|  |
| --- |
| **Pong 3D** |
| Trabalho referente à disciplina Computação Gráfica — SCC 650, ministrada pela profa. dr. Maria Cristina F. de Oliveira e sob auxílio do estágiario PAE Erick Gómez Nieto. |
|  |
| Descrição de projeto e desenvolvimento da aplicação, envolvendo fortemente conceitos de computação gráfica. |
|  |
| **Alunos: Rodrigo Almeida Bergamo Ferrari - 8006421 Adriano Belfort de Sousa - 7960706 Ilan Sales Galvão de Figueiredo - 7656321** |
| **17 de abr.** |
|  |





1. **Introdução**

Programas e aplicações contemporâneas estão repletas de conceitos de computação gráfica. Dentre as atividades mais notáveis, destacam-se a síntese de imagens, aplicações que envolvem processamento de imagens, visualização computacional, entre diversas outras. Nesse projeto, serão trabalhados conceitos de modelagem 3D, *rendering*, *viewing* e, também, bem associado com os conceitos teóricos usados em transformações gráficas, o desenvolvimento e uso de uma *engine* física.

Essa aplicação consiste em um pong 3D, tendo como alguns de seus principais aspectos:

* Elaborar um jogo de pong, envolvendo interações e regras desse jogo;
* Um espaço retangular delimitado por paredes brancas, indicando qual a região que a bola pode trafegar;
* Pequenos planos controlados pelos jogadores que serão usados para evitar que a bola atravesse o "gol" de cada um dos lados;
* Uma bola que ficará em movimento durante todo o jogo.

Uma *Engine* Física estará constantemente envolvida nos eventos da aplicação, dado que esses eventos devem ocorrer de forma contínua e são basicamente a simulação de eventos físicos que deveriam acontecer na realidade. Ainda nesse contexto, os parâmetros da engine física podem ser mudados para, por exemplo, possibilitar a elaboração de "modos alternativos de simulação", por exemplo, com uma gravidade, resistência do ar acrescentadas ao jogo, alteração dos parâmetros de perda/ganho de energia nas colisões etc..

Adicionalmente, ao menos as etapas de planejamento e projeto irão envolver conceitos de redes de computadores, ao buscar-se desenvolver um jogo multiplayer. O jogo principal irá operar apenas no modo multiplayer, implementado via internet e com um servidor próprio para gerenciamento dos jogadores.

Duas justificativas para o desenvolvimento do jogo multiplayer são:

1. Não será necessário implementar uma inteligência artificial para executar o jogo no modo *1-player*. O jogo irá executar integralmente com 2 jogadores.

2. Será concedida aos usuários a deleitável[[1]](#footnote-1) experiência de um jogo multiplayer online, podendo ser um momento compartilhado com amigos geograficamente próximos ou distantes.

