

ARHockey: Um Jogo em Realidade Aumentada Baseada em Projetores

Beatriz N. S. Vieira*

Camila Theodoro*

Lucas P. Trias*

Fábio R. de Miranda**

Romero Tori**

Centro Universitário Senac, Bacharelado em Ciência da Computação, São Paulo – SP, Brasil

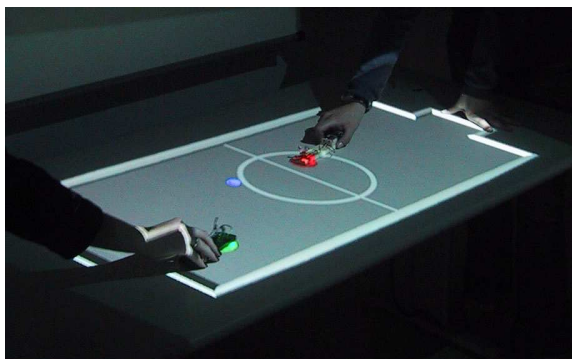


Figura 1: ARHockey em funcionamento

Resumo

Este trabalho apresenta o projeto e a implementação de prova de conceito de uma versão com realidade aumentada (RA) baseada em projetores do jogo conhecido como *air hockey* ou *shufflepuck*, comumente encontrado em fliperamas ou áreas de entretenimento de shopping centers.

A RA baseada em projetores permite a participação de um número maior de pessoas num jogo porque dispensa o uso de equipamentos individuais de visualização, e apresenta para os desenvolvedores possibilidades interessantes de alterações dinâmicas dos elementos do jogo que não são possíveis no jogo real.

Palavras-Chave: jogos eletrônicos, realidade aumentada, simulação física, sistemas projetor-câmera

Abstract

This work presents the design and implementation of a projector-based augmented reality version of the game known as *air hockey* or *shufflepuck*, commonly found in arcade halls and malls.

Projector-based AR allows a larger number of people to take part in a game because there's no need for individual visualization equipment, and opens up interesting possibilities to the game developer by allowing dynamic changes in some of the game elements, that would not be possible in the real game.

Keywords: augmented reality, physics simulation, projector-camera systems, video games.

Contatos:

*{beatriz.vieira, camila.theodoro, lucas.trias}@senacsp.edu.br

**{fabio.rmiranda, romero.tori}@sp.senac.br

1. Introdução

O projeto *AR Hockey* (*augmented reality hockey*) tem como objetivo criar um jogo multijogador relacionado ao ambiente físico em que se possa participar sem empecilhos e que possa explorar a inserção de elementos virtuais para criar novas possibilidades de jogo. Este trabalho apóia-se fortemente em realidade aumentada [Azuma 1997] e no jogo *air hockey* original.

1.1 Organização do texto

Na seção 2, é apresentado um retrospecto de trabalhos que tratam de jogos em realidade aumentada, e estes trabalhos são comparados com o que aqui se apresenta. O projeto é explicado em maiores detalhes na seção 3, e na seção 4 são apresentados resultados do estado atual do projeto e as conclusões.

1.2 Realidade aumentada

A realidade aumentada (RA), é uma variação da realidade virtual (RV) que se caracteriza por integrar o ambiente virtual ao ambiente real, utilizando-se de recursos computacionais para inserir elementos sintéticos na percepção que usuários têm do mundo físico [Azuma 1997]. Esta tecnologia pode ser aplicada via diversos meios de visualização, sendo que cada um deles possui suas próprias características e graus de liberdade. Basicamente, existem quatro classes de RA [Raskar et al. 2005]: *Optical See-Through*, que utiliza um HMD (*head mounted display*) com uma lente semi-opaca que permite a visão direta da realidade ao mesmo tempo em que serve de anteparo para a projeção dos elementos sintéticos aumentadores; *Video See-Through*, que utiliza um HMD que apresenta

diretamente aos olhos do usuário as imagens reais captadas em tempo-real por uma câmera misturada com elementos virtuais; *Monitor-based AR*, que utiliza monitores convencionais para apresentar as imagens reais e virtuais misturadas; e *Projector-based AR*, que utiliza um ou mais projetores para inserir elementos aumentadores da realidade sobre anteparos existentes no ambiente real (objetos ou telas adequadas à projeção, por exemplo).

