

Utilizando Realidade Virtual e Wiimote para a Criação de Jogos Voltados à Reabilitação

Silvano M. Malfatti Edson A. G. Coutinho* Selan R. dos Santos*

Fundação Universidade do Tocantins, Brasil

* Universidade Federal do Rio Grande do Norte, Brasil

Abstract

This tutorial aims to describe the necessary steps to develop games for rehabilitation of patients using virtual environments and interaction by Wiimote. In order to accomplish that, we discuss the Wiiuse library, showing some uses for its structures and methods to create virtual environments which provide body movement-based interaction. At last, this tutorial describes some case studies about using Wiimote in rehabilitation games.

Keywords: Virtual Reality, Wiimote, Games, Rehabilitation.

Resumo

O tutorial proposto tem como objetivo descrever os passos necessários para o desenvolvimento de jogos destinados a reabilitação de pacientes utilizando ambientes virtuais com interação via Wiimote. Para tanto, será abordada a biblioteca Wiiuse com o intuito de demonstrar o uso de suas estruturas e funções para a criação de ambientes virtuais com suporte a interação baseada em movimentos corporais. Por fim, este tutorial descreve alguns estudos de caso demonstrando o uso do Wiimote em jogos de reabilitação.

Palavras chave: Realidade Virtual, Wiimote, Jogos, Reabilitação.

Contatos:

{silvanomalfatti, *alyppyo}@gmail.com
*selan@dimap.ufrn.br

1. Introdução

A utilização de jogos com outras finalidades que apenas entretenimento já não é uma novidade. Hoje em dia podem ser encontrados diversos estudos acadêmicos e aplicações comerciais que utilizam jogos para outros fins, sendo grande parte deles para treinamento de habilidades [Michael and Chen 2006].

Um dos principais fatores que colaboram para isso é que um jogo, assim como a Realidade Virtual em geral, permite que diversas situações possam ser experimentadas através de simulações, sem os ônus materiais envolvidos realmente nelas. Além disso, não se pode deixar de frisar o caráter lúdico da atividade,

que geralmente é o seu ponto-chave e um dos principais responsáveis por um maior nível de engajamento com a simulação. Esse potencial existente nos jogos pode ser aproveitado para diversas áreas, incluindo a saúde.

A reabilitação motora é por sua natureza um processo repetitivo e essa repetição tende a reduzir a motivação dos pacientes [Burdea 2003]. Tomando por base essa premissa, podemos perceber que o uso de jogos é uma alternativa viável, pois quando bem trabalhada, pode vir a contribuir para neutralizar a desmotivação causada pela monótona repetição de movimentos. Neste caso, os movimentos repetitivos podem ser utilizados como entrada para o jogo.

A proposta deste tutorial é ser um guia simplificado para a criação de jogos para serem utilizados em terapias de reabilitação motora utilizando o Wiimote e a biblioteca Wiiuse. Inicialmente, abordamos as principais características deste tipo de jogo, incluindo alguns requisitos de planejamento, bem como a questão da motivação. Logo em seguida, na Seção 3, pode ser encontrada uma breve descrição do Wiimote e do seu funcionamento. Os detalhes da biblioteca Wiiuse estão dispostos na Seção 4 e após isso, na Seção 5, é sugerido um conjunto de etapas para a criação deste tipo de jogo. Para finalizar, são apresentados alguns estudos de caso e, em seguida, as considerações finais.

2. Jogos de Reabilitação

Jogos voltados à reabilitação necessitam de um planejamento bem mais cuidadoso do que jogos destinados apenas ao entretenimento. Isso se deve não só ao fato do conteúdo abordado no jogo fazer parte do tratamento do jogador, mas principalmente pelo caráter especial do seu público-alvo.

Pessoas que estão passando pelo processo de reabilitação geralmente possuem sequelas físicas e emocionais que devem ser levadas em consideração durante o projeto do jogo. Ao desenvolver um jogo voltado à reabilitação é preciso saber que o tratamento não envolve somente a recuperação da parte física, pois fatores psicológicos também estão envolvidos.

Jogos com metas muito difíceis de serem alcançadas, por exemplo, não são aconselháveis neste caso. Isso ocorre por se tratar de pessoas que possuem

alguma função prejudicada apresentam uma tendência maior em se sentirem frustradas ou até mesmo à depressão, exatamente por não poder realizar a função em questão como desejado. Logo, é aconselhável encorajá-las o máximo possível, recompensando-se seus feitos positivos no decorrer do jogo e relevar sutilmente seus erros.

Quando tratamos de reabilitação de problemas motores, o papel da fisioterapia já está bem definido [Nudo et al. 1996]. A fisioterapia tradicional consiste repetição de sequências de movimentos passivos, que pode vir a se tornar tedioso ao longo da terapia. Todavia, o método de movimentação passiva estimula principalmente as articulações [Lynch et al. 2005], sendo amplamente utilizado. Cabe então ao jogo aproveitar esse processo e transformá-lo, fazendo com que o que antes poderia ser considerado rotineiro e cansativo passe a ser uma das fontes de motivação do tratamento.