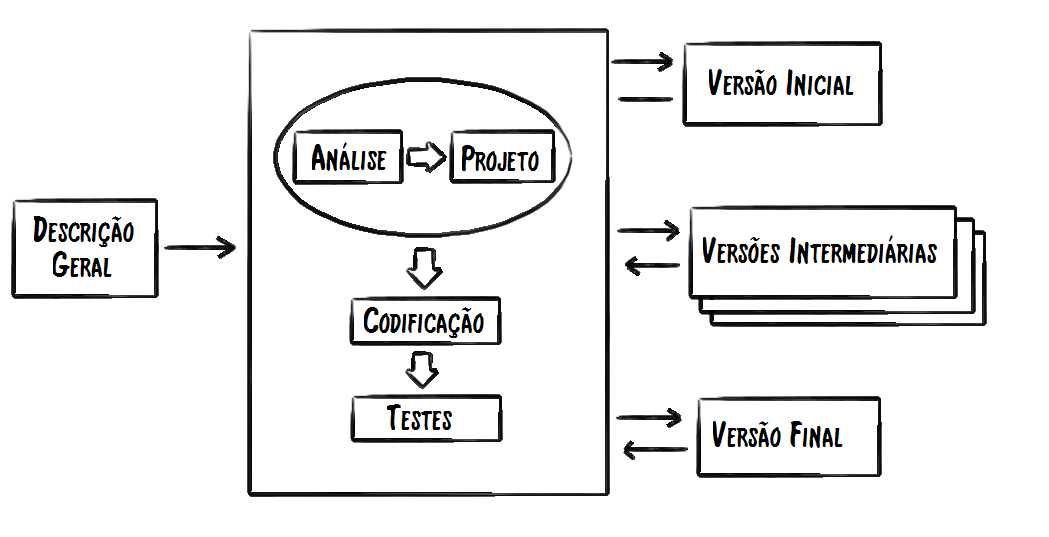
1. **Objetivos**

* **Objetivos Gerais**
* Projetar e desenvolver um jogo de Pong 3D simples com suporte para modo de jogo multiplayer online via web.
* **Objetivos Específicos**
* Aplicar conceitos de computação gráfica para criar um mundo composto por um volume retangular e simular a interação de planos paralelos e uma bola durante uma partida de pong - deve haver um ganho de energia nas colisões crescente ao longo do tempo.
* Estender a aplicação desenvolvida para que ela possua a funcionalidade multiplayer online via web.

1. **Procedimento de Desenvolvimento**

**1. Modelo de Desenvolvimento**

Será usado o modelo de desenvolvimento de software incremental para desenvolver a aplicação. Essa abordagem será utilizada principalmente porque as técnicas de implementação de aplicações gráficas ainda são desconhecidas para os desenvolvedores, mas também para permitir que novos requisitos sejam acrescentados à aplicação enquanto ela é desenvolvida.

 Com o auxílio do esquema apresentado na figura 1, pode-se entender com maior clareza a utilidade desse modelo para a aplicação a ser desenvolvida:

A descrição geral consiste do conjunto de requisitos básicos do sistema; As etapas de análise e projeto serão caracterizadas principalmente pela análise de novos requisitos e descoberta de novas técnicas de implementação necessárias para a implementação; As etapas de codificação e testes são essenciais para o desenvolvimento da aplicação — de fato, a etapa de implementação que evolui o desenvolvimento da aplicação.

Figura 1: Modelo de Desenvolvimento Incremental

Serão definidos três grandes ciclos de desenvolvimento a partir do dia 10/05, com duração de 2 semanas cada. Cada um desses ciclos deverá possuir um conjunto de objetivos específicos bem definido, e ao final do ciclo, serão definidos os objetivos e requisitos para o ciclo seguinte. Dentro de cada um dos ciclos, serão divididas tarefas entre os desenvolvedores (alunos) de forma que essas tarefas possuam um acoplamento mínimo.

**2. Descrição do Formato de Apresentação dos Resultados ao Final de um Ciclo**

Um conjunto de dados estará disponível ao final de cada um dos ciclos de desenvolvimento e antes do início do primeiro ciclo. A saber:

* Conjunto de requisitos já cumpridos;
* Versão da aplicação;
* Conjunto de requisitos a serem cumpridos no ciclo seguinte;
* Mapeamento de subconjuntos de requisitos dependentes em tarefas;
* Atribuição das tarefas a desenvolvedores.

Em cada um dos instantes descritos, os itens acima serão documentados e anexados ao documento final de projeto da aplicação na seção IV (Resultados Obtidos e Discussão). A primeira realização dos itens acima deve ser feita antes do início do primeiro ciclo, exceto o item "Conjunto de requisitos já cumpridos".

**3. Descrição Geral**

A aplicação deverá modelar um pong, composto por um delimitador de volume perimetral, uma bola e dois paralelepípedos controlados por dois jogadores distintos. As interações físicas modeladas serão semelhantes às interações físicas do mundo real — movimentação e colisões.

Associado à *engine* física, colisões entre dois objetos vão depender de um conjunto de atributos, sendo eles: Coeficientes de elasticidade dos materiais dos corpos envolvidos na colisão, a massa e as velocidades dos corpos.

