image(cam, 0, 0);
}

```

May 7.  
**Code Entry**

Trouble -  
shooting  
/ Trippy Cam  
accident

Trippy Cam v.1  
.5

```

import processing.video.*;

Capture cam;

void setup() {

    size(1280, 720);
    background(178, 212, 223);
    String[] cameras = Capture.list();
    if (cameras.length == 0) {
        println("There are no cameras available for capture.");
        exit();
    } else {
        println("Available cameras:");
        for (int i = 0; i < cameras.length; i++) {
            println(cameras[i]);
        }
    }
}

```

```

        cam = new Capture(this, cameras[0]);
        cam.start();
    }

}

void draw() {
    int r = 0;
    int g = 0;
    int b = 0;

    cam.updatePixels();
    if (cam.available() == true) {
        cam.read();
    }
    cam.loadPixels();
    for (int i = 0; i < cam.pixels.length; i++) {
        int c = cam.pixels[i];
        r = (c>>16)&0x000000FF;
        g = (c>>8)&0x000000FF;
        b = c&0x000000FF;

        if (abs(r-g) < 20 && abs(g-b) < 20 && abs(r-b) < 20) {
            int average = (r+g+b)/3;
            if (average > 225) {
                r = 255;
                g = 255;
                b = 255;
            } else if (average > 185) {
                r = 200;
                g = 200;
                b = 200;
            } else if (average > 125) {
                r = 126;
                g = 126;
                b = 126;
            } else if (average > 50) {
                r= ((round(r/60))*60)+20;
                g= ((round(g/40))*40)+10;
                b= ((round(b/30))*30)+20;
            } else {
                r= ((round(r/60))*60)+20;
                g= ((round(g/40))*40)+10;
                b= ((round(b/30))*30)+20;
            }
        } else {
            r= ((round(r/60))*60)*round(900000000/(r+1));
            g= ((round(g/40))*40)*round(900000000/(g+1));
            b= ((round(b/30))*30)*round(900000000/(b+1));
        }
        cam.pixels[i]=(r<<16)|(g<<8)|b;
    }

    for (int i = 0; i < cam.pixels.length; i++) {
        if (i % width == (width-1) /*right border*/ || i % width == 0 /*left
border*/ || i < width /*top row*/ || i > width*(height - 1) /*bottom row*/)
{
        } else {

            int c = cam.pixels[i];
            r = (c>>16)&0x000000FF;
            g = (c>>8)&0x000000FF;
            b = c&0x000000FF;
            int uL = cam.pixels[i-(width+1)]; /*up left RGB*/
            int rUL=(uL>>16)&0x000000FF;
            int gUL=(uL>>8)&0x000000FF;
            int bUL=uL&0x000000FF;
            int uR = cam.pixels[i+(width-1)]; /*up right RGB*/
            int rUR=(uR>>16)&0x000000FF;
            int gUR=(uR>>8)&0x000000FF;
            int bUR=uR&0x000000FF;
            int dL = cam.pixels[i+(width+1)]; /*down left RGB*/
            int rDL=(dL>>16)&0x000000FF;
            int gDL=(dL>>8)&0x000000FF;
            int bDL=dL&0x000000FF;
            int dR = cam.pixels[i+(width-1)+(width+1)]; /*down right RGB*/
            int rDR=(dR>>16)&0x000000FF;
            int gDR=(dR>>8)&0x000000FF;
            int bDR=dR&0x000000FF;
            int cL = cam.pixels[i+(width+1)+(width+1)]; /*center left RGB*/
            int rCL=(cL>>16)&0x000000FF;
            int gCL=(cL>>8)&0x000000FF;
            int bCL=cL&0x000000FF;
            int cR = cam.pixels[i+(width-1)+(width-1)]; /*center right RGB*/
            int rCR=(cR>>16)&0x000000FF;
            int gCR=(cR>>8)&0x000000FF;
            int bCR=cR&0x000000FF;
            int cC = cam.pixels[i+(width-1)+(width-1)+(width+1)]; /*center center RGB*/
            int rCC=(cC>>16)&0x000000FF;
            int gCC=(cC>>8)&0x000000FF;
            int bCC=cC&0x000000FF;
            int rAvg = (r+rUL+rUR+rDL+rDR+rCL+rCR+rCC)/8;
            int gAvg = (g+gUL+gUR+gDL+gDR+gCL+gCR+gCC)/8;
            int bAvg = (b+bUL+bUR+bDL+bDR+bCL+bCR+bCC)/8;
            cam.pixels[i]=(rAvg<<16)|(gAvg<<8)|bAvg;
        }
    }
}

```

	<pre> int bUL=uL&amp;0x000000FF;  int u = cam.pixels[i-width]; /*up RGB*/ int rU=(u&gt;&gt;16)&amp;0x000000FF; int gU=(u&gt;&gt;8) &amp;0x000000FF; int bU=u&amp;0x000000FF;  int uR = cam.pixels[i-(width-1)]; /*up right RGB*/ int rUR=(uR&gt;&gt;16)&amp;0x000000FF; int gUR=(uR&gt;&gt;8) &amp;0x000000FF; int bUR=uR&amp;0x000000FF;  int l = cam.pixels[i-1]; /*left RGB*/ int rL=(l&gt;&gt;16)&amp;0x000000FF; int gL=(l&gt;&gt;8) &amp;0x000000FF; int bL=l&amp;0x000000FF;  int ri = cam.pixels[i+1]; /*right RGB*/ int rRi=(ri&gt;&gt;16)&amp;0x000000FF; int gRi=(ri&gt;&gt;8) &amp;0x000000FF; int bRi=ri&amp;0x000000FF;  int dL = cam.pixels[i+(width-1)]; /*down left RGB*/ int rDL=(dL&gt;&gt;16)&amp;0x000000FF; int gDL=(dL&gt;&gt;8) &amp;0x000000FF; int bDL=dL&amp;0x000000FF;  int d = cam.pixels[i+width]; /*down RGB*/ int rD=(d&gt;&gt;16)&amp;0x000000FF; int gD=(d&gt;&gt;8) &amp;0x000000FF; int bD=d&amp;0x000000FF;  int dR = cam.pixels[i+width]; /*down right RGB*/ int rDR=(dR&gt;&gt;16)&amp;0x000000FF; int gDR=(dR&gt;&gt;8) &amp;0x000000FF; int bDR=dR&amp;0x000000FF;  if (cam.pixels[i] != d) {     r = rD;     g = gD;     b = bD; }  cam.pixels[i]=(r&lt;&lt;16)   (g&lt;&lt;8)   b; } }  image(cam, 0, 0); } </pre>
<u>May 7.</u> <b>Code Entry</b>  Messing Around  <i>Trippy Cam - Computer Vision</i>	<pre> import processing.video.*;  Capture cam;  void setup() {      size(1280, 720);     background(178, 212, 223);     String[] cameras = Capture.list();     if (cameras.length == 0) {         println("There are no cameras available for capture.");         exit();     } else {         println("Available cameras:");     } } </pre>