Visando amenizar o caráter repetitivo inerente às terapias de recuperação motora, os jogos de reabilitação devem procurar utilizar tais movimentos como entrada a ser convertida em comandos. Com base nos comandos inseridos, o jogador pode ter um resultado imediato de suas ações como receber um bônus, passar de uma fase ou aumentar seu nível. Isso pode manter sua motivação elevada mesmo que os seus movimentos ainda estejam aquém dos esperados para o término da terapia.

3. Wiimote: Características e Funcionamento

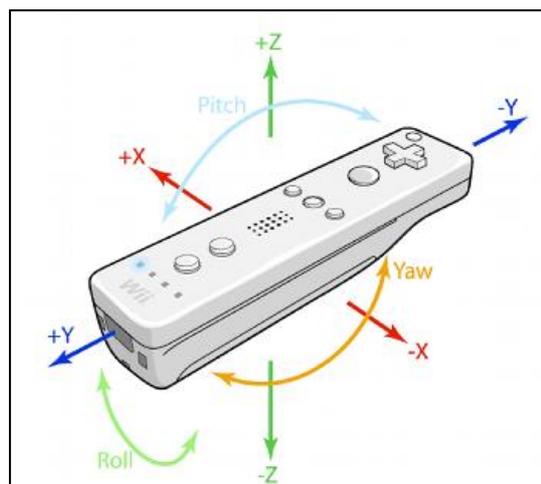
O console Wii da Nintendo, juntamente com o Xbox 360, foram os precursores de uma nova geração de videogames cuja interatividade não se limita apenas a ações disparadas por intermédio de botões. Nestas plataformas, a jogabilidade se estende aos movimentos efetuados pelo corpo do jogador que pode ser capturado através de recursos como processamento de imagens ou dispositivos equipados com giroscópios e acelerômetros.

No caso do Wii, o reconhecimento dos movimentos realizados pelo jogador é feito pelo controle, denominado Wii Remote ou Wiimote, que conta com um acelerômetro responsável pela detecção de movimentos nos eixos X, Y e Z. O Wiimote é equipado também com um sensor infravermelho de alta resolução responsável pelo rastreamento de posições que atua com base em LEDs especiais posicionados pelo ambiente. A Figura 1 apresenta o Wiimote e os movimentos reconhecidos pelo mesmo.

Além do suporte nativo no se que se refere ao reconhecimento de movimentos como aceleração e rotação nos eixos X, Y e Z, o Wiimote conta com uma série de acessórios que permitem ampliar a interatividade entre o jogador e os aplicativos. Dentre

os principais acessórios é possível citar o Nunchuk, Sensor Bar e o Motion Plus.

Figura 1: Movimentos reconhecidos pelo Wiimote.



O Nunchuk consiste de um controle direcional analógico também equipado com um acelerômetro. Ao conectar este acessório ao Wiimote, o participante poderá utilizar as duas mãos para interagir com o jogo. Além de aumentar a interatividade, o conjunto formado pelo Nunchuk e Wiimote permite a criação de novas formas de jogabilidade como é o caso do jogo Wii Boxing no qual o participante deve executar movimentos combinados para atacar ou se defender do oponente. A Figura 2 apresenta o Nunchuk associado ao Wiimote.



Figura 2: Acessório Nunchuk conectado ao Wiimote.

Outro acessório fundamental é a Sensor Bar, que consiste basicamente de dois LEDs infravermelhos utilizados como referência para jogos que necessitam do rastreamento de posições. A principal limitação neste caso consiste em manter a barra de LEDs no raio de ação do sensor infravermelho contido no Wiimote.

Também é possível acoplar na parte inferior do Wiimote o Motion Plus. Este acessório foi desenvolvido com o intuito de oferecer uma precisão maior no reconhecimento de movimentos do jogador. Para tanto, além do uso de acelerômetros, o Motion Plus vem equipado com um giroscópio que permite o

reconhecimento preciso de 1:1 no que se refere a movimentos de rotação nos eixos X, Y e Z. Portanto, ao desenvolver aplicativos baseados neste acessório é possível programar os movimentos do jogo com base no valor real das rotações realizadas pelo jogador.

Todos os dados gerados pelo Wiimote e seus acessórios são transmitidos para o console através de uma conexão via Bluetooth. O protocolo de comunicação utilizado é o mesmo empregado em outros periféricos do tipo Bluetooth. [Pheatt e Goering 2010].

Ao adotar este padrão de comunicação para o Wiimote, os projetistas permitiram também que este dispositivo pudesse se conectar a outros equipamentos como, por exemplo, um computador. Sendo assim, ao utilizar um computador para executar a busca por dispositivos Bluetooth em um raio de ação onde se encontra um Wiimote, este poderá ser reconhecido como um dispositivo padrão de interação, assim como um mouse ou teclado.

