

Module 07

Revision: Earth Sun Moon Application

CS 106 Winter 2020

Earth Sun Moon

- Three versions:

1. ESM
2. ESM_Dom
3. ESM_Dom2

Earth Sun Moon

- ESM covers:
 - Geometric Context for drawing earth, sun and moon
 - One record for storing earth and moon data
 - Two dom elements (textfield and button) to update number of stars in the background
 - Loading images for earth and sun

Earth Sun Moon

- ESM_Dom covers:
 - More dom elements (sliders) to control:
 - Earth's distance from sun
 - Speed of earth's rotation about Sun
 - Speed of earth's rotation about it's axis
 - Moon's distance from earth
 - Speed of moon's rotation about earth
- Sliders update the record and the transformations fetch updated values from the record

Earth Sun Moon

- ESM_Dom2 covers:
 - Text field and button to add planets with/without moons to the solar system
 - Planet data is stored in a record which is pushed into an array

ESM – Global Variables

```
let sun; // variable to hold image of the sun
let earth; // variable to hold image of the earth
let noOfStars = 100; // number of stars to display
let sizeOfSun = 0.7; // scaling factor for the image of sun

let stars = new Array(noOfStars); // An Array to hold the x,y coordinates
//of the stars

let angle = 0; // angle for the rotation, initially set 0
//and updated in draw
```

ESM – Global Variables

```
// Dom Variables
```

```
let noOfStarsIP;
```

```
// textbox to update the number of stars
```

```
let planetEarth = {
```

```
// Record to hold data about the earth  
// and its moon
```

```
  size : 0.06,
```

```
// scaling factor for the earth's image
```

```
  distFromSun: 200,
```

```
  rotSpeed : 1,
```

```
  revSpeed : 4,
```

```
  moonExists : true,
```

```
  moonDist : 50,
```

```
  moonRotSpeed : 1,
```

```
  moonSize : 10,
```

```
};
```

ESM

```
function preload(){  
  // loads images for sun and moon. Other planets and moon just have a colour  
  sun = loadImage("data/sun.png");  
  earth = loadImage("data/earth.png");  
}
```


ESM

```
function createStars(){
  // Randomly generate locations for stars and store them in an array in the form of vector
  for (let i=0;i<noOfStars;i++){
    stars[i] = createVector(int(random(width)), int(random(height)));
  }
}

function updateNoOfStars (){
  // Takes a user entered number from the textfield and calls createStars() to randomly generate stars
  noOfStars = int(noOfStarsIP.value());
  stars = new Array(noOfStars);
  createStars();
}

function drawStars(){
  // called from draw, it loops through the array and draw a str as ellipse for each location available
  fill(255, 255, 255);
  noStroke();
  for (let i=0;i<noOfStars;i++){
    ellipse(stars[i].x, stars[i].y, 2, 2);
  }
}
```

ESM

```
function drawMoon(planet){  
  // Given a planet's record, it draws a white moon of that planet using the moon paramters  
  fill(255, 255, 255);  
  push();  
  rotate(angle * planet.moonRotSpeed);  
  translate(planet.moonDist, 0);  
  ellipse(0, 0, planet.moonSize, planet.moonSize);  
  pop();  
}
```

ESM

```
function setup() {
  createCanvas(730, 730);
  createStars(); // Create initial set of stars (default: 100)

  // Setup text field, button and callback function to update number of stars to display in the system
  createP("");
  noOfStarsIP = createInput(noOfStars);
  noOfStarsBT = createButton('submit');
  noOfStarsBT.mousePressed(updateNoOfStars);

  // All are shapes and images are drawn using CENTER mode and all angles are in degrees
  rectMode(CENTER);
  angleMode(DEGREES);
  imageMode(CENTER);
}
```

