Brief

A popular parable from the book *Art & Fear* (David Bales) goes like this:

> *The ceramics teacher announced on opening day that he was dividing the class into two groups. All those on the left side of the studio, he said, would be graded solely on the quantity of work they produced, all those on the right solely on its quality.*
>
> *His procedure was simple: on the final day of class he would bring in his bathroom scales and weigh the work of the "quantity" group: fifty pounds of pots rated an "A", forty pounds a "B", and so on. Those being graded on "quality", however, needed to produce only one pot – albeit a perfect one – to get an "A".*
>
> *Well, came grading time and a curious fact emerged: the works of highest quality were all produced by the group being graded for quantity. It seems that while the "quantity" group was busily churning out piles of work – and learning from their mistakes – the "quality" group had sat theorizing about perfection, and in the end had little more to show for their efforts than grandiose theories and a pile of dead clay.*

Motivated by this parable, this **individual** assignment encourages you to make several small generative works without worrying about quality. You will be asked to reflect the topics from class in your sketches, but you won't be otherwise graded on quality, only your percentage completion. Your goal is to share one original generative art piece (sketch), along with its code, with your instructors and (optionally, but encouraged) your peers, each week.

In total, you'll share **5** sketches by Spring Break.

For the purposes of this class, the week ends **Friday, 11:59pm Eastern time**. To fully enumerate the deadlines for Spring '25, they are:

- January 24
- January 31
- February 7
- February 14
- February 21
- February 28

Note that there are 6 deadlines but only 5 required sketches. That's on purpose: life is unpredictable, and sometimes you need to take a mulligan on a homework assignment. So you get one freebie (lowest grade will be dropped). However, unless you have exceptional circumstances (e.g., a family member dies, or you become very ill), no late work will be accepted.

Detailed Instructions

Each of your sketches must illustrate the **theme technique** for that week. This year's theme techniques are:

1.	January 24	Directed Randomness
2.	January 31	[Your choice]
3.	February 7	Recombination
4.	February 14	Grammars
5.	February 21	Cellular Automata
6.	February 28	Recursive Subdivision

Ask the instructor if you're not sure if your sketch uses the technique. One way to be sure is to attend the Friday "workshops" each week and follow along with the lecture on your own computer, which will give you starter code that uses the technique.

In addition, each sketch must respond to a **creative prompt**. See below ("Prompts") for a complete list of prompts.

Your "deliverable" will be a **well-commented p5.js sketch**, and a **Piazza post** – ideally shared with the whole class, but you can post privately to the instructors if you prefer – that contains the following information:

- The tag **#weeklysketch<N>** (where <N> is the week you are submitting)
- The subject "**Weekly Sketch: <Prompt> (#<P>)**", where <Prompt> is the text of the creative prompt you chose, and **<P>** is the number for that prompt.
    - Example: If I'm posting for Prompt 11 (radial symmetry), my subject would be

      Weekly Sketch: Radial Symmetry (#11)
- **Body**: Your message should contain:
    - AT THE VERY TOP: any content warnings, such as flashing lights, sounds, or unsettling imagery (please do not use any explicitly violent or sexual imagery in this class). Some people experience migraines or seizures triggered by certain images and sounds, so it's important to provide a warning.
    - A link to your p5.js code (in "edit" mode) as the very first line after that. See below ("Sharing p5.js sketches") for instructions.
    - **Attribution:** if you used any code or images outside your own or the standard p5.js library, including the use of forums like StackOverflow, video tutorials like Coding Train, or any code assistant tools like ChatGPT, you MUST indicate which parts of your code come from which sources.
    - Your output for the Design Recipe steps 1-4 (these may, alternatively, be included in comments within your code):

        - **1. Intention**: what is your program supposed to do (effects) under what conditions (events)?
        - **2. State types**: what global variables do you need to achieve the intended effect? What are their data types (e.g. text, color, number, boolean)?
        - **3. Required library functions:** what p5.js (or general Javascript) functionality do you need? If you plan to use an external library, this is also where you should include its documentation and point out what specific functions you need from it.
        - **4. Instructions for running your example** (in lieu of i/o tests). This can just be “press play and watch for at least 30 seconds” or “click on the canvas repeatedly”, but if the user needs to interact in a certain way to see the full scope of your sketch, include instructions here.
    - A few sentences explaining how you interpreted the prompt, what your goal was, how you approached it, and any adjustments you made along the way.
    - 1-4 screenshots showing what your code generates.