```

        for (int i = 0; i < cameras.length; i++) {
            println(cameras[i]);
        }
        cam = new Capture(this, cameras[0]);
        cam.start();
    }
}

void draw() {
    int r = 0;
    int g = 0;
    int b = 0;

    cam.updatePixels();
    if (cam.available() == true) {
        cam.read();
    }
    cam.loadPixels();
    for (int i = 0; i < cam.pixels.length; i++) {
        int c = cam.pixels[i];
        r = (c>>16)&0x000000FF;
        g = (c>>8)&0x000000FF;
        b = c&0x000000FF;
        int average = (r+g+b)/3;
        if (abs(r-g) < 20 && abs(g-b) < 20 && abs(r-b) < 20) {

            if (average > 225) {
                r = 255;
                g = 255;
                b = 255;
            } else if (average > 185) {
                r = 200;
                g = 200;
                b = 200;
            } else if (average > 125) {
                r = 126;
                g = 126;
                b = 126;
            } else if (average > 50) {
                r = ((round(r/60))*60)+20;
                g = ((round(g/40))*40)+10;
                b = ((round(b/30))*30)+20;
            } else {
                r = ((round(r/60))*60)+20;
                g = ((round(g/40))*40)+10;
                b = ((round(b/30))*30)+20;
            }
        } else if (abs(r-g) < 100 && abs(g-b) < 100 && abs(r-b) < 100 && average
> 185 ) {
            r = 200;
            g = 200;
            b = 200;
        } else {
            r= ((round(r/60))*60)+round(900000000/(r+1));
            g= ((round(g/40))*40)+round(900000000/(g+1));
            b= ((round(b/30))*30)+round(900000000/(b+1));
        }
        cam.pixels[i]=(r<<16)|(g<<8)|b;
    }

    for (int i = 0; i < cam.pixels.length; i++) {
        if (i % width == (width-1) /*right border*/ || i % width == 0 /*left
border*/ || i < width /*top row*/ || i > width*(height - 1) /*bottom row*/)
{
        } else {

```

```

int uL = cam.pixels[i-(width+1)]; /*up left RGB*/
int rUL=(uL>>16)&0x000000FF;
int gUL=(uL>>8)&0x000000FF;
int bUL=uL&0x000000FF;

int u = cam.pixels[i-width]; /*up RGB*/
int rU=(u>>16)&0x000000FF;
int gU=(u>>8)&0x000000FF;
int bU=u&0x000000FF;

int uR = cam.pixels[i-(width-1)]; /*up right RGB*/
int rUR=(uR>>16)&0x000000FF;
int gUR=(uR>>8)&0x000000FF;
int bUR=uR&0x000000FF;

int l = cam.pixels[i-1]; /*left RGB*/
int rL=(l>>16)&0x000000FF;
int gL=(l>>8)&0x000000FF;
int bL=l&0x000000FF;

int ri = cam.pixels[i+1]; /*right RGB*/
int rRi=(ri>>16)&0x000000FF;
int gRi=(ri>>8)&0x000000FF;
int bRi=ri&0x000000FF;

int dL = cam.pixels[i+(width-1)]; /*down left RGB*/
int rDL=(dL>>16)&0x000000FF;
int gDL=(dL>>8)&0x000000FF;
int bDL=dL&0x000000FF;

int d = cam.pixels[i+width]; /*down RGB*/
int rD=(d>>16)&0x000000FF;
int gD=(d>>8)&0x000000FF;
int bD=d&0x000000FF;

int dR = cam.pixels[i+width]; /*down right RGB*/
int rDR=(dR>>16)&0x000000FF;
int gDR=(dR>>8)&0x000000FF;
int bDR=dR&0x000000FF;

if (cam.pixels[i] != d) {
    r = rD;
    g = gD;
    b = bD;
}

cam.pixels[i]=(r<<16) | (g<<8) | b;
}
}
for (int i = 0; i < cam.pixels.length; i++) {
    int c = cam.pixels[i];
    r = (c>>16)&0x000000FF;
    g = (c>>8)&0x000000FF;
    b = c&0x000000FF;
    int average = (r+g+b)/3;
    if (average > 225) {
        r = 255;
        g = 255;
        b = 255;
    } else if (average > 185) {
        r = 200;
        g = 200;
        b = 200;
    } else if (average > 125) {
        r = 126;
    }
}

```

	<pre> g = 126; b = 126; } else if (average &gt; 50) { r = 55; g = 55; b = 55; } else { r = 25; g = 25; b = 25; } cam.pixels[i]=(r&lt;&lt;16)   (g&lt;&lt;8)  b; } if (random(55) &lt; 2) { tint(244, 25, 195); } else if (random(15) &lt; 2 ) { tint(200); } else { tint(80, 255, 55); } image(cam, 0, 0); } </pre>
<u>May 8.</u> <b>Code Entry</b>  Messing Around Final Trippy Cam  <i>Trippy Cam v.2</i>	<pre> import processing.video.*;  Capture cam;  void setup() {  size(1280, 720); background(178, 212, 223); String[] cameras = Capture.list(); if (cameras.length == 0) { println("There are no cameras available for capture."); exit(); } else { println("Available cameras:"); for (int i = 0; i &lt; cameras.length; i++) { println(cameras[i]); } cam = new Capture(this, cameras[0]); cam.start(); } }  void draw() { int r = 0; int g = 0; int b = 0;  cam.updatePixels(); if (cam.available() == true) { cam.read(); } cam.loadPixels(); for (int i = 0; i &lt; cam.pixels.length; i++) { int c = cam.pixels[i]; r = (c&gt;&gt;16)&amp;0x000000FF; g = (c&gt;&gt;8)&amp;0x000000FF; b = c&amp;0x000000FF; int average = (r+g+b)/3;  r = round(r*0.75); b = round(r*0.75); g = round(r*0.75); } } </pre>

```

r= ((round(r/60))*60)*round(900000000/(r+1));
g= ((round(g/40))*40)*round(900000000/(g+1));
b= ((round(b/30))*30)*round(900000000/(b+1));

cam.pixels[i]=(r<<16) | (g<<8) | b;
}

for (int i = 0; i < cam.pixels.length; i++) {
    if (i % width == (width-1) /*right border*/ || i % width == 0 /*left border*/
    || i < width /*top row*/ || i > width*(height - 1) /*bottom row*/)
    {
    } else {
        int uL = cam.pixels[i-(width+1)]; /*up left RGB*/
        int rUL=(uL>>16)&0x000000FF;
        int gUL=(uL>>8)&0x000000FF;
        int bUL=uL&0x000000FF;

        int u = cam.pixels[i-width]; /*up RGB*/
        int rU=(u>>16)&0x000000FF;
        int gU=(u>>8)&0x000000FF;
        int bU=u&0x000000FF;

        int uR = cam.pixels[i-(width-1)]; /*up right RGB*/
        int rUR=(uR>>16)&0x000000FF;
        int gUR=(uR>>8)&0x000000FF;
        int bUR=uR&0x000000FF;

        int l = cam.pixels[i-1]; /*left RGB*/
        int rL=(l>>16)&0x000000FF;
        int gL=(l>>8)&0x000000FF;
        int bL=l&0x000000FF;

        int ri = cam.pixels[i+1]; /*right RGB*/
        int rRi=(ri>>16)&0x000000FF;
        int gRi=(ri>>8)&0x000000FF;
        int bRi=ri&0x000000FF;

        int dL = cam.pixels[i+(width-1)]; /*down left RGB*/
        int rDL=(dL>>16)&0x000000FF;
        int gDL=(dL>>8)&0x000000FF;
        int bDL=dL&0x000000FF;

        int d = cam.pixels[i+width]; /*down RGB*/
        int rD=(d>>16)&0x000000FF;
        int gD=(d>>8)&0x000000FF;
        int bD=d&0x000000FF;

        int dR = cam.pixels[i+width]; /*down right RGB*/
        int rDR=(dR>>16)&0x000000FF;
        int gDR=(dR>>8)&0x000000FF;
        int bDR=dR&0x000000FF;

        if (cam.pixels[i] != d) {
            r = rD;
            g = gD;
            b = bD;
        }

        cam.pixels[i]=(r<<16) | (g<<8) | b;
    }
}

image(cam, 0, 0);
}

```

May 8  
**Code Entry**

Final Code  
Needed For  
Capstone

*Cartoon Cam*

