Earth As Art and Terrain Generator: Teacher Guide

Summary

Coding skill level:	**Advanced**
Recommended grade level:	**Grades 6-12 (U.S.), Years 7-13 (U.K.)**
Time required:	**50 minutes per module**
Number of modules:	**2 modules**
Coding Language:	**Python, JavaScript**

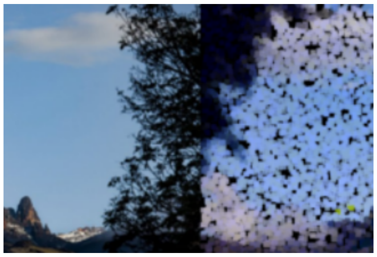
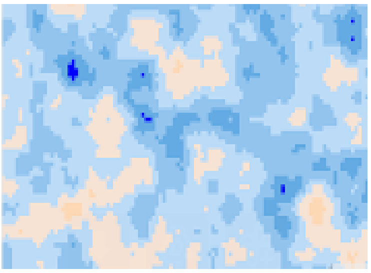
Teacher Guide Outline

Welcome!

Hour of Code
Earth As Art and Terrain Generator

Did you know that in 1960, NASA put the United State's first Earth-observing environmental satellite into orbit? Since then, these satellites have been providing us new perspectives, invaluable information, and beautiful images of Earth. The images captured by these satellites are truly a work of Art, which we will explore further in today's coding adventure!

Students will begin by watching a short video clip that explains how satellites capture images of Earth. Next, they'll browse through the gallery of *Earth as Art* photographs. They'll use this as inspiration to create either of the two coding projects: *Earth As Art* or *Terrain Generator*.

Project	Description	Example
	Students will use *Python* and *processing.py* as they learn how to apply color tinting and explore advanced image processing effects like a Pointillism filter.	
	Students will explore how to create pseudo-random noise using code. Let's see who can simulate a nearby geological feature, like a river or mountain range!	

Each project includes a tutorial to help guide your students' creative process, a sample project, ideas on how to expand on their project, and code snippets to help them get started. **Note:** *Earth As Art* uses *Python* and *processing.py* and *Terrain Generator* uses *JavaScript* and *p5.js*.

How to Prepare

This activity is designed for self-directed learning. Your role will be to help students individually and facilitate as they complete the activities. The best way to prepare is to:

1. **Familiarize yourself with the material.** After selecting your Tynker lesson, read through this teacher guide and complete the activity before assigning it to students. This will allow you to troubleshoot anything in advance and plan for potential questions from your students.

2. **Get students excited about coding.** Inspire students and get them excited for the Hour of Code event. Here is a link to resources such as inspirational videos and posters from the Hour of Code website: https://hourofcode.com/us/promote/resources#videos
3. **OPTIONAL: Sign up for a teacher account.** Although an account is NOT required, creating a free teacher account will allow you to access teacher guides, answer keys, and tons of additional resources. You'll also be able to create free accounts for your students, monitor their progress, and see their projects.
4. **OPTIONAL: Create student accounts.** From your teacher account, you can easily create free student accounts for all your students. This will allow them to save their projects and progress, so they can continue coding when they get home! Again, this is not necessary to complete the Earth As Art and Terrain Generator lesson.
5. **OPTIONAL: Print certificates to hand out.** While signed in to your Tynker teacher account, you can print certificates by clicking on a classroom from your teacher dashboard, clicking the "Gradebook" tab, going to "Hour of Code," and clicking the "Print All Certificates" button. This will only print certificates for student accounts assigned to the selected classroom.

Activity

Overview

Objectives

Students will...