4. Utilizando a Biblioteca Wiiuse para o Desenvolvimento de Jogos com Suporte a Wiimote

Um dos requisitos necessários para promover a interação entre o Wiimote com aplicativos desenvolvidos para computadores é a necessidade de uma biblioteca que realize o gerenciamento da conexão com os controles, bem como gerencie o processo de comunicação entre as duas pontas.

Dentre as APIs existentes para a realização deste trabalho é possível citar a biblioteca Wiiuse, desenvolvida e mantida por uma comunidade independente [Wiiuse 2011]. Através da Wiiuse é possível estabelecer a conexão de um ou mais controles com uma aplicação desktop, bem como capturar movimentos de rotação, aceleração e posição do Wiimote e seus acessórios. As seções seguintes demonstram como utilizar esta biblioteca e os passos necessários para promover a comunicação entre um computador com o Wiimote.

4.1 Configuração do Ambiente

O primeiro passo para iniciar a programação com a Wiiuse refere-se à configuração do ambiente que será utilizado para o desenvolvimento das aplicações. Neste caso, será utilizada a IDE Visual C++ 2008 Express Edition [Visual Studio 2008].

A preparação desta IDE para a programação com a Wiiuse consiste basicamente em configurar o caminho para os arquivos de cabeçalho e bibliotecas estáticas que serão necessários à etapa de compilação. Todos estes procedimentos podem ser realizados através do comando `Options` encontrado no menu `Tools` como apresentado pela Figura 3.

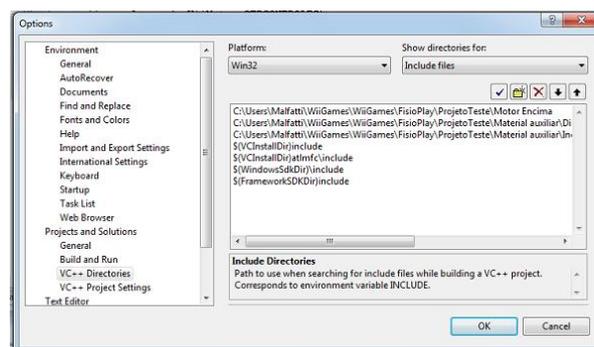


Figura 3: Tela de configuração do Visual C++ Express Edition para a programação com a biblioteca Wiiuse.

Além de especificar o caminho dos arquivos é necessário também especificar as bibliotecas estáticas que serão utilizadas pelo projeto. Este procedimento deve ser realizado através da opção `Properties` definida pelo menu `Project`. A Figura 4 demonstra a declaração das bibliotecas estáticas a serem utilizadas pelo projeto em questão.

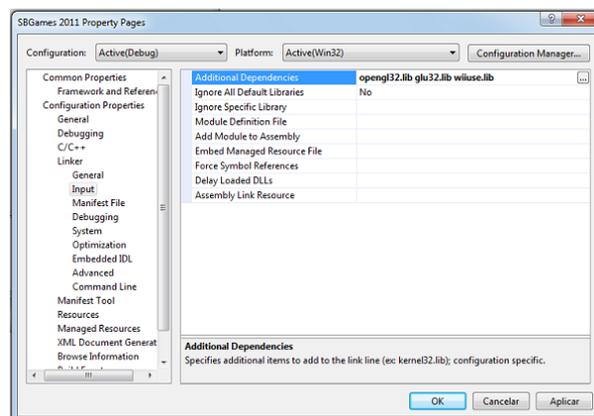


Figura 4: Declaração de bibliotecas estáticas no Visual C++ 2008 Express Edition.

Para o projeto apresentado na Figura 4 também foram incluídas as duas bibliotecas da API gráfica OpenGL necessárias à programação do mundo virtual.

4.2 Conectando o Wiimote ao Sistema Operacional

Antes de tudo, para que a biblioteca Wiiuse possa estabelecer conexão com o Wiimote, é preciso que o mesmo tenha sido registrado e reconhecido pelo sistema operacional.

No caso do Windows, existem duas formas de realizar este procedimento, através da opção de adicionar um novo hardware oferecida pelo próprio sistema ou utilizando um software específico para o gerenciamento de dispositivos Bluetooth.

Para efetuar o reconhecimento do dispositivo a partir do próprio Windows, basta abrir o Painel de

Controle do mesmo e selecionar a opção “Adicionar um Dispositivo”. Neste momento mantenha os botões 1 e 2 do Wiimote pressionados até que o mesmo apareça como um dispositivo reconhecido. A Figura 5 demonstra esta etapa.

