UTILIZAÇÃO DE DISPOSITIVOS MÓVEIS COM ACELERÔMETRO PARA CONTROLE DE APLICAÇÕES

Rodrigo Veiga da Silva[1], Francisco Assis da Silva[2]

[1] Discente da Faculdade de Informática da UNOESTE; [2] Professor da Faculdade de Informática da UNOESTE.

RESUMO

Nos dias de hoje, observa-se um grande crescimento na necessidade de controlar aplicativos à distância fazendo uso de tecnologias que permitam a comunicação sem fio, possibilitando uma maior mobilidade ao controlador. Frente a tais necessidades, pensou-se no desenvolvimento de um aplicativo que permita a criação de um joystick virtual que seja capaz de controlar alguns aplicativos por meio de um dispositivo móvel. Dentre os estudos realizados nesse ambiente, está a manipulação dos dados do acelerômetro, o envio desses dados bem como a maneira de enviar esses dados ao Sistema Operacional Windows. Para que isso seja possível, faz-se necessário que este dispositivo disponha das tecnologias Bluetooth e Acelerômetro. O objetivo desse projeto é controlar aplicações de um computador por meio de um joystick virtual que receberá os dados provenientes de sensores de movimentos presentes no dispositivo móvel, bem como o acionamento dos botões do mesmo. Com isso, será possível controlar todas as aplicações que tenham suporte a joystick como jogos, simuladores e softwares de navegação.
Palavras-Chave: Acelerômetro, Bluetooth, Python, Dispositivos Móveis.

USE OF MOBILE DEVICES WITH ACCELEROMETERS TO CONTROL APPLICATIONS

ABSTRACT

Nowadays, there is a large growth in need to control remote applications using technology to control wireless communication, allowing greater mobility to the controller. Faced with such requirements, it was thought to develop an application that allows a virtual joystick creation capable of controlling some applications through a mobile device. It was studies the data manipulation from the accelerometer, the sending of such data and how to send data to the Windows OS. To make this possible, it is necessary that this device has accelerometer and Bluetooth technologies. The project goal is to control computer applications using a virtual joystick that will receive data from sensors of movements and buttons present in the mobile device. It will possible to control all applications that have support for joystick as games, simulators and navigation softwares.
Keywords: Accelerometer, Bluetooth, Python, Mobile Devices

INTRODUÇÃO

A evolução das tecnologias digitais traz um novo cenário à utilização do telefone celular. Apesar de atualmente reunir inúmeros recursos não mais focados em efetuar e receber chamadas, os celulares ainda são vistos apenas como telefones móveis. No entanto, a busca do ser humano em armazenar e organizar informações de forma cada vez mais rápida e eficiente faz com que os aparelhos celulares ganhem cada vez mais funcionalidades (ALVES, 2007).

Para o desenvolvimento deste projeto foram investigadas algumas tecnologias e recursos, como: o acelerômetro do celular N95 da Nokia, a linguagem utilizada para acessar os dados do acelerômetro, maneiras como fazer a conexão Bluetooth com o celular, qual API deveria ser utilizada para tal comunicação e como interpretar e executar os dados dos eixos do acelerômetro e os botões pressionados. Durante as investigações tentou-se criar um *joystick* virtual no controlador de jogos do Sistema Operacional Windows para que este se encarregasse de executar os comandos, de acordo com os dados recebidos do celular. A partir desses estudos foi construída uma aplicação contendo dois módulos, um que é executado em um aparelho celular e o outro em um computador pessoal.

Os dois módulos que compõem a aplicação se comunicam entre si utilizando *socket*, a conexão entre os módulos é Bluetooth. Essa comunicação é responsável por realizar o envio dos dados obtidos a partir do acelerômetro e dos botões deste aparelho celular, visando a interação do usuário com jogos e aplicativos.

O módulo que é executado no aparelho celular utiliza a linguagem Python S60 que permite a captura dos dados do acelerômetro e dos botões para que sejam enviados ao outro módulo (executando no computador). Este segundo módulo foi desenvolvido utilizando a linguagem Java e possibilita que os dados enviados pelo celular sejam encaminhados a um de dois tipos de controle dos jogos e aplicativos: via teclado ou via controlador de jogos.