```
import processing.video.*;

Capture cam;

void setup() {
    size(1280, 720);
    background(178, 212, 223);
    String[] cameras = Capture.list();
    if (cameras.length == 0) {
        println("There are no cameras available for capture.");
        exit();
    } else {
        println("Available cameras:");
        for (int i = 0; i < cameras.length; i++) {
            println(cameras[i]);
        }
        cam = new Capture(this, cameras[0]);
        cam.start();
    }
}

void draw() {
    int r = 0;
    int g = 0;
    int b = 0;

    cam.updatePixels();
    if (cam.available() == true) {
        cam.read();
    }
    cam.loadPixels();
    for (int i = 0; i < cam.pixels.length; i++) {
        int c = cam.pixels[i];
        r = (c>>16)&0x000000FF;
        g = (c>>8)&0x000000FF;
        b = c&0x000000FF;

        r = (round((r/55)*25.6))+42;
        g = (round((g/40)*25.6))+34;
        b = (round((b/50)*25.6))+52;

        cam.pixels[i]=(r<<16)|(g<<8)|b;
    }

    for (int i = 0; i < cam.pixels.length; i++) {
        if (i % width == (width-1) /*right border*/ || i % width == 0 /*left border*/ || i < width /*top row*/ || i > width*(height - 1) /*bottom row*/)
        {
        } else {
            int c = cam.pixels[i];
            r = (c>>16)&0x000000FF;
            g = (c>>8)&0x000000FF;
            b = c&0x000000FF;

            int uL = cam.pixels[i-(width+1)]; /*up left RGB*/
            int rUL=(uL>>16)&0x000000FF;
            int gUL=(uL>>8)&0x000000FF;
            int bUL=uL&0x000000FF;

            int u = cam.pixels[i-width]; /*up RGB*/
            int rU=(u>>16)&0x000000FF;
            int gU=(u>>8)&0x000000FF;
```

```

int bU=u&0x000000FF;

int uR = cam.pixels[i-(width-1)]; /*up right RGB*/
int rUR=(uR>>16)&0x000000FF;
int gUR=(uR>>8)&0x000000FF;
int bUR=uR&0x000000FF;

int l = cam.pixels[i-1]; /*left RGB*/
int rL=(l>>16)&0x000000FF;
int gL=(l>>8)&0x000000FF;
int bL=l&0x000000FF;

int ri = cam.pixels[i+1]; /*right RGB*/
int rRi=(ri>>16)&0x000000FF;
int gRi=(ri>>8)&0x000000FF;
int bRi=ri&0x000000FF;

int dL = cam.pixels[i+(width-1)]; /*down left RGB*/
int rDL=(dL>>16)&0x000000FF;
int gDL=(dL>>8)&0x000000FF;
int bDL=dL&0x000000FF;

int d = cam.pixels[i+width]; /*down RGB*/
int rD=(d>>16)&0x000000FF;
int gD=(d>>8)&0x000000FF;
int bD=d&0x000000FF;

int dR = cam.pixels[i+width]; /*down right RGB*/
int rDR=(dR>>16)&0x000000FF;
int gDR=(dR>>8)&0x000000FF;
int bDR=dR&0x000000FF;

if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d == dL
&& dL == l && l == cam.pixels[i]) {
    } else if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d
== dL && dL == l && l != cam.pixels[i]) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (uL == u && u == uR && uR == l && ri == rD && rD == d && d
== dL && dL != l && l != cam.pixels[i] && ri != cam.pixels[i]) {
        if (random(10) < 5) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else {
            r = rL;
            b = bL;
            g = gL;
        }
    } else if (uL != u && u == uR && uR == l && ri != rD && rD == d && d
== dL && dL == l && l != cam.pixels[i] && ri != cam.pixels[i]) {
        if (random(10) < 5) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else {
            r = rL;
            b = bL;
            g = gL;
        }
    } else if (u == uL && uL == l && l == dL && dL == d && cam.pixels[i]
!= l) {
        r = rL;
        b = bL;
    }
}

```

```

        g = gL;
    } else if (u == uR && uR == ri && ri == dR && dR == d && cam.pixels[i] != ri) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (ri == uR && uR == u && u == uL && uL == l && cam.pixels[i] != ri) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (ri == dR && dR == d && d == dL && dL == l && cam.pixels[i] != ri) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (ri == dR && dR == d && cam.pixels[i] != ri) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (ri == uR && uR == u && cam.pixels[i] != ri) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (l == dL && dL == d && cam.pixels[i] != l) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (l == uL && uL == u && cam.pixels[i] != l) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (cam.pixels[i] != uL && cam.pixels[i] != u && cam.pixels[i] != uR && cam.pixels[i] != ri && cam.pixels[i] != rD && cam.pixels[i] != d && cam.pixels[i] != dL && cam.pixels[i] != l) {
        if (random(10) < 5) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else {
            r = rL;
            b = bL;
            g = gL;
        }
        cam.pixels[i]=(r<<16) | (g<<8) | b;
    }
}

image(cam, 0, 0);
}

```

May. 8  
**Code Entry**

Messing  
around /  
Based off  
color scheme  
from previous  
code

Cartoon Cam /  
Alternate  
Version -

```

import processing.video.*;

Capture cam;

void setup() {

    size(1280, 720);
    background(178, 212, 223);
    String[] cameras = Capture.list();
    if (cameras.length == 0) {
        println("There are no cameras available for capture.");
        exit();
    } else {

```

*Crayon Cam*

```
println("Available cameras:");
for (int i = 0; i < cameras.length; i++) {
    println(cameras[i]);
}
cam = new Capture(this, cameras[0]);
cam.start();
}

void draw() {
    int r = 0;
    int g = 0;
    int b = 0;

    cam.updatePixels();
    if (cam.available() == true) {
        cam.read();
    }
    cam.loadPixels();
    for (int i = 0; i < cam.pixels.length; i++) {
        int c = cam.pixels[i];
        r = (c>>16)&0x000000FF;
        g = (c>>8)&0x000000FF;
        b = c&0x000000FF;

        r= (round(r/60))*60;
        g= (round(g/60))*60;
        b= (round(b/60))*60;

        cam.pixels[i]=(r<<16) | (g<<8) | b;
    }

    for (int i = 0; i < cam.pixels.length; i++) {
        if (i % width == (width-1) /*right border*/ || i % width == 0 /*left border*/ || i < width /*top row*/ || i > width*(height - 1) /*bottom row*/)
        {
            } else {
                int c = cam.pixels[i];
                r = (c>>16)&0x000000FF;
                g = (c>>8)&0x000000FF;
                b = c&0x000000FF;

                int uL = cam.pixels[i-(width+1)]; /*up left RGB*/
                int rUL=(uL>>16)&0x000000FF;
                int gUL=(uL>>8)&0x000000FF;
                int bUL=uL&0x000000FF;

                int u = cam.pixels[i-width]; /*up RGB*/
                int rU=(u>>16)&0x000000FF;
                int gU=(u>>8)&0x000000FF;
                int bU=u&0x000000FF;

                int uR = cam.pixels[i-(width-1)]; /*up right RGB*/
                int rUR=(uR>>16)&0x000000FF;
                int gUR=(uR>>8)&0x000000FF;
                int bUR=uR&0x000000FF;

                int l = cam.pixels[i-1]; /*left RGB*/
                int rL=(l>>16)&0x000000FF;
                int gL=(l>>8)&0x000000FF;
                int bL=l&0x000000FF;

                int ri = cam.pixels[i+1]; /*right RGB*/
                int rRi=(ri>>16)&0x000000FF;
```

