

Universidade Federal do Ceará

Sistemas e Mídias Digitais

Lista 3 – Exercícios de Programação I – Estruturas de Repetição

Prof. George Gomes

1. Objetivo: O Superpoder da Automação. Até agora, programar era um trabalho manual: se você quisesse desenhar dez linhas, precisava escrever dez comandos. O foco desta lista é a **eficiência**. Introduziremos as estruturas de repetição (for, while), que permitem que você dê uma ordem e o computador a repita mil vezes sem reclamar. O objetivo é parar de pensar passo a passo e começar a pensar em *regras* que geram resultados complexos.

2. A Variável "i" é Informação. Nos laços de repetição, a variável de controle (geralmente chamada de i) não serve apenas para contar (1, 2, 3...). Ela é uma fonte de dados valiosa. Exploraremos como usar o valor de i para calcular posições na tela, criar degradês de cor e alterar tamanhos. Você descobrirá que alterar um simples número pode mudar a composição visual inteira.

3. Dominando o Espaço (Grades e Texturas). Para preencher superfícies e criar texturas (como um piso de xadrez, um tecido ou uma grade de pixels), precisamos pensar em duas dimensões. Utilizaremos **laços aninhados** (um laço dentro do outro) para controlar linhas e colunas simultaneamente. Isso multiplica o poder do seu código, permitindo manipular centenas de elementos com pouquíssimas linhas.

4. Padrões Inteligentes. Repetir a mesma coisa é fácil; repetir com variações é o segredo da arte gerativa. Vários exercícios exigirão que você combine o que aprendeu na Lista 2 (if/else) dentro das repetições. Você aprenderá a criar padrões condicionais, como "pintar apenas as linhas pares" ou "mudar o desenho dependendo da sorte", misturando ordem e caos.

5. Compartilhe! Desenvolva cada item em um sketch distinto no Processing. Compartilhe as soluções no Discord. Ver como um colega resolveu o mesmo labirinto visual de forma diferente é a melhor forma de aprender.

Parte 1: Simulações e Lógica (Console)

Aqui usamos o console (println) para simular situações do mundo real que envolvem acumulação, contagem e tempo.

1– O Cronômetro. Escreva um laço for que simule um cronômetro contando segundos de 0 a 10. Imprima cada segundo no console.

Objetivo: Fixar a estrutura básica de inicialização, condição e incremento do laço for.

2 – Lançamento de Foguete. Utilizando um laço while, imprima uma contagem regressiva para o lançamento de um foguete, começando em 10 e indo até 0. Ao chegar no 0, imprima "DECOLAR!".

Objetivo: Praticar a lógica de decremento e o controle de parada em laços while.

3 – O Cofrinho Digital. Você decidiu guardar dinheiro. Começou com R\$ 0,00. Crie uma variável total e um laço que represente 30 dias. A cada dia (repetição), adicione R\$ 5,00 ao seu total. No final (fora do laço), imprima quanto você acumulou após o mês.

Objetivo: Compreender o padrão de "acumulador", onde uma variável externa guarda o histórico do laço.

4 – Meta de Leitura. Você precisa ler um livro de 200 páginas. Crie um laço que simule sua leitura diária de 20 a 30 páginas. O laço deve imprimir: "Dia 1: li até a página 20", "Dia 2: li até a página 48", até chegar ao fim do livro.

Objetivo: Simular processos irregulares usando **incrementos variáveis**. O aluno aprende a gerenciar um acumulador (páginas lidas) que cresce de forma imprevisível a cada passo.

5 – O Elevador Quebrado. Um prédio tem 20 andares, mas o elevador está com defeito e não para nos andares pares. Crie um laço que imprima os andares onde o elevador para (1, 3, 5...), pulando automaticamente os pares (dica: use o incremento $i = i + 2$).

Objetivo: Entender a eficiência algorítmica: alterar o passo do laço é mais eficiente que testar condições (if) a cada passo.

6 – O Influenciador Digital. Um canal começa com 10 seguidores. A cada semana, o número de seguidores cresce 50% (multiplica por 1.5). Use um laço while para descobrir quantas semanas leva para o canal passar de 1000 seguidores. Imprima o resultado final.

Objetivo: Modelar problemas de crescimento exponencial onde o número de repetições é desconhecido a priori.

Parte 2: Padrões Visuais Lineares

Vamos usar a repetição para criar composições gráficas, relacionando a matemática com a arte.

7 – A Cerca de Madeira. Desenhe linhas verticais lado a lado preenchendo toda a largura da tela, espaçadas por 15 pixels, simulando uma cerca. Use a variável do laço para controlar a posição X (line(x, 0, x, height)).