1. ACELERÔMETRO

Os acelerômetros são sensores que medem aceleração de um objeto. A aceleração é uma medida de quão rapidamente a velocidade varia e pode ser obtida segundo uma, duas ou três direções, utilizando acelerômetros *uni*, *bi* e *tri axiais*, respectivamente (FIGUEIREDO, 2007). São classificados de acordo com a sensibilidade definida pela aceleração máxima e mínima que o acelerômetro pode perceber e a quantidade de eixos existentes no sensor, que pode variar de um a três eixos, com isso o acelerômetro pode perceber a aceleração em todas as direções do espaço (FERNANDES, 2008). Os acelerômetros não medem somente aceleração, mas também inclinação, rotação, vibração, colisão e gravidade (FIGUEIREDO, 2007).

1.1. Funcionamento do Acelerômetro

A orientação é descrita na forma dos eixos *x*, y e *z*, eles podem assumir valores inteiros no intervalo de -128 a 127 (GOMES, 2008). O que vai determinar os valores para os eixos é a posição do aparelho celular em relação a ele mesmo, tais como: com a tela voltada para frente e apontando para direita ou esquerda, para cima ou para baixo e com a tela voltada para cima ou para baixo.

O eixo *x* assume um valor positivo quando virado com a tela para frente e apontado para a direita, e um valor negativo com a tela para frente e apontado para a esquerda, como é mostrado na Figura 1.

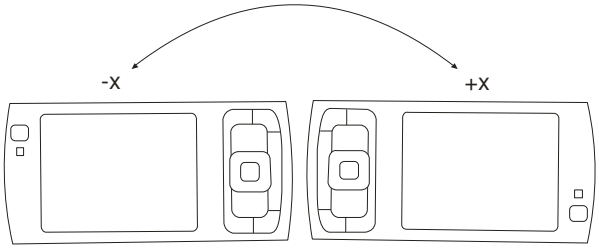

Figura 1: Posição do celular que ilustra o eixo *x* do acelerômetro.

O eixo *y* assume um valor positivo quando virado com a tela para frente e apontado para cima, e um valor negativo com a tela para frente e apontado para baixo, como mostrado na Figura 2.

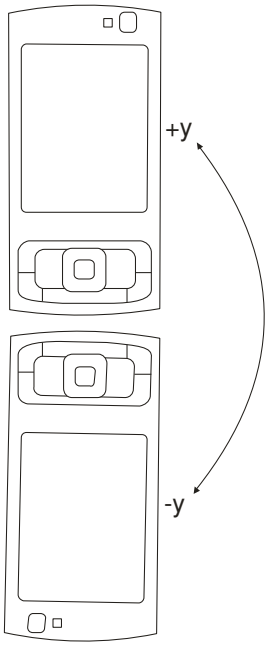

Figura 2: Posição do celular que ilustra o eixo *y* do acelerômetro.

Já o eixo *z* assume um valor positivo quando a tela é virada para baixo, e um valor negativo quando a tela está virada para cima, como mostra a Figura 3.

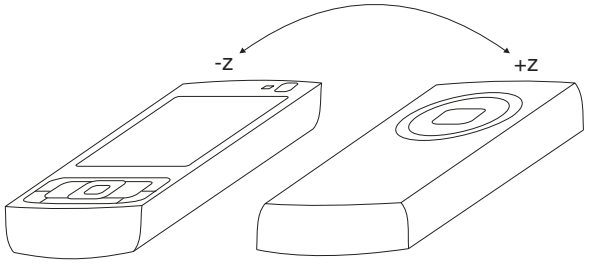

Figura 3: Posição do celular que ilustra o eixo *z* do acelerômetro.

1.2. Tipos de Acelerômetro

Existem vários tipos de acelerômetro, cada um com suas características, vantagens e desvantagens. Os acelerômetros mecânicos são os mais comuns, mas esses acelerômetros estão sendo substituídos por acelerômetros eletromecânicos, um exemplo é o acelerômetro MENS (FIGUEIREDO, 2007), mostrado na Figura 4.