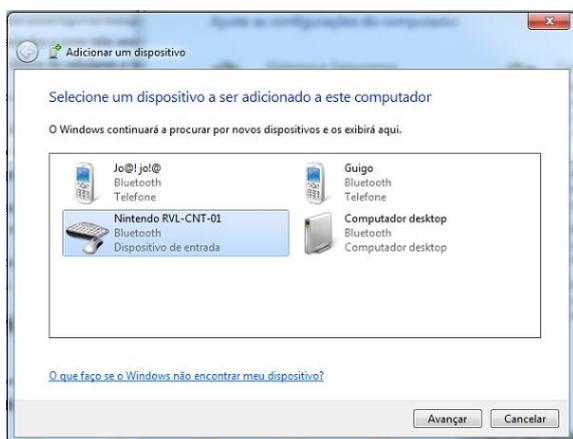


Figura 5: Reconhecimento de um Wiimote a partir do Windows.

Após o reconhecimento do dispositivo é preciso ainda realizar o pareamento do mesmo. O Windows oferece diferentes formas para realizar este procedimento, porém como o Wiimote não possui teclado, a única forma para realizar a conexão é através da opção “Emparelhar sem usar código” como mostra a Figura 6.

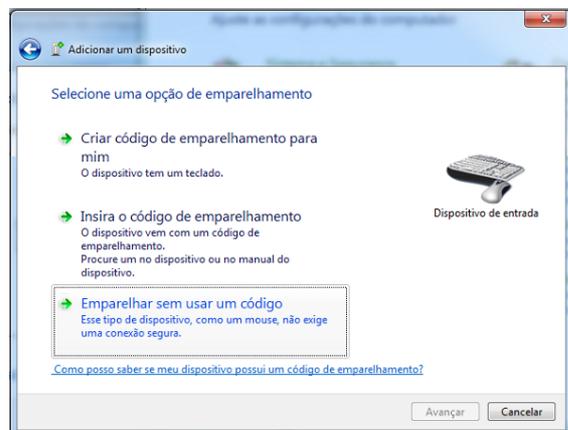


Figura 6: Procedimento necessário para emparelhar o Wiimote com Windows sem o uso de código.

Apesar da facilidade para reconhecer os controles, foram observados alguns problemas de comunicação entre Wiimotes e Windows para aplicações que necessitam utilizar mais de um controle simultaneamente.

Para estes casos é recomendável utilizar um software específico para o gerenciamento de dispositivos Bluetooth, como por exemplo, o Bluesoleil apresentado na Figura 7, que pode ser encontrado gratuitamente junto com o pacote de

softwares distribuído por alguns fabricantes de adaptadores Bluetooth.

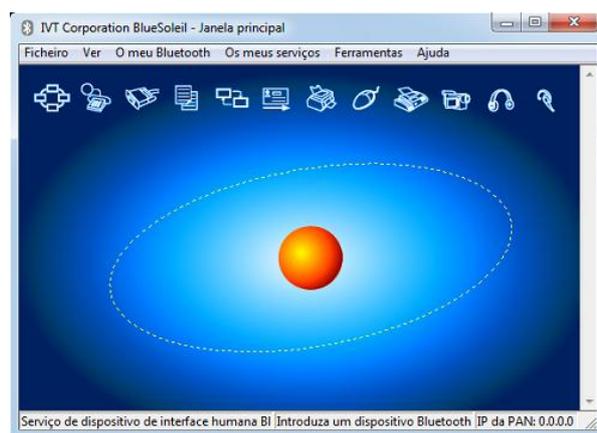


Figura 7: Interface do software BlueSoleil.

O processo de identificação do Wiimote através do Bluesoleil é muito parecido com o do Windows. Primeiro é preciso realizar uma busca por dispositivos Bluetooth na área para que o Wiimote possa ser encontrado. A segunda etapa consiste em identificar o tipo de serviço Bluetooth oferecido pelo Wiimote, que neste caso será reconhecido como um mouse. A Figura 8 apresenta a realização deste procedimento.

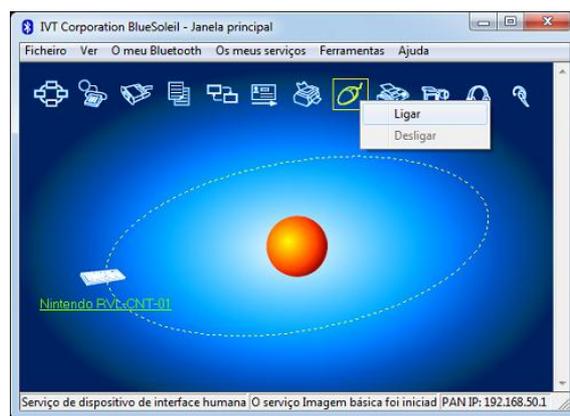


Figura 8: Conexão com um Wiimote através do Bluesoleil.

Após o reconhecimento do Wiimote via Windows ou Bluesoleil já é possível utilizar a biblioteca Wiiuse para o desenvolvimento das aplicações. A seção seguinte demonstra os recursos desta biblioteca.