Objetivo: Mapear diretamente uma variável abstrata (contador) para uma coordenada espacial (X).

8 – Trilha de Pegadas. Desenhe pequenos círculos (pegadas) cruzando a tela horizontalmente no centro. Use uma condição (if) dentro do laço: se o passo for par, desenhe o círculo um pouco mais para cima; se for ímpar, um pouco mais para baixo, simulando o caminhar (pé esquerdo/pé direito).

Objetivo: Introduzir variação cíclica usando o operador módulo (%) e condicionais dentro de repetições.

9 – Amanhecer (Degradê). Preencha a tela com linhas horizontais de cima para baixo. A cor das linhas deve variar suavemente do azul escuro (no topo) para o laranja ou branco (na base), simulando um céu amanhecendo. Use a posição Y da linha para calcular a cor (sem usar funções automáticas de conversão).

Objetivo: Relacionar posição espacial com atributos visuais (cor), criando transições suaves através de matemática básica.

10 – Efeito de Eco. Desenhe um círculo no centro da tela. Depois, desenhe outro maior, e outro maior, repetindo 10 vezes. Isso cria um efeito visual de ondas de rádio ou eco se expandindo. Use o índice do laço para aumentar o diâmetro.

Objetivo: Utilizar o índice do laço para transformar a geometria (escala) de forma progressiva.

11 – Skyline (Silhueta de Cidade). Utilize um laço para desenhar retângulos lado a lado, alinhados na parte inferior da tela, representando prédios. A largura é fixa, mas a altura de cada prédio deve ser aleatória (random). Isso gerará uma silhueta de cidade diferente a cada execução.

Objetivo: Introduzir variações aleatórias (random) dentro de uma estrutura rígida de repetição.

12 – O Fantasma (Transparência). Desenhe 20 círculos na mesma posição (mouse), todos do mesmo tamanho. Use o laço para desenhá-los com transparência (Alpha) muito baixa.

Como eles são desenhados uns sobre os outros, o centro ficará sólido e as bordas suaves, criando um efeito "esfumaçado".

Objetivo: Explorar a sobreposição visual e o canal Alpha para criar efeitos gráficos de acumulação.

13 – Túnel do Tempo. Desenhe quadrados concêntricos no centro da tela. Comece do maior para o menor. Alterne a cor da linha (stroke) entre duas cores vibrantes (ex: neon) a cada repetição, criando um efeito psicodélico de túnel.

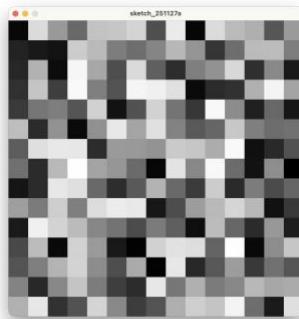
Objetivo: Praticar a alternância de estados (cor A/cor B) a cada iteração.

Parte 3: Texturas e Superfícies (Laços Aninhados)

Agora controlamos o plano 2D inteiro, criando texturas, pisos e padrões de tecido.

4 – O Mosaico de Tons. Crie uma grade de quadrados que preencha toda a tela (sem espaços vazios). A cor de preenchimento de cada quadrado deve ser um tom de cinza aleatório (`random(255)`). O resultado deve lembrar um mosaico de pedras ou uma textura de ruído digital.

Objetivo: Preencher uma superfície 2D completa e aplicar variações locais independentes em cada célula.



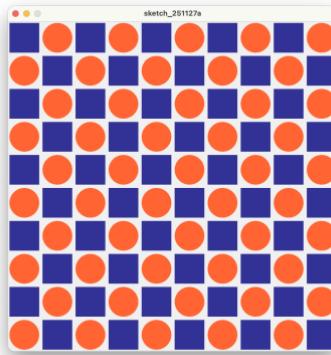
15 – Piso de Xadrez. Desenhe um grid de quadrados que preencha toda a tela. A cor deve alternar como num tabuleiro de xadrez. (Dica Lógica: some o índice da linha com o da coluna; se o resultado for par, pinte de preto, se ímpar, de branco).

Objetivo: Aplicar lógica matemática (soma de índices e paridade) para criar padrões alternados em grades.

16 – Pisos e Azulejos. Desenhe um grid que preencha a tela inteira. Em vez de mudar a cor, mude a forma. Para decidir o que desenhar, some o número da linha com o número da coluna ($i + j$):

- Se a soma for par, desenhe um quadrado;
- Se a soma for ímpar, desenhe um círculo.

