

# HEADLIGHTS\_ON



**By: Danny Ng Liu and Farid Fattahli**

# OUR DMV RULE

It is very dangerous to drive in the night or in places where there are no visibility without headlights on. Without headlights, it can make it difficult to see the traffic lights, pedestrians and other cars coming your way and can inflict harm if you are careless. Having a headlights on can be the difference between life or death of others or yourself.

# CONTROL AND TIMER VARIABLES

Variables containing the word speed controls the speed of the dashes, trees, stars and clouds.

MinusOne and the timers variables are created to simulate a makeshift timer.

```
float speedOfDash = 3;  
float speedOfTrees = 1;  
float speedOfStars = 0.2;  
float speedOfClouds = 0.4;  
float minusOne = 500;
```

```
float timer = 1;  
float timer2 = 1;  
float timer3 = 1222;  
//float timer4 = 1222;
```

# POSITION OF TREES, CLOUDS, DASHES AND STARS

Most of these variables contain the x-axis values which later then can be used to update and these shapes.

```
float dash1 = 40;
float dash1q = 35;
float dash1w = 85;
float dash1e = 90;

float dash2 = 130;
float dash2q = 125;
float dash2w = 175;
float dash2e = 180;

float dash3 = 220;
float dash3q = 215;
float dash3w = 265;
float dash3e = 270;

float dash4 = 310;
float dash4q = 305;
float dash4w = 355;
float dash4e = 360;

float dash5 = 410;
float dash5q = 405;
float dash5w = 455;
float dash5e = 460;

float a = 157;
float b = 211;
float c = 245;
float d = 255;

float treeLog1 = 300;
float tree1q = 290;
float tree1w = 310;
float tree1e = 330;

float treeLog2 = 50;
float tree2 = 59;

float treeLog3 = 200;
float tree3 = 210;

float treeLog4 = 410;
float tree4 = 390;

float white = 1;
float white2 = 255;

float star1 = 123;
float star2 = 321;
float star3 = 232;
float star4 = 353;
float star5 = 21;
float star6 = 167;
float star7 = 297;
float star8 = 430;
float star9 = 452;
float star10 = 470;
float star11 = 512;
float star12 = 378;
float star13 = 312;
float star14 = 296;

float cloud1 = 200;
float cloud2 = 180;
float cloud3 = 240;
float cloud4 = 260;
float cloud5 = 340;
float cloud6 = 360;
float cloud7 = 40;
float cloud8 = 60;
float cloud9 = 440;
float cloud10 = 460;
float cloud11 = 550;
float cloud12 = 570;
float cloud13 = 680;
float cloud14 = 700;
float cloud15 = 750;
float cloud16 = 770;
```

# HOW THE BACKGROUND CHANGES PT.1

It starts off with the program going from broad daylight slowly darkening. The background is being changed by multiple 0.995 with variables a,b,c and d decreasing the background color changing it to black. Same with the clouds, the clouds are getting darker because the color is being multiplied by 0.9949. The rest of the code just created the clouds and changes the position, moving the clouds to the left slowly.

```
} else {  
  
  a = a*0.995;  
  b = b*0.995;  
  c = c*0.995;  
  d = d*0.995;  
  background(a,b,c,d);  
  
  white2 = white2*0.9949;  
  fill(white2,white2,white2);  
  arc(cloud1, 100, 40, 40, PI, TWO_PI);  
  arc(cloud2, 100, 40, 40, PI, TWO_PI);  
  arc(cloud3, 60, 40, 40, PI, TWO_PI);  
  arc(cloud4, 60, 40, 40, PI, TWO_PI);  
  arc(cloud5, 140, 40, 40, PI, TWO_PI);  
  arc(cloud6, 140, 40, 40, PI, TWO_PI);  
  arc(cloud7, 30, 40, 40, PI, TWO_PI);  
  arc(cloud8, 30, 40, 40, PI, TWO_PI);  
  arc(cloud9, 40, 40, 40, PI, TWO_PI);  
  arc(cloud10, 40, 40, 40, PI, TWO_PI);  
  arc(cloud11, 110, 40, 40, PI, TWO_PI);  
  arc(cloud12, 110, 40, 40, PI, TWO_PI);  
  arc(cloud13, 150, 40, 40, PI, TWO_PI);  
  arc(cloud14, 150, 40, 40, PI, TWO_PI);  
  arc(cloud15, 60, 40, 40, PI, TWO_PI);  
  arc(cloud16, 60, 40, 40, PI, TWO_PI);  
  
  cloud1 -= speedOfClouds;  
  cloud2 -= speedOfClouds;  
  cloud3 -= speedOfClouds;  
  cloud4 -= speedOfClouds;  
  cloud5 -= speedOfClouds;  
  cloud6 -= speedOfClouds;  
  cloud7 -= speedOfClouds;  
  cloud8 -= speedOfClouds;  
  cloud9 -= speedOfClouds;  
  cloud10 -= speedOfClouds;  
  cloud11 -= speedOfClouds;  
  cloud12 -= speedOfClouds;  
  cloud13 -= speedOfClouds;  
  cloud14 -= speedOfClouds;  
  cloud15 -= speedOfClouds;  
  cloud16 -= speedOfClouds;
```