4.3 Programando com a Biblioteca Wiiuse

A biblioteca Wiiuse é composta por uma série de funções e estruturas destinadas ao armazenamento de dados e captura de eventos gerados pelo Wiimote. Neste tutorial, serão abordados os principais recursos desta biblioteca, entretanto, a documentação completa da mesma pode ser obtida através da página do projeto.

O primeiro passo para a programação de interação baseada no uso de Wiimotes consiste em criar referências para cada um dos dispositivos conectados

ao computador. Para tanto, a biblioteca Wiiuse define uma estrutura denominada `wiimote` que contém as propriedades do dispositivo que poderão ser exploradas via código.

Como a biblioteca foi desenvolvida para oferecer suporte a diversos controles ao mesmo tempo, muitas funções oferecidas pela Wiiuse trabalham com um vetor de ponteiros para a estrutura `wiimote`, como é o caso da função `wiiuse_init()` demonstrada no código a seguir.

```
wiimote** vrWiimotes = NULL;
vrWiimotes = wiiuse_init(QTDCONTROLES);
```

Como é possível observar através deste fragmento de código, a função responsável pela inicialização do vetor de ponteiros do tipo `wiimote` recebe como parâmetro um valor referente à quantidade de controles que serão referenciados. Neste caso, foram criadas duas referências para Wiimotes, representadas pelos ponteiros `vrWiimotes[0]` e `vrWiimotes[1]`. Nesta etapa, ainda não há a conexão com o dispositivo, apenas a alocação da memória necessária para as estruturas.

O próximo passo consiste na preparação das estruturas para o estabelecimento da conexão. A biblioteca Wiiuse permite que o desenvolvedor especifique qual será o mecanismo de conexão utilizado para o registro dos controles ao sistema.

Através deste recurso, o programador pode especificar se o gerenciamento das conexões será feito pelo próprio Windows ou através de um software auxiliar como o Bluesoleil. O código a seguir demonstra a realização deste procedimento.

```
wiiuse_set_bluetooth_stack(vrWiimotes, QTDCONTROLES, WIIUSE_STACK_BLUESOLEIL);
```

No exemplo de código anterior, definiu-se o software Bluesoleil como o mecanismo utilizado para o registro dos controles ao sistema. Outra opção seria utilizar a constante `WIIUSE_STACK_MS`, para definir o Windows ou outros gerenciadores de conexão.

Após as etapas de alocação de memória às estruturas de referência e configuração do modo de conexão Bluetooth, já é possível solicitar a conexão com os controles. Para realizar este procedimento é preciso primeiramente verificar quantos controles estão disponíveis para a conexão. Este procedimento é realizado através da função `wiiuse_find()`.

Para cada Wiimote encontrado é preciso então usar a função `wiiuse_connect()` para que a conexão seja estabelecida. O exemplo de código a seguir demonstra a realização da busca e conexão com os controles.

```
1: int iWiimotes =
wiiuse_find(vrWiimotes, QTDCONTROLES,
TEMPOBUSCA);
2: if (iWiimotes)
3: {
4:   int conectados =
wiiuse_connect(vrWiimotes, iWiimotes);
5: }
```

A função de busca recebe como parâmetro o vetor de estruturas, a quantidade total de controles que se deseja manter conexão e o tempo máximo (em segundos) de busca por tais controles. Ao finalizar a execução, a função de busca retorna o número total de Wiimotes encontrados e disponíveis para conexão.

Se um ou mais controles forem encontrados então a função `wiiuse_connect()` pode ser acionada para o estabelecimento da conexão (linha 4). Esta função também retorna um valor inteiro para informar o número de Wiimotes conectados com sucesso.

A partir deste momento já é possível realizar a troca de dados com os controles. Uma das ações frequentemente utilizadas após o estabelecimento da conexão consiste em acender no controle o LED correspondente ao identificador do mesmo no vetor de estruturas como mostra o fragmento de código a seguir.

```
wiiuse_set_leds(vrWiimotes[0],
                WIIMOTE_LED_1);

wiiuse_set_leds(vrWiimotes[1],
                WIIMOTE_LED_2);
```

Este recurso é interessante para ajudar o jogador a identificar o seu personagem na tela, no caso de jogos multiplayer. Também é possível utilizar a constante `WIIMOTE_LED_NONE` para desligar os LEDs ligados previamente.

Vale a pena salientar que ao final da execução do programa é recomendável encerrar as conexões previamente estabelecidas. Esta operação pode ser realizada através da função `wiiuse_cleanup()` que recebe os mesmos parâmetros utilizados com a função de conexão.

Além disso, sempre que a conexão estiver ativa, é preciso então verificar se algum evento foi gerado pelos dispositivos. Para tanto, a biblioteca Wiiuse trabalha com um sistema de *polling* não bloqueante que necessita ser constantemente verificado para que eventos recentes possam ser tratados. Esta tarefa é realizada pela função `wiiuse_poll()`.