```

int gRi=(ri>>8)&0x000000FF;
int bRi=ri&0x000000FF;

int dL = cam.pixels[i+(width-1)]; /*down left RGB*/
int rDL=(dL>>16)&0x000000FF;
int gDL=(dL>>8)&0x000000FF;
int bDL=dL&0x000000FF;

int d = cam.pixels[i+width]; /*down RGB*/
int rD=(d>>16)&0x000000FF;
int gD=(d>>8)&0x000000FF;
int bD=d&0x000000FF;

int dR = cam.pixels[i+width]; /*down right RGB*/
int rDR=(dR>>16)&0x000000FF;
int gDR=(dR>>8)&0x000000FF;
int bDR=dR&0x000000FF;

if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d == dL
&& dL == l && l == cam.pixels[i]) {
} else if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d
== dL && dL == l && l != cam.pixels[i]) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (uL == u && u == uR && uR == l && ri == rD && rD == d && d
== dL && dL != l && l != cam.pixels[i] && ri != cam.pixels[i]) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
        g = gL;
    }
} else if (uL != u && u == uR && uR == l && ri != rD && rD == d && d
== dL && dL == l && l != cam.pixels[i] && ri != cam.pixels[i]) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
        g = gL;
    }
}
else if (u == uL && uL == l && l == dL && dL == d && cam.pixels[i]
!= l) {
    r = rL;
    b = bL;
    g = gL;
} else if (u == uR && uR == ri && ri == dR && dR == d && cam.pixels[i]
!= ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == uR && uR == u && u == uL && uL == l && cam.pixels[i]
!= ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == dR && dR == d && d == dL && dL == l && cam.pixels[i]
!= ri) {
    r = rRi;
}

```

```

        b = bRi;
        g = gRi;
    } else if (ri == dR && dR == d && cam.pixels[i] != ri) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (ri == uR && uR == u && cam.pixels[i] != ri) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (l == dL && dL == d && cam.pixels[i] != l) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (l == uL && uL == u && cam.pixels[i] != l) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (cam.pixels[i] != uL && cam.pixels[i] != u && cam.pixels[i]
!= uR && cam.pixels[i] != ri && cam.pixels[i] != rD && cam.pixels[i] != d &&
cam.pixels[i] != dL && cam.pixels[i] != l) {
        if (random(10) < 5) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else {
            r = rL;
            b = bL;
            g = gL;
        }
        cam.pixels[i]=(r<<16) | (g<<8) |b;
    }
}

image(cam, 0, 0);
}

```

May. 8

### Code Entry

Attempt at  
outlining  
color blobs

*Cartoon Cam*

```

import processing.video.*;

Capture cam;

void setup() {

    size(1280, 720);
    background(0);
    String[] cameras = Capture.list();
    if (cameras.length == 0) {
        println("There are no cameras available for capture.");
        exit();
    } else {
        println("Available cameras:");
        for (int i = 0; i < cameras.length; i++) {
            println(cameras[i]);
        }
        cam = new Capture(this, cameras[0]);
        cam.start();
    }
}

void draw() {
    int r = 0;
    int g = 0;
    int b = 0;
}

```

```

cam.updatePixels();
if (cam.available() == true) {
    cam.read();
}
cam.loadPixels();
for (int i = 0; i < cam.pixels.length; i++) {
    int c = cam.pixels[i];
    r = (c>>16)&0x000000FF;
    g = (c>>8)&0x000000FF;
    b = c&0x000000FF;

    r = (round((r/55)*25.6))+42;
    g = (round((g/40)*25.6))+34;
    b = (round((b/50)*25.6))+52;

    cam.pixels[i]=(r<<16) | (g<<8) | b;
}

for (int i = 0; i < cam.pixels.length; i++) {
    if (i % width == (width-1) /*right border*/ || i % width == 0 /*left border*/
    || i < width /*top row*/ || i > width*(height - 1) /*bottom row*/)
    {
        } else {

            int c = cam.pixels[i];
            r = (c>>16)&0x000000FF;
            g = (c>>8)&0x000000FF;
            b = c&0x000000FF;
            int average = (r + g + b )/3;
            int uL = cam.pixels[i-(width+1)]; /*up left RGB*/
            int rUL=(uL>>16)&0x000000FF;
            int gUL=(uL>>8)&0x000000FF;
            int bUL=uL&0x000000FF;

            int u = cam.pixels[i-width]; /*up RGB*/
            int rU=(u>>16)&0x000000FF;
            int gU=(u>>8)&0x000000FF;
            int bU=u&0x000000FF;

            int uR = cam.pixels[i-(width-1)]; /*up right RGB*/
            int rUR=(uR>>16)&0x000000FF;
            int gUR=(uR>>8)&0x000000FF;
            int bUR=uR&0x000000FF;

            int l = cam.pixels[i-1]; /*left RGB*/
            int rL=(l>>16)&0x000000FF;
            int gL=(l>>8)&0x000000FF;
            int bL=l&0x000000FF;

            int ri = cam.pixels[i+1]; /*right RGB*/
            int rRi=(ri>>16)&0x000000FF;
            int gRi=(ri>>8)&0x000000FF;
            int bRi=ri&0x000000FF;

            int dL = cam.pixels[i+(width-1)]; /*down left RGB*/
            int rDL=(dL>>16)&0x000000FF;
            int gDL=(dL>>8)&0x000000FF;
            int bDL=dL&0x000000FF;

            int d = cam.pixels[i+width]; /*down RGB*/
            int rD=(d>>16)&0x000000FF;
            int gD=(d>>8)&0x000000FF;
            int bD=d&0x000000FF;

```

```

int dR = cam.pixels[i+width]; /*down right RGB*/
int rDR=(dR>>16)&0x000000FF;
int gDR=(dR>>8)&0x000000FF;
int bDR=dR&0x000000FF;

if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d == dL
&& dL == l && l == cam.pixels[i]) {
} else if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d
== dL && dL == l && l != cam.pixels[i]) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (uL == u && u == uR && uR == l && ri == rD && rD == d && d
== dL && dL != l && l != cam.pixels[i] && ri != cam.pixels[i]) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
        g = gL;
    }
} else if (uL != u && u == uR && uR == l && ri != rD && rD == d && d
== dL && dL == l && l != cam.pixels[i] && ri != cam.pixels[i]) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
        g = gL;
    }
}
} else if (u == uL && uL == l && l == dL && dL == d && cam.pixels[i]
!= l) {
    r = rL;
    b = bL;
    g = gL;
} else if (u == uR && uR == ri && ri == dR && dR == d && cam.pixels[i]
!= ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == uR && uR == u && u == uL && uL == l && cam.pixels[i]
!= ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == dR && dR == d && d == dL && dL == l && cam.pixels[i]
!= ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == dR && dR == d && cam.pixels[i] != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == uR && uR == u && cam.pixels[i] != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (l == dL && dL == d && cam.pixels[i] != l) {
    r = rRi;
    b = bRi;
}

```