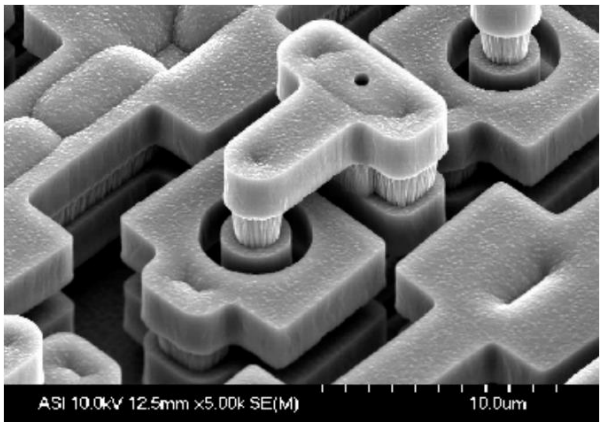

Figura 4: Foto do acelerômetro MENS.
Fonte: Fernandes (2008).

Figueiredo (2007) relaciona alguns tipos de acelerômetro:

- Acelerômetros Capacitivos: são componentes que possuem um condensador entre a massa e a estrutura de suporte e são sensíveis às mudanças na capacidade entre eles;
- Acelerômetros Piezoelétricos: utilizam-se de cristais sintéticos ou naturais que conduzem carga quando são comprimidos. Quando o cristal vibra, ele fica submetido a forças de atração, gerando carga;
- Acelerômetros Piezoresistivos: ao invés de gerar carga, como o Piezoelétrico, ele gera uma resistência;
- Acelerômetros Eletromecânicos: podem medir a aceleração em um, dois ou três eixos. Esse sensor é baseado em uma estrutura de silicone, com membranas em forma de pentes fixos ou móveis. A aceleração é calculada a partir do

deslocamento dos elementos móveis que estão associados aos eixos.

2. BLUETOOTH

Bluetooth é uma tecnologia que utiliza transmissão via rádio por meio de ondas curtas, aproximadamente 9 a 10 metros, e que tem como objetivo principal, interligar qualquer dispositivo de computação, comunicação e eletrônico, independente de marca, modelo ou sistema utilizado (BLUETOOTH, 2009).

Esta tecnologia possibilita que dispositivos móveis atendam a uma rede local sem que estejam conectados por cabos, respeitando apenas o limite físico das ondas curtas. Mas, a tecnologia Bluetooth não tem a simples finalidade de eliminar cabos de conexão de uma rede, com seu avanço também é possível fazer, por exemplo, um telefone celular se conectar, sincronizar ou ainda controlar dispositivos eletrônicos como televisão, sistemas de alarme, computadores, entre outros, dentro de casa ou escritório (BLUETOOTH, 2009).

2.1. Surgimento do Bluetooth

Miller (2001) descreve que em 1994, chefiado pelo Dr. Sven Mattison, da Suécia, e o Dr. Jaap Haartsenm, da Holanda, iniciaria uma pesquisa financiada pela Ericsson, a fim de eliminar os fios utilizados entre celulares e acessórios, principalmente os fones de ouvido, para que pudesse utilizar o celular sem tirá-lo do bolso ou da cintura.

Ederli (2003), conta que o nome Bluetooth é homenagem a um rei dinamarquês chamado de Harald Blatand, em inglês, Harold Bluetooth (Haroldo Dente-Azul), ele é conhecido por unificar tribos dinamarquesas, e por esse motivo o nome Bluetooth foi escolhido, pois a tecnologia procura unir dispositivos de diferentes tipos.

2.2. Bluetooth SIG – Grupo de interesse Especial

Após o surgimento do Bluetooth, outras empresas começaram a mostrar interesse nesse projeto e logo passaram a fornecer apoio e, em 1998, foi criado o SIG (*Special Interest Group*), formado pelas empresas Ericsson, Nokia, IBM, Toshiba e Intel. Com empresas de diferentes ramos permitiu-se o desenvolvimento de padrões garantindo o uso e a interoperabilidade da tecnologia nos mais variados dispositivos (MILLER, 2001).

2.3. Vantagens e Desvantagens

Miller (2001) afirma que alguns pontos devem ser levados em consideração, tais como, conveniência de conexão, pois assim que um dispositivo Bluetooth encontra outro, os dois dispositivos tentam fazer uma sincronia, automaticamente. Dentro desta conveniência destacam-se ainda a substituição de cabos e a criação de redes *Ad-hoc* pessoais, interligando os principais dispositivos de computação e comunicação. Ainda com vantagem, todos esses serviços são relativamente de baixo custo de implementação, pois utilizam um rádio transmissor em um micro *chip*, e o baixo gasto de energia perfeito para dispositivos portáteis. Entre as desvantagens destacam-se o curto alcance para a conexão, cerca de 10 metros, e a quantidade de dispositivos interligados ao mesmo tempo, um número limitado comparando-se a outras tecnologias, inclusive as próprias redes cabeadas.