Associado à dinâmica geral da partida, o jogo deverá manter contabilizadas as pontuações dos jogadores. O jogador poderá escolher uma dentre várias posições possíveis para a posição da câmera e deverá existir um modo de pausa, o qual permitirá que o jogador tome completo controle da câmera, alterando posição, *viewpoint* e vetor *viewup* da câmera como quiser.

Por fim, será implementado um servidor que disponibilize um serviço de conexão entre jogadores que desejam jogar o jogo um contra o outro. O servidor irá fornecer os endereços dos jogadores um para o outro, de forma que eles possam estabelecer uma conexão para jogar o jogo um contra o outro. A aplicação irá implementar um protocolo de comunicação que permita essa funcionalidade.

1. **Resultados Obtidos e Discussão**

**1. Dados Levantados antes do Primeiro Ciclo de Desenvolvimento**

**1.1. Requisitos**

* **Requisitos de Modelagem de Objetos do Mundo**

**1.1.1.** A aplicação deve modelar um mundo 3D composto por um volume delimitado perimetralmente e em uma das extremidades por um material branco translúcido. O restante do mundo deve ser de uma coloração uniforme que o distingua do material de delimitação, preferencialmente preta.

**1.1.2.** A aplicação deve modelar uma bola de pong com dimensões aceitáveis.

**1.1.3.** A aplicação deve modelar um plano que ficará na extremidade aberta do volume.

**1.1.4.** A aplicação deve permitir que o plano se movimente controlado pelo usuário, com dois graus de liberdade, mantendo-se sempre perpendiculares ao volume delimitador.

* **Requisitos de Viewing**

**1.1.5.** A aplicação deve possuir uma viewport estática que veja todo o volume delimitador, com a câmera localizada exatamente ao centro do volume, mas em uma posição superior no eixo vertical.

**1.1.6.** A aplicação deve possuir duas viewports estáticas que vejam o volume de posições laterais, com as câmeras localizadas um pouco acima do volume no eixo vertical.

**1.1.7.** A aplicação deve possuir uma viewport estática que veja o volume de uma posição com o plano controlado pelo usuário entre a câmera e o volume.

**1.1.8.** A aplicação deve possuir um modo o qual a posição, o *viewpoint* e o vetor *viewup* possam ser alterados.

* **Requisitos da *Engine* Física**

**1.1.9.** A aplicação deve modelar o movimento da bola no interior do volume, permitindo um modo o qual o usuário possa movimentar a bola no interior do volume.

**1.1.10.** A aplicação deve modelar a colisão da bola com as paredes do volume. Se o modo de operação for como descrito no requisito 1.1.11, a bola deve simplesmente parar e não atravessar o delimitador ou o plano.

**1.1.11.** A aplicação deve modelar a colisão da bola com o plano, levando em consideração o ângulo de impacto da bola com a raquete, e as velocidades[[2]](#footnote-2) da bola e do plano no instante da colisão.

* **Requisitos de Menus e de Jogabilidade**

**1.1.12.** A aplicação deve possuir um modo de jogo normal, consistindo do volume definido no modo de jogo normal e cuja bola começa ao centro do volume, com alguma velocidade inicial dada **V0 .** Esse modo de jogo normal será apenas uma espécie de *training mode*, nas etapas posteriores do projeto.

**1.1.13.** A aplicação deve possuir um menu (menu 0 - futuras referências) que possua as opções:

* One Player;
* Multiplayer;
* Instructions;
* Options;
* Credits;
* Exit;

**1.1.14.** Ao ser selecionada a opção *Exit* no menu 0, a aplicação deve encerrar.

**1.1.15.** Ao ser selecionada a opção *Credits* no menu 0, a aplicação deve redirecionar a tela para outra tela com a opção *Back*, que ao ser selecionada, faz a tela para retornar o menu 0; e com os créditos do trabalho (autores, profa., estagiário PAE, agradecimentos, eop.).

