

# Ladrilhos de Truchet

Jorge Cardoso

18 de Outubro de 2005

Built with L<sup>A</sup>T<sub>E</sub>X

# Conteúdo

<b>1</b>	<b>Resumo</b>	<b>1</b>
<b>2</b>	<b>Requisitos/Configuração</b>	<b>1</b>
<b>3</b>	<b>Introdução</b>	<b>1</b>
<b>4</b>	<b>Descrição do Programa</b>	<b>1</b>
<b>5</b>	<b>Implementação</b>	<b>2</b>
5.1	Desenhar um Ladrilho no Ecrã . . . . .	2
5.2	Preencher o Ecrã com Ladrilhos . . . . .	3
5.3	Escolher um Ladrilho Aleatoriamente . . . . .	4
5.4	Listagem do Código . . . . .	5
<b>6</b>	<b>Adaptações</b>	<b>6</b>

# 1 Resumo

O programa `Truchet` gera imagens recorrendo aos ladrilhos de Truchet.

## 2 Requisitos/Configuração

- Processing (BETA) [2]

## 3 Introdução

Em 1704, Sebastien Truchet<sup>1</sup>, um padre Dominicano, estudou os arranjos gráficos que poderiam ser obtidos recorrendo a um pequeno conjunto de ladrilhos. A Figura 1 mostra os ladrilhos originais utilizados por Truchet.

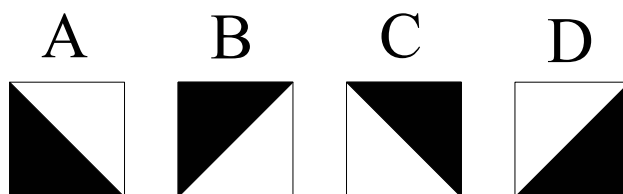


Figura 1: Ladrilhos originais de Truchet

Os ladrilhos de Truchet podem resultar em belas imagens quando dispostos numa grelha, como exemplificado na Figura 2.

A forma como os ladrilhos são dispostos na grelha origina imagens com determinadas características.

Existem muitas variações dos ladrilhos de Truchet. Uma das mais utilizadas é a substituição dos triângulos por arcos. Neste caso apenas se usam dois ladrilhos diferentes (Figura 3). A Figura 4 mostra o resultado da utilização destes ladrilhos.

## 4 Descrição do Programa

O programa `Truchet` gera uma composição aleatória de ladrilhos de Truchet, no ecrã. A probabilidade da ocorrência de cada um dos quatro ladrilhos é igual, i.e., a probabilidade de ocorrer um ladrilho é 25%. A dimensão de cada ladrilho e a dimensão do ecrã são definidas em variáveis, de forma a podermos facilmente gerar imagens com diferente número de ladrilhos.

---

<sup>1</sup>Foi também ele que criou o sistema de pontos utilizado para medir caracteres.

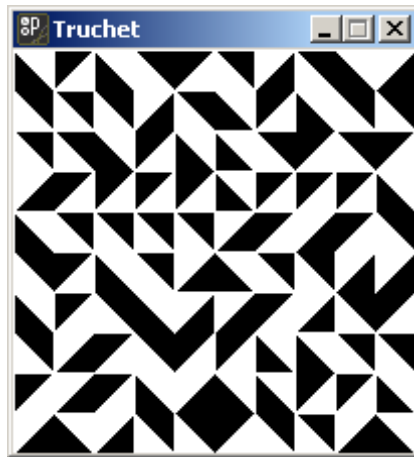


Figura 2: Composição aleatória com os ladrilhos originais de Truchet

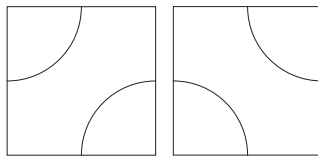


Figura 3: Ladrilhos formados por arcos

## 5 Implementação

### 5.1 Desenhar um Ladrilho no Ecrã

Cada um dos ladrilhos do Truchet pode ser facilmente desenhado no ecrã através de um triângulo preenchido a preto. A primitiva gráfica `triangle(x1, y1, x2, y2, x3, y3)` do Processing pode ser usada para o efeito.