1.3 O jogo convencional e o ARHockey

O jogo *air hockey* convencional, como ilustrado na Figura 2, consiste em uma competição entre dois jogadores que utilizam batedores para impulsionar um disco em direção ao gol adversário, sendo vetado ao jogador ultrapassar o meio da mesa de jogo. O jogo acontece sobre uma mesa retangular especial, que possui pequenos orifícios que expõem ar fazendo com que o atrito entre o disco e a mesa seja diminuído, de maneira a dar mais agilidade ao jogo e fazer com que o disco se movimente mais rapidamente.



Figura 2: Jogo *air hockey* convencional

Este projeto propõe explorar novas possibilidades de interação e apresentação oferecidas pela realidade aumentada, mantendo as regras originais do jogo.

Na versão do jogo em RA, é usada uma mesa comum em que são projetados objetos sintetizados por computação gráfica que servem como os diversos elementos de jogo (paredes da mesa, caçapas). Os jogadores deverão interagir com os objetos sintéticos, principalmente com o disco de *hockey* virtual, através de batedores reais incrementados com LEDs infravermelhos, cujos movimentos são acompanhados através do uso de uma *webcam* convencional e analisados por visão computacional, como demonstrado na Figura 3.

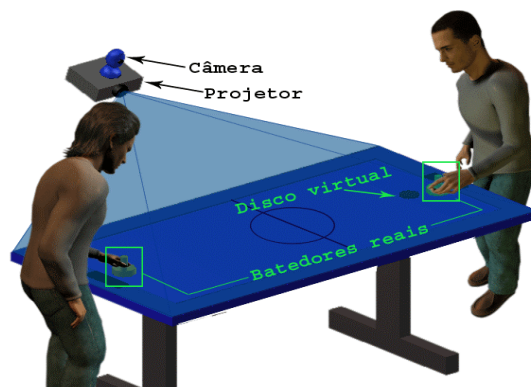


Figura 3: Estrutura do jogo com os elementos virtuais projetados sobre a mesa

2. Trabalhos Relacionados

Atualmente há uma grande busca por maior interatividade dos jogos com os jogadores. Há possibilidades interessantes e pouco exploradas que surgem da combinação de realidade aumentada e jogos, pois permitem combinar características da realidade que podem ser aproveitadas em dinâmicas inovadoras de jogo (dimensão espacial real, tangibilidade, disponibilidade, familiaridade do usuário) com a facilidade de se sintetizar elementos virtuais com base em técnicas empregadas tradicionalmente em jogos convencionais.

Há trabalhos anteriores que fazem uso da realidade aumentada em jogos e que serão citados a seguir. O *ARQuake* [ARQuake] é uma adaptação do jogo de tiro em primeira pessoa *Quake* para ser jogado ao ar livre. Nessa implementação, são utilizados *HMDs*, computadores portáteis e um sistema *GPS* para controle do jogo. Com esses equipamentos acoplados às costas do usuário, este pode se locomover no mundo real e jogar contra monstros virtuais.

O boliche *AR-Bowling* [Matyszczok et al 2004] é jogado com o auxílio de um *HMD* e uma luva. Marcadores são colocados no chão ou parede para que a cena de uma pista de boliche seja mostrada no visor. O jogador arremessa a bola contra os pinos virtuais e aqueles que têm contato com ela são derrubados.

O *Augmented Reality Chinese Checkers* [Cooper et al. 2004], um xadrez chinês, possui marcadores para determinar a posição e tamanho do tabuleiro e movimentar as peças. Cinco câmeras detectam esses marcadores e a cena virtual é mostrada no visor do *HMD* do jogador.

O *Ar²Hockey* [Ohshima et al. 1998], é também um jogo de *air hockey* em realidade aumentada, mas diferente do trabalho que aqui se apresenta em alguns aspectos importantes. Nesse projeto, os jogadores estão equipados com *head mounted displays* que permitem visualizar o mundo real e a imagem virtual exposta no visor ao mesmo tempo e um batedor para impulsionar o disco virtual. Um sensor magnético e uma câmera

são acoplados à cabeça do jogador a fim de ajustar a visão sintetizada para o jogador ao seu ponto de vista e determinar a posição de sua cabeça. Cada batedor possui *LEDs* infravermelhos para que sua posição seja captada por uma câmera direcionada diretamente à mesa. São utilizados três computadores para processar as duas imagens vindas das câmeras acopladas aos *HMDs* e o movimento dos batedores e um quarto computador para processar a posição da cabeça, renderizar as imagens e sons e controlar o sistema.

