# Ferramenta de Visualização para o Problema da Galeria de Arte

## Trabalho Prático 1

## Algoritmos II

**Lucas Vasconcelos Marra, Raphael Alves dos Reis**

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brasil

lucasmarra@ufmg.br, cap497@ufmg.br

# 1 Introdução

Este trabalho aborda a implementação de um algoritmo para resolver o Problema da Galeria de Arte, que envolve a triangulação de polígonos para determinar o número mínimo de guardas necessários para vigiar uma galeria de arte. Utiliza-se o método do corte de orelhas para a triangulação e desenvolve-se uma ferramenta interativa para visualização e acompanhamento da execução do algoritmo.

# 2 Teoria

## 2.1 Triangulação de Polígonos

A triangulação de um polígono consiste em dividir o polígono em triângulos, de forma que os triângulos resultantes não se sobreponham e seus vértices sejam os vértices do polígono original. A triangulação é uma técnica fundamental em geometria computacional com diversas aplicações, incluindo gráficos computacionais, mesclagem de superfícies e soluções para o Problema da Galeria de Arte.

### 2.1.1 Método do Corte de Orelhas

O método do corte de orelhas é um algoritmo clássico para a triangulação de polígonos simples. Uma "orelha"é um triângulo formado por três vértices consecutivos $v_i, v_{i+1}, v_{i+2}$ de um polígono, onde a diagonal $v_i v_{i+2}$ está inteiramente dentro do polígono. O processo de triangulação usando o corte de orelhas envolve os seguintes passos:

1. Identificar uma orelha no polígono.
2. Remover a orelha, formando um triângulo e reduzindo o número de vértices do polígono.
3. Repetir o processo até que o polígono seja completamente decomposto em triângulos.

O critério para identificar uma orelha é verificar se o ângulo formado pelos vértices $v_i, v_{i+1}, v_{i+2}$ é convexo e se nenhum outro vértice do polígono está dentro do triângulo formado por esses três vértices.

## 2.2 Coloração de Grafos

A coloração de grafos é o problema de atribuir cores aos vértices de um grafo de forma que vértices adjacentes (conectados por uma aresta) não tenham a mesma cor. O objetivo é minimizar o número de cores utilizadas.

### 2.2.1 3-Coloração de Grafos Planos

Um resultado importante na teoria dos grafos é o Teorema das Quatro Cores, que afirma que qualquer grafo plano pode ser colorido com no máximo quatro cores de forma que vértices adjacentes tenham cores diferentes. No caso de grafos triangulados (onde cada face é um triângulo), é possível colorir o grafo com no máximo três cores. Esse processo é conhecido como 3-coloração.

Para aplicar a coloração em um polígono triangulado, constrói-se um grafo de adjacência onde os vértices do grafo correspondem aos vértices do polígono e as arestas correspondem às conexões entre os vértices que formam os triângulos. Em seguida, utiliza-se um algoritmo de coloração para atribuir cores aos vértices de forma a minimizar o número de cores, garantindo que vértices adjacentes tenham cores diferentes.

### 2.2.2 Detecção de Casos Especiais e Heurística para Guardas Auxiliares

Em alguns casos, a 3-coloração pode não resolver completamente o problema, deixando vértices não coloridos. Isso ocorre porque a coloração com três cores pode não ser suficiente para cobrir todos os vértices de maneira adequada. Para resolver esses casos, é necessária uma heurística que utilize uma 4-coloração.

O exemplo no final deste relatório exemplifica um caso no qual é encontrado um guarda auxiliar, na segunda imagem, e sem ele o polígono não estaria completamente monitorado pelos guardas da 3-coloração.

A heurística para encontrar guardas auxiliares envolve os seguintes passos:

1. Identificar os vértices que não foram coloridos pela 3-coloração.
2. Aplicar uma 4-coloração, garantindo que todos os vértices sejam coloridos.

Embora essa heurística não garanta sempre uma solução ótima, pois podem haver guardas exatamente sobrepostos, ela assegura que não haverá espaços não monitorados pelos guardas. Isso é uma melhoria em relação à 3-coloração, que pode deixar vértices não coloridos e, consequentemente, áreas não monitoradas.

Essa abordagem, portanto, combina a simplicidade da 3-coloração com a robustez da 4-coloração para garantir uma cobertura completa do polígono.

# 3 Implementação

## 3.1 Importação das Bibliotecas

Utiliza-se `numpy` para manipulações matemáticas, `plotly` para visualização gráfica, `requests` para download de arquivos, `tarfile` para extração de arquivos compactados e `os` para operações no sistema de arquivos. A função `download_and_extract` baixa e extrai os arquivos das instâncias de polígonos simples aleatórios (Random Simple Polygon) disponíveis em https://www.ic.unicamp.br/~cid/Problem-instances/Art-Gallery/AGPVG/index.html.

