

RELATÓRIO FINAL DE ATIVIDADES

Projeto de Iniciação Científica: Visualizador Dinâmico para Estruturas de Representação de Subdivisões Planares

Orientador: Prof. Dr. Pedro J. de Rezende
Orientado: Flávio Ivan da Silva

Resumo

Este relatório contém uma descrição das atividades realizadas ao longo do último ano de trabalho neste projeto de Iniciação Científica enfocando, para tanto, o cumprimento das atividades previstas no cronograma de execução originalmente estabelecido. Além das atividades realizadas, este relatório contém descrições de aspectos teóricos e práticos referentes à implementação do visualizador da estrutura Half-Edge, que é o produto de software gerado ao fim deste projeto.

Índice:

| | |
|--|----|
| 1. Introdução..... | 2 |
| 2. Cronograma de execução..... | 3 |
| 3. Descrição das atividades realizadas..... | 4 |
| 4. Manipulação e descrição da topologia dos sólidos representados pela estrutura de dados Half-edge..... | 6 |
| 5. Especificação do tipo abstrato de dados Half-edge..... | 11 |
| 6. Descrição do Visualizador de Subdivisões Planares | 16 |
| 7. Testes e exemplo de uso..... | 19 |
| 8. Atividades extras..... | 22 |
| 9. Conclusão..... | 22 |

1. Introdução

Em linhas gerais, o projeto que originou esta Iniciação Científica estabelecia para a mesma os seguintes propósitos: estudo e compreensão da estrutura de dados Half-Edge manipulada pelos operadores de Euler e elaboração de uma ferramenta de software que permitisse a um usuário criar desenhos planares e visualizar, simultaneamente, a construção da estrutura Half-Edge correspondente ao desenho criado.

Este relatório descreve, portanto, o que foi realizado neste projeto ao longo do último ano, isto é, de agosto de 2004 a julho de 2005. A seguir são apresentadas resumidamente as principais atividades deste período.

Durante o segundo semestre do ano de 2004, o orientado cursou um total de 7 disciplinas da graduação, totalizando 26 créditos, e obteve uma média ponderada de 7.8. Simultaneamente, realizou estudos focados nos principais tópicos relacionados ao presente projeto. Nas férias de verão, participou do Programa de Verão do IMPA (Instituto Nacional de Matemática Pura e Aplicada) no Rio de Janeiro, onde cursou a disciplina “Conceitos Básicos de Computação Gráfica”, obtendo menção final ‘A’. Paralelamente, nesse período, foi iniciado o desenvolvimento do Visualizador de Subdivisões Planares, que era o produto de software a ser construído até o final desta iniciação científica.

No primeiro semestre de 2005, o orientado cursou outras 6 disciplinas da graduação, somando novamente 26 créditos, e é esperada a obtenção de uma média ponderada de aproximadamente 9.0. É importante destacar que uma das disciplinas cursadas nesse período é da pós-graduação do Instituto de Computação (oferecida em conjunto com a graduação), intitulada “Introdução a Visão Computacional”, cuja menção final esperada é também ‘A’. Ao mesmo tempo, neste último semestre, foi concluída a implementação do software, conforme planejado.

Ressaltamos também, neste ponto, que fizemos um pedido de renovação de bolsa, cujo projeto trata da elaboração e desenvolvimento de um novo visualizador, aproveitando o conhecimento acumulado neste primeiro ano, e utilizando uma nova modelagem geométrica para visualizar uma outra estrutura de dados: a Quad-Edge.

Conteúdo deste relatório: a próxima seção deste relatório recupera o cronograma de execução estabelecido no projeto original; a seguir, é descrito em detalhes o cumprimento de cada uma das tarefas contidas neste cronograma. As seções que se seguem abordam aspectos teóricos importantes para o correto entendimento do projeto, e também descrevem o software obtido ao término desta Iniciação Científica.

2. Cronograma de execução

| | 2004 | | | | | | 2005 | | | | | | |
|--|------|-----|-----|-----|-----|-----|------|-----|-----|-----|-----|-----|-----|
| | Jul | Ago | Set | Out | Nov | Dez | Jan | Fev | Mar | Abr | Mai | Jun | Jul |
| Conhecimento de Unix e Linux; | ■ | | | | | | | | | | | | |
| Domínio da linguagem C++; | ■ | | | | | | | | | | | | |
| Familiaridade com OpenGL; | | ■ | | | | | | | | | | | |
| Domínio do toolkit Qt; | | ■ | | | | | | | | | | | |
| Noções de modelagem geométrica; | | ■ | ■ | | | | | | | | | | |
| Domínio do uso dos operadores de Euler; | | | ■ | | | | | | | | | | |
| Conhecimento de estruturas de representação de modelagem geométrica; | | | ■ | | | | | | | | | | |
| Projeto do módulo de interface interativa; | | | | ■ | ■ | | | | | | | | |
| Implementação do módulo da estrutura Half-Edge (e testes); | | | | | ■ | | | | | | | | |
| Implementação do módulo de acionamento de operadores de Euler (e testes); | | | | | | ■ | ■ | | | | | | |
| Visualizador da subdivisão planar gerada por operadores de Euler (e testes); | | | | | | | ■ | ■ | | | | | |
| Visualizador da estrutura Half-Edge (e testes); | | | | | | | | ■ | ■ | ■ | | | |
| Integração de todos os módulos; | | | | | | | | | | ■ | ■ | | |
| Testes da versão final integrada; | | | | | | | | | | | ■ | ■ | |
| Relatório Final; | | | | | | | | | | | | ■ | |
| Preparação de apresentação para o Congresso de Iniciação Científica; | | | | | | | | | | | | | ■ |

3. Descrição das atividades realizadas

Inicialmente, foram realizadas atividades de leitura da bibliografia contida no projeto original para aquisição de conhecimentos teóricos sobre Linux, Unix e linguagem de programação C++. Por outro lado, foram desenvolvidos programas de complexidade simples e intermediária usando C++ em ambiente Linux, a fim de proporcionar familiaridade com essa linguagem e esse sistema operacional.

Em seguida, foi feito o tutorial do toolkit Qt. É preciso ressaltar que a realização desse tutorial apresentou-se mais difícil e demorada do que o esperado, mas teve o benefício de gerar o domínio do ambiente Qt, fato esse fundamental para o bom desenvolvimento de várias das atividades restantes para a conclusão do projeto.

Já os estudos da teoria de modelagem geométrica, operadores de Euler e estruturas de representação, atividades seguintes do cronograma, se deram basicamente pelo estudo do livro “Introduction to Solid Modeling”, de Marti Mäntylä. No entanto, foram necessários estudos extras em topologia a fim de melhorar a compreensão sobre o papel da estrutura de representação Half-Edge e dos operadores de Euler, sendo que o resultado destes estudos está descrito na seção **Manipulação e descrição da topologia de sólidos representados pela estrutura de dados Half-edge**.