Uma outra implementação eletrônica do jogo *air hockey*, porém não baseada em realidade aumentada, é o *Airhockey Over a Distance* [Mueller et al. 2006]. Nesta versão os usuários jogam com discos reais, como se fosse um jogo de *air hockey* tradicional. Há uma fenda na metade da mesa por onde os discos desaparecem e são coletados por uma máquina de jogo. Um sensor capta a direção e velocidade do disco no momento em que estes colidiram depois de passarem pela metade da mesa, e envia estas informações pela *internet* para onde estão os outros jogadores. Há mecanismos que se encarregam de lançar os discos para o jogador remoto de forma coerente com o arremesso feito pelo primeiro jogador. Também há uma câmera com boa resolução que capta as imagens de cada um dos jogadores e as transmite pela rede para que seja projetada aos adversários, para que se tenha a impressão de que se joga um jogo de *air hockey* verdadeiro, com os jogadores presentes.

Os projetos citados acima, com exceção do *Airhockey Over a Distance*, utilizam, ao invés de projetores, *head mounted displays* para visualizar a cena virtual. Alguns deles ainda utilizam equipamentos adicionais como luvas e mais de um computador. Para captação de movimento e localização de posições utilizam, como neste projeto, câmeras e marcadores. Em geral, as soluções apresentadas têm custo e complexidade de montagem mais elevados do que a abordada no *AR Hockey*.

3. O Projeto

O *ARHockey* é implementado usando realidade aumentada baseada em projetores, de maneira a permitir que não só os jogadores, mas todos os presentes também possam acompanhar as partidas. Tal abordagem libera os jogadores de equipamentos acoplados ao corpo (como o *HMD*) que possam prejudicar seu conforto, retirar sua atenção ou mobilidade, e elevar o custo do projeto devido ao valor do equipamento.

Esse tipo de RA é objeto crescente de estudo e muitas áreas o têm adotado. É possível citar como exemplo um protótipo de um sistema para realizar cirurgias crânio-maxilo-facial que utiliza a realidade aumentada baseada em projetores [Hoppe et al. 2002].

O projeto foi dividido em dois objetivos principais. O primeiro objetivo, assunto deste artigo, foi

implementar o jogo através de recursos de RA de maneira fiel às regras originais. O segundo é explorar as possibilidades proporcionadas pela RA, como por exemplo movimentar gols e caçapas ao longo do jogo, ou realizar modificações dinâmicas no formato da mesa, ou então em atributos como o centro de massa do disco para alterar sua trajetória, além de implementar novas regras e modos de jogo, como por exemplo, criar mini-jogos que façam uso dos batedores como dispositivos de entrada.

A implementação do jogo foi dividida em subsistemas com competências individuais e independentes. O subsistema de simulação física, responsável pelo controle da interação física entre os objetos, o de rastreamento, responsável por identificar a posição dos batedores, e o de apresentação, responsável por gerar as imagens do ambiente de jogo.

3.1 Funcionamento geral do jogo

O *ARHockey* é jogado por dois participantes munidos de batedores e um plano horizontal (de preferência uma mesa) no qual o ambiente do jogo será projetado. Na mesa virtual projetada há um disco que deve ser impulsionado pelos jogadores em direção ao gol adversário, localizado na extremidade oposta da mesa. O fim da partida é dado quando um dos jogadores atinge a marca de sete gols.

Uma câmera captura imagens da cena do jogo que são usadas para localizar os batedores. Após a simulação física da interação dos batedores com os objetos virtuais, é gerada uma representação gráfica do estado atual. Esta representação é projetada sobre a superfície horizontal de projeção.

3.2 Os subsistemas

O jogo é dividido em três subsistemas básicos: apresentação, rastreamento e simulação física. Cada um deles é responsável pelo processamento de funcionalidades estanques que compõem o jogo e se integram apenas por interfaces de acesso bem definidas. Assim é possível diminuir o acoplamento entre os subsistemas, que serão melhor descritos a seguir.

3.2.1 Apresentação

O subsistema apresentação é responsável pela representação gráfica dos objetos da cena tridimensional a serem projetados na superfície física.

A função do subsistema apresentação é criar uma janela de visualização e nela desenhar os objetos virtuais da cena, ou seja, a mesa e o disco. Para a criação desta janela foi utilizado o *Ogre*, um *engine 3D* que constrói a cena através de grafos de cena. Uma cena é composta por luz, câmera e entidades (objetos).

Os objetos que representam a mesa e os batedores foram salvos no formato *mesh* (formato específico do *Ogre*) para que pudessem ser incorporados ao grafo de cena. Esses arquivos *mesh* podem ser obtidos a partir de exportadores que são adicionados a alguns programas de modelagem 3D.