- Demonstrate an understanding of loops, functions, variables, and arrays
- Use *Python* and *Processing.py* to create an image with a pointillist effect
- Use *JavaScript* and *p5.js* to create an image that looks like elevated terrain

Materials

- **For web:** Computers, laptops, or Chromebooks (1 per student)
- **For mobile:** iPads or Android tablets (1 per student)

Vocabulary

- **Code:** The language that tells a computer what to do
- **Sequence:** The order in which steps or events happen
- **Command:** A specific action or instruction that tells the computer to do something
- **Loop:** An action that repeats one or more commands over and over
- **Function:** A set of known actions that the computer can perform
- **HTML Canvas:** The HTML element that is used to draw graphics

- **Variable:** A memory location that is used to store data in a program. The value of variables can vary, or change
- **Pointillism:** An art technique that uses multiple tiny dots to create a scene or subject

U.S. Standards

- **CCSS-ELA:** RI.6.4, RI.6.7, SL.6.1, RI.7.4, SL.7.1, SL.8.1, RI.8.4, RI.9-10.5, RI.11-12.6
- **CCSS-Math:** MP.1
- **CSTA:** 2-AP-13, 2-AP-17, 3A-AP-17, 3B-AP-11
- **CS CA:** 6-8.AP.13, 6-8.AP.16, 6-8.AP.17, 9-12.AP.12, 9-12.AP.16
- **Illinois CS:** 6-8.AP.14, 6-8.AP.18, 9-10.AP.17, 11-12.AP.13
- **ISTE**: 1.c, 1.d, 4.d, 5.c, 5.d, 6.b

U.K. Standards

National Curriculum in England (computing):

- **Key Stage 3 (Years 7-9)**
  - Create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
  - Understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns
- **Key Stage 4 (Years 10+)**
  - Develop their capability, creativity and knowledge in computer science, digital media and information technology
  - Develop and apply their analytic, problem-solving, design, and computational thinking skills

Getting Started **(20 minutes)**

- Play this short video about Landsat 8, a satellite that orbits the Earth and captures images: https://www.youtube.com/watch?v=xBhorGs8uy8
- Time permitting, ask students to explore the Earth As Art image gallery: https://eros.usgs.gov/image-gallery/earth-as-art

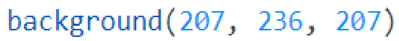
DIY Module **(30 minutes)**

Hour of Code
Earth As Art and Terrain Generator

This lesson has 2 DIY (do-it-yourself) modules that your students can choose from. Facilitate as students complete the modules on their own:

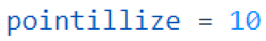
Earth As Art and Terrain Generator

- Students will use *Earth as Art* satellite photos as inspiration to create either an *Earth As Art* project or a *Terrain Generator* project. Note that one project uses Python and processing.py (Earth As Art) and the other project uses JavaScript and p5.js (Terrain Generator).
- Are students using a saved image for the Earth As Art project?
    - One way to upload the image onto the Tynker Text Code Editor is to save the image onto their desktop. Then drag and drop the image into the left-side panel. This method is explained in the first 20 seconds here: https://www.youtube.com/watch?v=ZgeGAU4t9Vs
- Do students want to change the background color in their Earth As Art project?
    - To change the color, students will need to change the **RGB** values inside the **background()** function. Inform students that **RGB** (red, green, blue) values must be between 0 and 255. Here's an example that will create a green-tinted background:



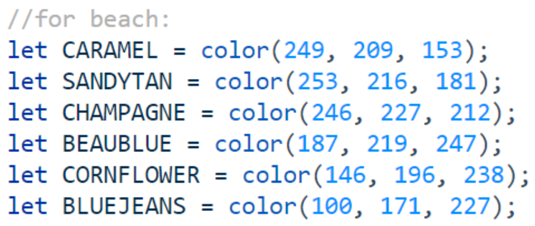
```
background(207, 236, 207)
```

- Do students want larger/smaller points in their Earth As Art project?
    - Ask them to change the value of the **pointillize** value. Example:



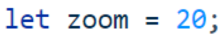
```
pointillize = 10
```

    - A larger value makes the points larger, whereas a smaller value makes the points smaller.
    - Optional: Encourage students to explore these different values: 5, 10, 20.
- Do students want to change the terrain colors in their Terrain Generator project?
    - First, ask students to locate an RGB table online. Next, ask them to change the **RGB** values inside the **color()** function. Inform students that **RGB** (red, green, blue) values must be between 0 and 255. Here's an example that will create beach-themed colors:



```
//for beach:
let CARAMEL = color(249, 209, 153);
let SANDYTAN = color(253, 216, 181);
let CHAMPAGNE = color(246, 227, 212);
let BEAUBLUE = color(187, 219, 247);
let CORNFLOWER = color(146, 196, 238);
let BLUEJEANS = color(100, 171, 227);
```