Por outro lado, era de fundamental importância compreender claramente a estrutura de dados Half-edge antes de implementá-la e, para isso, foi gerada uma especificação que pode ser vista na seção **Especificação do tipo abstrato de dados Half-edge**.

Uma vez compreendido o modo como funcionam tanto os operadores de Euler como a estrutura de representação Half-Edge, restava utilizar o Qt a fim de implementar os componentes do visualizador; a saber: a interface gráfica interativa, contendo uma tela dentro da qual o usuário poderia criar subdivisões planares e outra tela dentro da qual poderia, simultaneamente, visualizar a estrutura Half-edge correspondente às subdivisões criadas; a estrutura de dados Half-Edge que armazena a representação topológica das subdivisões planares criadas pelo usuário; e os operadores de Euler que pudessem ser acionados da interface gráfica e que levassem uma subdivisão planar criada pelo usuário à sua representação dentro da Half-edge.

Assim, as atividades seguintes do cronograma já dizem respeito ao desenvolvimento do visualizador propriamente dito e começam com o projeto do módulo de interface interativa. Após a implementação da interface, e seguindo o cronograma, ocorreram as atividades de implementação dos módulos da estrutura Half-Edge e dos operadores de Euler, concluindo assim o primeiro semestre de trabalho neste projeto de iniciação científica.

No início do segundo semestre de projeto, foi desenvolvido o módulo que permitia visualizar uma estrutura de dados Half-Edge, sendo que o principal desafio neste ponto foi readaptar o projeto de Interface proposto

anteriormente, e utilizar as funcionalidades providas pelo Qt a fim de desenhar a estrutura de dados de modo hierárquico, conforme havíamos planejado, a fim de permitir um maior realismo na sua representação visual. Neste ponto é importante destacar que o curso de computação gráfica, realizado no IMPA e descrito na seção **Atividades extras**, contribuiu bastante para a implementação dos elementos gráficos da tela de interface.

Por outro lado, após concluir o módulo que permitia o desenho da Half-Edge, foi necessário implementar o módulo de visualização das subdivisões planares criadas pelo usuário. A seção **Descrição do Visualizador de Subdivisões Planares** aborda aspectos de interface e de integração de todos os módulos.

Após a integração de todos os módulos, foi possível testar o primeiro protótipo do software produzido, o que permitiu excluir diversos erros não identificados até esse ponto. A seção **Testes e exemplo de uso** esclarece melhor como o visualizador pode ser usado.

Embora o programa já contasse, a essa altura, com as funcionalidades inicialmente especificadas, resolvemos tirar proveito da inversibilidade de cada um dos operadores de Euler já implementados e inserir duas novas funcionalidades que estão relacionadas e que contribuiriam para significativa melhoria do sistema. Trata-se dos botões Undo e Redo, e da gravação de arquivos contendo subdivisões planares criadas pelo usuário, que poderiam, portanto, ser utilizadas novamente mesmo após o fechamento do programa.

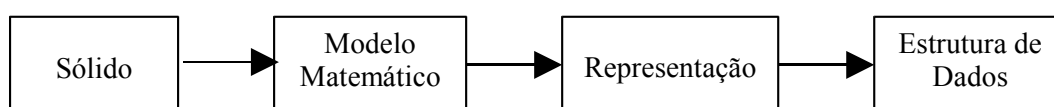
Nos últimos dois meses de trabalho produzimos um novo projeto de iniciação científica para o próximo ano, preparamos o relatório final e a apresentação para o XII Congresso de Iniciação Científica da Unicamp.

4. Manipulação e descrição da topologia de sólidos representados pela estrutura de dados Half-edge

Esta seção descreve como se dá a representação da topologia de sólidos através da estrutura de dados Half-edge, bem como define os operadores necessários à criação e manipulação dessa representação topológica, que são os operadores de Euler.

4.1. Representação de Sólidos

A representação de um sólido para fins computacionais requer um modelo matemático que transforme o sólido em uma representação armazenável em alguma estrutura de dados. Isso é ilustrado pela figura a seguir:



A idéia central neste projeto é criar uma aplicação que permita armazenar e visualizar informações referentes à topologia de subdivisões planares, também chamadas de sólidos planares. Assim, para qualquer sólido planar criado pelo usuário, deve ser usado um modelo matemático que permita obter uma representação, de tamanho finito, de sua topologia para armazená-la numa estrutura de dados. Para alcançar esse objetivo, será utilizado como modelo matemático o modelo de fronteira, cuja representação resultante será armazenada na estrutura de dados Half-edge.

Além disso, é importante observar que a representação de um sólido obtida do modelo de fronteira pode conter informações geométricas associadas a fim de permitir uma visualização da topologia do sólido representado.

4.2. Modelo de fronteira

O modelo de fronteira cria uma representação para a topologia de um sólido através da descrição da sua superfície de contorno como sendo um conjunto de faces poligonais. Cada face poligonal também é descrita por suas curvas de contorno as quais, por sua vez, são descritas por uma cadeia de arestas cujos extremos são dois vértices. Dessa forma, o modelo de fronteira faz uma descrição de todas as faces, arestas e vértices de um sólido, e também uma descrição de como estes elementos estão relacionados entre si no que diz respeito a adjacências.

Duas considerações se fazem necessárias a respeito do modelo de fronteira: a primeira é que as faces em que o modelo divide a superfície externa de um sólido não são necessariamente polígonos de arestas retas, podendo ocorrer que as arestas de contorno destes polígonos sejam curvas; a segunda é que o modelo de fronteira pode armazenar informações sobre curvas que descrevem arestas e coordenadas de vértices de modo a permitir uma visualização (geométrica) da topologia do sólido representado.

4.3. Manipulação da Estrutura de Dados

Na prática, o armazenamento da topologia de um sólido dentro da estrutura de dados é feito de maneira direta conforme mostrado na figura abaixo:



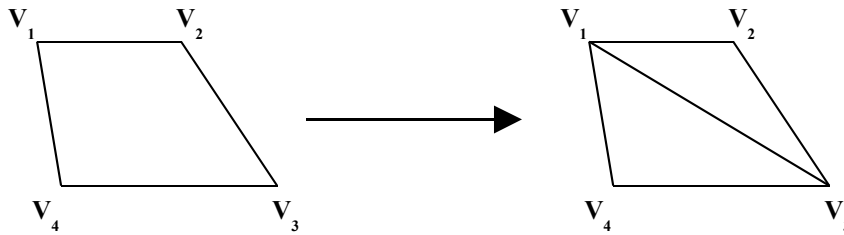
Portanto, é necessário um conjunto de operadores topológicos que criem uma representação para a topologia de um sólido recebido como entrada e a armazenem na estrutura de dados Half-edge. Além disso, esses operadores devem ser gerais o suficiente para permitir o armazenamento da topologia de qualquer sólido.