**1.1.16.** Ao ser selecionada a opção *Instructions* no menu 0, a aplicação deve redirecionar a tela para outra tela contendo as instruções do jogo (controles do plano e descrição geral de como o jogo é executado). Ele deve possuir um botão *Back to Initial Menu*, que faz retornar para o menu 0 se selecionado. Se necessário, poderá existir o botão *Next Page* para redirecionar a tela para outra página contendo mais instruções. Nas páginas seguintes, poderão existir também as opções *Next Page* e *Previous Page*.

**1.1.17.** Ao ser selecionada a opção *One Player* no menu inicial, a aplicação deve redirecionar a tela para um menu secundário (menu OP), composto das seguintes opções:

* New Game;
* Training Mode;
* Back;

**1.1.18.** Ao ser selecionada a opção Back no menu OPT, a aplicação deve redirecionar a tela para o menu 0.

**1.1.19.** Ao ser selecionada a opção *Training Mode* no menu OP, a aplicação deve levar o jogador para uma viewport como descrita no requisito 1.1.7. e com a bola no centro do volume. Após uma mensagem "START!" ser mostrada na tela, a bola começará a se mover com uma velocidade inicial **V0**.Cada vez que a bola sair do volume pelo lado que deveria ser protegido pelo plano controlado pelo jogador, ela volta instantaneamente para o centro do volume e o processo reinicia.

**1.1.20.** Ao selecionar a opção *Options* no menu OP, a aplicação deve redirecionar a tela para um menu (menu 0) contendo possibilidade de alterar as teclas de controle do plano e de comandos executados pelo jogador. Ele também deve possuir o botão *Back to Initial Menu*, que ao ser selecionado, faz retornar para o menu 0.

**2. Dados Levantados após o Primeiro Ciclo de Desenvolvimento**

**2.1. Requisitos**

* **Requisitos de Menus e Rede**

**2.1.1.** Ao ser selecionada a opção *Multiplayer* no menu 0, a aplicação deve automaticamente tentar conectar ao servidor. Se o servidor não estiver disponível, a aplicação deve abrir um diálogo com a mensagem *Server temporalially unavailable*. Se o servidor estiver disponível, a aplicação deve redirecionar a tela para outro menu (menu MULT) que mostre uma lista de *lobbys* criados. Abaixo dessa janela, devem existir as opções *Create Lobby* e *Join Lobby*.

**2.1.2.** Cada elemento da lista de *lobbys* criados deve ser visível na forma:

**2.1.2.** Ao ser selecionada a opção *Create Lobby* no menu MULT, uma janela deve ser aberta pedindo para que o jogador insira o nome do *lobby*. Após a janela ser confirmada com um nome válido (não-nulo), um novo lobby deve ser mandado para o servidor e a aplicação deve redirecionar a tela para outra tela que possua:

* Lista de jogadores conectados (o *host* ocupa a primeira posição e as demais recebem com a palavra *Open*);
* Opção de dificuldade;
* Opção de habilitar *cheats*;
* Botão *Start Game*;
* Outras opções v1d4 l0k4.

**2.1.3.** O servidor da aplicação deve estar sempre pronto a receber novos *lobbys* de jogadores. Cada novo *lobby* deve ser inserido em uma lista de *lobbys* criados mantida no servidor.

**2.1.4.** Quando um jogador se conecta a um lobby, ele estabelece uma conexão privada diretamente com o jogador criador do *lobby* (*host*). O servidor também deve estar ciente dessa ação para atualizar a lista de *lobbys* vista pelos outros jogadores. A aplicação deve redirecionar a tela para uma tela semelhante à tela descrita no requisito 2.1.2., mas com as opções não-editáveis (apenas podendo ser visualizadas).

**3. Discussão**

Por o projeto não estar concluído — de fato, ele nem mesmo foi iniciado ainda — a discussão ainda não pode ser realizada com precisão.

1. Deleitável: adj. Que consegue causar deleite; em que há ou provoca prazer e satisfação; deleitoso. Fonte: Calibri 10. [↑](#footnote-ref-1)
2. É importante lembrar que a velocidade é uma grandeza vetorial. Quando ela é mencionada, nos referimos a um vetor **V** = (Vx,Vy,Vz). [↑](#footnote-ref-2)