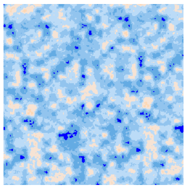
- Do students want to zoom in/out in their Terrain Generator project?
    - Ask them to change the value of the **zoom** value. Here's what it looks like:



```
let zoom = 20;
```

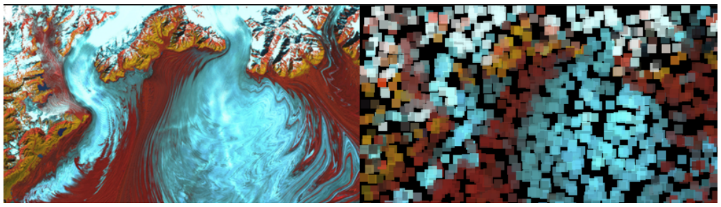
- A larger value will create a zoomed in effect. Here's an example of a beach when the "zoom" variable is set to a value of 200:

- A smaller value zooms out on the terrain. Here's an example of a beach when the "zoom" variable is set to a value of 20:



- Optional: Encourage students to explore these different values: 10, 100, 200.

- When students finish the Terrain Generator project, make sure they move their mouse around the screen. Notice how the terrain changes!
- Did students finish early? Encourage them to experiment with their code, get creative, and add their own unique twists. For example, here's what the Earth As Art project could look like if squares were used:



Extended Activities (10 minutes each)

Lead a discussion with your students:

- At the end of the time allotted for the *Earth As Art* and *Terrain Generator* projects, ask students to share their work and explain how they overcame any bugs or difficulties.
- *Optional:* Do a gallery walk as a class, visiting one another's work and asking questions of the artist (the student).

Going Beyond an Hour

If your students enjoyed an Hour of Code, they're sure to enjoy the rest of what Tynker has to offer! Tynker offers a complete premium solution for schools to teach computer science. Over 400 hours of lessons are available to take K-8 students from block coding to advanced text coding. We offer tons of resources for teachers, including comprehensive guides, free webinars, and a forum to connect with other educators.

More Hour of Code Activities

Tynker offers many other tutorials for the Hour of Code, including STEM Hour of Code lessons that you can integrate into the subjects you already teach. Check out the main Tynker Hour of Code page to see all the tutorials!

Do More with Tynker

With Tynker, kids don't just acquire programming skills--they explore the world of possibilities that coding opens up. Tynker has several interest-driven learning paths that make coding fun, both inside and outside the classroom:

- **Coding and Game Design:** Your students can use Tynker Workshop, a powerful tool for crafting original programs to make games, stories, animations, and other projects. They can even share their work with other kids in the Tynker Community.
- **Drones and Robotics:** Tynker integrates with connected toys, including Parrot drones and Lego WeDo robotics kits, so kids can see their code come to life.
- **Minecraft:** Tynker integrates with Minecraft so your students can learn coding through a game they love. Tynker offers skin and texture editing, as well as a custom Mod Workshop that lets kids try their original code in Minecraft.

Tynker for Schools

Used in over 90,000 schools, our award-winning platform has flexible plans to meet your classroom, school, or district needs. All solutions include:

- Grade-specific courses that teach visual coding, JavaScript, Python, robotics and drones
- A library of NGSS and Common Core compliant STEM courses that are great for project-based learning
- Automatic assessment and mastery charts for whole schools and individual classes and students
- Easy classroom management with Google Classroom and Clever integration
- Professional training, free webinars and other teacher training resources

Need help getting Tynker started at your school? Contact us to learn more about teaching programming at your school with Tynker!