Assim, para se chegar a um conjunto de operadores topológicos que respeitem essas condições, é necessário descobrir operadores que funcionem de maneira independente e permitam: a criação da topologia de um novo sólido; a alteração desta topologia inserindo ou removendo elementos como arestas e vértices; e, por fim, a remoção da topologia de um sólido de dentro de uma Half-edge.

Isso é obtido, no entanto, dividindo os operadores topológicos buscados em duas categorias distintas: locais e globais. Os operadores locais são aqueles que alteram a topologia de apenas uma parte de um sólido, ao passo que os globais alteram globalmente sua topologia.

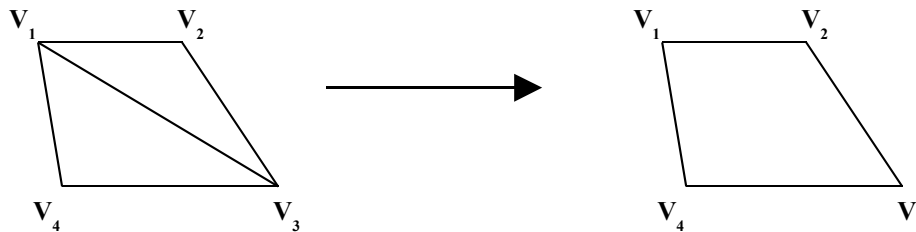
4.3.1. Operadores topológicos locais

Para um polígono cuja fronteira é uma cadeia de arestas ligadas duas a duas por vértices, e cujo número de vértices é maior do que três, pode-se definir a seguinte operação: escolha dois vértices não adjacentes por aresta e crie uma nova aresta com extremos nos vértices escolhidos, formando duas novas cadeias de arestas distintas. A seguir tem-se uma visualização desta operação:



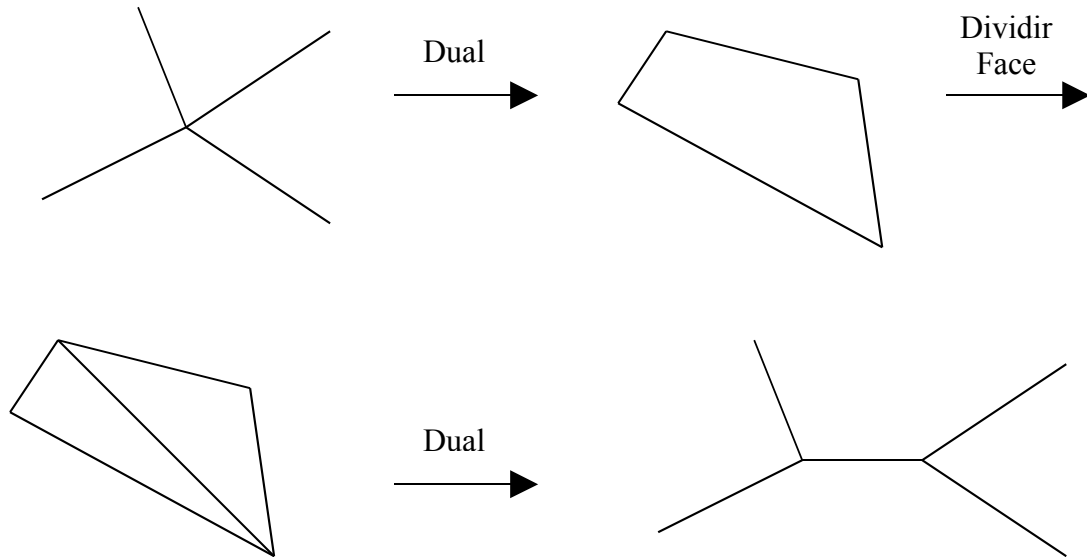
Supondo que este polígono seja a face de um determinado sólido no modelo de fronteira, é facilmente observável que o efeito desta operação é local, resultando apenas na divisão desta face em duas novas faces.

Analogamente, a operação inversa faz a junção de duas faces em uma, conforme mostra a figura abaixo:



Dessa forma, foram obtidos dois operadores topológicos locais e, na busca por outros operadores será introduzido o conceito de modelo de fronteira dual. Modelo dual é aquele obtido a partir da conversão dos vértices do modelo original em faces do modelo dual, sendo que as arestas incidentes em um vértice do modelo original são convertidas em arestas de contorno da face correspondente do modelo dual.

Assim, podemos considerar o efeito da operação de dividir faces, descrita acima, quando aplicada no modelo dual, o que pode ser visualizado pela figura que se segue:

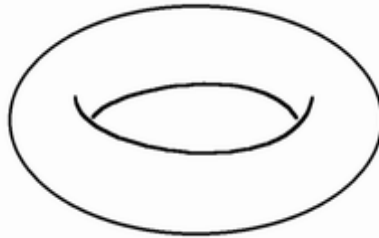


A figura acima permite concluir que a operação de dividir uma face em duas quando aplicada no modelo dual tem o efeito de dividir um vértice em dois, separados por uma nova aresta. Além disso, sua inversa também pode ser definida como tendo o efeito de eliminar uma aresta e um vértice quando aplicada sobre dois vértices adjacentes por uma aresta.

Essas quatro operações são suficientes para obter, a partir de um modelo com apenas um vértice e uma face, qualquer modelo com topologia equivalente à topologia de uma esfera, isto é, topologia de sólidos que, como a esfera, não têm furos.

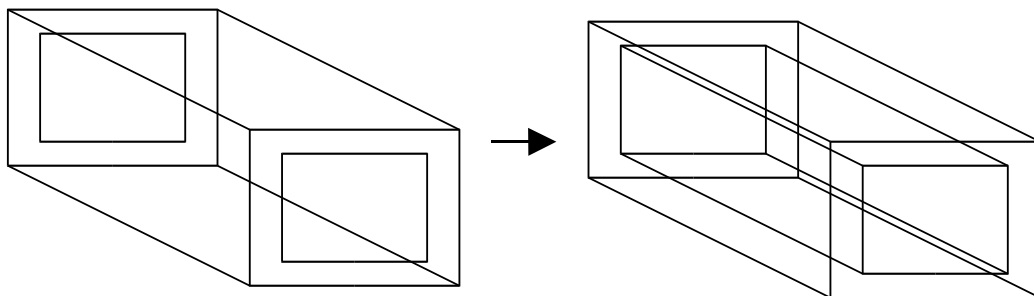
4.3.2 Operadores topológicos globais:

Além dos operadores topológicos locais, são necessários operadores que permitam alterar propriedades topológicas globais de um modelo de fronteira, isto é, que permitam ao modelo de fronteira representar sólidos topologicamente equivalentes não somente à esfera, que não tem furos, mas também ao toro (veja figura abaixo), ao bi-toro, etc.



toro

Dessa forma, vamos introduzir um operador topológico global que faça a união de duas faces poligonais da superfície de um sólido de modo a gerar um furo através da superfície (alteração de gênero). O efeito da aplicação desse operador global é mostrado na figura abaixo:



Analogamente, pode ser definido o operador inverso que faça a remoção do furo (reduzindo o gênero), gerando duas novas faces na superfície externa do sólido.