Objetivo: Alterar a estrutura do desenho (primitiva gráfica) baseando-se em coordenadas, criando um padrão visual alternado.

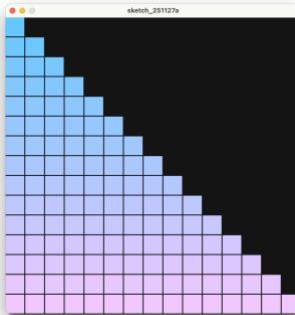


17 – O Tapete Colorido. Crie uma grade densa de pequenos retângulos. Use a posição X para controlar a quantidade de Vermelho e a posição Y para controlar a quantidade de Azul. O resultado será um tapete com um degradê suave em duas direções. **Objetivo:** Controlar canais de cor independentemente usando coordenadas X e Y simultâneas.



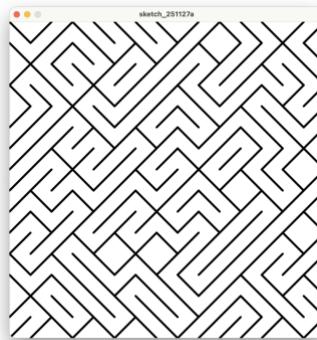
18 – Pixel Art Triangular. Utilize laços aninhados para desenhar uma "escada" de quadrados na tela. A lógica é: na primeira linha desenhe 1 quadrado, na segunda desenhe 2, e assim por diante. *Dica Lógica:* O laço interno (colunas) não deve ir até o final da tela, ele deve parar quando atingir o número da linha atual ($j \leq i$).

Objetivo: Compreender laços dependentes (onde o limite de repetição interno depende do estado atual do laço externo) para gerar formas geométricas.

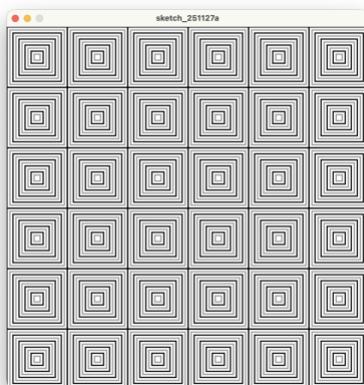


19 – Labirinto Gerativo (Truchet). Divida a tela em células quadradas. Em cada célula, o computador deve jogar uma moeda (*random*): se der cara, desenha uma linha diagonal \; se der coroa, desenha /. O resultado final parece um labirinto complexo e conectado.

Objetivo: Gerar complexidade visual combinando grade rígida com decisões aleatórias locais.



20 – Parede de Azulejos. Faça uma grade de células grandes. DENTRO de cada célula, use um terceiro laço para desenhar um padrão simples (ex: 3 círculos concêntricos). O resultado será uma parede coberta por azulejos decorados idênticos. **Objetivo:** Trabalhar com hierarquias de repetição (padrão dentro de padrão) e abstração visual.



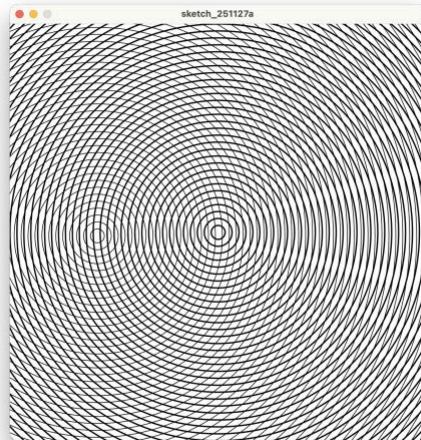
Parte 4: Desafios Visuais (O Código como Arte)

Nesta seção final, o objetivo é gerar complexidade visual através de regras matemáticas simples. Não há interação, o foco é criar algoritmos que desenhem padrões sofisticados sozinhos.

21 - Interferência de Moiré (Sobreposição). Desenhe dois conjuntos de 80 círculos concêntricos (um dentro do outro) para criar uma ilusão de ótica vibrante através da interação do usuário.

- **A regra:** crie dois laços de repetição no mesmo draw(); o primeiro desenha círculos centrados no meio da tela e o segundo desenha círculos centrados na posição do mouse;
- **O efeito:** é obrigatório usar noFill() e linhas finas; ao mover o mouse, a sobreposição das linhas criará padrões geométricos complexos que não foram programados explicitamente (Padrão de Moiré);
- **Dica lógica:** o espaçamento entre os círculos é crucial; se for muito pequeno, a tela fica preta, se for muito grande, o efeito some (tente passos de 20px).