Help

Need help? Below you'll find answers to frequently asked questions about the Earth As Art and Terrain Generator lesson.

What is Hour of Code?

The Hour of Code is a global learning event in which schools and other organizations set aside an hour to teach coding. No prior coding experience from you or your students is needed! The event is held every December during Computer Science Education Week. You can also organize an Hour of Code year-round. The goal of the Hour of Code is to expand access to computer science education for people of all backgrounds. Learning computer science helps students develop logic and creativity, and prepares them for the changing demands of the 21st century. Tynker has been a leading provider of lessons for the Hour of Code since the event began in 2013. Since then, over 100 million students from 180 countries have finished an Hour of Code.

How can Tynker help me manage my Hour of Code?

Tynker has several free features for registered teachers that will help you manage your Hour of Code. If you set your students up with a Tynker classroom, you will be able to track their progress and print Hour of Code completion certificates for them to keep.

How do I open the DIY modules?

Have your students go to this URL: tynker.com/hour-of-code. Next, direct them to the activity you want them to complete.

Who is this activity for?

Earth As Art and *Terrain Generator* is intended for students in grades 6-12 (U.S.) and Years 7-13 (U.K.) with some coding experience.

Do I need to create Tynker accounts for my students?

No, you do not need to create Tynker accounts for your students.

What devices do I need?

- **For web:** Computers, laptops, or Chromebooks (1 per student) with an internet connection
- **For mobile:** iPads or Android tablets (1 per student) with an internet connection
- If not enough devices are available, students can work in pairs on the same device

What will my students learn?

Students will reinforce coding concepts and experiment with their code as they complete either the Earth As Art project or the Terrain Generator project. In this process, students will develop debugging and logical reasoning skills.

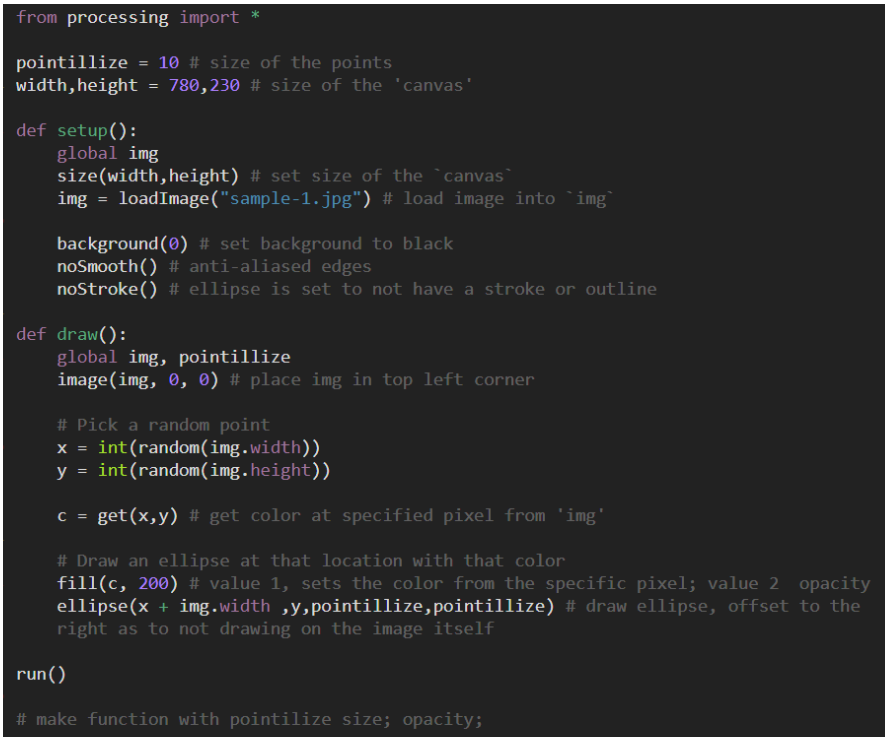
What are some helpful websites I can refer to?