3.2.2 Rastreamento

O sistema de rastreamento é o responsável pela garantia de coerência entre o ambiente de jogo real e a cena virtual. É ele quem captura imagens da área de jogo e identifica a posição dos batedores dotados de LEDs infravermelhos.

O funcionamento correto deste sistema é o que permite que os jogadores efetivamente interajam com o disco projetado. A captura de imagens da mesa real é feita através de uma *webcam* comum, que possui uma resposta ligeiramente além da banda visível no limite inferior do espectro, permitindo que o infravermelho próximo da luz visível apareça nas imagens. Para acentuar o contraste entre luz infravermelha e não-infravermelha é usado um filtro de plástico, que permite somente a passagem de luz do primeiro tipo. Com isso a imagem final que a câmera captura é basicamente escura, com dois pontos claros, os *LEDs* (*Light Emitting Diodes*) infravermelhos que são instalados nos batedores, como pode ser visto na **Figura 4**.

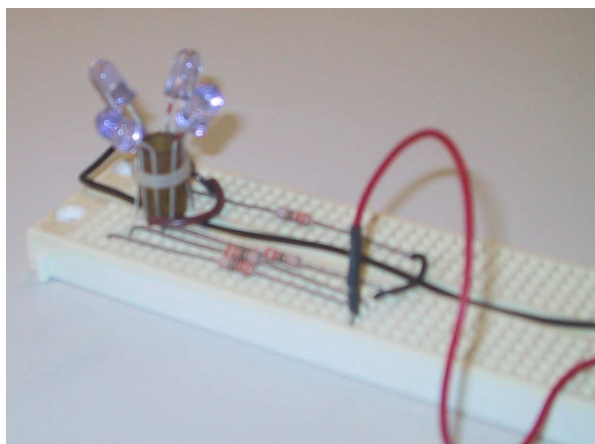


Figura 4: Protótipo de marcador com quatro LEDs

Estes *LEDs* podem ser camuflados para não atrapalharem a dinâmica do jogo, e emitem luz no espectro que é aceito pelo filtro de luz visível. Deste modo consegue-se uma imagem nítida da posição dos batedores na cena de jogo. Posteriormente esta imagem é tratada por um algoritmo de visão, implementado usando a biblioteca *OpenCV* [OpenCV], que é descrito na seção seguinte, e o rastreamento retorna a posição dos batedores no espaço para o subsistema de simulação física.

Foram encontrados problemas de desempenho no subsistema de rastreamento. A *webcam* utilizada é destinada a consumidores comuns, e tenta automatizar grande parte dos ajustes que podem ser feitos em

relação ao brilho, contraste da imagem e tempo de exposição na captura. Uma consequência disso é que, quando a imagem capturada é escura, a câmera aumenta o tempo de exposição, e assim a taxa de atualização das imagens cai muito. Como no primeiro protótipo implementado todo o sistema estava atrelado ao rastreamento, a simulação e a apresentação atingiam taxas de atualização de 8 quadros por segundo, que é muito pouco para uma simulação física de qualidade. A solução para esta deficiência consiste em usar uma linha de execução separada para o sistema de rastreamento, o que permite desacoplá-lo do resto do jogo. Tal abordagem não resolve efetivamente o problema da baixa taxa de captura da câmera, mas permite que a simulação e a apresentação sejam executadas com alto desempenho. Considera-se que a baixa taxa de atualização da câmera não representa um problema para o sistema de rastreamento, uma vez que nas imagens dos vídeos capturados a estas taxas, o movimento aparenta ser bastante contínua.

Algoritmo de Visão

A posição dos batedores é estimada encontrando-se a posição de sua representação nas imagens capturadas pela câmera e efetuando-se transformações de correção de paralaxe sobre os pontos encontrados.

Estas transformações projetivas mapeiam coordenadas do plano da mesa de projeção identificadas na imagem capturada durante uma fase de calibração para pontos cujas coordenadas são conhecidas no sistema de coordenadas da imagem capturada. O efeito deste tratamento pode ser visto na **Figura 5**.