Os seguintes operadores sobre a estrutura Half-edge são suficientes para representar a topologia de qualquer sólido: um operador que crie uma nova representação de um sólido contendo um vértice e uma face externa, sem aresta; outro que remova um sólido contendo apenas um vértice e uma face externa; os quatro operadores topológicos locais e os dois operadores topológicos globais descritos acima. Estes operadores em conjunto formam os operadores de Euler, e serão especificados na seção que se segue.

5. Especificação do tipo abstrato de dados Half-edge

5.1. Elementos: o tipo abstrato de dados Half-edge contém registros de cinco tipos distintos: solid, face, loop, halfedge e vertex, hierarquizados nessa ordem.

5.2. Estrutura: o tipo abstrato de dados Half-edge é uma estrutura hierárquica heterogênea, na qual existem cinco níveis distintos, cada um deles formado por uma lista duplamente ligada de registros de um dos seguintes tipos: solid, face, loop, halfedge e vertex. A descrição desses níveis é dada a seguir:

Nível 1: contém uma lista duplamente ligada de registros do tipo solid. Cada registro do tipo solid contém dois apontadores para registros do tipo solid e um apontador para um registro do Nível 2.

Nível 2: contém uma lista duplamente ligada de registros do tipo face. Cada registro do tipo face contém um apontador para um registro do Nível 1, dois apontadores para registros do tipo face e um apontador para um registro do Nível 3.

Nível 3: contém uma lista duplamente ligada de registros do tipo loop. Cada registro do tipo loop contém um apontador para um registro do Nível 2, dois apontadores para registros do tipo loop e um apontador para um registro do Nível 4.

Nível 4: contém uma lista duplamente ligada de registros do tipo halfedge. Cada registro do tipo halfedge contém um apontador para um registro do Nível 3, dois apontadores para registros do tipo halfedge e um apontador para um registro do Nível 5.

Nível 5: contém uma lista duplamente ligada de registros do tipo vertex. Cada registro do tipo vertex contém um apontador para um registro do Nível 4 e dois apontadores para registros do tipo vertex.

O tipo abstrato de dados Half-edge, como descrito acima, permite armazenar informações sobre a topologia de um conjunto de sólidos. O modo como sua estrutura hierárquica é utilizada para armazenar essas informações está descrito a seguir:

Nível 1 – lista duplamente ligada de sólidos: cada sólido tem apontadores para os sólidos anterior e posterior, e um apontador para uma de suas faces no Nível 2.

Nível 2 – lista duplamente ligada de faces: cada face tem um apontador para o sólido do Nível 1 ao qual esta face pertence, apontadores para as faces anterior e posterior, e apontador para um de seus loops no Nível 3.

Nível 3 – lista duplamente ligada de loops: cada loop tem um apontador para a face do Nível 2 à qual este loop pertence, apontadores para os loops anterior e posterior, e apontador para uma de suas halfedges no Nível 4.

Nível 4 – lista duplamente ligada de halfedges: cada halfedge tem um apontador para o loop do Nível 3 ao qual esta halfedge pertence, dois apontadores para as halfedges anterior e posterior e um apontador para o vértice inicial dessa halfedge no Nível 5.

Nível 5 – lista duplamente ligada de vértices: cada vértice tem um apontador para a halfedge do Nível 4 ao qual este vértice pertence, e apontadores para os vértices anterior e posterior.

Observação: para se ter uma visualização geométrica da estrutura topológica representada pelo tipo abstrato de dados Half-edge, é necessário agregar coordenadas a todos os vértices da estrutura, e traçar segmentos de reta (ou de curvas) ligando cada dois vértices adjacentes.

5.3. Operações: as operações sobre o tipo abstrato de dados Half-edge são equivalentes aos operadores de Euler.

Podemos abreviar os operadores de Euler como:

- MVFS (Make Vertex Face Solid)
- KVFS (Kill Vertex Face Solid)
- MEV (Make Edge Vertex)
- KEV (Kill Edge Vertex)
- MEF (Make Edge Face)
- KEF (Kill Edge Face)
- MEKR (Make Edge Kill Ring)
- KEMR (Kill Edge Make Ring)
- MFKRH (Make Face Kill Ring Hole)
- KFMRH (Kill Face Make Ring Hole).

A descrição das **pré**-condições e **pós**-condições para cada um dos dez operadores de Euler é dada a seguir:

MVFS:

Pré – especificação das coordenadas de um vértice.

Pós – passa a existir um novo sólido, que contém apenas uma face, um loop vazio e um vértice cujas coordenadas são as especificadas na pré-condição.

KVFS:

Pré – especificação de um sólido composto de um vértice, um loop vazio e uma face.

Pós – o sólido especificado deixa de existir.

MEV:

Pré – especificação de um vértice v e das coordenadas de um novo vértice w .

Pós – a estrutura Half-edge contém um novo vértice w e uma aresta entre v e w .

KEV:

Pré – especificação de dois vértices v e w que estão ligados por uma aresta, sendo que w contém apenas uma aresta (que o liga a v).

Pós – a estrutura Half-edge não contém o vértice w e nem a aresta que unia v a w .

MEF:

Pré – especificação de dois vértices v e w que pertencem a um mesmo loop, sendo que deve existir um caminho C entre eles.

Pós – passa a existir uma aresta e' ligando v a w , e uma nova face formada pelo fechamento do caminho C através da aresta e' .

KEF:

Pré – especificação de dois vértices ligados por uma aresta e' que pertençam a um loop fronteiroço de uma face f .

Pós – a aresta e' e a face f deixam de existir.

MEKR:

Pré – especificação de dois vértices pertencentes a loops distintos l_1 e l_2 de uma mesma face.

Pós – passa a existir uma aresta ligando os dois vértices especificados e o loop l_2 deixa de existir.

KEMR:

Pré – especificação de dois vértices v e w adjacentes e pertencentes a um mesmo loop, sendo que existe um caminho C_1 que sai de v e retorna a v sem passar por w e outro caminho C_2 que sai de w e retorna a w sem passar por v .

Pós – a aresta que ligava v a w e o loop a que eles pertenciam deixam de existir, e dois novos loops passam a existir: um deles contém v e os vértices do caminho C_1 e o outro contém w e os vértices do caminho C_2 .

MFKRH:

Pré – especificação de uma aresta a_1 que ligue os loops externos l_1 e l_2 de um furo, sendo que a face do furo é determinada pelo loop l_1, a_1, l_2 .

Pós – a face do furo, cujo loop de contorno é l_1, a_1, l_2 , e o loop l_1, a_1, l_2 deixam de existir, e surgem duas novas faces f_1 e f_2 cujos contornos externos são os loops l_1 e l_2 .

