VIT-AP UNIVERSITY

DAY AND NIGHT VISION

Computer Graphics
Course Code: CSE 2006
Slot: (L19+L20+L29+L30+L45+L46)
Group No: 20
Guided by: Dr.Reeja SR

SUBMITTED BY

Batchu Priyanka (20BCE7272)
Sashank Kottha (20BCE7287)
Varanasi L.V.S.K.B.Kasyap (20BCE7315)
Chaparala Sri Manasa (20BCE7345)
Nallamothu Dharanija (20BCE7360)

ABSTRACT:

The study was focused on the study of the creation of our imagination using coding language. The study examines the extent to which the object can be created, how to apply the motion mode for the object, application of shading effect, lighting effect, how to apply the mouse click response. This data was analyzed based on functions and coordinates.

The study reveals the output of beautiful imagination. It was recommended that learning and knowing of function usage results in increase of creativity mindset. Application of learnt functions results the new shine or sparkle to eyes.

INTRODUCTION:

ABOUT TOOL:

P5. Js Web Editor

- User Friendly (It is easy to import the image and use the code).
- It is easy to handle and edit the code by sharing the link when working as team.

ABOUT THE PROJECT:

- setup()
  To create canvas of required size
  Syntax: CreateCanvas(size,size)

- draw()
  This is called directly after the setup(). The draw() function continuously executes the lines of code contained inside its block until the program is stopped.
  Syntax: draw()

- fill()
  Sets the colour used to fill shapes
  syntax: fill(values)

- ellipse()
  Draws an ellipse to the screen. By default the first two parameters set the location of the center of the ellips, and the third and fourth parameters set the shapes width and height
  Syntax: ellipse(x,y,w,[h])

- triangle()

  Draws a triangle to the canvas. The first two arguments specify the first point, the middle two arguments specify the second point, and the last two arguments specify the third point.

  Syntax: triangle(x1,y1,x2,y2,x3,y3)

- vertex()
  All shapes are constructed by connecting a series of vertices. vertex() is used to specify the vertex coordinates for points, lines, triangles, quads, and polygons. It is used exclusively within the beginShape() and endShape() functions.
  Syntax:
  vertex(x,y)
  vertex(x,y,z)

- beginShape() and endShape()
  Using the beginShape() and endShape() functions allow creating more complex forms. beginShape() begins recording vertices for a shape and endShape() stops recording.
  Syntax:
  beginShape();
  vertex(x,y)
  ….
  ….
  endShape();

- strokeWeight()
  Sets the width of the stroke used for lines, points and the border around shapes. All widths are set in units of pixels.
  Syntax: strokeWeight(weight)

- stroke()
  Sets the color used to draw lines and borders around shapes.
  Syntax: stroke(value)

- rect()
  Draws a rectangle in the canvas. By default, the first two parameters set the location of the upper-left corner, the third sets the width, and the fourth sets the height.
  Syntax: rect(x,y,w,h)

- circle()
  Draws a circle on the screen. By default, the first two parameters set the location of the centre of the circle, the third sets the diameter of the circle.
  Syntax: circle(x,y,d)

- translate()
  Specifies an amount to displace objects within the display window. The x parameter specifies left/right translation, the y parameter specifies up/down translation.
  Syntax: translate(x,y)

- rotate()
  Rotates a shape by the amount specified by the angle parameter.
  Syntax: rotate(angle) - angle specified in radians or degrees

- mousePressed()
  The mousePressed() function is called once after every time a mouse button is pressed. The mouseButton variable can be used to determine which button has been pressed.
  Syntax: mousePressed([event])

- random()

  If two arguments are given, returns a random number from the first argument up to (but not including) the second argument.

  Syntax: random([min], [max])

DESCRIPTION OF THE RESULT:

It is a Day and Night view.

Here the output is a Day and Night view in the nature. The view is described as the place near the road which is surrounded by mountains, trees, and the birds flying in the sky. There is also a helicopter in the sky. We can observe the changing of the day and night mode based on the sun and the moon. Whenever the sun is entering, the view is Day view. And when the moon is entering then we can observe the night view. Here is the beautiful scenery of the moving clouds in the sky. Here we can also observe movements in the grass which indicates the blowing air. In the sky the birds are producing based on the mouse Click option and using some functions we can create the movement effects for the birds. Day And Night view are differentiated based on the transition of colors in the background and observation of sun color and moon color.