2.4. Piconet e Scatternet

A integração de dispositivos dentro de um mesmo segmento, formam o que se chama de Piconet, uma arquitetura mais simples onde cada dispositivo assume um papel, mestre ou escravo, sendo o mestre o dispositivo que controla seus respectivos escravos. O número máximo de dispositivos que podem formar um Piconet é oito,

existindo apenas um mestre. Como esse número é muito pequeno, existe o Scatternet que é a técnica que permite sobrepor vários Piconets, aumentando os pontos de comunicação (MILLER, 2001).

2.5. Conexões Bluetooth

Existem algumas formas de uso de conexão Bluetooth. A Figura 5 ilustra algumas formas de conexão Bluetooth: *Data Access Point*, *Cable Replacement* e *Peer-to-peer Networking*.

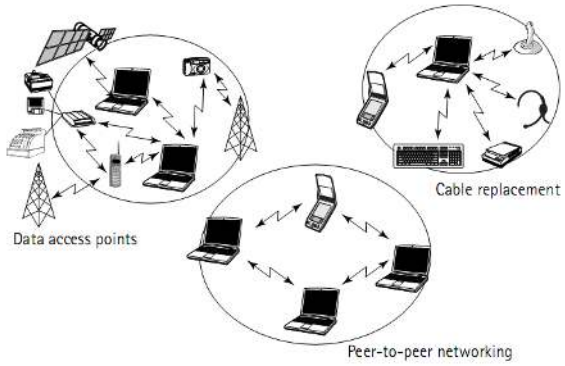

Figura 5: Conexões Bluetooth.

As conexões ilustradas na Figura 5:

- *Data Access Point*: permite que dispositivos que tenham Bluetooth se conectem entre si e com a Internet.
- *Cable Replacement*: permite a substituição de cabos por um *chip* Bluetooth que transmite informações numa frequência especial de rádio a um chip receptor Bluetooth.
- *Peer-to-peer Networking*: uma Rede *Ad-hoc* que permite que os dispositivos pessoais troquem informações e sincronizem um ao outro automaticamente. Como exemplo, tem-se atualizações de um calendário em um PDA para um calendário de um computador.

3. SOCKET

Processos que rodam em máquinas diferentes se comunicam uns com os outros enviando mensagens por meio de *sockets*. Um *socket* é a interface entre a camada de aplicação e a de transporte dentro de uma máquina, é também denominado como uma interface de programação de aplicação entre a aplicação e a rede. Para que dois ou mais processos se comuniquem, eles enviam e recebem mensagens de uma rede através de um *socket*. Por exemplo, quando um processo deseja enviar mensagens a outro processo, ele empurra a mensagem na rede através do *socket*. Esse processo emissor admite que exista uma infra-estrutura de transporte do outro lado que transportará a mensagem pela rede até o *socket* do processo destinatário. Ao chegar ao destinatário, a mensagem passa através do *socket* do processo receptor que, então, executa alguma operação sobre a mensagem. A Figura 6 ilustra o funcionamento de um *socket* (KUROSE; ROSS, 2006).

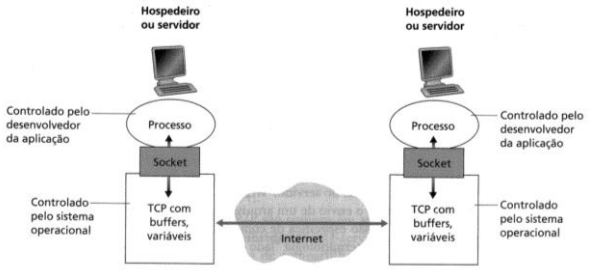

Figura 6: Funcionamento de um socket.

4. IMPLEMENTAÇÃO DA APLICAÇÃO

A aplicação desenvolvida foi divida em dois módulos, o primeiro será executado em um aparelho celular e o segundo em um computador como mostra o diagrama da Figura 7. Nas subseções seguintes estão descritas as tecnologias envolvidas no desenvolvimento da aplicação deste trabalho bem como seus módulos.

