Activity 3.3 - Geocentric Orbit

Overview

In this activity, you will use methods from the `Random` and `Math` classes to animate an object orbiting the Earth.

Instructions

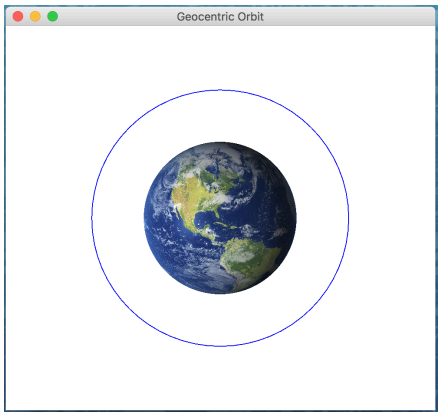
Getting Started

1. Create an "`Activity3.3`" folder and open it in VScode.
2. Import the `Orbit.java` and `earth.png` starter files (see Activity 3.1 for a reminder of how to import files)
3. When you first run the program, you should see Earth in the center of the frame. Modify `Orbit.java` to do the following.

Specifications

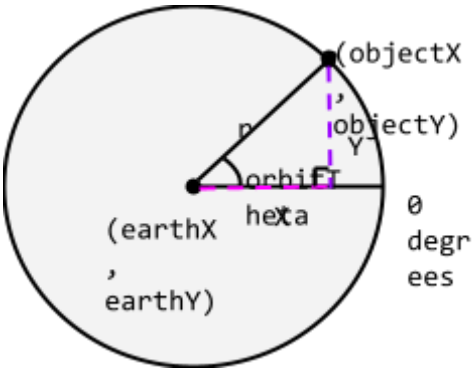
1. Define an `orbitRadius` variable. Assign it the minimum of ⅓ the `width` and ⅓ the `height` of the screen (use `Math.min` and the provided `width` and `height` variables).

2. Use this radius to draw the orbit around the Earth. **Important**: The given `earthX` and `earthY` variables define the **middle** of the Earth, not the top-left corner. We are doing this to simplify our orbit position calculations.



3. To keep things simple to start, we will just draw a circle orbiting around the Earth. Define an `objectRadius` variable and assign it ⅛ the `earthRadius`.

4. Use trigonometric functions to calculate your object's x and y coordinates.



Y is the vertical distance from earth to our object. X is the horizontal distance from earth to our object. r is the radius of the orbit.

$$cos(\theta) = \frac{adjacent}{hypotenuse} \Rightarrow cos(\theta) = \frac{X}{r} \Rightarrow X = r * cos(\theta)$$

$$sin(\theta) = \frac{opposite}{hypotenuse} \Rightarrow sin(\theta) = \frac{Y}{r} \Rightarrow Y = r * sin(\theta)$$

So,

```
objectX = earthX + r * cos(theta)
objectY = earthY - r * sin(theta)
```

5. Draw your object (an oval, with `fillOval`) using the `objectRadius`, `objectX`, and `objectY` values you have calculated.

6. Create a random color and use that to draw your object.

7. In the Orbit() constructor, use Random to get a random orbitDelta (direction and magnitude of the orbit angle change in each step).

8. If you have time, replace your oval object with an image of your choice.

Terminology Identification

In your code add comments identifying examples of the following: pseudo-random number, static method, dot operator. These should be identified with an @keyterm tag within the comment.

Code Review

When you are finished with this activity, pair up with a classmate and review each other's code to make sure it meets all the requirements.

Submission

After completing the assignment, use the assignment link in Canvas and follow the submission instructions there. You will upload your `Orbit.java` file and submit your reflection in the "Comments" box.

Reflection Requirements

Write a one paragraph reflection describing your experience with this activity. The reflection should also include the name of your code review partner AND something interesting you found in their code. Please review the activity rubric for more details.