Dado um ponto  $(x, y)$  no ecrã correspondente ao canto superior esquerdo de um ladrilho, cada um dos ladrilhos de Truchet (A, B, C e D) pode ser desenhado da seguinte forma (`dimensaoLadrilho` contém a dimensão do ladrilho):

```
Ladrilho A triangle(x, y,  
    x+dimensaoLadrilho, y+dimensaoLadrilho,  
    x, y+dimensaoLadrilho)
```

```
Ladrilho B triangle(x, y,  
    x+dimensaoLadrilho, y,  
    x, y+dimensaoLadrilho);
```

```
Ladrilho C triangle(x, y,
```

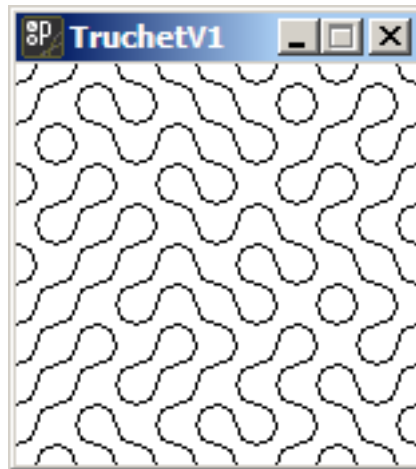


Figura 4: Ladrilhos com arcos

```
x+dimensaoLadrilho, y,  
x+dimensaoLadrilho, y+dimensaoLadrilho)
```

```
Ladrilho D triangle(x+dimensaoLadrilho, y,  
x+dimensaoLadrilho, y+dimensaoLadrilho,  
x, y+dimensaoLadrilho);
```

## 5.2 Preencher o Ecrã com Ladrilhos

Uma vez que sabemos à partida a dimensão do ecrã (para facilitar vamos definir um ecrã quadrado, i.e., com a largura igual à altura) e sabemos também a dimensão de cada ladrilho (um ladrilho é também quadrado), podemos determinar também a posição de todos os ladrilhos no ecrã.

No caso de um ecrã de dimensão 100x100 e ladrilhos de dimensão 10x10 e considerando que posicionamos os ladrilhos através do canto superior esquerdo, sabemos que vamos ter ladrilhos nas posições (0, 0), (10, 0), (20, 0), (30, 0), (40, 0), (50, 0), (60, 0), (70, 0), (80, 0), (90, 0), que correspondem aos ladrilhos na primeira linha; e nas posições (0, 10), (10, 10), ... , (90, 10), que correspondem à segunda linha, etc. O último ladrilho estará na posição (90, 90). A Figura 5 mostra o posicionamento dos ladrilhos no ecrã.

Assim sendo podemos facilmente gerar todas as posições dos ladrilhos recorrendo a dois ciclos `for`:

```
for (int x = 0; x < 100; x = x + 10) {  
    for (int y = 0; y < 100; y = y + 10) {  
        // desenhar ladrilho na posição (x, y)  
    }  
}
```

}

Este ciclo desenha ladrilhos de dimensão 10 pixeis, num ecrã de dimensão 100 pixeis.

