

Relatório final – MI Prograãmação

Tutor: Thiago Mariano

Vagner José dos Santos

Vagnerjose01@gmail.com

1. Introdução

Uma empresa de segurança chamada Forte Seguro solicitou a implementação de um programa capaz de exibir o melhor caminho para uma rota de coleta e entrega saindo do estacionamento da empresa. O objetivo do software é calcular o menor caminho entre pontos específicos indicados em um grafo que representa um mapa da região de atuação da empresa com os vários caminhos possíveis. O projeto consiste em automatizar a escolha da trajetória.

2. Fundamentação teórica

Para simular os pontos e percursos pelos quais os transportes da Forte Seguro poderiam passar foram utilizados os conceitos de grafos. Em matemática, um grafo é um conjunto composto por dois subconjuntos $V(G)$ e $A(G)$ (Vértices do grafo e Arestas do grafo respectivamente, em que $V(G)$ é um conjunto finito não vazio e $A(G)$ é uma família de pares não ordenados de elementos, não necessariamente distintos de $V(G)$).

Devido a sua complexidade e aplicabilidade, a teoria dos grafos tem sido bastante estudada desde o início de sua elaboração no século XVIII. Desde então essa teoria foi reformulada várias vezes porque das novidades provenientes dos estudos. Graças a esse esforço, foram criados diversos algoritmos para solucionar problemas envolvendo grafos, entre eles os algoritmos de Prim, Floyd, Dijkstra, Kruskal, Boruvka, Ford, entre outros.

Em programação, um grafo é uma estrutura de dados muito importante capaz de representar interações e relacionamentos entre entidades de diversos contextos. Como o desenvolvimento de software é uma área do conhecimento que tem por objetivo desenvolver ferramentas para facilitar tarefas de outras áreas, seus conceitos, tais como grafo, são aplicados de forma a resolver o problema da maneira mais eficiente.

No caso do Forte Seguro, o algoritmo utilizado foi o Algoritmo de Dijkstra, cuja finalidade é resolver o *Problema do Caminho mais Curto* em grafos de arestas com pesos positivos.

Como um dos requisitos do programa, foi solicitado a execução em browsers web, logo, foi necessário implementar a interface gráfica em applet. Applet é uma software que é executado no contexto de um programa maior.

Para representar vértices e arestas na tela do programa foram utilizadas primitivas geométricas disponibilizadas por algumas APIs do java.

3. Metodologia

A fim de solucionar o problema proposto foram feitas reuniões, nas quais os fatos, as ideias, as questões e as metas acerca do desenvolvimento do software foram discutidas. Como a solução se trata da aplicação de um algoritmo pronto, não houve muito o que discutir nesse sentido, deixando como demanda apenas a adaptação ao contexto.

Por padrão, o desenvolvimento foi baseado no modelo MVC (Model View Controller) objetivando maior organização do código fidelidade aos padrões Grasp.

A linguagem de programação usada nesse projeto foi o java.

Por ter sido desenvolvido em dupla, surgiu a ideia de usar uma ferramenta de gerenciamento de projetos chamada Trello, na qual ambos os membros puderam se manter informado sobre o andamento da implementação do seu parceiro. Assim não houve risco de ambos desenvolverem a mesma parte do código, causando uma perda de tempo.

Além do Trello, um repositório de arquivos foi usado para armazenar as versões do software em desenvolvimento. Isso permitiu que houvesse maior eficiência na manipulação do projeto.

4. Resultados e Discussões

Por conta dos requisitos do programa ForteSeguro, é necessário usar o algoritmo de Dijkstra duas vezes para calcular o caminho mais curto entre o estacionamento e o destino, pois esse caminho é a soma do caminho mais curto entre o estacionamento e o ponto de coleta com o caminho mais curto entre o ponto de coleta e o destino.

Para armazenar esse caminho, existe uma classe chamada caminhoPercorrido que possui uma lista com os vértices em sequência por ordem de visita. Essa informação é promovida pela classe Dijkstra.

A interface gráfica do programa também foi desenvolvida, porém devido a um acidente de última hora, algumas classes foram perdidas e não teve como recuperar porque do limite do prazo de entrega.

A implementação por trás da interface gráfica é simples e complexa. Pois exige a manipulação de objetos do tipo Graphics2D Para desenhar o grafo na tela, mas após um período de utilização essa utilização se torna intuitiva.

Uma tela é tratada pelas bibliotecas gráficas do java como um sistema de coordenadas no plano cartesiano, no qual o eixo x é horizontal e o eixo y é vertical. O ponto O (origem) representa o canto superior esquerdo da tela.

Tendo esse conhecimento se torna fácil localizar componentes no painel. Sendo assim, o auxílio de um mouseListener pôde gerenciar o funcionamento do programa através de clicks com o botão direito do mouse que aciona um menu popup que possui as opções de operação do programa.

Algumas ideias para customização do programa surgiram, como por exemplo, colocar uma imagem do mapa de Feira de Santana como background do painel da tela principal. Isso promoveria uma experiência mais interativa ao usuário ao usar o aplicativo.

5. Conclusão

A utilização de grafos para representar situações do dia-a-dia em determinados contextos é uma forma eficiente de resolver problemas, uma vez que permite a obtenção de uma referência bem abstrata da simulação do problema quando comparado com outras estruturas de dados. O algoritmo de Dijkstra, assim como outros algoritmos elaborados para manipulação da teoria dos grafos, pode ajudar a resolver muitos problemas, tanto de campo quanto institucionais. A combinação da teoria dos grafos junto ao paradigma de programação orientada a objetos é uma forte ferramenta de abstração de casos.

Referências Bibliográficas

<http://www.rc.unesp.br/tmelo/diss-polyanna.pdf> -acesso em 14/05/2016

http://www.ime.usp.br/~pf/algoritmos_para_grafos/ -acesso em 14/05/2016