# HOW THE BACKGROUND CHANGES PT.2

Remember from before? The variable `minusOne` is set to 500 and it has been slowly decreasing due to `minusOne -= 1`. Once it gets below 0, it will set the background to black and the stars start to come out. White is being updated by multiplying it by 1.05 to increase the whiteness. The variable `timer` is then used as an makeshift timer. The rest of the code is used to create the stars and change the position to move it to the left.

(`timer3` was set to 1222; and is decreasing by 1)

```
minusOne -= 1;
timer3 -= 1;
//timer4 -= 1;

//System.out.println("the timer4 is: " + timer4);
if (minusOne < 0) {
    background(0);
    timer = timer*1.001;
    //stars
    white = white*1.05;
    fill(white,white,white);
    circle(star1, 83, 3);
    circle(star2, 152, 4);
    circle(star3,112,5);
    circle(star4, 21, 6);
    circle(star5, 175, 7);
    circle(star6,54,3);
    circle(star7,87,4);
    circle(star8, 11, 5);
    circle(star9,179,6);
    circle(star10,59,7);
    circle(star11,92,3);
    circle(star12,56,4);
    circle(star13,164,5);
    circle(star14,30,6);

    star1 -= speedOfStars;
    star2 -= speedOfStars;
    star3 -= speedOfStars;
    star4 -= speedOfStars;
    star5 -= speedOfStars;
    star6 -= speedOfStars;
    star7 -= speedOfStars;
    star8 -= speedOfStars;
    star9 -= speedOfStars;
    star10 -= speedOfStars;
    star11 -= speedOfStars;
    star12 -= speedOfStars;
    star13 -= speedOfStars;
    star14 -= speedOfStars;
```

# HOW THE BACKGROUND CHANGES PT.3

The if-statements on the left are used to detect if the clouds are outside of the display (400,400). If it is true, the clouds will be set to a certain x-axis value which is then slowly moved to the left to create this infinite loop.

```
if(cloud8 < -10) {  
    cloud8 = 820;  
    cloud7 = 800;  
}  
if(cloud1 < -10) {  
    cloud1 = 870;  
    cloud2 = 850;  
}  
if(cloud4 < -10) {  
    cloud4 = 920;  
    cloud3 = 900;  
}  
if(cloud6 < -10) {  
    cloud6 = 970;  
    cloud5 = 950;  
}  
if(cloud10 < -10) {  
    cloud10 = 1020;  
    cloud9 = 1000;  
}  
if(cloud12 < -10) {  
    cloud12 = 1070;  
    cloud11 = 1050;  
}  
if(cloud14 < -10) {  
    cloud14 = 1120;  
    cloud13 = 1100;  
}  
if(cloud16 < -10) {  
    cloud16 = 1170;  
    cloud15 = 1150;  
}
```

# HOW THE BACKGROUND CHANGES PT. 4

We are back to the timer from the beginning of this big if-statement. Once the timer is greater than 2, the background will slowing recover and go back to the broad daylight setting. The stars are then going to slowly disappear from the background. This is when timer3 is activated because it is less than 0. It will set white to 255 because without this, white is some large number because the if-statements before are still running. Without including this will take a long unexpected time to change the color of the stars. It will also set timer3 to 99999999 so that timer3 won't activate again anytime soon. I call this a "One time activation".

```
System.out.println(timer);
if(timer > 2) {
    System.out.println(d);
    if(d < 255) {
        a = a*1.005;
        b = b*1.005;
        c = c*1.005;
        d = d*1.005;
        //System.out.println("got here");
        background(a,b,c,d);
    }
    System.out.println("white: " + white);
    white = white*0.94;
    fill(white,white,white);
    circle(star1, 83, 3);
    circle(star2, 152, 4);
    circle(star3,112,5);
    circle(star4, 21, 6);
    circle(star5, 175, 7);
    circle(star6,54,3);
    circle(star7,87,4);
    circle(star8, 11, 5);
    circle(star9,179,6);
    circle(star10,59,7);
    circle(star11,92,3);
    circle(star12,56,4);
    circle(star13,164,5);
    circle(star14,30,6);

    star1 -= speedOfStars;
    star2 -= speedOfStars;
    star3 -= speedOfStars;
    star4 -= speedOfStars;
    star5 -= speedOfStars;
    star6 -= speedOfStars;
    star7 -= speedOfStars;
    star8 -= speedOfStars;
    star9 -= speedOfStars;
    star10 -= speedOfStars;
    star11 -= speedOfStars;
    star12 -= speedOfStars;
    star13 -= speedOfStars;
    star14 -= speedOfStars;
}
```