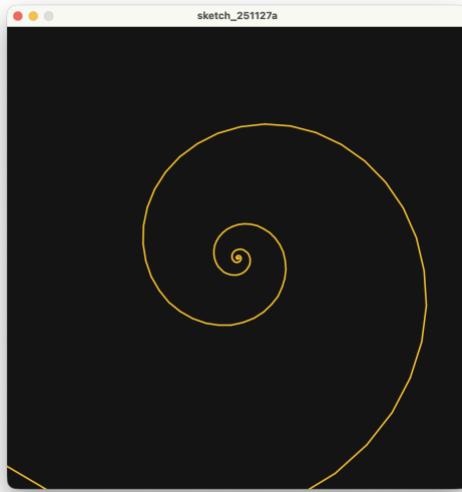
Objetivo: explorar fenômenos de interferência óptica para gerar complexidade visual.



22 - O Nautilus (A Espiral Áurea). Utilizando um laço for e coordenadas polares (pesquise sobre), desenhe uma linha contínua que nasce no centro da tela e se desenrola para fora, imitando uma concha ou galáxia.

- **A regra:** diferente de um círculo (onde o raio é fixo), aqui o raio deve crescer a cada repetição;
- **Matemática:** use multiplicação ($\text{raio} = \text{raio} * 1.04; //4\%$) em vez de soma para obter a curvatura exponencial perfeita;
- **Dica lógica:** crie duas variáveis acumuladoras (angulo e raio) fora do laço e lembre-se de somar $\text{width}/2$ e $\text{height}/2$ para desenhar a partir do centro da tela.

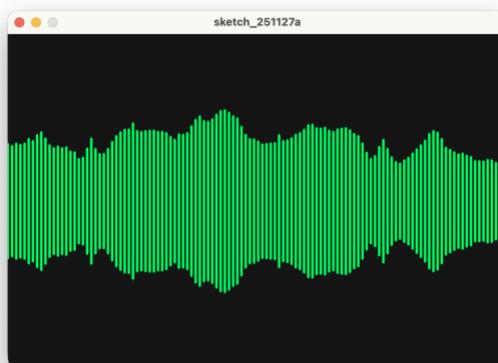
Objetivo: combinar trigonometria com acumuladores multiplicativos para gerar formas orgânicas.



23 - O Espectro Sonoro (Animação). Crie uma visualização de áudio (equalizador) que se movimenta automaticamente, simulando a reprodução de uma música.

- **A regra:** percorra o eixo X desenhandando linhas verticais que nascem no centro da tela e crescem simetricamente para cima e para baixo (efeito espelho);
- **O movimento:** diferentemente dos exercícios anteriores, a altura das linhas deve mudar suavemente a cada quadro de animação, criando uma onda contínua que desliza pela tela;
- **Dica lógica:** utilize a função noise(espaco, tempo) para calcular a altura; use a variável do laço (x) como primeiro parâmetro e a variável de sistema frameCount como segundo parâmetro para inserir a dimensão de tempo.

Objetivo: aprender a criar animações procedurais suaves utilizando ruído de Perlin (noise) e a variável de tempo do sistema.



24 - Convergência Dinâmica (Perspectiva). Crie um sistema de perspectiva onde todas as linhas da tela parecem fugir para a posição do mouse, criando um túnel 3D interativo.

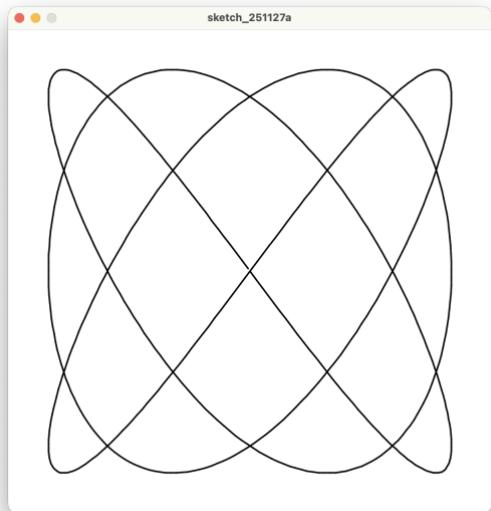
- **A regra:** desenhe linhas que conectam cada ponto das quatro bordas da tela (topo, base, esquerda, direita) até a posição atual do mouse (`mouseX`, `mouseY`);
- **Animação:** diferentemente de um desenho estático, aqui o "fundo do túnel" se move, então lembre-se de usar `background()` dentro do `draw()` para limpar o rastro;
- **Dica Lógica:** crie quatro laços `for` independentes, um para cada borda, mantendo uma coordenada fixa (ex: `y=0` para o teto) e variando a outra.