## 3.2 Funções Básicas Geométricas

### 3.2.1 Leitura do Polígono

A função `read_polygon` lê os vértices de um arquivo no formato descrito a seguir, tal qual como no endereço original.

Cada arquivo de instância consiste em uma linha dividida em duas partes. A primeira parte é um valor inteiro que representa o número de vértices do polígono. A segunda parte é uma sequência dos vértices em sentido anti-horário. Cada vértice é representado por suas coordenadas x e y, cada uma escrita como o quociente de dois inteiros `int/int`.

Como exemplo, aqui está a representação de um quadrado com coordenadas (1, 1), (50, 1), (50, 50) e (1, 50):

```
4   1/1   1/1   100/2   1/1   500/10   50/1   1/1   100/2
```

### 3.2.2 Verificação de Interseção de Arestas

As funções `are_edges_intersecting` e `on_segment` verificam se duas arestas se cruzam, garantindo que o polígono é simples.

### 3.2.3 Detecção de Orelhas

As funções `is_convex`, `is_ear` e `is_point_in_triangle` são utilizadas para identificar e validar orelhas no polígono.

## 3.3 Visualização dos Triângulos

### 3.3.1 Triangulação do Polígono

A função `triangulate` implementa o método do corte de orelhas para dividir o polígono em triângulos. Durante o processo, os vértices são analisados iterativamente, identificando e removendo orelhas até que o polígono esteja completamente triangulado. Frames são criados para cada etapa da triangulação, permitindo visualização passo-a-passo.

### 3.3.2 Plotagem dos Triângulos

A função `plot_triangles` utiliza `plotly` para exibir o polígono e os triângulos resultantes. Adicionalmente, vértices são coloridos para minimizar o número de cores, e guardas principais são destacados.

# 4 Escolha do Polígono

## 4.1 Listagem e Filtragem das Instâncias

As funções `listar_quantidades_vertices`, `filtrar_instancias_por_vertices` e `selecionar_e_carregar_instancia` permitem listar e selecionar polígonos com base na quantidade de vértices desejada.

# 5 Visualização da Solução

## 5.1 Triangulação e Plotagem

A triangulação é executada com visualização passo-a-passo, e os vértices que representam as câmeras são selecionados e exibidos.

# 6 Resultados

## 6.1 Triangulação

O polígono é corretamente triangulado utilizando o método do corte de orelhas. Este método garante que o polígono seja decomposto em triângulos sem sobreposição, utilizando vértices do polígono original.

## 6.2 Coloração dos Vértices

A coloração dos vértices é realizada para minimizar o número de cores utilizadas, permitindo a identificação das posições ideais para as câmeras. A 3-coloração é aplicada inicialmente, e quando há vértices que não podem ser coloridos com três cores, a heurística de 4-coloração é utilizada para garantir que todos os vértices sejam monitorados.

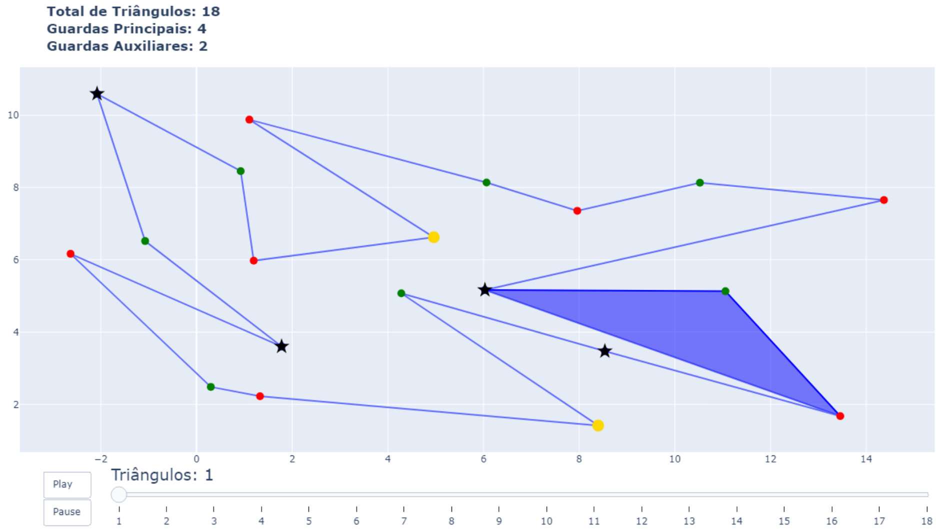
### 6.3 Visualização

A ferramenta implementada permite visualizar o processo de triangulação e coloração dos vértices, auxiliando na compreensão dos conceitos teóricos envolvidos. As visualizações passo-a-passo oferecem uma forma interativa de observar como os triângulos são formados e como as cores são atribuídas aos vértices.