IMPLEMENTATION:

```
var a;
var b;
var c;
var s = 500;
var m = 1000;
var speed = 1.5;
let xChange = 0;
let birds = [];
let theta;

function preload()
{
	helicopterIMG = loadImage("helicopter.png");
}

function setup() {
  createCanvas(600,600);
  grass = new yard();
  //image(helicopterIMG, 10, 10);
}
```

```
function draw() {
  //background(60,128,230);
  background(0,0, a);
    image(helicopterIMG, 300, 20);
  helicopterIMG.resize(250, 80);


  if ((s > 0) && (s < 500)) {
    a = c
  }
  if ((s > -555) && (s < 0)) {
    a = b
  }


//SUN
  fill (255, 240,102);
  ellipse (210, s, 100, 100);
  s = s - speed
  if (s <= -555){
   s = 500;
  }
  c = (-1.8*(s-300));
```

```
//MOON
  fill (240);
  ellipse (210, m, 100, 100);
  m = m - speed

  if (m <= -555){
   m = 500;
  }

  b = -300+m*1.8;
    var galaxy = {
  locationX : random(width),
  locationY : random(height),
  size : random(1,6)
}
      ellipse(mouseX ,mouseY, galaxy.size, galaxy.size);
  ellipse(galaxy.locationX ,galaxy.locationY, galaxy.size, galaxy.size);

//MOUNTAINS
//further mountain
  fill(139,69,19);
  strokeWeight(0);
```

```
triangle(360, 130, 760, 500, -40, 500);

fill(213, 212, 255);

strokeWeight(0);

        beginShape();

                  vertex(360, 130);

                  vertex(485, 246);

                  vertex(390, 200);

                  vertex(360, 250);

        vertex(320, 217);

        vertex(225, 255);

        endShape(CLOSE);


fill(255,255,255);

stroke(255,255,255);

ellipse(100+xChange,100,55,40);

ellipse(130+xChange,100,55,40);

ellipse(120+xChange,130,55,40);

ellipse(150+xChange,120,55,40);

ellipse(180+xChange,110,55,40);

ellipse(160+xChange,90,40,40);


xChange = xChange + 0.8;
```

```
  for (let i = 0; i < birds.length; i++) {
    birds[i].fly();
    birds[i].display();
  }
//closer one
  fill(169,63,8);
  strokeWeight(0);
  triangle(100, 180, 500, 500, -260, 500);
  fill(231, 241, 255);
  strokeWeight(0);
          beginShape();
                    vertex(100, 180);
                    vertex(225, 280);
                    vertex(145, 250);
                    vertex(120, 290);
          vertex(70, 260);
          vertex(-20, 286);
          endShape(CLOSE);

//HILLS
  fill(60, 145, 57);
```

```
ellipse(50, 580, 1000, 400);

fill(69, 168, 66);

ellipse(440, 600, 1000, 400);

fill(150, 75, 0);

rect(190,479,30,30);

fill(102,255,102);

triangle(202,404,116,479,288,479);

triangle(202,374,116,462,288,459);

triangle(202,309,116,430,288,430);

triangle(202,285,116,405,288,405);

fill(255,192,203);

circle(183,431,5);

 circle(155,409,5);

 circle(238,389,5);

 circle(239,370,5);

 circle(237,452,5);

 circle(162,471,5);

 circle(183,431,5);

circle(223,465,5);

 circle(197,326,5);

 circle(188,394,5);

circle(171,354,5);
```