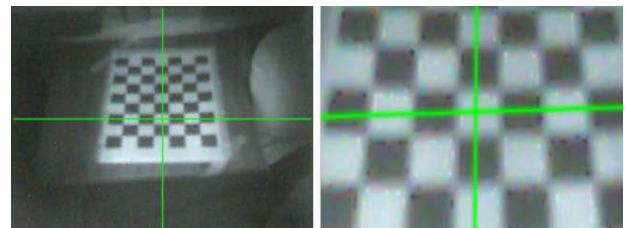


Figura 5: Imagem original (à esquerda) e transformada para um sistema de coordenadas que corresponde ao plano da mesa (à direita)

Este recurso permite que, mesmo que a câmera seja posicionada obliquamente em relação ao plano de projeção, não haja distorção na estimativa da posição dos batedores em coordenadas da mesa. O processo de obtenção das coordenadas iniciais dos pontos brancos (batedores) na imagem é resultado da execução de um processo de tratamento nas imagens que é descrito a seguir.

Inicialmente a imagem do batedor é capturada pela câmera e convertida para tons de cinza, em seguida é limiarizada para que se converta de tons de cinza para preto-e-branco puros. Posteriormente, é aplicada uma sequência de operações de abertura morfológica [Barrera et al. 1994] que visam aparar eventuais arestas

e excentricidades (Figura 6) e segmentar melhor a imagem de maneira a determinar a posição do LED. Estas podem se formar devido a um ofuscamento da câmera que ocorre quando ela se encontra dentro de uma faixa de 10° ao redor do elemento emissivo do LED. Nesta faixa a radiação do LED é mais intensa e o tempo lento de adaptação da câmera a esta nova intensidade faz com que ocorra ofuscamento.

Após a sequência de aberturas, é identificado o contorno da representação do batedor e seu centro, calculando-se a média das posições dos píxeis que compõem o contorno. Finalmente, este centro calculado é usado como dado de entrada em um filtro de Kalman [Kalman 1960], que atenua ruído e imperfeições na estimativa da posição dos batedores.

O filtro de Kalman é um ferramental estatístico usado para atenuar variações em sistemas de filtragem linear usados para previsão. Ele se baseia no histórico de variação de um sistema para prever um estado futuro. Quando há uma medição real do estado previsto, os valores são usados pelo filtro para corrigir seu estado interno de modo a amenizar os erros de previsão ao longo do tempo.

Um fluxograma ilustrado de uma iteração do algoritmo de visão é mostrado na Figura 12



Figura 6: Fluxograma de uma iteração do sistema de rastreamento

3.2.3 Simulação Física

A simulação física é responsável pela verossimilhança entre o comportamento dos elementos virtuais projetados e o aquele que se esperaria de elementos reais.

Toda movimentação, atrito e choques que acontecem durante as partidas são simuladas de acordo com as leis da física que regem este tipo de eventos, permitindo um nível de realismo que pode ser bastante grande. Opcionalmente, os objetos podem possuir

características físicas diferentes das características convencionais, permitindo mais possibilidades de entretenimento.

Simular a interação física entre os componentes em um ambiente não é uma tarefa fácil devido ao grande número de variáveis envolvidas. No jogo há um subsistema específico para ela, baseado na biblioteca *ODE - Open Dynamics Engine* [ODE], de código aberto, sendo que o acesso a esta é feito através do wrapper em C++ *OgreOde* [OgreOde]. Esta é responsável pelo controle do mundo de simulação

A cada iteração, esta biblioteca recalcula a posição dos objetos de acordo com a interação destes no ambiente. A atualização da simulação no *ODE* pode ser feita de duas formas: utilizando o *WorldStep* ou o *WorldQuickStep*, que são dois métodos diferentes de integração das equações diferenciais ordinárias que descrevem o movimento dos corpos rígidos que são simulados pela biblioteca. Estes dois métodos garantem níveis diferentes de precisão. O primeiro, garante maior precisão, porém aloca mais recursos e é mais lento. O segundo, que é utilizado neste projeto, garante uma velocidade de processamento maior e menor alocação de recursos, mas trabalha em uma precisão um pouco menor.

Para que a simulação funcione, todos os objetos que interagem de alguma forma devem estar definidos para o simulador do *ODE* no mesmo espaço e também no mesmo mundo. No *ODE* um objeto é representado por duas partes, o corpo e a geometria. O corpo rígido do objeto possui massa, direção, posição, orientação e velocidade, sendo o responsável pelas ações do objeto, por exemplo, ser suscetível à gravidade. A geometria, por outro lado, é responsável pela colisão entre os objetos, representando apenas seus contornos. Sendo assim, objetos estáticos que não sofrem reação, mas ainda assim influenciam na colisão (como as paredes da mesa de jogo) apenas precisam de uma representação geométrica, enquanto os objetos dinâmicos simulados (como o disco) necessitam tanto da geometria quanto do corpo.