# HOW THE BACKGROUND CHANGES PT. 5

Once white is less than 40, it will replace the background to the original background color which is like a sky blue color. It will also increase the clouds color showing the clouds once again.

```
if (white < 40) {  
    background(a,b,c,d);  
}  
  
    white2 = white2*1.005;  
fill(white2,white2,white2);  
arc(cloud1, 100, 40, 40, PI, TWO_PI);  
arc(cloud2, 100, 40, 40, PI, TWO_PI);  
arc(cloud3, 60, 40, 40, PI, TWO_PI);  
arc(cloud4, 60, 40, 40, PI, TWO_PI);  
arc(cloud5, 140, 40, 40, PI, TWO_PI);  
arc(cloud6, 140, 40, 40, PI, TWO_PI);  
arc(cloud7, 30, 40, 40, PI, TWO_PI);  
arc(cloud8, 30, 40, 40, PI, TWO_PI);  
arc(cloud9, 40, 40, 40, PI, TWO_PI);  
arc(cloud10, 40, 40, 40, PI, TWO_PI);  
arc(cloud11, 110, 40, 40, PI, TWO_PI);  
arc(cloud12, 110, 40, 40, PI, TWO_PI);  
arc(cloud13, 150, 40, 40, PI, TWO_PI);  
arc(cloud14, 150, 40, 40, PI, TWO_PI);  
arc(cloud15, 60, 40, 40, PI, TWO_PI);  
arc(cloud16, 60, 40, 40, PI, TWO_PI);  
  
cloud1 -= speedOfClouds;  
cloud2 -= speedOfClouds;  
cloud3 -= speedOfClouds;  
cloud4 -= speedOfClouds;  
cloud5 -= speedOfClouds;  
cloud6 -= speedOfClouds;  
cloud7 -= speedOfClouds;  
cloud8 -= speedOfClouds;  
cloud9 -= speedOfClouds;  
cloud10 -= speedOfClouds;  
cloud11 -= speedOfClouds;  
cloud12 -= speedOfClouds;  
cloud13 -= speedOfClouds;  
cloud14 -= speedOfClouds;  
cloud15 -= speedOfClouds;  
cloud16 -= speedOfClouds;  
  
System.out.println("white2 is: " + white2);
```

# HOW THE BACKGROUND CHANGES PT.6

Timer2 is also another makeshift timer, this is the last part of the code that is executed and once timer2 is greater than 2, all the variables get updated back to the original values to loop the entire program again.

```
System.out.println(timer2);
timer2 = timer2*1.001;
if(timer2 > 2) {
    timer = 1;
    timer2 = 1;
    minusOne = 500;
    a = 157;
    b = 211;
    c = 245;
    d = 255;
    white = 1;
    star1 = 123;
    star2 = 321;
    star3 = 232;
    star4 = 353;
    star5 = 21;
    star6 = 167;
    star7 = 297;
    star8 = 430;
    star9 = 452;
    star10 = 470;
    star11 = 512;
    star12 = 378;
    star13 = 312;
    star14 = 296;
    timer3 = 1222;
    //timer4 = 1222;
    white2 = 255;
}
}
```