```

        g = gRi;
    } else if (l == uL && uL == u && cam.pixels[i] != l) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else if (cam.pixels[i] != uL && cam.pixels[i] != u && cam.pixels[i]
!= uR && cam.pixels[i] != ri && cam.pixels[i] != rD && cam.pixels[i] != d &&
cam.pixels[i] != dL && cam.pixels[i] != l) {
        if (random(10) < 5) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else {
            r = rL;
            b = bL;
            g = gL;
        }
    }

    cam.pixels[i]=(r<<16) | (g<<8) | b;

    if (average != 0) {
        if (abs((rU + gU + bU)/3 - average) > 20) {
            rU = 0;
            gU = 0;
            bU = 0;
            cam.pixels[i-width]=(rU<<16) | (gU<<8) | bU;
        }if (abs((rUL + gUL + bUL)/3 - average) > 20) {
            rUL = 0;
            gUL = 0;
            bUL = 0;
            cam.pixels[i-(width+1)]=(rUL<<16) | (gUL<<8) | bUL;
        } else if (abs((rL + gL + bL)/3 - average) > 20) {
            rL = 0;
            gL = 0;
            bL = 0;
            cam.pixels[i-1]=(rL<<16) | (gL<<8) | bL;
        } else if (abs((rRi + gRi + bRi)/3 - average) > 20) {
            rRi = 0;
            gRi = 0;
            bRi = 0;
            cam.pixels[i+1]=(rRi<<16) | (gRi<<8) | bRi;
        }if (abs((rUR + gUR + bUR)/3 - average) > 20) {
            rUR = 0;
            gUR = 0;
            bUR = 0;
            cam.pixels[i+1]=(rUR<<16) | (gUR<<8) | bRi;
        } else if (abs((rD + gD + bD)/3 - average) > 20) {
            rD = 0;
            gD = 0;
            bD = 0;
            cam.pixels[i-(width-1)]=(rD<<16) | (gD<<8) | bD;
        } else if (abs((rDL + gDL + bDL)/3 - average) > 20) {
            rDL = 0;
            gDL = 0;
            bDL = 0;
            cam.pixels[i+(width-1)]=(rDL<<16) | (gDL<<8) | bDL;
        } else if (abs((rDR + gDR + bDR)/3 - average) > 20) {
            rDR = 0;
            gDR = 0;
            bDR = 0;
            cam.pixels[i+width]=(rDR<<16) | (gDR<<8) | bDR;
        }
    }
}

```

	<pre>         }          image(cam, 0, 0);     } } </pre>
<b>May. 10</b> <b>Code Entry</b>  Found minute errors in code that contributed to color blob outlining to not work correctly  <i>Cartoon Cam</i>	<pre> import processing.video.*;  Capture cam;  void setup() {      size(1280, 720);     background(0);     String[] cameras = Capture.list();     if (cameras.length == 0) {         println("There are no cameras available for capture.");         exit();     } else {         println("Available cameras:");         for (int i = 0; i &lt; cameras.length; i++) {             println(cameras[i]);         }         cam = new Capture(this, cameras[0]);         cam.start();     } }  void draw() {     int r = 0;     int g = 0;     int b = 0;      cam.updatePixels();     if (cam.available() == true) {         cam.read();     }     cam.loadPixels();     for (int i = 0; i &lt; cam.pixels.length; i++) {         int c = cam.pixels[i];         r = (c&gt;&gt;16)&amp;0x000000FF;         g = (c&gt;&gt;8)&amp;0x000000FF;         b = c&amp;0x000000FF;          r = (round((r/55)*25.6))+42;         g = (round((g/40)*25.6))+34;         b = (round((b/50)*25.6))+52;          cam.pixels[i]=(r&lt;&lt;16) (g&lt;&lt;8) b;     }      for (int i = 0; i &lt; cam.pixels.length; i++) {         if (i % width == (width-1) /*right border*/    i % width == 0 /*left border*/    i &lt; width /*top row*/    i &gt; width*(height - 1) /*bottom row*/){         } else {              int c = cam.pixels[i];             r = (c&gt;&gt;16)&amp;0x000000FF;             g = (c&gt;&gt;8)&amp;0x000000FF;             b = c&amp;0x000000FF;             int average = (r + g + b )/3;              int uL = cam.pixels[i-(width+1)]; /*up left RGB*/             int rUL=(uL&gt;&gt;16)&amp;0x000000FF; </pre>

```

int gUL=(uL>>8)&0x000000FF;
int bUL=uL&0x000000FF;

int u = cam.pixels[i-width]; /*up RGB*/
int rU=(u>>16)&0x000000FF;
int gU=(u>>8) &0x000000FF;
int bU=u&0x000000FF;

int uR = cam.pixels[i-(width-1)]; /*up right RGB*/
int rUR=(uR>>16)&0x000000FF;
int gUR=(uR>>8)&0x000000FF;
int bUR=uR&0x000000FF;

int l = cam.pixels[i-1]; /*left RGB*/
int rL=(l>>16)&0x000000FF;
int gL=(l>>8) &0x000000FF;
int bL=l&0x000000FF;

int ri = cam.pixels[i+1]; /*right RGB*/
int rRi=(ri>>16)&0x000000FF;
int gRi=(ri>>8)&0x000000FF;
int bRi=ri&0x000000FF;

int dL = cam.pixels[i+(width-1)]; /*down left RGB*/
int rDL=(dL>>16)&0x000000FF;
int gDL=(dL>>8)&0x000000FF;
int bDL=dL&0x000000FF;

int d = cam.pixels[i+width]; /*down RGB*/
int rD=(d>>16)&0x000000FF;
int gD=(d>>8) &0x000000FF;
int bD=d&0x000000FF;

int dR = cam.pixels[i+(width+1)]; /*down right RGB*/
int rDR=(dR>>16)&0x000000FF;
int gDR=(dR>>8)&0x000000FF;
int bDR=dR&0x000000FF;

if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d == dL
&& dL == l && l == cam.pixels[i]) {
} else if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d
== dL && dL == l && l != cam.pixels[i]) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (uL == u && u == uR && uR == l && ri == rD && rD == d && d
== dL && dL != l && l != cam.pixels[i] && ri != cam.pixels[i]) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
        g = gL;
    }
} else if (uL != u && u == uR && uR == l && ri != rD && rD == d && d
== dL && dL == l && l != cam.pixels[i] && ri != cam.pixels[i]) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
    }
}

```