In order, completing a weekly sketch consists of the following activities:

1. Choosing a prompt to respond to (see “Prompts” below).
2. Thinking of an idea for how to use the technique to respond to the prompt
3. Following the design recipe for your idea, and documenting the steps as you go.
4. **Make sure your code is well-commented!** Include a header at the top with your name, the date, and attribution for all pieces of the code, and make sure every constant and global variable has a comment explaining what it is used to represent. Add comments throughout your code, at your discretion, whenever it is not self-explanatory.
5. Drafting a Piazza post and running down the checklist to make sure you’ve included all the required pieces.
6. Post your sketch!
7. Take a look at your classmates’ sketches and leave constructive critique, according to the guidelines below (Commenting on Classmates’ Sketches).

Grading

Each week, the instructor/TA will collect data from Piazza into a table and enter a grade and feedback on your sketch into gradescope. Grades out of 10 points will be assigned as follows:

- 10: Sketch submitted, includes all required parts, uses the theme technique, and responds to a creative prompt
- 7: Sketch submitted, uses the theme technique, but does not respond to a particular prompt OR misses some required pieces in Piazza post
- 5: Sketch submitted, but does not use the theme technique

- 0: No sketch submitted
- 0: Missing attribution for parts of code not written by the individual student.

Remember that your lowest grade will be dropped.

Prompts

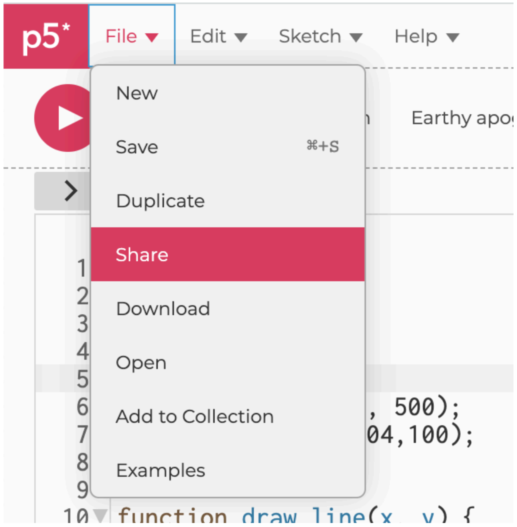
The following prompts come from Genuary, an online event wherein artists create generative art throughout the month of January. You must also incorporate one of these prompts into each sketch you submit, but the choice of prompt is up to you – there isn't a set prompt for each week. You are also free to interpret the prompt in whatever way you like.

1. Drifting clouds
2. Draw a line. Wrong answers only.
3. Glitch Art (see Wikipedia: Glitch Art)
4. Make a grid of at least 100 permutations of something
5. Use sound
6. Infinite Scroll
7. Replicate a natural concept (e.g. gravity, flocking, path following, plant growth)
8. Aesemic (Dictionary: Asemic, Wikipedia: Asemic writing)
9. 500 lines
10. Cellular automata
11. Radial symmetry
12. Come up with some rules, then follow them by hand on paper.
13. Self portrait
14. Noise and/or interference patterns
15. Generative Architecture
16. Text (See, e.g., concrete poetry)
17. One process grows, another process prunes
18. Increase the randomness along the Y-axis
19. No loops
20. Monochrome gradients without lines
21. Code Golf: How little code can you write to make something interesting?
22. Use an API (e.g. the weather). Here's a huge list of free public APIs.
23. Make something that will look completely different in a year.
24. Something that reminds you of home (e.g. a city skyline or geological feature)
25. Negative space
26. Collage
27. Organic output using only rectangular shapes
28. Video game tileset
29. Isometric perspective
30. Zooming in
31. Mythology

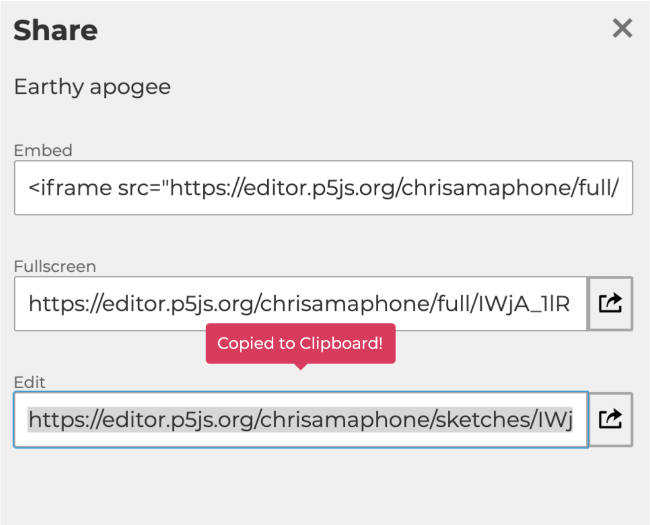
Sharing p5.js Sketches

To save and share sketches on p5.js, you will need to either create an account on the p5.js website or have a github account created already. Use these credentials to log in to the p5.js website before you edit.