KFMRH:

Pré – especificação de dois vértices x e y pertencentes aos loops l_1 e l_2 externos às faces distintas f_1 e f_2 .

Pós – as faces f_1, f_2 e os loops l_1, l_2 deixam de existir, e surgem: uma aresta a_1 unindo os vértices x e y , um loop composto de l_1, a_1 e l_2 , e uma face de um furo cujo contorno externo é o loop l_1, a_1, l_2 .

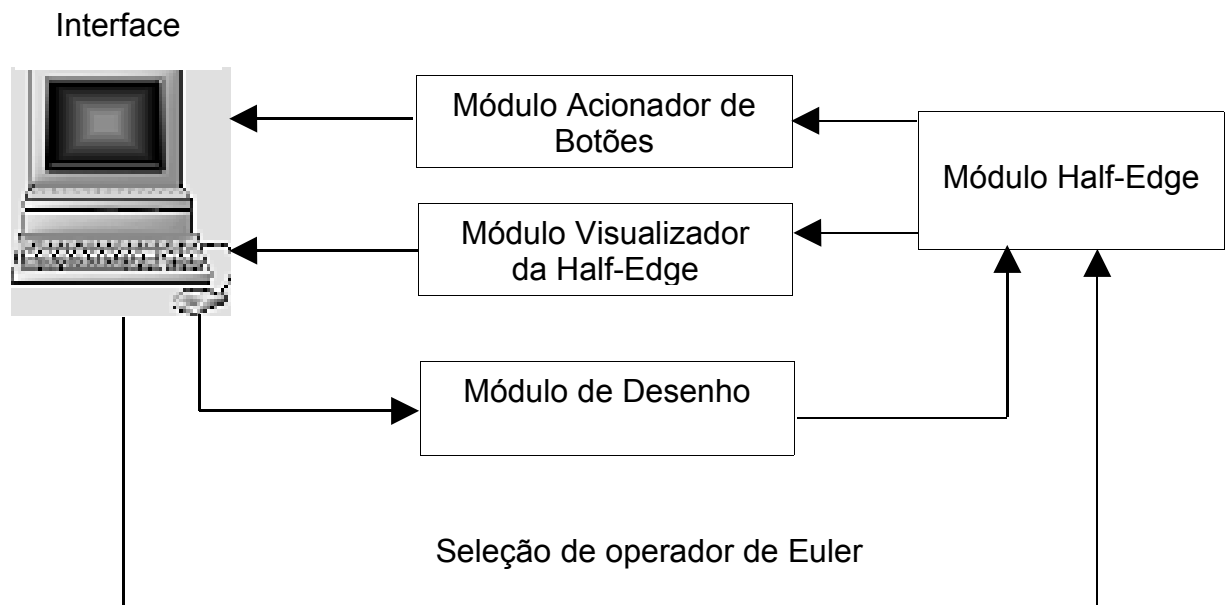
6. Descrição do Visualizador de Subdivisões Planares

O projeto original desta Iniciação Científica estabelecia como objetivo, além do estudo e compreensão da estrutura Half-Edge e dos Operadores de Euler, a implementação de um software que permitisse a um usuário criar desenhos planares em uma tela e visualizar, simultaneamente, a estrutura Half-Edge correspondente ao desenho feito. Além disso, o usuário deveria ter possibilidade de percorrer a estrutura de dados mostrada, o que significaria percorrer também o desenho criado.

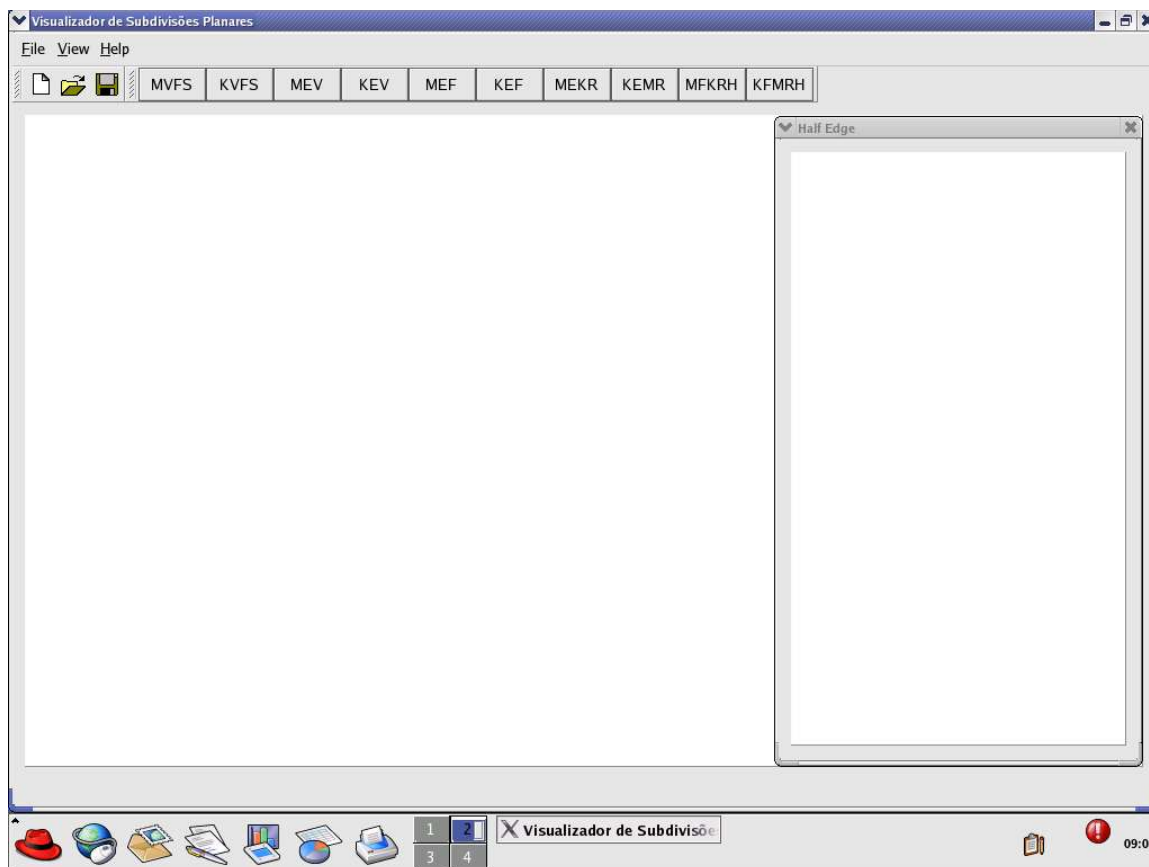
Porém, além desta especificação de software, ainda foi estabelecido que o mesmo deveria ser produzido usando linguagem de programação C++ e ambiente de desenvolvimento Qt. Essa implementação, por sua vez, foi dividida em módulos de modo que tornasse mais organizado o processo de implementação.