ESM

```
function draw() {
  background('black');
  drawStars(); // Draw the stars from the array generated through createStars()

  fill(255, 0, 0);
  angle += 0.25; // angle incremented by 0.25 for each call to render

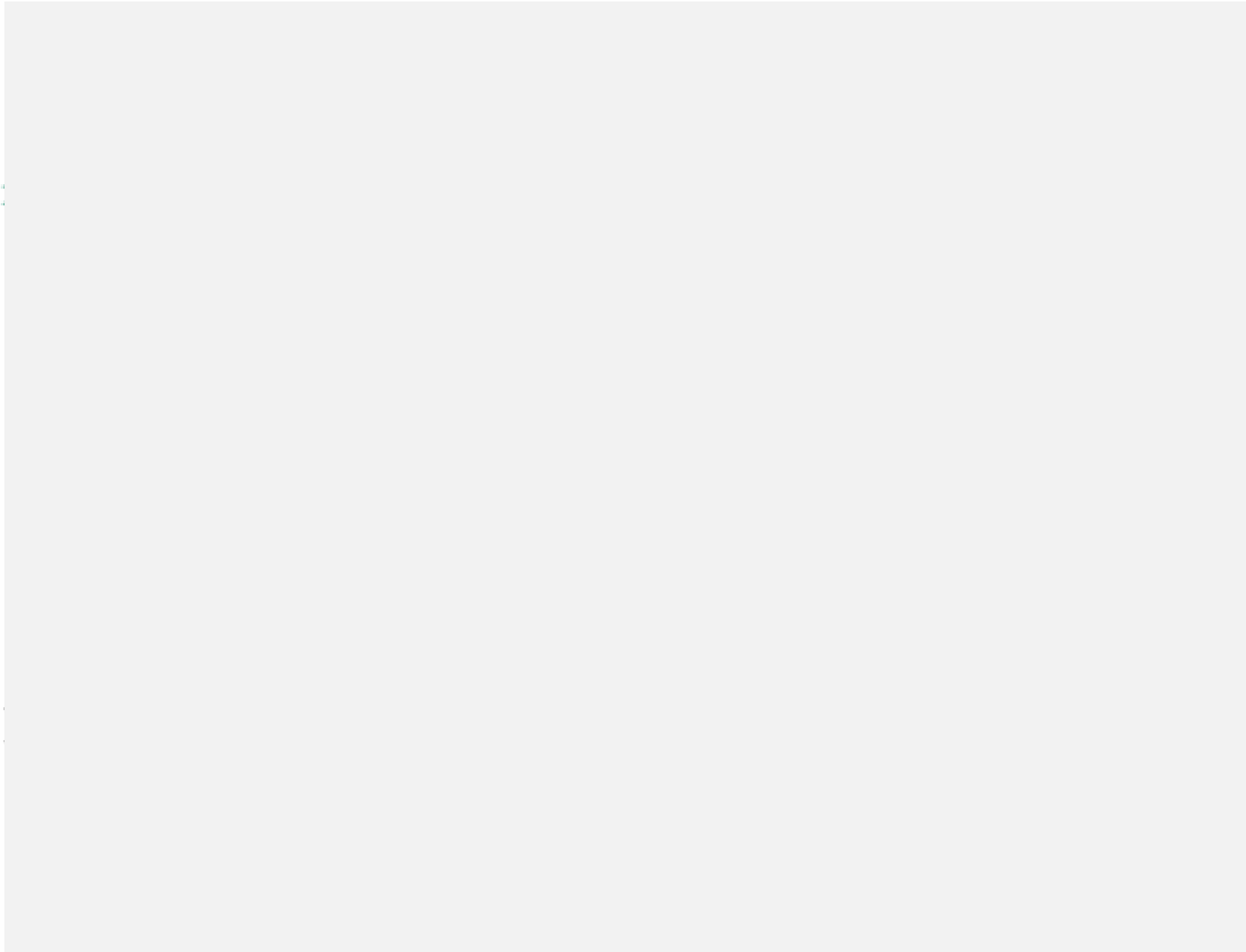