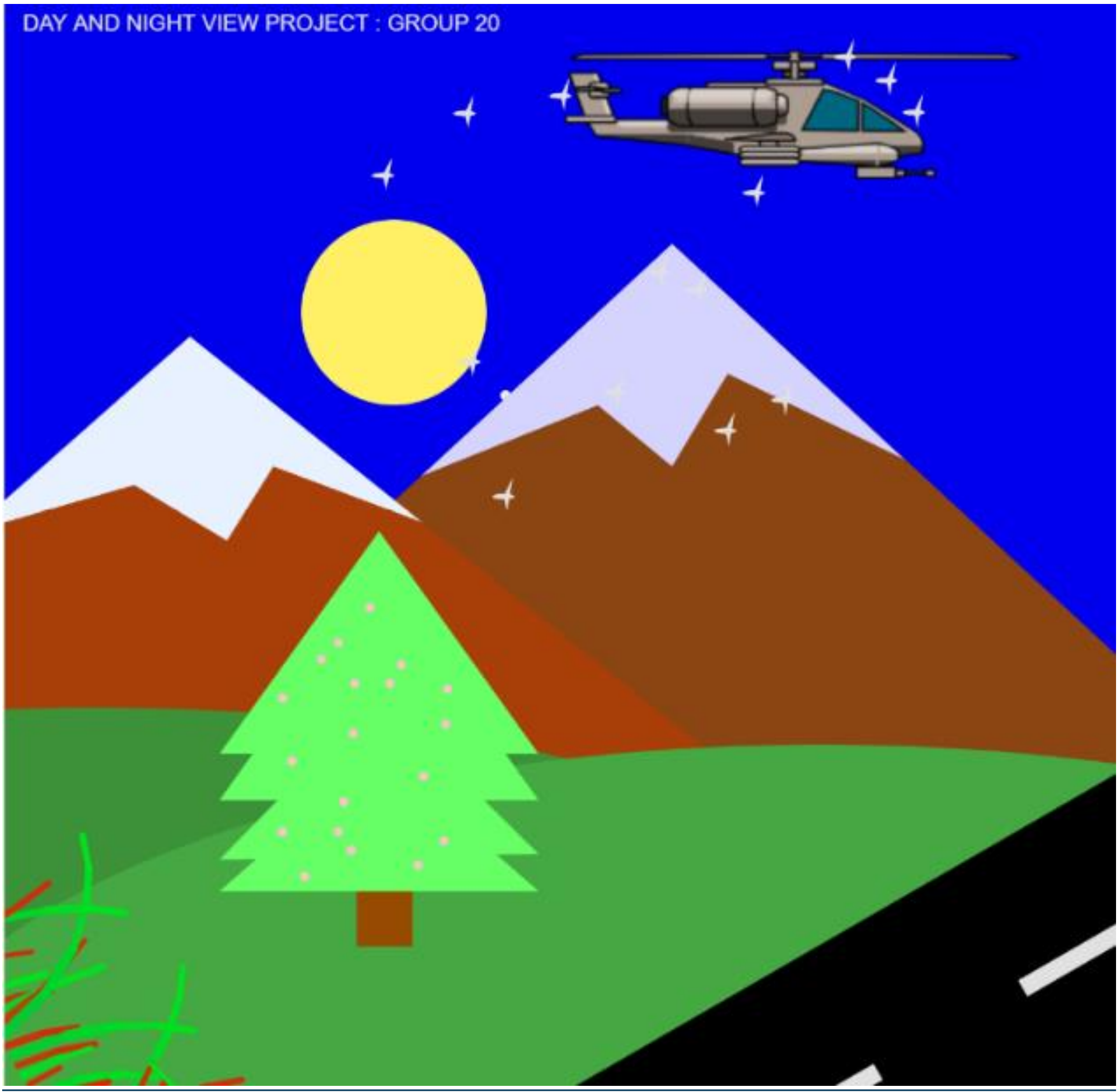
```
   circle(214,357,5);

   circle(226,417,5);

   circle(150,375,5);

   circle(150,447,5);

  circle(208,367,5);

   circle(180,345,5);

   circle(180,447,5);

   circle(187,457,5);

   circle(189,367,5);

   circle(150,375,5);

   circle(150,447,5);

   circle(187,457,5);

fill(0);

  text("(" + mouseX + ", " + mouseY + ")", mouseX, mouseY);

  stroke(0);

          fill(225);

          text("DAY AND NIGHT VIEW PROJECT : GROUP 20",10,15);

  fill(0);

  translate(width / 2, height / 2);

rotate(PI / 3.0);

rect(250,-225,552,552);

  fill(225);
```

```
rect(320,-190,10,90);
rect(320,-10,10,90);
   grass.update();
}

class Bird {
  constructor(x, y) {
    this.x = x;
    this.y = random(height * 0.25, height * 0.75);
  }

  fly() {
    this.x = this.x + 0.5;
    this.y = this.y + -0.2;
  }

  display() {
    {
      fill(225);
      //for (let i = 0; i > height * 0.25 && i < height *0.25; i++);
      triangle(this.x, this.y, this.x + 2, this.y + 8, this.x + 5, this.y - 10)
      ellipse(this.x, this.y, 12, 3)
    }
  }
}

function mousePressed() {
  let b = new Bird(random(-10,0), random(0,height));
  birds.push(b);
}
function yard() {
  this.grass = [];
  this.roff = [];
  this.rwave = [];
  this.size = [];
  this.seg = [];
  this.index = 0;
  this.population = 100;

  for (let x = 0; x < width; x += width / this.population) {
    this.index += 1;
    this.grass.push(x);
    this.roff.push((this.index * 0.065) + 0.015);
    this.rwave.push(0);
    this.size.push(random(35, 15));
    this.seg.push(0.85);
  }
  this.update = function() {
    for (let i = 0; i < this.index; i++) { // draw many blades
      let len = this.size[i];
      push();
      translate(this.grass[i], height * 0.65);
      this.blade(len, i);
      pop();
    }
  }
  this.blade = function(len, ind) {
    if (ind / 2 === int(ind / 2)) {
      this.roff[ind] += 0.0025;
      stroke(0, 255 - (len * 1.5), len * 1.5, 255);
      rot = map(noise(this.roff[ind]), 0, 1,
        -QUARTER_PI * 0.75, QUARTER_PI * 0.75);
    }
```