# DASHES AND TREES

This part of the code just creates the color for the road, the water and the grass. It also fills in the roads with the yellow dashes and has the trees on the background. These if conditions are the same, if they are not on the display anymore, their x-axis positions are changed to a x-axis off the screen to the right.

```
//grass
fill(96,128,63,255);
rect(0,200,400,60);

//street
fill(85,81,72,255);
rect(0,260,400,60);
fill(241,202,34);
quad(dash1, 295, dash1q, 285, dash1w, 285, dash1e, 295);
quad(dash2, 295, dash2q, 285, dash2w, 285, dash2e, 295);
quad(dash3, 295, dash3q, 285, dash3w, 285, dash3e, 295);
quad(dash4, 295, dash4q, 285, dash4w, 285, dash4e, 295);
quad(dash5, 295, dash5q, 285, dash5w, 285, dash5e, 295);

dash1 -= speedOfDash;
dash1q -= speedOfDash;
dash1w -= speedOfDash;
dash1e -= speedOfDash;

dash2 -= speedOfDash;
dash2q -= speedOfDash;
dash2w -= speedOfDash;
dash2e -= speedOfDash;

dash3 -= speedOfDash;
dash3q -= speedOfDash;
dash3w -= speedOfDash;
dash3e -= speedOfDash;

dash4 -= speedOfDash;
dash4q -= speedOfDash;
dash4w -= speedOfDash;
dash4e -= speedOfDash;

dash5 -= speedOfDash;
dash5q -= speedOfDash;
dash5w -= speedOfDash;
dash5e -= speedOfDash;

treeLog1 -= speedOfTrees;
tree1q -= speedOfTrees;
tree1w -= speedOfTrees;
tree1e -= speedOfTrees;

treeLog2 -= speedOfTrees;
tree2 -= speedOfTrees;

treeLog3 -= speedOfTrees;
tree3 -= speedOfTrees;

treeLog4 -= speedOfTrees;
tree4 -= speedOfTrees;

if (dash1e < 0) {
  dash1 = 410;
  dash1q = 405;
  dash1w = 455;
  dash1e = 460;
}

if (dash2e < 0) {
  dash2 = 410;
  dash2q = 405;
  dash2w = 455;
  dash2e = 460;
}

if (dash3e < 0) {
  dash3 = 410;
  dash3q = 405;
  dash3w = 455;
  dash3e = 460;
}

if (dash4e < 0) {
  dash4 = 410;
  dash4q = 405;
  dash4w = 455;
  dash4e = 460;
}

if (dash5e < 0) {
  dash5 = 410;
  dash5q = 405;
  dash5w = 455;
  dash5e = 460;
}

if (treeLog1 < -30) {
  treeLog1 = 450;
  tree1q = 439;
  tree1w = 459;
  tree1e = 479;
}

if (treeLog2 < -30) {
  treeLog2 = 450;
  tree2 = 459;
}

if (treeLog3 < -30) {
  treeLog3 = 450;
  tree3 = 459;
}

if (treeLog4 < -30) {
  treeLog4 = 450;
  tree4 = 429;
}

//water
fill(50, 209, 252);
rect(0,320,400,90);
```

# MODELS

Lastly, these code are used to create the model of the car and the trees for the background.

```
//triangle tree
fill(92,78,67,255);
rect(treeLog1,170,20,60);
fill(41,52,21,255);
triangle(treeLq, 200, treeLw, 120, treeLe, 200);
//circle tree
fill(92,78,67,255);
rect(treeLog2,190,20,60);
fill(37,48,13,255);
circle(tree2, 180, 60);
//oval circle
fill(92,78,67,255);
rect(treeLog3,150,20,60);
fill(200,219,73,255);
ellipse(tree3, 140, 40, 90);
//square tree
fill(92,78,67,255);
rect(treeLog4,180,20,60);
fill(136,155,58,255);
rect(tree4, 150, 60, 60);

car1();

}
```

```
void car1() {
//body
fill(166,39,42);
rect(80,245,140,25,5);
//wheels
fill(1);
arc(100, 270, 25, 25, PI, PI+TWO_PI);
fill(56,44,46,255);
arc(100, 270, 20, 20, PI, PI+TWO_PI);
fill(1);
arc(200, 270, 25, 25, PI, PI+TWO_PI);
fill(56,44,46,255);
arc(200, 270, 20, 20, PI, PI+TWO_PI);
//roof
fill(166,39,42);
noStroke();
triangle(95, 245, 125, 230, 125, 245);
triangle(160, 245, 160, 230, 190, 245);
rect(125,230,35,15);
rect(125,245,35,25);
fill(202,227,243,255);
triangle(100, 245, 130, 230, 130, 245);
triangle(155, 245, 155, 230, 185, 245);
rect(138,231,17,14);
rect(130,231,5,14);
ellipse(218,250,5,10);
if(d < 70) {
    fill(217,182,117,255);
    ellipse(218,250,5,10);
    rect(223,249, 5, 2);
    translate(222,244);
    rotate(TWO_PI - PI*3/18);
    rect(0,0, 5, 2);
    rotate(PI*6/18);
    rect(6,8, 5, 2);
}else {
    fill(146,158,167);
    ellipse(218,250,5,10);
}
}
```