  // Translate to the center of the canvas and draw sun, planets and moons
  translate(width/2, height/2);
  push();
  push();
  scale(sizeOfSun);
  image(sun, 0, 0);
  pop();

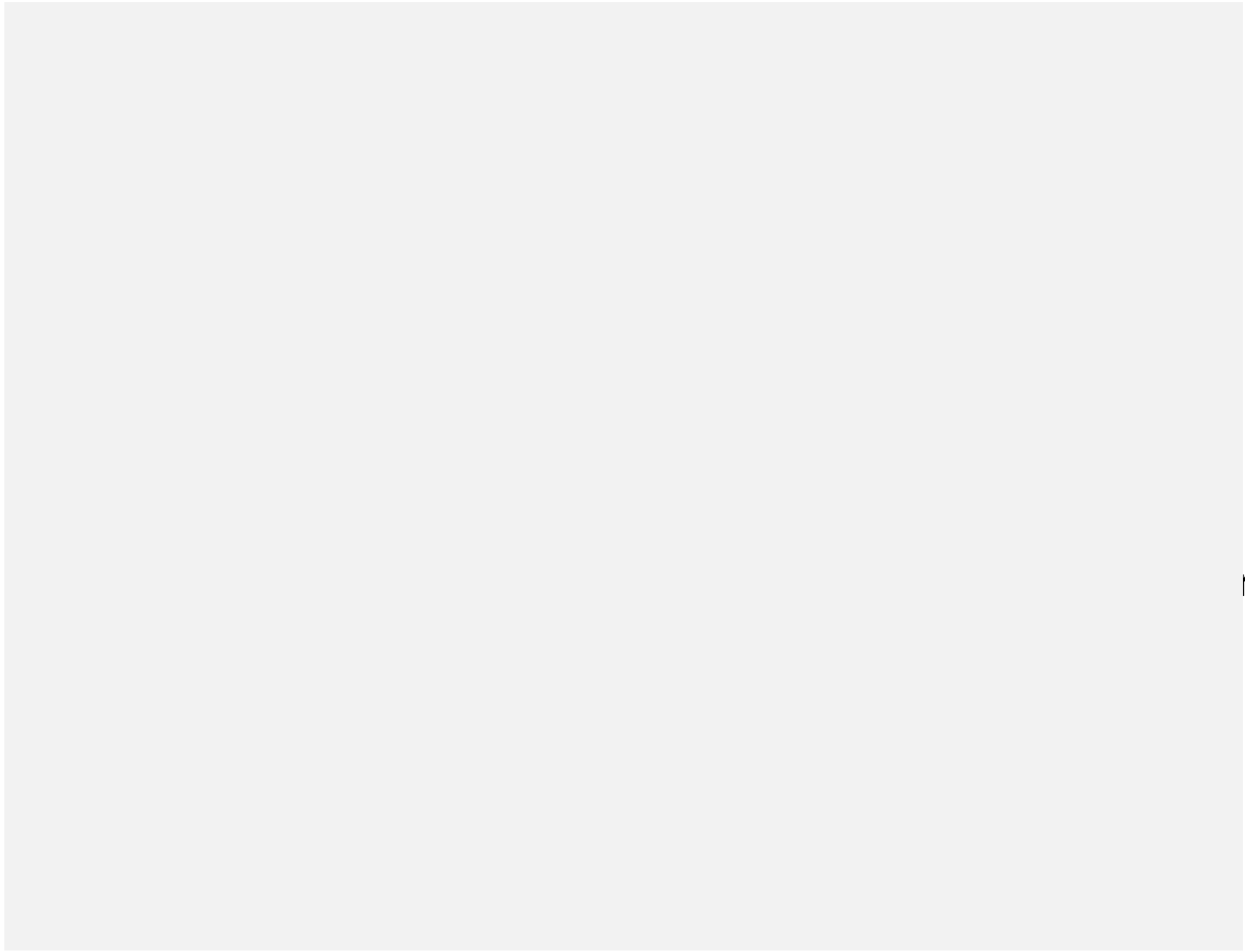
  rotate(angle * planetEarth.rotSpeed);
  translate(planetEarth.distFromSun, 0);