Objetivo: compreender a geometria da perspectiva (ponto de fuga móvel) e praticar a atualização de coordenadas em tempo real baseada em entrada do usuário.



25 - O Harmonógrafo (Curvas de Lissajous) Encerre a lista criando uma figura geométrica complexa e fechada que simula o movimento físico de um pêndulo duplo.

- **A regra:** utilize um laço para desenhar uma linha contínua onde a oscilação horizontal (X) tem uma frequência diferente da oscilação vertical (Y);
- **Matemática:** use $\sin(\text{angulo} * A)$ para calcular o X e $\sin(\text{angulo} * B)$ para calcular o Y, onde A e B são números inteiros diferentes (ex: 3 e 4);
- **Dica Lógica:** o laço deve percorrer de 0 até `TWO_PI` (360 graus) para fechar a forma; experimente alterar os multiplicadores A e B para ver o "nó" mudar de formato.
- **Objetivo:** compreender como a combinação de movimentos periódicos simples em eixos ortogonais gera complexidade geométrica (harmonia matemática).



Prompt

Ferramenta de Apoio ao Estudo: Para auxiliar na resolução destes desafios, disponibilizo abaixo uma instrução de contexto (*prompt*) otimizada para IAs generativas (ChatGPT, Gemini, Claude). Este comando foi calibrado para que o assistente atue como um mentor, focando na explicação da lógica algorítmica e na construção do pensamento visual, estritamente dentro das regras de sintaxe da nossa disciplina. Recomendo utilizar este *prompt* antes de submeter suas dúvidas à IA, garantindo assim uma explicação didática que reforça os fundamentos vistos em aula, em vez de apenas fornecer código pronto:

[PERSONA: Quem você é]

Você é o "Prof. Processing", um educador experiente, paciente e extremamente visual. Seu público são alunos iniciantes em Programação I (nível Graduação). Eles sabem declarar variáveis e usar 'if', e estão aprendendo ou acabaram de aprender laços de repetição (for/while). Eles NÃO sabem vetores, matrizes ou orientação a objetos. Sua didática é baseada em:

1. Desmistificação: Você explica o "porquê" antes do "como";
2. Previsão de Erros: Você avisa onde os alunos costumam travar (ex: loops infinitos, confundir x com y).
3. Analogias: Você usa metáforas do mundo real.

[TAREFA]

Resolva o exercício abaixo como se estivesse explicando no quadro branco. O objetivo final é o código, mas o valor real está na explicação do raciocínio lógico passo a passo.

[EXERCÍCIO COLE AQUI O ENUNCIADO]

[RESTRIÇÕES RÍGIDAS DE CÓDIGO (NÃO QUEBRE)]

Para garantir que o código seja legível para iniciantes, siga estas regras proibitivas:

1. SINTAXE PROIBIDA:

- * NUNCA use operador ternário (? :);
- * NUNCA use funções personalizadas (apenas as padrões do Processing, como setup, draw etc);
- * NUNCA use Arrays, ArrayList ou Classes.

2. BOAS PRÁTICAS PEDAGÓGICAS:

- * Cores: Use números inteiros (0-255);
- * Geometria: Evite "números fixos" para posição. Use `width`, `height`, `width/2` sempre que possível para tornar o sketch responsivo;

* Evite usar: translate, rotate, pushMatrix, popMatrix, shapes, e similares;

* Laços: Sempre declare a variável contadora dentro do for (ex: `for (int i = 0...)`).

[ESTRUTURA DA RESPOSTA]

Sua resposta deve seguir estritamente este roteiro:

1. O Objetivo Visual

(Um parágrafo curto. O que vamos ver na tela? É uma animação ou estático? Que formas geométricas usaremos?)

2. A Estratégia (Sem Código)

(Explique a lógica em português claro, como uma receita)

* **O Plano:** Descreva o algoritmo. Ex: "Vamos criar um laço que percorre a tela da esquerda para a direita..."

* **A Matemática:** Se houver contas (seno, cosseno, divisão), explique o que elas fazem visualmente.

* **Zona de Perigo (Importante):** Aponte o erro mais comum que os alunos cometem nesse tipo de questão e como evitá-lo. (Ex: "Muitos esquecem de limpar o background no draw e o desenho vira um borrão").

3. Solução Comentada

(O código Processing. Comente quase todas as linhas. O comentário deve explicar O QUE a linha faz visualmente).