Below are websites you might find helpful:

- **Earth As Art Website:** https://eros.usgs.gov/image-gallery/earth-as-art
- **Earth As Art eBook:** https://www.nasa.gov/pdf/703154main_earth_art-ebook.pdf
- **Information about NASA's Landsat Program:** https://landsat.gsfc.nasa.gov/
- **How Landsat Works:** https://www.youtube.com/watch?v=xBhorGs8uy8

Do you have a sample solution for Earth As Art?

Yes, below is a sample solution for the *Earth As Art* project:

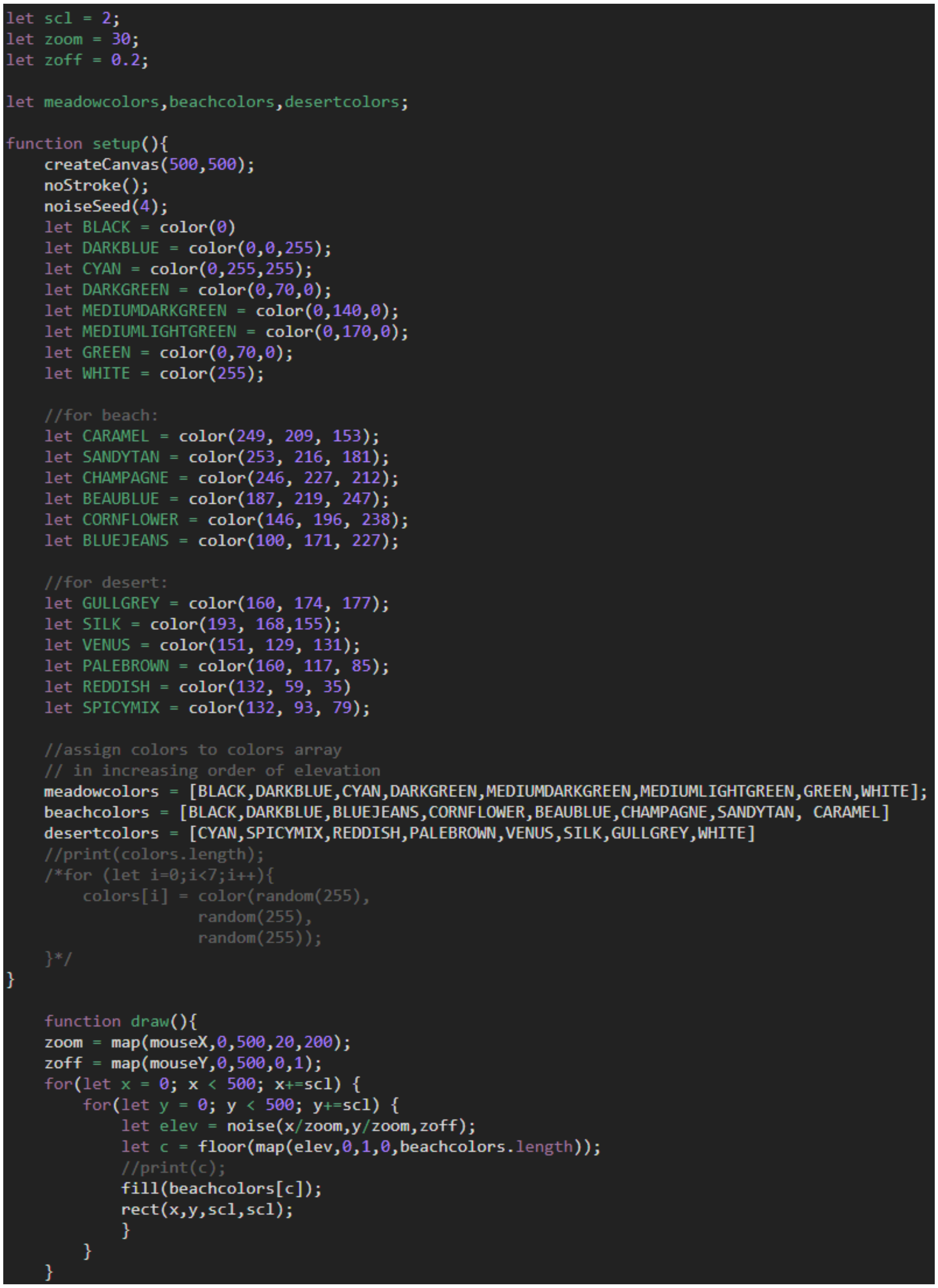


```
from processing import *

pointillize = 10 # size of the points
width,height = 780,230 # size of the 'canvas'

def setup():
    global img
    size(width,height) # set size of the `canvas`
    img = loadImage("sample-1.jpg") # load image into `img`

    background(0) # set background to black
    noSmooth() # anti-aliased edges
    noStroke() # ellipse is set to not have a stroke or outline

def draw():
    global img, pointillize
    image(img, 0, 0) # place img in top left corner

    # Pick a random point
    x = int(random(img.width))
    y = int(random(img.height))

    c = get(x,y) # get color at specified pixel from 'img'

    # Draw an ellipse at that location with that color
    fill(c, 200) # value 1, sets the color from the specific pixel; value 2  opacity
    ellipse(x + img.width ,y,pointillize,pointillize) # draw ellipse, offset to the
    right as to not drawing on the image itself

run()

# make function with pointilize size; opacity;
```