Os principais módulos estabelecidos, portanto, foram: interface gráfica interativa, estrutura de dados Half-Edge, módulo de desenho, módulo visualizador da Half-Edge e módulo acionador de botões.

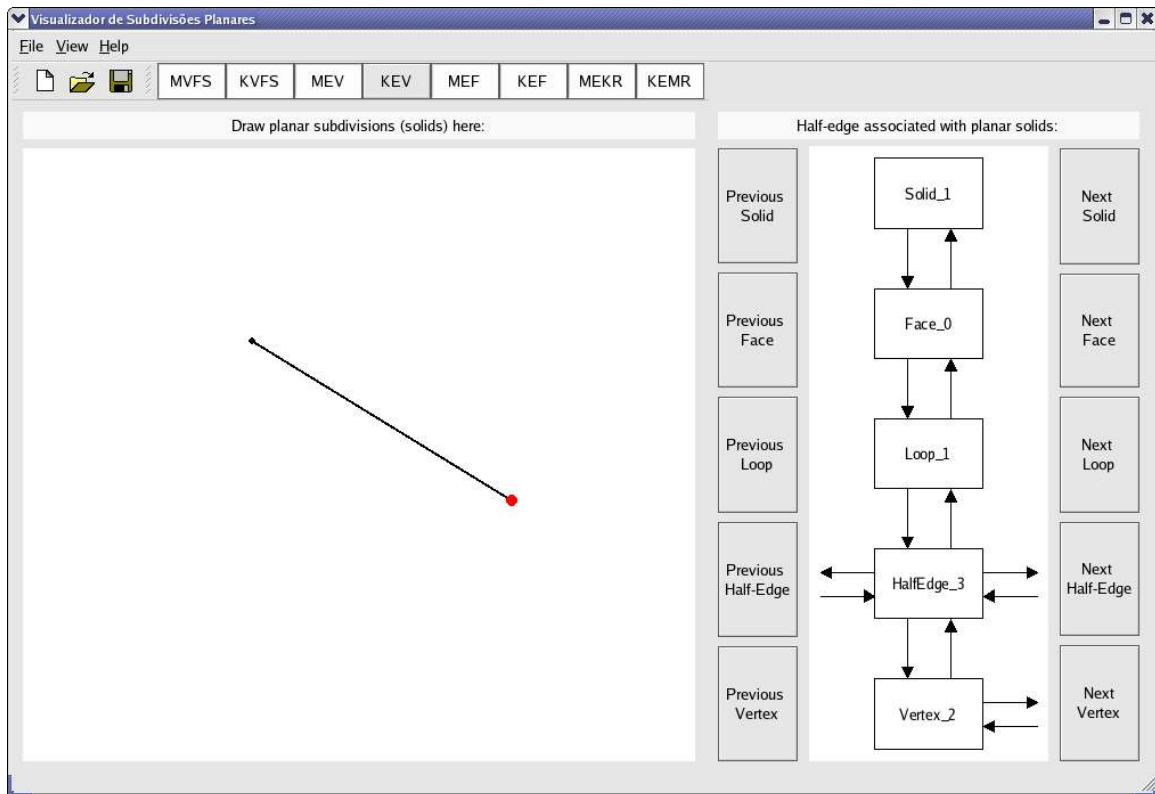
Conforme descrito no item **Descrição das atividades realizadas**, os módulos citados acima foram sendo construídos um a um, até que chegou o momento de integrá-los no software final. Isto posto, é importante mostrar como se dá a interligação entre os módulos, o que pode ser visto na figura abaixo:



Merece destaque aqui uma descrição de como foi desenvolvida a interface gráfica interativa. Isto porque a idéia central era desenvolver uma interface o mais simples e intuitiva possível para um usuário com noções básicas de informática e de sistemas operacionais baseados em janelas. Com isto em mente, ao fim do primeiro semestre de trabalho tínhamos produzido o protótipo da interface que é mostrado abaixo:



No entanto, após concluir a implementação da estrutura Half-Edge e do módulo que a desenhava, sentimos a necessidade de incluir botões na tela que permitissem ao usuário navegar pela estrutura mostrada. Por outro lado, para implementar o módulo de desenhos planares pelo usuário, novas alterações pareceram razoáveis, de modo a integrar harmonicamente todas as funcionalidades na tela principal do programa. Dessa forma, após a conclusão do projeto acabamos por obter uma nova tela de interface, contendo significativas mudanças em relação à tela inicialmente projetada. A seguir, é mostrada uma imagem da interface final do sistema.



Da figura acima é possível observar os três módulos que interagem com a interface gráfica do sistema: à direita o módulo de desenho no qual o usuário pode inserir novos elementos com a ajuda do mouse; à direita o módulo que mostra a estrutura de dados Half-Edge na configuração atual, sendo que o usuário pode usar os botões à esquerda e à direita da representação para navegar por seus elementos, sempre que possível; por fim, na parte superior o módulo que aciona os operadores de Euler.

Em relação a este último módulo, o usuário só poderá acionar o operador de Euler cujo uso faz sentido em um dado instante (i.é., cujas pré-condições estão satisfeitas). Como exemplo, a figura acima possui o operador KEV (Kill Edge Vertex) habilitado nesta tela, uma vez que a única opção viável com esta configuração é desfazer a aresta contida na tela de desenho.