Ao final de sua execução, a função de *polling* retorna um valor inteiro que representa o total de dispositivos Wiimote que possuem novos eventos. O exemplo de código a seguir demonstra como utilizar a biblioteca Wiiuse para verificar eventos gerados pelo Wiimote.

```

1: if ( wiiuse_poll(vrWiimotes,
             QTDCONTROLES) )
2: {
3:   for(int iIndice=0;
           iIndice<QTDCONTROLES; iIndice++)
4:   {
5:     if (vrWiimotes[iIndice] &&
           vrWiimotes[iIndice]->event)
6:     {
7:       //Identificar e tratar evento
8:     }
9:   }

```

Por ser uma função que necessita ser acionada constantemente, o melhor local para realizar tal procedimento consiste no laço principal da aplicação ou internamente a uma função chamada pelo mesmo. Sendo assim, sempre que a função de *poll* é chamada, o estado da variável *event* pertencente à estrutura *wiimote* é alterado. Ao sofrer uma alteração a variável de evento pode assumir algum dos seguintes estados:

- **WIIUSE_NONE**: nenhum evento foi gerado no período verificado.
- **WIIUSE_EVENT**: significa que um evento de botão, orientação ou movimentação ocorreu.
- **WIIUSE_DISCONNECT**: o dispositivo foi desconectado.
- **WIIUSE_NUNCHUK_INSERTED**: o acessório Nunchuk foi conectado ao dispositivo.

A menos que o valor de *event* seja igual à **WIIUSE_NONE**, será necessário tomar alguma atitude em resposta ao tipo de evento gerado. Eventos do tipo **WIIUSE_EVENT** exigem uma verificação detalhada no estado dos botões, estruturas de orientação e posição como mostra o fragmento de código a seguir.

```

1: case WIIUSE_EVENT:
2: {
3:   if (IS_PRESSED(prWiiMotes[iIndex],
                   WIIMOTE_BUTTON_TWO))
4:   {
5:     bAceleracao = TRUE;
6:   }
7:   if (IS_RELEASED(prWiiMotes[iIndex],
                    WIIMOTE_BUTTON_TWO))
8:   {
9:     bAceleracao = FALSE;
10:  }
11:  if ( WIIUSE_USING_ACC
        (prWiiMotes[iIndex]) )
12:  {
13:    fIncAngulo =
        prWiiMotes[iIndex]
        ->orient.pitch/100;
14:  }
15:  break;
16: }

```

Neste exemplo, após o evento ter sido detectado, utiliza-se uma série de macros definidas pela biblioteca para verificar se existem botões pressionados ou liberados recentemente, ou mesmo se o acelerômetro está em uso para capturar o ângulo de rotação do tipo *pitch*. Além disso, é possível também acessar dados de diversas estruturas a fim de verificar se os seus atributos sofreram alguma alteração.

Dentre as principais estruturas a serem testadas durante um evento do tipo **WIIUSE_EVENT** é possível citar:

- **orient**: armazena dados referentes a orientação do acelerômetro representada pelos campos *roll*, *pitch* e *yaw*. Os dois primeiros variam de -180 graus a 180 graus. O Campo *yaw* varia de -26 a 26 graus, porém para que esta dado esteja disponível é preciso que o rastreamento via infravermelho esteja habilitado.
- **gforce**: contém a força da gravidade exercida sobre os eixos x, y e z. O valor varia de -3 a +3 unidades em ponto flutuante.
- **ir**: estrutura responsável pelo sistema de posicionamento atuando com base no infravermelho. Cada Wiimote pode armazenar dados de posicionamento (x,y,z) para um espaço de coordenadas virtual. Além disso, também é possível obter a posição real do Wiimote no plano de coordenadas (x,y) para até quatro pontos de referência representados pela estrutura `dot` [4].

Através dos valores fornecidos por estas estruturas é possível captar movimentos do corpo do jogador ou equipamentos utilizados pelos mesmos, como aparelhos destinados a exercícios de pernas, quadril, braços e ombros.

A biblioteca *Wiiuse* disponibiliza também diversas funções que permitem verificar ou alterar o estado do controle, além de configurar a sensibilidade do mesmo. Muitas operações de configuração dependem de uma verificação prévia a respeito do estado atual de alguns recursos do dispositivo. As funções listadas a seguir retornam informações importantes sobre os recursos do Wiimote.

- **WIIUSE_USING_ACC(wiimote*)**: retorna 0 ou 1 para indicar se o acelerômetro está ou não habilitado no momento.
- **WIIUSE_USING_EXP(wiimote*)**: retorna 1 se existe algum acessório extra conectado ao Wiimote.
- **WIIUSE_USING_IR(wiimote*)**: retorna 1 se o infravermelho esta habilitado no momento.

- **WIIUSE_USING_SPEAKER(wiimote*)**: retorna 1 se o alto-falante estiver habilitado.