```

        g = gL;
    }
} else if (u == uL && uL == l && l == dL && dL == d && cam.pixels[i] != 1) {
    r = rL;
    b = bL;
    g = gL;
} else if (u == uR && uR == ri && ri == dR && dR == d && cam.pixels[i] != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == uR && uR == u && u == uL && uL == l && cam.pixels[i] != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == dR && dR == d && d == dL && dL == l && cam.pixels[i] != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == dR && dR == d && cam.pixels[i] != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == uR && uR == u && cam.pixels[i] != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (l == dL && dL == d && cam.pixels[i] != l) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (l == uL && uL == u && cam.pixels[i] != l) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (cam.pixels[i] != uL && cam.pixels[i] != u && cam.pixels[i] != uR && cam.pixels[i] != ri && cam.pixels[i] != rD && cam.pixels[i] != d && cam.pixels[i] != dL && cam.pixels[i] != l) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
        g = gL;
    }
}
cam.pixels[i]=(r<<16) | (g<<8) | b;

if (average != 0) {
    if (abs((rU + gU + bU)/3 - average) > 1 && (rU + gU + bU) != 0) {
        rU = 0;
        gU = 0;
        bU = 0;
        cam.pixels[i-width]=(rU<<16) | (gU<<8) | bU;
    } else if (abs((rUL + gUL + bUL)/3 - average) > 1 && (rUL + gUL + bUL) != 0) {
        rUL = 0;
        gUL = 0;
        bUL = 0;
    }
}

```

```

        cam.pixels[i-(width+1)]=(rUL<<16) | (gUL<<8) | bUL;
    } else if (abs((rL + gL + bL)/3 - average) > 1 && (rL + gL + bL) != 0) {
        rL = 0;
        gL = 0;
        bL = 0;
        cam.pixels[i-1]=(rL<<16) | (gL<<8) | bL;
    } else if (abs((rRi + gRi + bRi)/3 - average) > 1 && (rRi + gRi + bRi) != 0) {
        rRi = 0;
        gRi = 0;
        bRi = 0;
        cam.pixels[i+1]=(rRi<<16) | (gRi<<8) | bRi;
    } else if (abs((rUR + gUR + bUR)/3 - average) > 1 && (rUR + gUR + bUR) != 0) {
        rUR = 0;
        gUR = 0;
        bUR = 0;
        cam.pixels[i+1]=(rUR<<16) | (gUR<<8) | bRi;
    } else if (abs((rD + gD + bD)/3 - average) > 1 && (rD + gD + bD) != 0) {
        rD = 0;
        gD = 0;
        bD = 0;
        cam.pixels[i-(width-1)]=(rD<<16) | (gD<<8) | bD;
    } else if (abs((rDL + gDL + bDL)/3 - average) > 1 && (rDL + gDL + bDL) != 0) {
        rDL = 0;
        gDL = 0;
        bDL = 0;
        cam.pixels[i+(width-1)]=(rDL<<16) | (gDL<<8) | bDL;
    } else if (abs((rDR + gDR + bDR)/3 - average) > 1 && (rDR + gDR + bDR) != 0) {
        rDR = 0;
        gDR = 0;
        bDR = 0;
        cam.pixels[i+(width+1)]=(rDR<<16) | (gDR<<8) | bDR;
    }
}
}

image(cam, 0, 0);
}

```

May. 10  
**Code Entry**

Tweaked code  
a bit to look  
a better with  
thicker  
lines.

*Cartoon Cam*

```

import processing.video.*;

Capture cam;

void setup() {
    size(1280, 720);
    background(0);
    String[] cameras = Capture.list();
    if (cameras.length == 0) {
        println("There are no cameras available for capture.");
        exit();
    } else {
        println("Available cameras:");
        for (int i = 0; i < cameras.length; i++) {
            println(cameras[i]);
        }
        cam = new Capture(this, cameras[0]);
    }
}

```

```

        cam.start();
    }
}

void draw() {
    int r = 0;
    int g = 0;
    int b = 0;

    cam.updatePixels();
    if (cam.available() == true) {
        cam.read();
    }
    cam.loadPixels();
    for (int i = 0; i < cam.pixels.length; i++) {
        int c = cam.pixels[i];
        r = (c>>16)&0x000000FF;
        g = (c>>8)&0x000000FF;
        b = c&0x000000FF;

        r = (round((r/55)*25.6))+42;
        g = (round((g/40)*25.6))+34;
        b = (round((b/50)*25.6))+52;

        cam.pixels[i]=(r<<16) | (g<<8) | b;
    }

    for (int i = 0; i < cam.pixels.length; i++) {
        if (i % width == (width-1) /*right border*/ || i % width == 0 /*left border*/
            || i < width /*top row*/ || i > width*(height - 1) /*bottom row*/)
        {
        } else {

            int c = cam.pixels[i];
            r = (c>>16)&0x000000FF;
            g = (c>>8)&0x000000FF;
            b = c&0x000000FF;
            int average = (r + g + b )/3;

            int uL = cam.pixels[i-(width+1)]; /*up left RGB*/
            int rUL=(uL>>16)&0x000000FF;
            int gUL=(uL>>8)&0x000000FF;
            int bUL=uL&0x000000FF;

            int u = cam.pixels[i-width]; /*up RGB*/
            int rU=(u>>16)&0x000000FF;
            int gU=(u>>8)&0x000000FF;
            int bU=u&0x000000FF;

            int uR = cam.pixels[i-(width-1)]; /*up right RGB*/
            int rUR=(uR>>16)&0x000000FF;
            int gUR=(uR>>8)&0x000000FF;
            int bUR=uR&0x000000FF;

            int l = cam.pixels[i-1]; /*left RGB*/
            int rL=(l>>16)&0x000000FF;
            int gL=(l>>8)&0x000000FF;
            int bL=l&0x000000FF;

            int ri = cam.pixels[i+1]; /*right RGB*/
            int rRi=(ri>>16)&0x000000FF;
            int gRi=(ri>>8)&0x000000FF;
            int bRi=ri&0x000000FF;
        }
    }
}

```

```

int dL = cam.pixels[i+(width-1)]; /*down left RGB*/
int rDL=(dL>>16)&0x000000FF;
int gDL=(dL>>8)&0x000000FF;
int bDL=dL&0x000000FF;

int d = cam.pixels[i+width]; /*down RGB*/
int rD=(d>>16)&0x000000FF;
int gD=(d>>8)&0x000000FF;
int bD=d&0x000000FF;

int dR = cam.pixels[i+(width+1)]; /*down right RGB*/
int rDR=(dR>>16)&0x000000FF;
int gDR=(dR>>8)&0x000000FF;
int bDR=dR&0x000000FF;
if (average != 0) {
    if (abs((rU + gU + bU)/3 - average) > 9 && (rU + gU + bU) != 0) {
        rU = 0;
        gU = 0;
        bU = 0;
        cam.pixels[i-width]=(rU<<16) | (gU<<8) | bU;
    } else if (abs((rUL + gUL + bUL)/3 - average) > 9 && (rUL + gUL +
bUL) != 0) {
        rUL = 0;
        gUL = 0;
        bUL = 0;
        cam.pixels[i-(width+1)]=(rUL<<16) | (gUL<<8) | bUL;
    } else if (abs((rL + gL + bL)/3 - average) > 9 && (rL + gL + bL) !=
0) {
        rL = 0;
        gL = 0;
        bL = 0;
        cam.pixels[i-1]=(rL<<16) | (gL<<8) | bL;
    } else if (abs((rRi + gRi + bRi)/3 - average) > 9 && (rRi + gRi +
bRi) != 0) {
        rRi = 0;
        gRi = 0;
        bRi = 0;
        cam.pixels[i+1]=(rRi<<16) | (gRi<<8) | bRi;
    } else if (abs((rUR + gUR + bUR)/3 - average) > 9 && (rUR + gUR +
bUR) != 0) {
        rUR = 0;
        gUR = 0;
        bUR = 0;
        cam.pixels[i+1]=(rUR<<16) | (gUR<<8) | bUR;
    } else if (abs((rD + gD + bD)/3 - average) > 9 && (rD + gD + bD) !=
0) {
        rD = 0;
        gD = 0;
        bD = 0;
        cam.pixels[i-(width-1)]=(rD<<16) | (gD<<8) | bD;
    } else if (abs((rDL + gDL + bDL)/3 - average) > 9 && (rDL + gDL +
bDL) != 0) {
        rDL = 0;
        gDL = 0;
        bDL = 0;
        cam.pixels[i+(width-1)]=(rDL<<16) | (gDL<<8) | bDL;
    } else if (abs((rDR + gDR + bDR)/3 - average) > 9 && (rDR + gDR +
bDR) != 0) {
        rDR = 0;
        gDR = 0;
        bDR = 0;
        cam.pixels[i+(width+1)]=(rDR<<16) | (gDR<<8) | bDR;
    }
}
if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d == dL

```