No projeto, os batedores são compostos apenas por uma geometria, pois estes possuem apenas influência na simulação e nenhum tipo de reação. Já o disco é formado por duas geometrias, um cubo para representar sua base e uma esfera para representar o seu diâmetro e um corpo, pois o disco possui reação física e é o principal componente da simulação.

A mesa é composta por várias geometrias que representam seus contornos para fins de colisão no simulador físico e seu fundo. Essas geometrias são carregadas para o sistema por meio de um arquivo de configuração em *Extensible Markup Language (XML)*, utilizando a biblioteca *TinyXml* [TinyXml], para garantir flexibilidade e facilidade de modificações no formato da mesa.

O arquivo *XML* de configuração das geometrias da mesa possui uma estrutura bastante simples. O nó raiz possui sub-nós que representam cada uma das mesas que podem ser carregadas no jogo. Dentro desses sub-nós é possível configurar cada uma das paredes e barreiras presentes na mesa de acordo com possibilidade de movimentação (móvel ou não), posição, tamanho, proprietário (no caso dos gols, a que jogador pertence) e *mesh* (no caso das barreiras, qual sua representação visual), como pode ser visto na estrutura da Figura 7.

```
<arhockey>
  <table name="testTable" mesh="tableTexture.mesh">
    <wall name="wall" mobility="still">
      <size x="100" y="8" z="5"/>
      <position x="0" y="0" z="50"/>
    </wall>
    <wall name="gol" mobility="still">
      <size x="5" y="8" z="100"/>
      <position x="50" y="0" z="0"/>
      <gol owner="player1"/>
    </wall>
    <wall name="barrear1" mobility="still">
      <size x="25" y="15" z="5"/>
      <position x="25" y="1.4" z="50"/>
      <mesh name="barrear.mesh"/>
    </wall>
  </table>
</arhockey>
```

Figura 7: Exemplo do *XML* de configuração

A estrutura de geometrias de colisão do sistema pode ser vista esquematicamente na Figura 8.

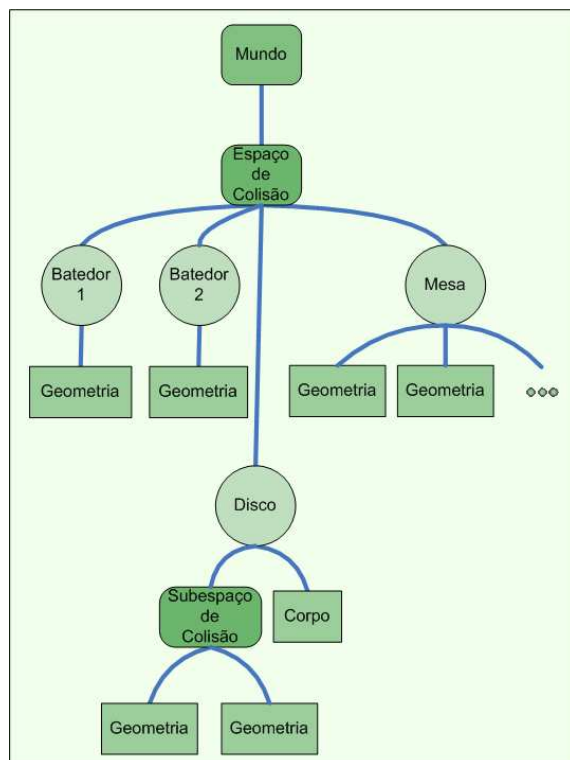


Figura 8: Grafo de cena da biblioteca *ODE*

A estrutura do sistema de colisão é dividida em quatro partes principais: iniciação, fechamento, simulação e colisão. A iniciação é responsável pela

criação do mundo e ajuste da gravidade. Já o fechamento trata da devolução dos recursos utilizados por este sistema. A simulação é um passo que representa um avanço de tempo no mundo físico que é simulado, e é responsável pelo reposicionamento dos objetos no ambiente. Por fim, a colisão governa o comportamento dos objetos quando eles colidem. Esta função cria contatos, que é o modo da biblioteca *ODE* representar um contato de colisão, e especifica que tipo de reação esses objetos têm, como o atrito ou a suavidade.

O subsistema de simulação usa um par de coordenadas para posicionar o batedor no mundo físico, este par de coordenadas representa a posição em que se encontra a mão do usuário e é obtido através do subsistema de rastreamento. A simulação também disponibiliza a posição e rotação de cada objeto dinâmico participante da simulação para que o subsistema de apresentação seja capaz de representar estes objetos no área de desenho.