Com base no estado atual do dispositivo é possível então acionar as funções que permitem configurar recursos e calibrar o dispositivo. Algumas destas funções são:

- **wiimote_set_orient_threshold(wiimote*, float)**: permite configurar um determinado valor em graus que será considerado para que uma modificação na orientação do controle seja considerada um novo evento ou não.
- **wiimote_set_accel_threshold(wiimote*, float)**: permite configurar a sensibilidade referente à geração de novos eventos para a aceleração.

5. Etapas para a Criação de um Jogo de Reabilitação com Realidade Virtual e Suporte ao Wiimote

O desenvolvimento de um jogo destinado à reabilitação inicia pela observação dos exercícios realizados pelos pacientes de um determinado grupo ou tipo de patologia.

Posteriormente, é preciso então em adaptar um ambiente virtual aos mecanismos utilizados pelo paciente ou mesmo criar uma situação que simule o exercício em questão.

Após a ideia do jogo ter sido definida o processo de desenvolvimento é dividido em etapas compostas pela modelagem dos objetos pertencentes ao mundo virtual e a programação. Estas duas atividades podem acontecer em paralelo, tendo em vista que os programadores podem utilizar objetos temporários que serão substituídos na versão final.

Outro ponto importante que deve ser ressaltado é a escolha da tecnologia ou motor gráfico que será utilizado na programação. Um dos pontos importantes a considerar no momento da escolha diz respeito aos formatos de objetos e imagens suportados pelo mesmo. É preciso ter certeza que o motor gráfico seja capaz de trabalhar com os objetos gerados pelas ferramentas utilizadas pelos modeladores.

No caso de jogos cuja interação será realizada através do Wiimote, o motor gráfico deverá permitir a vinculação da biblioteca Wiiuse bem como customizações ou adaptações necessárias ao desenvolvimento de jogos voltados à reabilitação.

Dentre os motores gráficos disponíveis atualmente é possível citar, por exemplo, o Delta 3D, Ogre, OpenSceneGraph e EnCIMA [Malfatti 2009].

6. Estudos de Caso

Esta seção descreve algumas aplicações desenvolvidas com o intuito de atuar na recuperação da coordenação motora e equilíbrio.

6.1 Taxi Driver

O jogo *Taxi Driver* pertence ao projeto Fisioplay, desenvolvido pelo DJT (Desenvolvedores de Jogos do Tocantins) e vinculado ao curso de Sistemas de Informação da Fundação Universidade do Tocantins [Fisioplay 2011].

Este projeto tem como objetivo o desenvolvimento de jogos baseados em Realidade Virtual e Wiimote para o tratamento de patologias que de alguma forma comprometam o movimento dos pacientes.

Um dos primeiros aplicativos desenvolvidos no projeto Fisioplay foi o *Taxi Driver*. Neste jogo o usuário incorpora um motorista de táxi cuja tarefa consiste em transportar passageiros espalhados em uma cidade fictícia.

Para dar mais realismo à jogabilidade proposta, utilizou-se um elemento de interface tangível juntamente com o Wiimote: um volante vendido separadamente juntamente com o kit de acessórios esportivos para Wii como mostra a Figura 9.



Figura 9: Interação com o jogo *Taxi Driver* via Wiimote.

Além de ter que girar o volante para definir a direção do veículo, o jogador também interage com os botões para realizar operações como acelerar, frear e engatar marcha à ré.

O *Taxi Driver* é destinado a pacientes vítimas de patologias que enrijecem os membros de apenas um dos lados do corpo. Nestas situações o paciente tende a evitar o uso de tais membros, dando preferência para a realização de movimentos com o lado saudável. Portanto, no *Taxi Driver*, o jogador necessitará utilizar ambas as mãos, não sendo possível ignorar o lado do corpo que possui maior dificuldade de movimentação.

6.2 Helicopter

O *Helicopter* [Alankus et al 2010] é um jogo desenvolvido em parceria pelo Departamento de Engenharia e Ciência da Computação da Universidade de Washington e pelo Departamento de Engenharia de Elétrica e de Computação da Universidade da Califórnia. Voltado para reabilitação de pessoas que sofreram um Acidente Vascular Encefálico (AVE), utiliza dois Wiimotes para controlar um helicóptero, como pode ser visualizado na Figura 10.



Figura 10: Interação com o jogo *Helicopter* utilizando dois Wiimotes posicionados no braço e no antebraço.

Os movimentos necessários são simples e cognitivamente leves, utilizando a flexão do cotovelo para interagir com o jogo. O helicóptero pode voar para cima ou para baixo desviando de obstáculos e coletando tanques de gasolina para evitar a queda.

Este jogo possui a particularidade de encorajar o jogador a realizar a repetição completa do movimento mais do que qualquer outro. Isso se deve ao fato de que a dinâmica do jogo para pilotar o helicóptero consiste em um conjunto de movimentos contínuos que requer movimentação constante por parte do paciente.