When you're ready to share your creation and code, go to **File -> Share**, and copy the **Edit link** to paste into your Piazza post.



```
, 500);
04,100);
function draw_line(x, y) {
```



Share

Earthy apogee

Embed: <iframe src="https://editor.p5js.org/chrisamaphone/full/

Fullscreen: https://editor.p5js.org/chrisamaphone/full/IWjA_1IR

Edit: https://editor.p5js.org/chrisamaphone/sketches/IWj

Commenting on Classmates' Sketches

Class participation is worth some percentage of your grade in this class (see the syllabus for details). Some of that participation can come from online interaction within the course communication channels, and offering constructive feedback on your peers' creative work is a great opportunity to earn that credit.

If someone shares their work with the whole class, I encourage you to leave them comments by replying in the thread where they made their post. Critique is welcome, but please be respectful! Here are some guidelines for giving your classmates constructive, respectful feedback on their work:

- Take a moment to really study the piece and its code before commenting. Examine your subjective reaction and any associations you might have to your personal memories (does the pattern remind you of a piece of clothing or a building you have seen?), but also try to formulate objective observations, like you're trying to describe the piece to someone who can't see it.

- If you like what they did, get specific about **what** you like. For example, don't just say "looks cool!," say, e.g., "I like the way the overlapping lines create the illusion of movement."
- Similarly, if you think the piece is boring, unimaginative, or unappealing in some way, get specific about **what** doesn't work for you. Try not to give them specific advice based on what you would do, but instead describe the cause-and-effect relationship between their choices for the piece and the effects it has on you. For example, instead of saying "This is kind of dull - try adding some color!", you could say "the greyscale palette doesn't grab my attention; my eyes can't figure out where to focus."
    - Always make sure your comments are about the **work** and not the **person** who created it. Personal insults will not be tolerated.
- You can comment on many different aspects of the work, including:
    - The code
    - The appearance of the results shown in the screenshots
    - The way the results change, move, or animate in the rendered version
    - The relationship between this piece and the student's prior work, noticing recurring themes or improved skill across multiple pieces.

Getting Inspiration

Aside from checking out your classmates' posts, you can also search your favorite social media site (e.g. Mastodon, Instagram, Twitter, Tumblr) for the #genuary hashtag, or do a google search for "genuary generative art". Here are a few examples I personally enjoy:

- Ram-n's responses to Genuary 2021: https://ram-n.github.io/Genuary_2021/
- Roni Kaufman on Mastodon: https://genart.social/@ronikaufman/media
- Loren Schmidt on Mastodon: https://genart.social/@lorenschmidt/media
- Ayliean, a Scottish math/art creator: https://ayliean.com/math-art
- Solving Sol, a collaborative repository of code implementations of Sol Lewitt instructions: https://solvingsol.com/
- Feel free to share your own inspiration! (Please always give attribution to any artists whose work directly inspires you!)

Getting Help

If you're struggling with getting your code to work well enough for a complete submission, here are a few things to try:

- Follow the design recipe! If you're stuck and not sure what to do next, the design recipe will tell you exactly what to do. This should always be your first line of defense.
- Review the lecture notes/lesson plans for the previous week, and try using the starter code there. (Remember to add an attribution for the starter code if you use it!)

- Go to the TA’s or instructor’s office hours if you need help with the fundamentals and techniques covered in class, or if you aren’t sure how to follow the next design recipe step.
- Book an appointment with CAMD’s Coding Clinic (they specifically have p5.js experience!)
- Ask a peer for help. Since there are no “right answers” to the weekly sketches, it’s perfectly fine for a classmate to help you (or even, with their permission, to use their previous sketch as a starting point for your own) – with the appropriate attribution, of course. However, you must be the primary “creative owner” of the piece that you post. If you’re not sure, a good test is that you should be able to explain, individually, all of the code that appears in your sketch.

Midsemester Show-and-Tell

One week before spring break, students will be asked to share in class 1 sketch they are proud of and/or wish to build on for a final project. Details TBD.