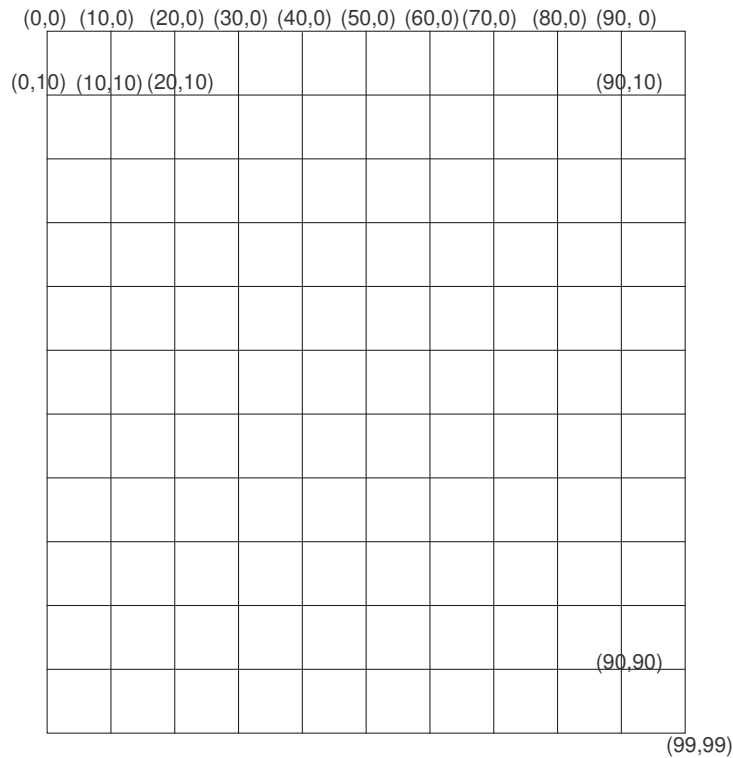


Figura 5: Posicionamento dos ladrilhos

### 5.3 Escolher um Ladrilho Aleatoriamente

Para escolhermos aleatoriamente qual o ladrilho a desenhar numa determinada posição podemos recorrer ao método `random()` do Processing. Se gerarmos um número aleatório entre 0 e 100 (`random(0, 100)`), podemos determinar o ladrilho da seguinte forma:

- Se o número gerado pertencer ao intervalo  $[0, 25[$ , então desenhamos o ladrilho A.
- Se o número gerado pertencer ao intervalo  $[25, 50[$ , então desenhamos o ladrilho B.
- Se o número gerado pertencer ao intervalo  $[50, 75[$ , então desenhamos o ladrilho C.

- Se o número gerado pertencer ao intervalo  $[75, 100[$ , então desenhamos o ladrilho D.

Desta forma garantimos que todos os ladrilhos são desenhados com a mesma probabilidade.

## 5.4 Listagem do Código

```
// Ladrilhos de Truchet
// Autor: Jorge Cardoso
// Data: 30 Julho 2005
// Versão 0.5

/* a dimensão da janela de visualização */
int dimensaoJanela = 200;

/* a dimensão do ladrilho */
int dimensaoLadrilho = 10;

/* Número de ladrilhos numa linha */
int grelha = dimensaoJanela/dimensaoLadrilho;

/* Número aleatório utilizado para escolher o ladrilho */
float rand;

/* Inicialização */
void setup() {
    size(dimensaoJanela, dimensaoJanela);
}

void draw() {
    /* primeiro vamos limpar a janela */
    background(0);

    /* se quisermos ver uma grelha, descomentar o código
    seguinte */
    /*
    stroke(255, 255, 255);
    for (int i = 0; i < dimensaoJanela; i = i +
        dimensaoLadrilho) {
        line(i, 0, i, dimensaoJanela);
        line(0, i, dimensaoJanela, i);
    }
    */
}
```

```

/* queremos os ladrilhos pintados a branco e também com o
   borda a branco */
stroke(255);
fill(255);

/* percorrer a grelha de ladrilhos */
for (int i = 0; i < dimensaoJanela; i = i +
    dimensaoLadrilho) {
    for (int j = 0; j < dimensaoJanela; j = j +
        dimensaoLadrilho) {

        /* vamos posicionar os ladrilhos aleatoriamente.
           */
        rand = random(0, 100);
        if (rand < 25) {
            // Ladrilho A
            triangle(i, j, i+dimensaoLadrilho, j+
                dimensaoLadrilho, i, j+dimensaoLadrilho);
        }
        else if (rand < 50) {
            //Ladrilho B
            triangle(i, j, i+dimensaoLadrilho, j, i, j+
                dimensaoLadrilho);
        }
        else if (rand < 75) {
            // Ladrilho C
            triangle(i, j, i+dimensaoLadrilho, j, i+
                dimensaoLadrilho, j+dimensaoLadrilho);
        }
        else if (rand < 100) {
            // Ladrilho D
            triangle(i+dimensaoLadrilho, j, i+
                dimensaoLadrilho, j+dimensaoLadrilho, i, j
                +dimensaoLadrilho);
        }
    }
}
noLoop(); // so queremos uma imagem
}

```

## 6 Adaptações

1. Alterar a percentagem de cada tipo de ladrilho desenhado, i.e., fazer com que a probabilidade de desenhar cada ladrilho seja diferente de 25%.
2. Gerar ladrilhos coloridos aleatoriamente.

3. Gerar ladrilhos coloridos aleatoriamente com um tom de cinza.
4. Utilizar os ladrilhos com arcos.

## Referências

- [1] Eric W. Weisstein, “Truchet Tiling” From MathWorld – A Wolfram Web Resource. <http://mathworld.wolfram.com/TruchetTiling.html>
- [2] Ben Fry and Casey Reas, “Processing (BETA)”, <http://processing.org/>