Entretanto, dois aspectos devem ser levados em consideração: tempo de uso e monotonicidade. Embora o jogo tenha incorporado satisfatoriamente os movimentos da reabilitação como controle, a repetição excessiva por um longo período de tempo pode ocasionar cansaço no jogador. Além disso, como se trata de uma atividade bem rotineira, após utilizar o jogo por diversas vezes o jogador tende a não se entusiasmar da mesma maneira como antes, sendo necessário certo equilíbrio em seu uso frequente.

6.3 Wiirehabilitation

O *Wiirehabilitation* [Decker et al. 2009] é um jogo para reabilitação dos movimentos do pulso. Desenvolvido por pesquisadores da Universidade de Rutgers, o jogo também faz uso do Wiimote, mas com uma forma indireta de interação. Três fitas de velcro com um material reflexivo são amarradas ao pulso, à mão e ao antebraço do jogador, como pode ser visto na Figura 11.



Figura 11: Fitas de velcro utilizadas para captar a movimentação do pulso do jogador.

Neste caso, o Wiimote funciona passivamente, não sendo manipulado diretamente pelo jogador. Circundado por LEDs infravermelhos, o Wiimote fica parado captando os raios infravermelhos refletidos pelas fitas presas ao jogador e gera as entradas para o jogo de acordo com a posição das fitas, que é alterada com as flexões exercidas no pulso. O jogo é similar ao Pong, e o jogador é responsável por movimentar a barra que evita que a bola saia dos limites da tela.

Pacientes que necessitam realizar reabilitação para o pulso geralmente não estão conseguindo flexioná-lo completamente ou não conseguem realizar um movimento contínuo. Por isso, um dos pontos principais do jogo é medir o grau de melhora do paciente baseado no quanto este consegue movimentar o pulso adequadamente.

7. Considerações Finais

A utilização de jogos para reabilitação é uma realidade atual que vem trazendo inúmeros benefícios não só para os pacientes, que tem mais uma fonte de motivação durante as sessões repetitivas da reabilitação, como também para os hospitais, os quais podem fazer uso de equipamentos economicamente viáveis e flexíveis que podem ser usado em inúmeros e diferentes tratamentos.

O Wiimote provê um tipo de interação natural que é deveras comum em terapias de reabilitação, proporcionando uma alternativa de interface considerável para se produzir jogos nesse contexto. Usado ativamente ou passivamente, fornece dados específicos sobre os movimentos que podem ser usados para estatísticas da terapia e medir o grau de avanço e melhoria do paciente.

A Wiiuse é uma biblioteca simples e que possui uma curva de aprendizado curta, que pode ser utilizada em diversos projetos desse gênero. Os exemplos citados nos estudos de caso não necessariamente a utilizam, mas podem ser implementados facilmente a partir dela.

Por fim, esse tutorial pode ser usado como uma referência básica para o desenvolvimento de jogos para a reabilitação com a Wiiuse, embora a biblioteca não se limite a apenas esse tipo de jogo. Vale salientar que existem diversas outras aplicações nas quais a Wiiuse e o Wiimote podem ser empregados.

8. Referências

- PHEAT, C. AND GOERING, S., 2010. Programming User Interfaces Using The Nintendo Wii Remote: Tutorial Presentation. *Journal of Computing Sciences in Colleges*. Volume 25 Issue 5, May.
- WIIUSE, 2011. [online] Disponível em: <http://sourceforge.net/projects/wiiuse/>. [Acessado em 14/09/2011].
- VISUAL STUDIO 2008, 2008. [online] Disponível em: <http://www.microsoft.com/visualstudio/enus/products/2008-editions/express>. [Acessado em 14/09/2011].
- MALFATTI, S., 2009. *EnCIMA – um motor para o desenvolvimento de aplicações de realidade virtual*. Dissertação de mestrado, instituto militar de engenharia.
- DECKER, J., LI, H., LOSOWYJ, D. AND PRAKASH, V., 2009. Wiihabilitation: Rehabilitation of Wrist Flexion and Extension Using a Wiimote-Based Game System.
- ALANKUS, G., LAZAR, A., MAY, M. AND KELLEHER, C., 2010. Towards Customizable Games for Stroke Rehabilitation.
- BURDEA, G. C., 2003. Virtual rehabilitation - Benefits and Challenges. *Methods Inf Med*.
- NUDO, R., WISE, B., SIFUENTES, F. AND MILLIKEN, G., 1996. Neural substrates for the effects of rehabilitative training on motor recovery after ischemic infarct. *Science*.
- LYNCH, D., FERRARO, M., KROL, J., TRUDELL, C. M., CHRISTOS, P. AND VOLPE, B. T., 2005. Continuous passive motion improves shoulder joint integrity following stroke. *Clin Rehabil*.
- FISIOPLAY, 2011. [online] Disponível em: <http://djt.unitins.br/Fisioplay/Fisioplay.htm>. [Acessado em 14/09/2011].
- MICHAEL, D. AND CHEN, S., 2006. *Serious Games: Games that Educate, Train, and Inform*. Boston: Thomson.