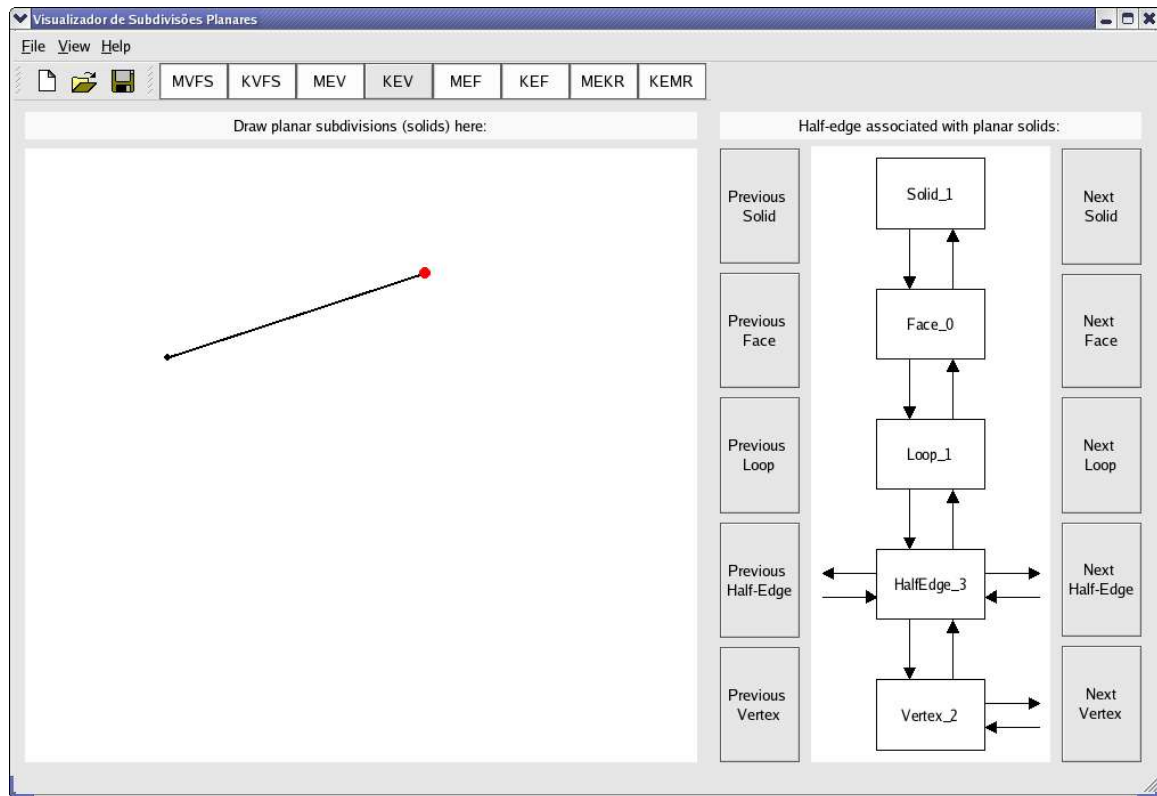
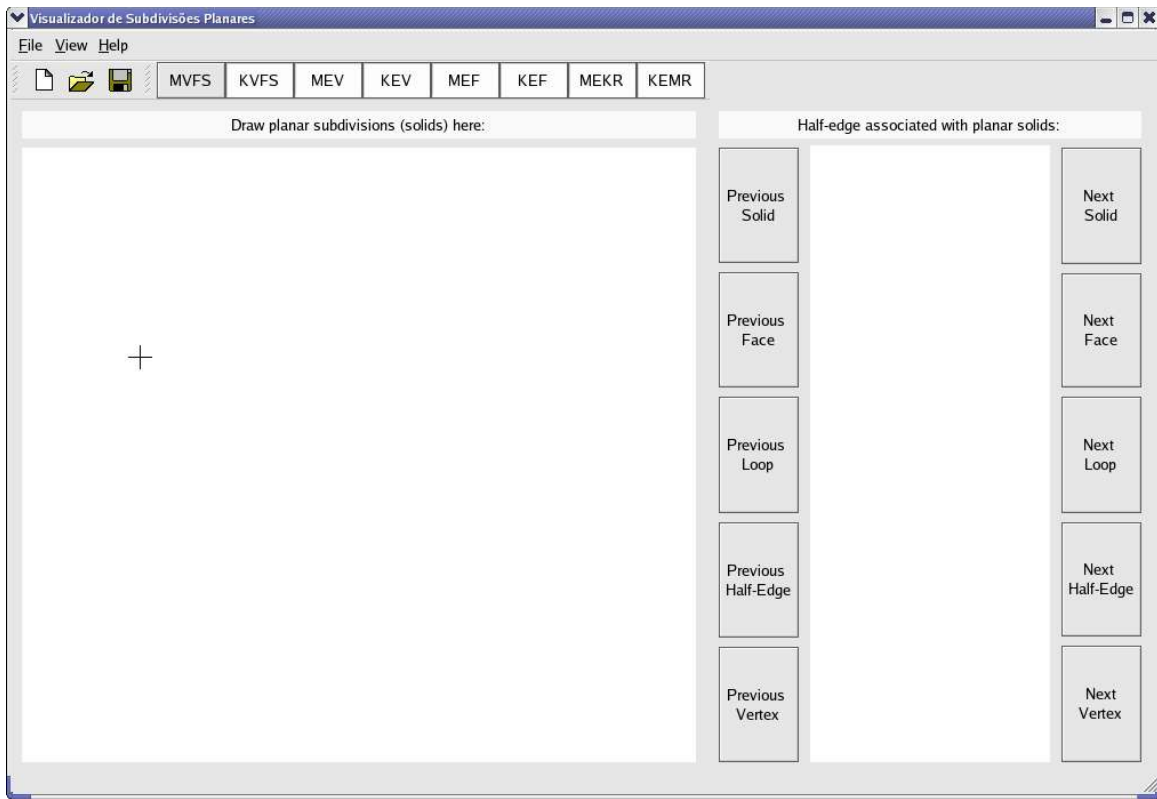
7. Testes e exemplo de uso