  push();
  rotate(-angle * planetEarth.revSpeed);
  scale(planetEarth.size);
  image(earth, 0, 0);
  pop();

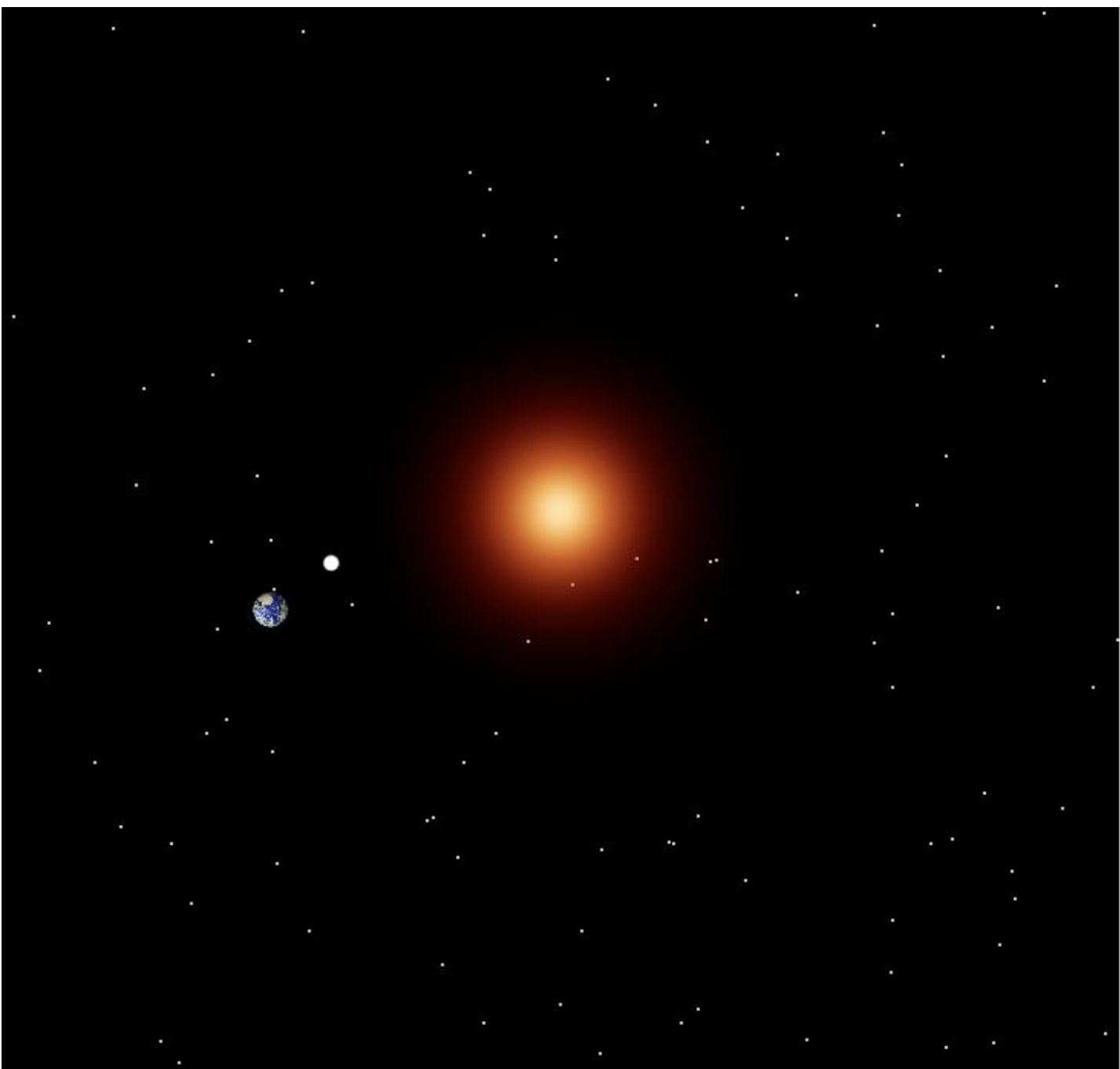
  if (planetEarth.moonExists == true)
    drawMoon(planetEarth); // draw moon if it exists

  pop();
}
```