Note: This is an open-ended project where students are encouraged to experiment with their code.

Do you have a sample solution for the Terrain Generator project?

Yes, below is a sample solution for the *Terrain Generator* project:



```
let scl = 2;
let zoom = 30;
let zoff = 0.2;

let meadowcolors,beachcolors,desertcolors;

function setup(){
    createCanvas(500,500);
    noStroke();
    noiseSeed(4);
    let BLACK = color(0)
    let DARKBLUE = color(0,0,255);
    let CYAN = color(0,255,255);
    let DARKGREEN = color(0,70,0);
    let MEDIUMDARKGREEN = color(0,140,0);
    let MEDIUMLIGHTGREEN = color(0,170,0);
    let GREEN = color(0,70,0);
    let WHITE = color(255);

    //for beach:
    let CARAMEL = color(249, 209, 153);
    let SANDYTAN = color(253, 216, 181);
    let CHAMPAGNE = color(246, 227, 212);
    let BEAUBLUE = color(187, 219, 247);
    let CORNFLOWER = color(146, 196, 238);
    let BLUEJEANS = color(100, 171, 227);

    //for desert:
    let GULLGREY = color(160, 174, 177);
    let SILK = color(193, 168,155);
    let VENUS = color(151, 129, 131);
    let PALEBROWN = color(160, 117, 85);
    let REDDISH = color(132, 59, 35)
    let SPICYMIX = color(132, 93, 79);

    //assign colors to colors array
    // in increasing order of elevation
    meadowcolors = [BLACK,DARKBLUE,CYAN,DARKGREEN,MEDIUMDARKGREEN,MEDIUMLIGHTGREEN,GREEN,WHITE];
    beachcolors = [BLACK,DARKBLUE,BLUEJEANS,CORNFLOWER,BEAUBLUE,CHAMPAGNE,SANDYTAN, CARAMEL]
    desertcolors = [CYAN,SPICYMIX,REDDISH,PALEBROWN,VENUS,SILK,GULLGREY,WHITE]
    //print(colors.length);
    /*for (let i=0;i<7;i++){
        colors[i] = color(random(255),
                    random(255),
                    random(255));
    }*/
}

    function draw(){
    zoom = map(mouseX,0,500,20,200);
    zoff = map(mouseY,0,500,0,1);
    for(let x = 0; x < 500; x+=scl) {
        for(let y = 0; y < 500; y+=scl) {
            let elev = noise(x/zoom,y/zoom,zoff);
            let c = floor(map(elev,0,1,0,beachcolors.length));
            //print(c);
            fill(beachcolors[c]);
            rect(x,y,scl,scl);
            }
        }
    }
```

Note: This is an open-ended project where students are encouraged to experiment with their code.

How can I contact the Tynker support team?

If you have any issues or questions, send us an email at support@tynker.com.