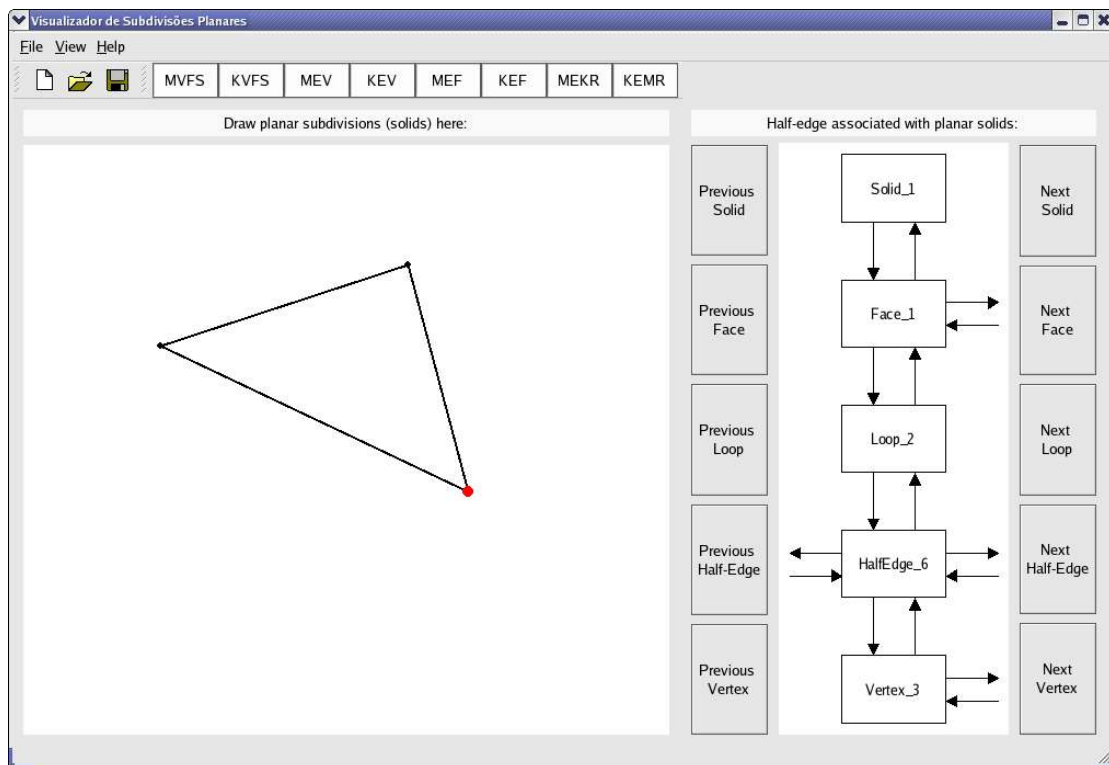
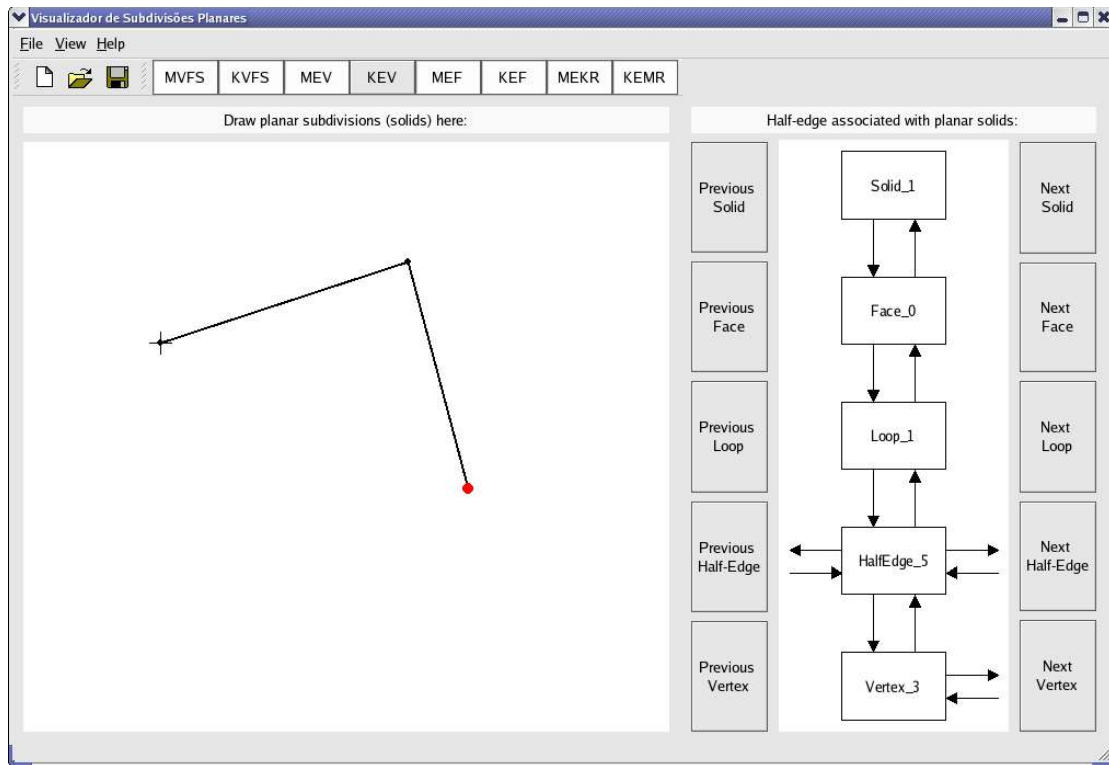
Alguns alunos do Instituto de Computação foram convidados a testar o sistema. Em geral, os mesmos não tiveram problemas com o uso, embora a falta de familiaridade com os assuntos abordados possa ter tornado o uso difícil a um usuário iniciante.

A seguir, mostramos um exemplo simples de uso do sistema, no qual criaremos o desenho de um triângulo, visualizando a estrutura Half-Edge ao longo do processo. Esse exemplo é constituído de quatro passos:

1. A tarefa inicial na criação de qualquer desenho planar consiste em inserir o primeiro vértice do novo desenho. Isto é feito através o operador MVFS (Make Vertex Face Solid).
2. Tendo iniciado o desenho, para criar uma nova aresta basta selecionar uma posição da tela e utilizar o operador MEV (Make Edge Vertex).
3. Para inserir outra aresta, basta repetir o passo 2 para um nova posição da tela de desenho.
4. Neste ponto o desenho contém três vértices e duas arestas. Logo, podemos selecionar os dois vértices das extremidades e utilizar o operador MEF (Make Edge Face) para obter uma face.

A seguir, são reproduzidas as telas de interface resultantes da execução destes quatro passos.





É possível observar das telas mostradas acima que a interação com o sistema é bastante simples. O desenho do triângulo, neste exemplo, foi obtido a partir de sucessivas escolhas de posições da tela, com o uso do mouse, e posterior utilização de um dos operadores de Euler cujo uso fazia sentido no instante dado.

8. Atividades extras

Conforme citado anteriormente, uma das atividades realizadas ao longo do último ano foi a participação no Programa de Verão 2005 do IMPA (Instituto de Matemática Pura e Aplicada), no Rio de Janeiro, onde foi cursada a disciplina “Conceitos Básicos Computação Gráfica”.

Os dois meses de duração deste curso foram importantes tanto para início das atividades de implementação do visualizador, como para aquisição de conhecimentos teóricos e práticos em Computação Gráfica, o que auxiliou bastante na posterior implementação das partes gráficas da interface interativa. Vale destacar também o bom rendimento da disciplina cursada, cuja menção final obtida foi ‘A’.

Outra importante atividade extra se deu no primeiro semestre de 2005, quando o orientado cursou uma disciplina de pós-graduação no Instituto de Computação da Unicamp. Embora a disciplina cursada “Introdução à Visão Computacional” não seja tema desta Iniciação Científica, a mesma envolve conceitos diretamente relacionados à Computação Gráfica e indiretamente a assuntos abordados neste projeto. Além disso, esta disciplina poderá ser validada quando o orientado ingressar no mestrado, ao término da graduação.

9. Conclusão

Ao término deste projeto de iniciação científica, alguns pontos merecem destaque. O primeiro é que tivemos êxito na realização das tarefas inicialmente propostas, cumprindo o prazo estabelecido para a execução das mesmas. O segundo ponto de destaque é a quantidade considerável de novos conhecimentos adquiridos pelo orientado ao término deste projeto, sendo que a maior parte destes não são abordados nas disciplinas de graduação em Engenharia de Computação. Em terceiro lugar, é importante destacar a qualidade dos resultados obtidos, sobretudo do Visualizador de Subdivisões Planares, o qual esperamos disponibilizar, com fins didáticos, para uso da comunidade acadêmica.

A versão final do software produzido será disponibilizada no seguinte endereço web:

<http://www.ic.unicamp.br/~rezende/SubdiViewer>

Consideramos, portanto, que o projeto foi concluído com sucesso, o que nos assegura que, concedida a renovação da bolsa pleiteada para um segundo ano de trabalho de iniciação científica, teremos sucesso na continuação desse projeto para uma nova estrutura de representação de subdivisões planares.