

# PCS 2290 - AP: Algoritmos na Prática

Ricardo Nakamura e Romero Tori

2015

## 1 Algoritmos no Processing

Na última aula, vimos que algoritmos são sequências de ações bem definidas e que ocorrem em um tempo limitado. Vimos também que programas de computador são algoritmos, cujas ações são de tipos bem específicos. Nesta aula, vamos realizar algumas atividades práticas envolvendo a criação de algoritmos computacionais e sua implementação no Processing.

### Atividade 1

Faça um programa no Processing correspondente ao algoritmo de desenhar polígonos proposto na aula passada.

Como já aprendemos através das atividades de laboratório de PCS 2190, um programa no Processing pode conter os blocos *setup* e *draw*. Os comandos do bloco *setup* são executados somente uma vez no início do programa. Por outro lado, o bloco *draw* corresponde a um laço de repetição que é executado continuamente durante a vida do programa. Desta forma, o algoritmo que descreve um programa do Processing contendo estes blocos é o seguinte:

```
executar os comandos do setup  
enquanto o programa está executando faça  
  ler teclado e mouse  
  executar os comandos do draw  
  atualizar a janela gráfica  
  se a janela gráfica foi fechada então  
    terminar a execução do programa
```

Podemos aproveitar esta estrutura ao implementarmos um algoritmo no Processing. Considere, por exemplo, o seguinte algoritmo para desenhar um tabuleiro de xadrez na tela:

```
pintar a tela de preto  
linha ← 0  
enquanto linha < 8 faça  
  coluna ← 0  
  enquanto coluna < 8 faça  
    se linha é par e coluna é ímpar ou coluna é par e linha é  
    ímpar então  
      desenhar um quadrado branco na posição (coluna,  
      linha)  
      coluna ← coluna + 1  
  linha ← linha + 1
```

Este algoritmo pode ser transformado no seguinte programa do Processing, sem utilizar os blocos *setup* e *draw*:

```
size(160,160);  
background(0);  
int linha = 0;  
noStroke();  
fill(255);  
while (linha < 8) {  
  int coluna = 0;  
  while (coluna < 8) {  
    if ((linha%2 == 0 && coluna%2 != 0) ||  
        (linha%2 != 0 && coluna%2 == 0)) {  
      rect(20*coluna, 20*linha, 20, 20);  
    }  
    coluna++;  
  }  
  linha++;  
}
```

O programa da listagem anterior desenha todo o tabuleiro de xadrez e então exibe o resultado na tela. Por outro lado, podemos observar que o bloco *draw* do Processing corresponde a um laço de repetição – que podemos utilizar para representar o laço de repetição mais externo do algoritmo.

Note que, neste caso, a janela gráfica será atualizada a cada linha desenhada e, como resultado, poderemos ver a imagem do tabuleiro ser desenhada progressivamente. Para tornar o efeito mais fácil de observar, acrescentamos também o comando *frameRate(1)*, fazendo com que a janela gráfica seja atualizada somente uma vez por segundo. A listagem a seguir mostra a nova versão do programa:

---

```
int linha = 0;

void setup() {
  size(160,160);
  background(0);
  noStroke();
  fill(255);
  frameRate(1);
}

void draw() {
  if (linha < 8) {
    int coluna = 0;
    while (coluna < 8) {
      if ((linha%2 == 0 && coluna%2 != 0) ||
          (linha%2 != 0 && coluna%2 == 0)) {
        rect(20*coluna, 20*linha, 20, 20);
      }
      coluna++;
    }
    linha++;
  }
}
```

---

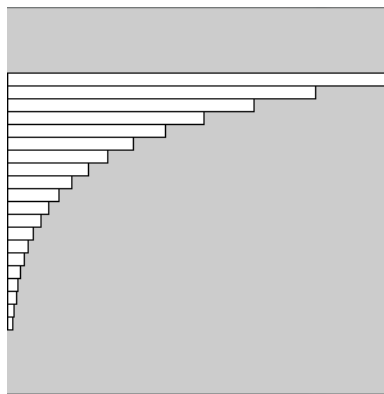
Compare as duas listagens. Você consegue enxergar a correspondência entre os comandos do Processing e o algoritmo original? Aproveite o tempo do laboratório para tirar suas dúvidas. Observe também que um comando do algoritmo pode corresponder a várias linhas do programa. Por exemplo, o comando “desenhar um quadrado branco na posição (coluna, linha)” corresponde a três linhas (você consegue identificar quais?)

## Atividade 2

Modifique o programa da listagem anterior para que seja possível ver o tabuleiro sendo desenhado quadrado a quadrado (ao invés de linha a linha). Você diria que este programa modificado corresponde a um novo algoritmo? Neste caso, escreva esse algoritmo em pseudo-linguagem.

## Atividade 3

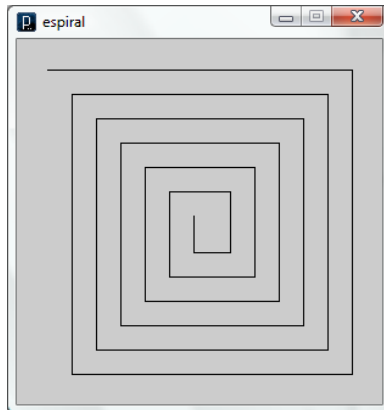
Escreva, em pseudo-linguagem, um algoritmo para desenhar uma pilha de barras, onde cada barra tem comprimento igual a 80% da anterior, como na figura abaixo. Em seguida, escreva um programa em Processing que implementa este algoritmo.



## Atividade 4

**DESAFIO:** Escreva, em pseudo-linguagem, um algoritmo para desenhar uma “espiral quadrada”, como na figura abaixo. Em se-

guida, escreva um programa em Processing que implementa este algoritmo.



## 2 Conclusão

O objetivo desta aula foi colocar em prática os conceitos de construção de algoritmos vistos anteriormente. Para isso, foram propostas várias atividades envolvendo a criação de algoritmos e sua implementação no Processing.

Pudemos observar também que, como algoritmos descrevem uma série de operações lógicas de forma abstrata, muitos detalhes (que não interferem na sua lógica) não são expressos neles. Estes detalhes irão depender da linguagem ou ferramenta de programação utilizada para implementar o algoritmo computacional.

### Atividade 5

Em sua opinião, é possível ter diferentes programas (na mesma linguagem) que correspondem exatamente ao mesmo algoritmo? Neste caso, onde estariam as diferenças?