



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Instituto de Ciências Exatas e de Informática

Computação Gráfica  
*Trabalho Prático I - Curva Paramétrica\**

Gabriel Luciano Gomes<sup>1</sup>

---

\*Readme dos algoritmos vistos em sala, implementados em Python.

<sup>1</sup>Aluno, Ciência da Computação, Brasil, glgomes@sga.pucminas.br.

**Sumário**

<b>1</b>	<b>Linguagem e Biblioteca Utilizada</b>	<b>3</b>
<b>2</b>	<b>Organização do Código Fonte</b>	<b>3</b>
<b>3</b>	<b>Utilização</b>	<b>4</b>
	<b>Referências</b>	<b>6</b>

## 1 LINGUAGEM E BIBLIOTECA UTILIZADA

Para desenvolver o programa, foi utilizado a linguagem Python 3.7, devido sua facilidade e quantidade de recursos ofertados. Junto a ela, foi utilizada a biblioteca PyQt5, que fornece a interface gráfica. Para informações ou download da linguagem ou biblioteca, basta acessar os links: *Python*, *PyQt5*.

## 2 ORGANIZAÇÃO DO CÓDIGO FONTE

O primeiro trabalho (implementação de transformações) foi incrementado com a funcionalidade de inserção de curvas paramétricas, logo este programa foi reutilizado. A implementação foi organizada em três arquivos diferentes. O principal deles é o "*main*", onde se encontra as implementações da interface gráficas e os dados persistentes utilizados no programa. O segundo é o "*implementacoes*", onde se encontram os algoritmos *DDA*, *Bresenhan*, *Cohen-Sutherland*, *Liang-Barsky* e *Curva de Bézier*, que são utilizados para plotagem das retas, circunferências, recortes de tela e curvas paramétricas. As transformações e preenchimentos foram realizados no arquivo *main*, pois demandam a utilização de métodos de interface que lá foram criados. Por fim, o arquivo "*dialogs*" é um arquivo puramente gráfico, onde são criadas as *designs* das caixas de textos para inserção dos valores das transformações das figuras. Vale resaltar também que as transformações são aplicadas em todas as figuras criadas.

Visto que existem alguns algoritmos implementados fora do arquivo "*implementacoes*", a tabela abaixo mapeia a posição deles no arquivo "*main*".

**Tabela 1 – Mapeamento de Algoritmos no arquivo "main"**

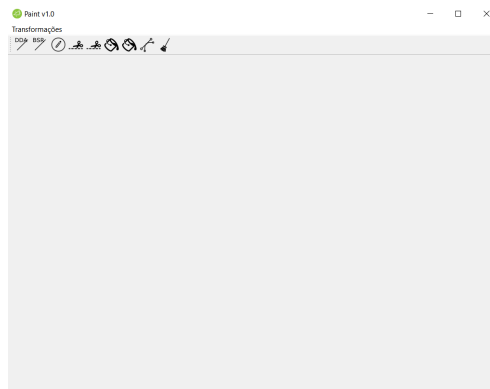
Algoritmo	Linha Início	Linha Fim
Transformação - Translação	273	287
Transformação - Escala	289	308
Transformação - Rotação	310	338
Transformação - Reflexão	340	367
Transformação - Cisalhamento	369	411
Boundary-Fill (Conectividade 4)	427	452
Flood-Fill (Conectividade 4)	454	479

Além disso, para a implementação da curvas paramétrica (Bézier), foram utilizados:

- Lista de tuplas: Guardar a posição dos pontos de controle
- Biblioteca: (lista de dicionários): pontos da curva gerados pelo algoritmo de Bézier

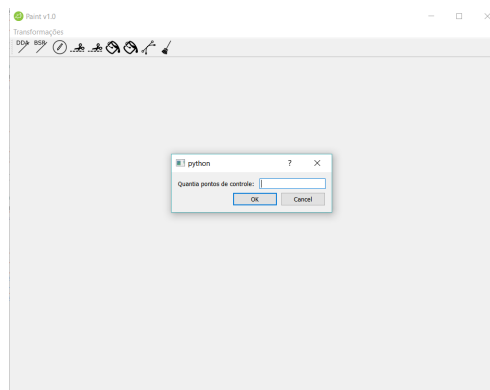
### 3 UTILIZAÇÃO

Para utilização do programa, basta executar o arquivo "*main*" que se encontra na pasta *paint* do arquivo (o Python deverá ser instalado para tal ação). Após isso, basta escolher uma das ferramentas na *toolbar* do programa, representadas por ícones, na parte superior da janela aberta. Caso seja necessário, mantenha o cursor sobre uma das funcionalidades e então será exibida uma mensagem indicando sua ação. A imagem abaixo representa o programa em execução e a *toolbar* mencionada.

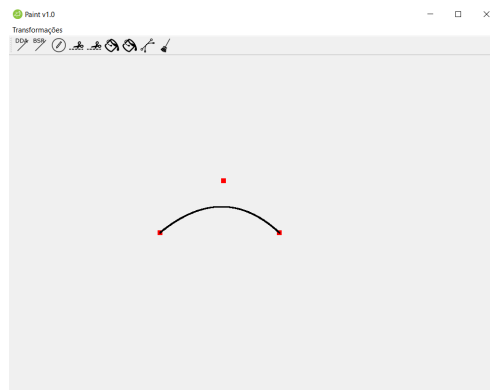


**Figura 1 – Programa executando - Janela Principal**

Após escolhida a função de curva paramétrica (Bezier), basta informar a quantidade de pontos de controle desejada e confirmar. Após isso, clique em pontos quaisquer desejados, com o botão esquerdo do mouse, para definir os pontos de controle. Após isso, a curva será plotada em relação aos controles estabelecidos. As imagens abaixo ilustram os passos descritos acima.



**Figura 2 – Programa executando - Quantia Pontos de Controle**



**Figura 3 – Programa executando - Definição dos pontos de Controle e curva plotada**

Caso seja necessário, o programa contém uma opção para resetar o *workflow*. Para isso, basta clicar no ultimo ícone da *toolbar*, representado como uma vassoura, ou pressionar Ctrl+L.

## **Referências**

PyQt PyQt5 Library. <<https://pypi.org/project/PyQt5/>>. Last Access: 15/09/2018.

Python Org. <<https://www.python.org>>. Last Access: 15/09/2018.

Roberto Gea. <<https://gist.github.com/Alquimista/1274149>>. Last Access: 10/10/2018.

ZetCode PyQt5 tutorial. <<http://zetcode.com/gui/pyqt5/>>. Last Access: 15/09/2018.