A representação das geometrias de cada objeto pode ser vista na Figura 9 representadas através de *wireframes* em volta da malha de polígonos que representa visualmente cada um dos objetos.

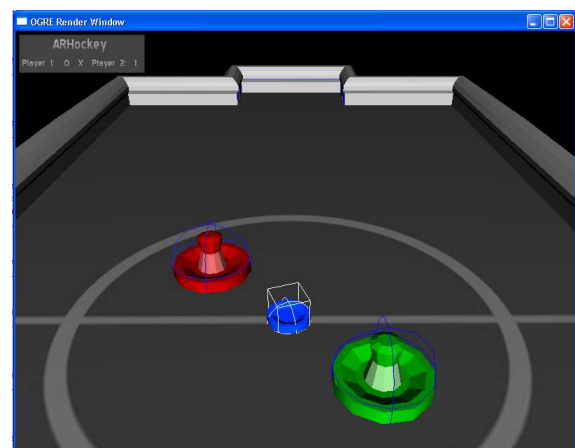


Figura 9: Representação "visual" e geométrica do jogo

Durante a fase de implementação do subsistema a detecção de colisão apresentou uma dificuldade. A biblioteca de colisão *ODE* 0.5, versão utilizada neste projeto, não possui suporte a geometrias cilíndricas, o que dificultou na definição de limites do disco. Para resolver esse problema, o disco do jogo passou a receber duas geometrias, a geometria responsável pela representação da base do disco, um paralelepípedo e a geometria responsável pelas laterais do disco, uma esfera. Juntas essas duas geometrias foram capazes de representar bem o formato do disco.

Uma outra solução seria gerar uma geometria através de malha de triângulos, porém essa funcionalidade está em desenvolvimento na biblioteca e ainda não possui tratamento para colisões de duas geometrias deste tipo, além de deixar a simulação um pouco mais lenta. Devido a essas restrições, essa solução foi descartada.

3.3 Integração dos subsistemas

O subsistema de apresentação solicita que o rastreador atualize a posição dos batedores. A simulação utiliza esta posição para atualizar o local onde a representação virtual do baterador se encontra no mundo físico e calcular possíveis efeitos dessa movimentação. Finalmente a apresentação usa os dados mais recentes da simulação para gerar uma representação gráfica do estado atual do jogo, como mostrado na Figura 10.

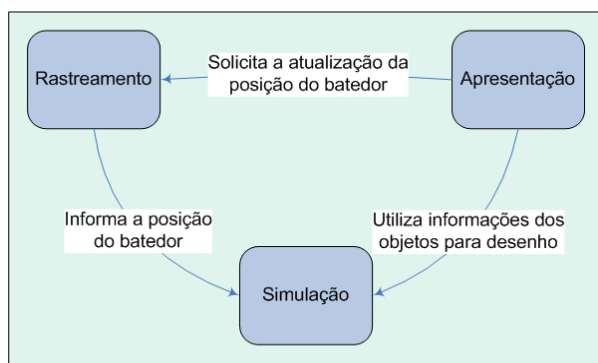


Figura 10: Diagrama de blocos que mostra a integração do sistema em um ciclo de execução.

4. Resultados

No protótipo atual, os três subsistemas estão integrados e com partes de sua funcionalidade operacional. A visualização teve suas funcionalidades básicas implementadas, como a visualização dos objetos e da pontuação de jogo, para dar suporte visual ao desenvolvimento do sistema de rastreamento e de simulação. A física do jogo, com base em simulação está operacional, o rastreamento tem todos seus estágios em funcionamento e estão sendo feitos aprimoramentos de desempenho e flexibilidade. A Fig. 4 mostra uma seção de testes do protótipo atual.



Figura 11: Seção de testes do protótipo atual do ARHockey

5. Conclusões e Trabalhos Futuros

Apesar de existirem implementações alternativas do jogo *air hockey* em realidade aumentada, que são baseadas em *HMDs* do tipo *optical see through*, a

utilização da Realidade Aumentada baseada em projetores abre novas e interessantes possibilidades, pois permite a visualização do jogo não só pelos jogadores participantes, mas também por outras pessoas. Deste modo consegue-se uma abordagem inovadora deste jogo e do desenvolvimento de jogos como um conceito, empregando técnicas que estão sendo lançadas no mercado nos dias de hoje, como rastreamento de jogadores, simulação física realista e integração do jogo ao ambiente do jogador. A utilização de apenas um equipamento de visualização e um equipamento de captura popular consegue garantir menor custo do sistema como um todo, se comparado com outros tipos de equipamentos de visualização mais caros e menos difundidos como o *HMD*.