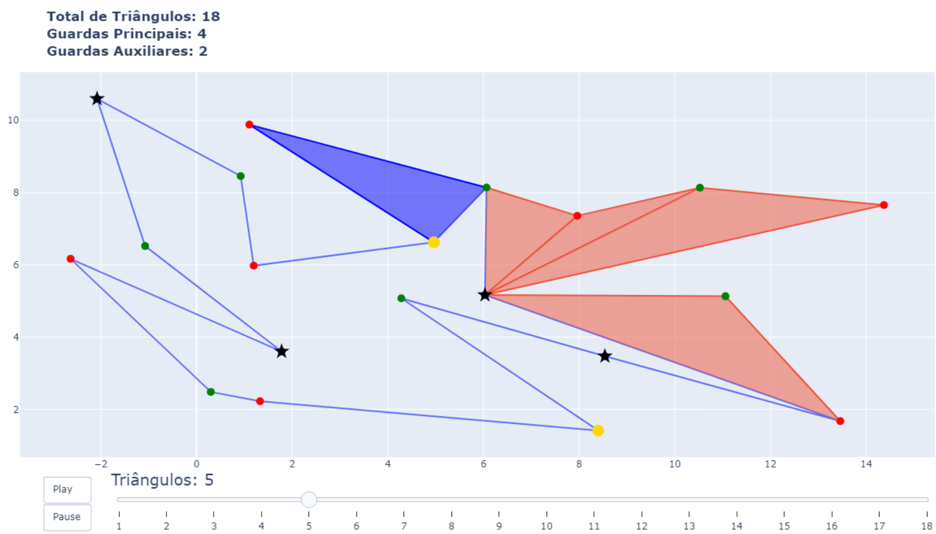
## 7 Considerações Finais

A implementação do algoritmo para o Problema da Galeria de Arte, utilizando o método do corte de orelhas, demonstrou-se eficaz para a triangulação de polígonos e determinação do número mínimo de guardas necessários. A ferramenta desenvolvida facilita a visualização e entendimento do processo, sendo um recurso valioso para o ensino de geometria computacional.

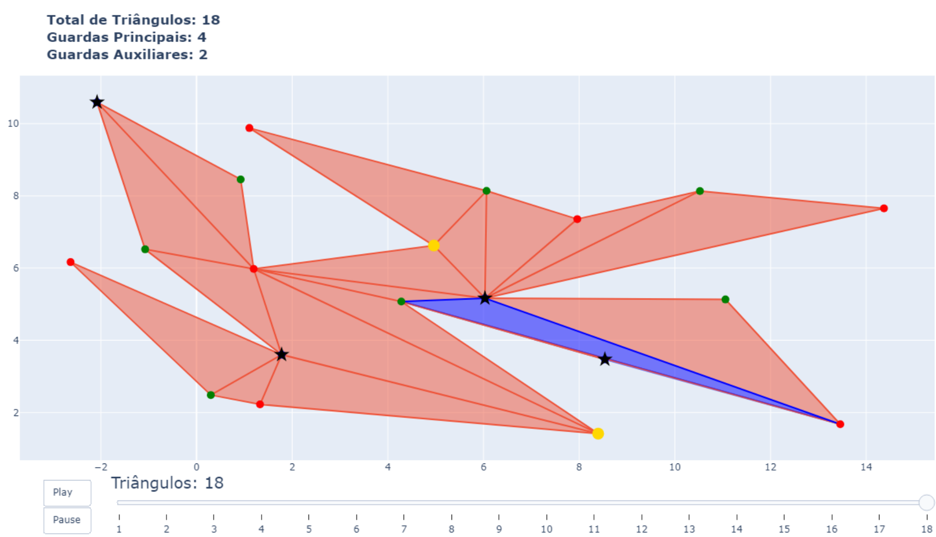
A coloração dos vértices, combinando a simplicidade da 3-coloração com a robustez da heurística de 4-coloração, garante que todas as áreas do polígono sejam monitoradas, mesmo em casos onde a 3-coloração não é suficiente. Embora a heurística de 4-coloração possa não fornecer uma solução ótima, ela assegura que não haja espaços não monitorados, o que é crucial para a aplicação prática do Problema da Galeria de Arte.




(a) Visualização inicial.




(b) Durante a triangulação.




(c) Triangulação completa.

Figura 1: Etapas da triangulação de um polígono.