```

    && dL == l && l == cam.pixels[i]) {
        } else if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d
== dL && dL == l && l != cam.pixels[i]) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else if (uL == u && u == uR && uR == l && ri == rD && rD == d && d
== dL && dL != l && l != cam.pixels[i] && ri != cam.pixels[i]) {
            if (random(10) < 5) {
                r = rRi;
                b = bRi;
                g = gRi;
            } else {
                r = rL;
                b = bL;
                g = gL;
            }
        } else if (uL != u && u == uR && uR == l && ri != rD && rD == d && d
== dL && dL == l && l != cam.pixels[i] && ri != cam.pixels[i]) {
            if (random(10) < 5) {
                r = rRi;
                b = bRi;
                g = gRi;
            } else {
                r = rL;
                b = bL;
                g = gL;
            }
        } else if (u == uL && uL == l && l == dL && dL == d && cam.pixels[i]
!= l) {
            r = rL;
            b = bL;
            g = gL;
        } else if (u == uR && uR == ri && ri == dR && dR == d && cam.pixels[i]
!= ri) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else if (ri == uR && uR == u && u == uL && uL == l && cam.pixels[i]
!= ri) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else if (ri == dR && dR == d && d == dL && dL == l && cam.pixels[i]
!= ri) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else if (ri == dR && dR == d && cam.pixels[i] != ri) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else if (ri == uR && uR == u && cam.pixels[i] != ri) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else if (l == dL && dL == d && cam.pixels[i] != l) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else if (l == uL && uL == u && cam.pixels[i] != l) {
            r = rRi;
            b = bRi;
            g = gRi;
        } else if (cam.pixels[i] != uL && cam.pixels[i] != u && cam.pixels[i]

```

```

!= uR && cam.pixels[i] != ri && cam.pixels[i] != rD && cam.pixels[i] != d &&
cam.pixels[i] != dL && cam.pixels[i] != l) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
        g = gL;
    }
}

cam.pixels[i]=(r<<16) | (g<<8) | b;

if (average != 0) {
    if (abs((rU + gU + bU)/3 - average) > 8 && (rU + gU + bU) != 0) {
        rU = 0;
        gU = 0;
        bU = 0;
        cam.pixels[i-width]=(rU<<16) | (gU<<8) | bU;
    } else if (abs((rUL + gUL + bUL)/3 - average) > 8 && (rUL + gUL +
bUL) != 0) {
        rUL = 0;
        gUL = 0;
        bUL = 0;
        cam.pixels[i-(width+1)]=(rUL<<16) | (gUL<<8) | bUL;
    } else if (abs((rL + gL + bL)/3 - average) > 8 && (rL + gL + bL) !=
0) {
        rL = 0;
        gL = 0;
        bL = 0;
        cam.pixels[i-1]=(rL<<16) | (gL<<8) | bL;
    } else if (abs((rRi + gRi + bRi)/3 - average) > 8 && (rRi + gRi +
bRi) != 0) {
        rRi = 0;
        gRi = 0;
        bRi = 0;
        cam.pixels[i+1]=(rRi<<16) | (gRi<<8) | bRi;
    } else if (abs((rUR + gUR + bUR)/3 - average) > 8 && (rUR + gUR +
bUR) != 0) {
        rUR = 0;
        gUR = 0;
        bUR = 0;
        cam.pixels[i+1]=(rUR<<16) | (gUR<<8) | bRi;
    } else if (abs((rD + gD + bD)/3 - average) > 8 && (rD + gD + bD) !=
0) {
        rD = 0;
        gD = 0;
        bD = 0;
        cam.pixels[i-(width-1)]=(rD<<16) | (gD<<8) | bD;
    } else if (abs((rDL + gDL + bDL)/3 - average) > 8 && (rDL + gDL +
bDL) != 0) {
        rDL = 0;
        gDL = 0;
        bDL = 0;
        cam.pixels[i+(width-1)]=(rDL<<16) | (gDL<<8) | bDL;
    } else if (abs((rDR + gDR + bDR)/3 - average) > 8 && (rDR + gDR +
bDR) != 0) {
        rDR = 0;
        gDR = 0;
        bDR = 0;
        cam.pixels[i+(width+1)]=(rDR<<16) | (gDR<<8) | bDR;
    }
}
}

```

	<pre>         }     }      image(cam, 0, 0); } </pre>
<u>May. 10</u> <b>Code Entry</b>  After Surveying People I Found The best I Could Do As Far As Outlining goes (Lines Are Less Thick Than I like, But There Is Less Noise).  <i>Cartoon Cam</i>	<pre> import processing.video.*;  Capture cam;  void setup() {     size(1280, 720);     background(0);     String[] cameras = Capture.list();     if (cameras.length == 0) {         println("There are no cameras available for capture.");         exit();     } else {         println("Available cameras:");         for (int i = 0; i &lt; cameras.length; i++) {             println(cameras[i]);         }     }     cam = new Capture(this, cameras[0]);     cam.start(); }  void draw() {     int r = 0;     int g = 0;     int b = 0;      cam.updatePixels();     if (cam.available() == true) {         cam.read();     }     cam.loadPixels();     for (int i = 0; i &lt; cam.pixels.length; i++) {         int c = cam.pixels[i];         r = (c&gt;&gt;16)&amp;0x000000FF;         g = (c&gt;&gt;8)&amp;0x000000FF;         b = c&amp;0x000000FF;          r = (round((r/55)*25.6))+42;         g = (round((g/40)*25.6))+34;         b = (round((b/50)*25.6))+52;          cam.pixels[i]=(r&lt;&lt;16)   (g&lt;&lt;8)   b;     }      for (int i = 0; i &lt; cam.pixels.length; i++) {         if (i % width == (width-1) /*right border*/    i % width == 0 /*left border*/    i &lt; width /*top row*/    i &gt; width*(height - 1) /*bottom row*/)         {         } else {              int c = cam.pixels[i];             r = (c&gt;&gt;16)&amp;0x000000FF;             g = (c&gt;&gt;8)&amp;0x000000FF;             b = c&amp;0x000000FF;             int average = (r + g + b )/3;              int uL =  cam.pixels[i-(width+1)]; /*up left RGB*/         }     } } </pre>