Os sistemas de visualização e de simulação conseguem manter uma taxa de atualização média de cem quadros por segundo, o que garante um bom desempenho, considerando que a taxa de atualização aceitável para aplicações interativas é de aproximadamente trinta quadros por segundo. A integração do sistema de rastreamento não altera a velocidade de atualização, pois este sistema está separado da linha de execução principal do sistema. Além disso foi aplicada a capacidade do filtro de Kalman de prever estados baseado em informações antigas para gerar previsões da posição dos batedores enquanto uma nova posição real ainda não pode ser medida. Deste modo consegue-se aumentar a taxa de atualização das posições dos batedores para a simulação sem que seja necessário aguardar uma nova medição. Como a sequência de previsões pode tender a se afastar do estado real no contexto do jogo, procura-se manter um número baixo de atualizações sem correção. Empiricamente chegou-se à conclusão de que uma previsão sem correção a cada duas corrigidas não causa distanciamento da realidade, e mantém a sensação de manipulação direta. Sendo assim, o sistema como um todo consegue garantir um bom desempenho e bem acima do mínimo tolerável para esse tipo de mídia.

Tendo em vista que o primeiro objetivo do projeto foi concluído, o jogo sofrerá melhorias e modificações em relação à sua versão convencional, como a adição de barreiras estáticas e dinâmicas à mesa. O sistema de rastreamento também sofrerá ajustes para garantir uma melhor performance e precisão.

Como trabalho futuro pretende-se implementar, alterações do formato da mesa e mudanças nas regras de jogo e inserção de novos objetivos aproveitando as possibilidades oferecidas pela realidade aumentada.

Referências

ARQUAKE. Disponível em:
<http://wearables.unisa.edu.au/projects/ARQuake/www/>
[Acesso em 02/06/2006].

AZUMA, R. T., 1997. A Survey of Augmented Reality. In: *Teleoperators and Virtual Environments* 6, 4

BANON, G. J. F., BARRERA, J., 1994. Bases da Morfologia Matemática para Análise de Imagens Binárias. Ed. IX Escola de Computação

COOPER, N. KEATLEY, A. DAHLQUIST, M. MANN, S. SLAY, H. ZUCCO, J. SMITH, R. AND THOMAS, B.H., 2004. Augmented Reality Chinese Checkers . In: *ACM SIGCHI International Conference on Advances in computer entertainment technology*.

HOPPE, H. KÜBLER, C. RACZKOWSKY, J. WÖRN, H. AND HABFELD, S., 2002. A Clinical Prototype System for Projector-Based Augmented Reality: Calibration And Projection Methods. In: *16th International Congress and Exhibition on Computer Assisted Radiology and Surgery (CARS)*.

KALMAN, R. E., 1960. A New Approach to Linear Filtering and Prediction Problems. In: *ACM SIGCHI International Conference on Advances in computer entertainment technology*.

MATYSZCZOK, C. RADKOWSKI, R. AND BERSSENBRUEGGE, J., 2004. AR-bowling: immersive and realistic game play in real environments using augmented reality. In: *ACM SIGCHI International Conference on Advances in computer entertainment technology*.

MUELLER, F. F. COLE, L. O'BRIEN, S. WALMINK, W. 2006. Airhockey Over a Distance. In: *CHI '06 extended abstracts on Human factors in computing systems CHI '06*.

ODE: OPEN DYNAMICS ENGINE. Disponível em: <http://www.ode.org> [Acesso em 05/04/06].

OGRE. Disponível em: <http://www.ogre3d.org/> [Acesso em 20/03/2006].

OGREODE. Disponível em: <http://www.ogre3d.org/wiki/index.php/OgreODE> [Acesso em 27/07/06].

OHSHIMA, T. SATOH, K. YAMAMOTO, H. AND TAMURA, H. 1988. AR² Hockey. In: *ACM SIGGRAPH 98 Conference abstracts and applications*.

OPENCV. Disponível em: <http://www.intel.com/technology/computing/opencv/index.htm> [Acesso em: 05/04/2006]

RASKAR, R. BIMBER, O. J., 2005. Spatial Augmented Reality Merging Real and Virtual Worlds

TINYXML. Disponível em: <http://www.grinninglizard.com/tinyxml/> [Acesso em 17/08/06].