r



100

submit

ESM_Dom

```
// Dom Variables
let noOfStarsIP;           // textbox to enter the number of stars for
                           // updatation
let earthDistFromSunSD;   // slider to update earth distance from sun
let earthRotSpeedSD;      // slider to update earth's rotation speed
let earthRevSpeedSD;     // slider to update earth's revolution speed
let moonDistSD;          // slider to update moon's distance to
                           // earth
let moonRotSpeedSD;      // slider to update speed of moon's
                           // rotation
```

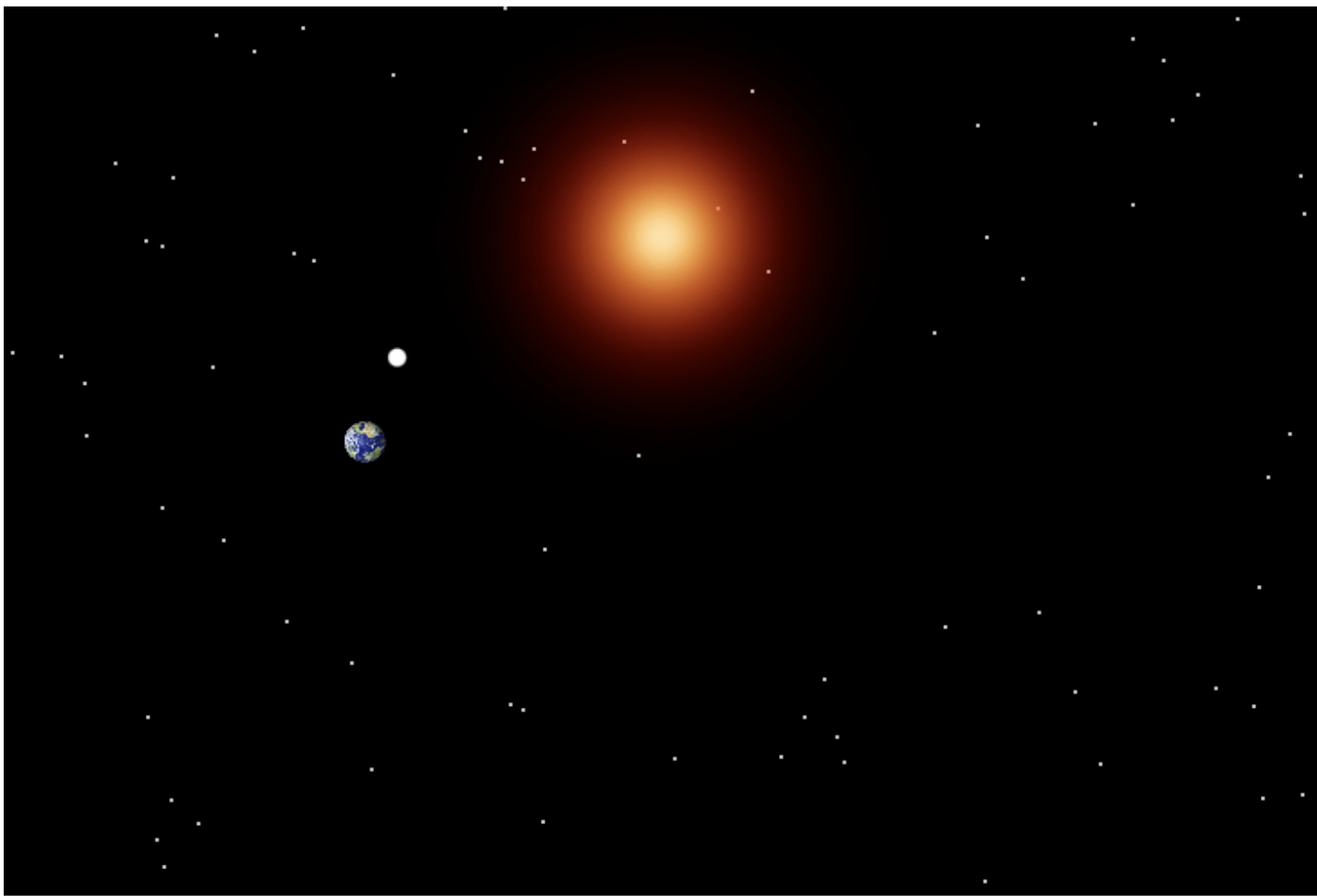

ESM_Dom

- Setup

```
// setup slider to control earth's speed of rotation about it's axis  
createDiv("Earth's Revolving Speed");  
earthRevSpeedSD = createSlider(3, 20, planetEarth.revSpeed);  
earthRevSpeedSD.style('width', '200px');  
  
// setup slider to control moon's distance from earth  
createDiv("Moon's Distane From Earth");  
moonDistSD = createSlider(30, 70, planetEarth.moonDist);  
moonDistSD.style('width', '200px');
```

ESM_Dom

```
function draw() {  
  planetEarth.distFromSun = earthDistFromSunSD.value();  
  planetEarth.rotSpeed = earthRotSpeedSD.value();  
  planetEarth.revSpeed = earthRevSpeedSD.value();  
  planetEarth.moonDist = moonDistSD.value();  
  planetEarth.moonRotSpeed = moonRotSpeedSD.value();  
  
  background('black');  
  drawStars(); // Draw the stars
```



100 submit

Earth's Distance From Sun



Earth's Rotation Speed



Earth's Revolving Speed



Moon's Distane From Earth



Moon's Rotation Speed