```

int rUL=(uL>>16)&0x000000FF;
int gUL=(uL>>8)&0x000000FF;
int bUL=uL&0x000000FF;

int u = cam.pixels[i-width]; /*up RGB*/
int rU=(u>>16)&0x000000FF;
int gU=(u>>8)&0x000000FF;
int bU=u&0x000000FF;

int uR = cam.pixels[i-(width-1)]; /*up right RGB*/
int rUR=(uR>>16)&0x000000FF;
int gUR=(uR>>8)&0x000000FF;
int bUR=uR&0x000000FF;

int l = cam.pixels[i-1]; /*left RGB*/
int rL=(l>>16)&0x000000FF;
int gL=(l>>8)&0x000000FF;
int bL=l&0x000000FF;

int ri = cam.pixels[i+1]; /*right RGB*/
int rRi=(ri>>16)&0x000000FF;
int gRi=(ri>>8)&0x000000FF;
int bRi=ri&0x000000FF;

int dL = cam.pixels[i+(width-1)]; /*down left RGB*/
int rDL=(dL>>16)&0x000000FF;
int gDL=(dL>>8)&0x000000FF;
int bDL=dL&0x000000FF;

int d = cam.pixels[i+width]; /*down RGB*/
int rD=(d>>16)&0x000000FF;
int gD=(d>>8)&0x000000FF;
int bD=d&0x000000FF;

int dR = cam.pixels[i+(width+1)]; /*down right RGB*/
int rDR=(dR>>16)&0x000000FF;
int gDR=(dR>>8)&0x000000FF;
int bDR=dR&0x000000FF;

if (average != 0) {
    if (abs((rU + gU + bU)/3 - average) > 9 && (rU + gU + bU) != 0) {
        rU = 0;
        gU = 0;
        bU = 0;
        cam.pixels[i-width]=(rU<<16) | (gU<<8) | bU;
    } else if (abs((rUL + gUL + bUL)/3 - average) > 9 && (rUL + gUL + bUL) != 0) {
        rUL = 0;
        gUL = 0;
        bUL = 0;
        cam.pixels[i-(width+1)]=(rUL<<16) | (gUL<<8) | bUL;
    } else if (abs((rL + gL + bL)/3 - average) > 9 && (rL + gL + bL) != 0) {
        rL = 0;
        gL = 0;
        bL = 0;
        cam.pixels[i-1]=(rL<<16) | (gL<<8) | bL;
    } else if (abs((rRi + gRi + bRi)/3 - average) > 9 && (rRi + gRi + bRi) != 0) {
        rRi = 0;
        gRi = 0;
        bRi = 0;
        cam.pixels[i+1]=(rRi<<16) | (gRi<<8) | bRi;
    } else if (abs((rUR + gUR + bUR)/3 - average) > 9 && (rUR + gUR + bUR) != 0) {

```

```

        rUR = 0;
        gUR = 0;
        bUR = 0;
        cam.pixels[i-(width-1)]=(rUR<<16) | (gUR<<8) | bUR;
    } else if (abs((rD + gD + bD)/3 - average) > 9 && (rD + gD + bD)
!=0) {
        rD = 0;
        gD = 0;
        bD = 0;
        cam.pixels[i-(width-1)]=(rD<<16) | (gD<<8) | bD;
    } else if (abs((rDL + gDL + bDL)/3 - average) > 9 && (rDL + gDL +
bDL) != 0) {
        rDL = 0;
        gDL = 0;
        bDL = 0;
        cam.pixels[i+(width-1)]=(rDL<<16) | (gDL<<8) | bDL;
    } else if (abs((rDR + gDR + bDR)/3 - average) > 9 && (rDR + gDR +
bDR) != 0) {
        rDR = 0;
        gDR = 0;
        bDR = 0;
        cam.pixels[i+(width+1)]=(rDR<<16) | (gDR<<8) | bDR;
    }
}
if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d == dL
&& dL == l && l == c) {
} else if (uL == u && u == uR && uR == ri && ri == rD && rD == d && d
== dL && dL == l && l != c) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (uL == u && u == uR && uR == l && ri == rD && rD == d && d
== dL && dL != l && l != c && ri != c) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
        g = gL;
    }
} else if (uL != u && u == uR && uR == l && ri != rD && rD == d && d
== dL && dL == l && l != c && ri != c) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
        g = gL;
    }
} else if (u == uL && uL == l && l == dL && dL == d && c != l) {
    r = rL;
    b = bL;
    g = gL;
} else if (u == uR && uR == ri && ri == dR && dR == d && c != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == uR && uR == u && u == uL && uL == l && c != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
}

```

```

} else if (ri == dR && dR == d && d == dL && dL == l && c != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == dR && dR == d && c != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (ri == uR && uR == u && c != ri) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (l == dL && dL == d && c != l) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (l == uL && uL == u && c != l) {
    r = rRi;
    b = bRi;
    g = gRi;
} else if (c != uL && c != u && c != uR && c != ri && c != rD && c != d && c != dL && c != l) {
    if (random(10) < 5) {
        r = rRi;
        b = bRi;
        g = gRi;
    } else {
        r = rL;
        b = bL;
        g = gL;
    }
}
}

cam.pixels[i]=(r<<16) | (g<<8) | b;

if (average != 0) {
    if (abs((rU + gU + bU)/3 - average) > 9 && (rU + gU + bU) != 0) {
        rU = 0;
        gU = 0;
        bU = 0;
        cam.pixels[i-width]=(rU<<16) | (gU<<8) | bU;
    } else if (abs((rUL + gUL + bUL)/3 - average) > 9 && (rUL + gUL + bUL) != 0) {
        rUL = 0;
        gUL = 0;
        bUL = 0;
        cam.pixels[i-(width+1)]=(rUL<<16) | (gUL<<8) | bUL;
    } else if (abs((rL + gL + bL)/3 - average) > 9 && (rL + gL + bL) != 0) {
        rL = 0;
        gL = 0;
        bL = 0;
        cam.pixels[i-1]=(rL<<16) | (gL<<8) | bL;
    } else if (abs((rRi + gRi + bRi)/3 - average) > 9 && (rRi + gRi + bRi) != 0) {
        rRi = 0;
        gRi = 0;
        bRi = 0;
        cam.pixels[i+1]=(rRi<<16) | (gRi<<8) | bRi;
    } else if (abs((rUR + gUR + bUR)/3 - average) > 9 && (rUR + gUR + bUR) != 0) {
        rUR = 0;
        gUR = 0;
        bUR = 0;
        cam.pixels[i-(width-1)]=(rUR<<16) | (gUR<<8) | bRi;
    }
}
}

```

```
        } else if (abs((rD + gD + bD)/3 - average) > 9 && (rD + gD + bD)
!=0) {
    rD = 0;
    gD = 0;
    bD = 0;
    cam.pixels[i-(width-1)]=(rD<<16) | (gD<<8) | bD;
} else if (abs((rDL + gDL + bDL)/3 - average) > 9 && (rDL + gDL +
bDL) != 0) {
    rDL = 0;
    gDL = 0;
    bDL = 0;
    cam.pixels[i+(width-1)]=(rDL<<16) | (gDL<<8) | bDL;
} else if (abs((rDR + gDR + bDR)/3 - average) > 9 && (rDR + gDR +
bDR) != 0) {
    rDR = 0;
    gDR = 0;
    bDR = 0;
    cam.pixels[i+(width+1)]=(rDR<<16) | (gDR<<8) | bDR;
}
}
image(cam, 0, 0);
}
```