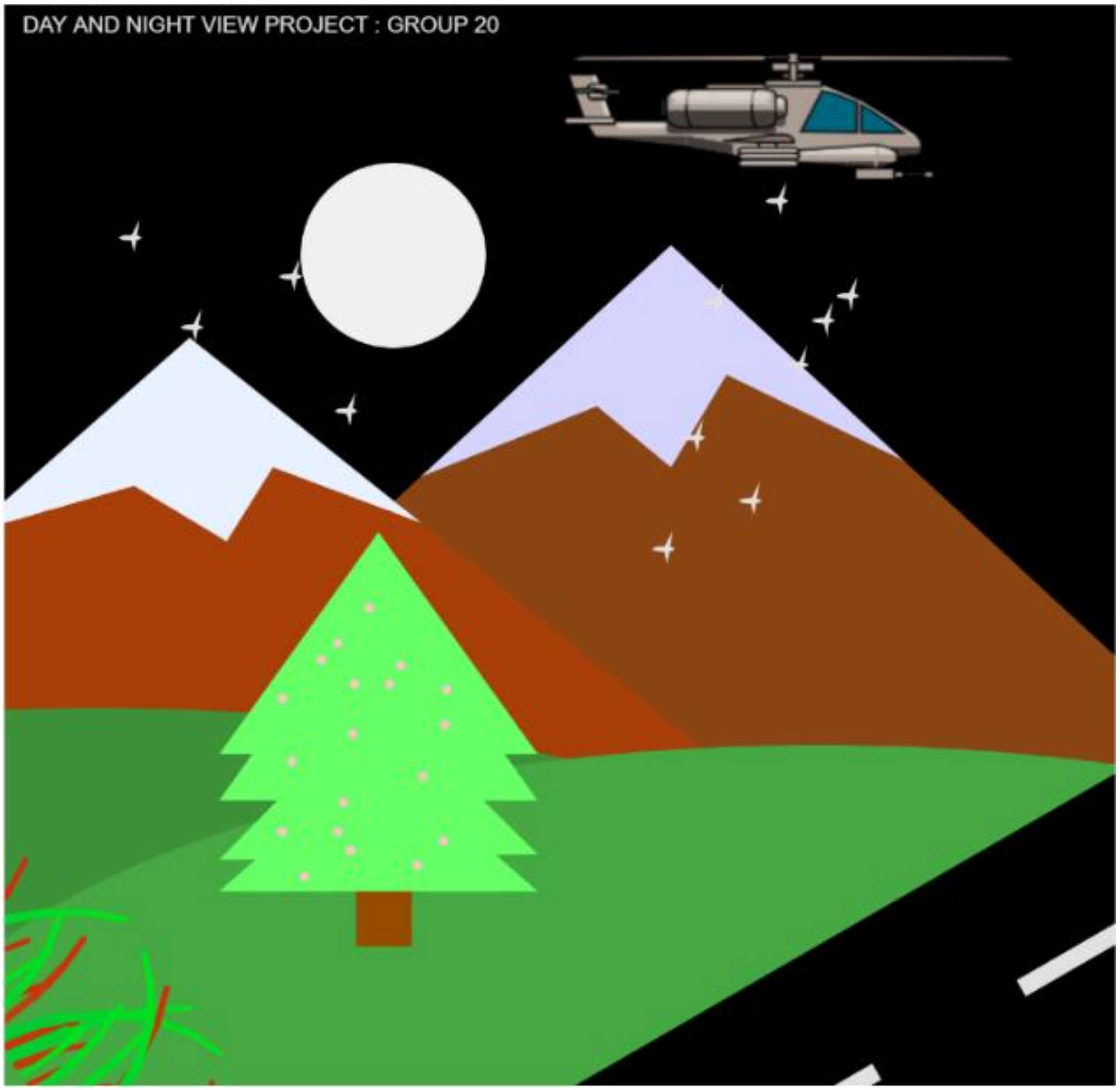
```
    if (ind / 2 != int(ind / 2)) {
      this.roff[ind] += 0.0025;
      stroke(255 - (len * 2.5), len * 2.5, 10, 255);
      rot = map(-sin(this.roff[ind]), -1, 1,
        -QUARTER_PI * 0.25, QUARTER_PI * 0.25);
    }
    strokeWeight(len * 2 * random(0.07, 0.11));
    rotate(rot);
    line(0, 0, 0, -len);
    translate(0, -len);

    if (len > 20) {
      this.blade(len * this.seg[ind], ind);
    }
  }
}
```

RESULT:

DAY VIEW:



NIGHT VIEW:



REFERENCE:

https://editor.p5js.org/GilbertoAlmeida/sketches/tJKcxlFmb