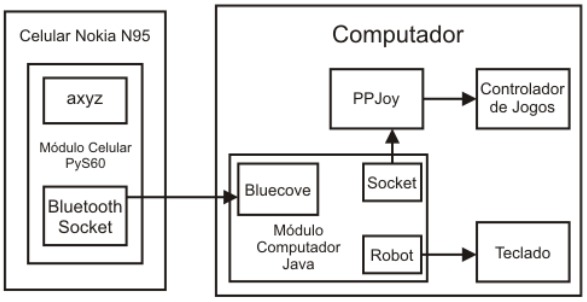

Figura 7: Diagrama do Projeto desenvolvido.

4.1. Java

Mählmann (2009), conta que em 1991, na Sun Microsystems, foi iniciado o *Green Project*.

O líder do projeto era James Gosling. O objetivo do projeto não era a criação de uma nova linguagem de programação, mas antecipar e planejar a "próxima onda" do mundo digital. Eles acreditavam que, em algum tempo, haveria uma convergência dos computadores com os equipamentos e eletrodomésticos comumente usados pelas pessoas no seu dia-a-dia. Esse grupo, formado por 13 pessoas, trabalhou arduamente durante 18 meses sem se comunicar com a Sun. Após esse período, James Gosling especificou uma nova linguagem de programação e decidiu batizá-la de "Oak".

A equipe achava que uma boa idéia seria controlar televisões e vídeo por meio de um equipamento, mas infelizmente era muito cedo para que pudessem viabilizar o negócio, era algo muito visionário e estava muito longe da capacidade do setor de televisão.

Em 1995, a Internet encontrava-se pronta para lidar com a tecnologia Java. Com isso, o navegador Netscape Navigator passou incorporar esta tecnologia.

Por ser versátil, eficiente, com plataforma portátil e segura, Java torna-se uma tecnologia ideal para rede de computadores, estando presente em mais de dois bilhões e meios de dispositivos:

- PCs;
- Telefones celulares e outros dispositivos portáteis;
- Smart cards;
- Além de set-top boxes, impressoras, webcams, jogos, sistemas de navegação para automóveis, terminais lotéricos, dispositivos médicos, estações de pagamento de estacionamento etc.

Essa linguagem foi utilizada na implementação do módulo do computador que é mostrado no diagrama da Figura 7.

4.1.1. API BlueCove

BlueCove é uma biblioteca Java para Bluetooth que é suportada pelo Mac OS X, WIDCOMM, BlueSoleil Bluetooth e Windows Móbile (BLUECOVE TEAM, 2009). Essa API foi utilizada para realizar a comunicação Bluetooth entre os módulos da aplicação.

4.1.2. Classe Robot

Essa classe é usada para gerar eventos de entrada para o sistema nativo para fins de testes automatizados, demonstrações de auto-execução, e outras aplicações onde o controle do mouse e do teclado é necessário. O objetivo primário da Robot é facilitar a realização de testes automatizados de implementações da plataforma Java (CLASS ROBOT, 2010). A classe Robot foi utilizada no módulo do computador para executar eventos do teclado.

4.2. Python

Python é uma poderosa linguagem de programação e fácil de aprender que possui estruturas de dados eficientes e adota uma abordagem simples e efetiva para a programação orientada a objetos. Tem uma sintaxe elegante e tipagem dinâmica, o que torna Python ideal para desenvolver aplicações de forma rápida em várias áreas e em muitas plataformas (VUN ROSSUM; DRAKE, 2009).

Há uma versão da linguagem Python para o Symbian OS, essa versão é o PyS60. Essa versão traz o poder e a produtividade da linguagem Python para a plataforma Symbian S60 3rd *edition* disponível no modelo N95 da Nokia. As ferramentas permitem o desenvolvimento rápido de aplicações e protótipos escritos em Python (PYTHON FOR S60, 2009).

O PyS60 possui a biblioteca axyz que é utilizada para capturar os eixos *x*, *y* e *z* do acelerômetro (GOMES, 2008). A Figura 8 ilustra um exemplo de captura dos dados do acelerômetro.

```
import e32
import axyz
def printout(x,y,z):
   print "x: %i y: %i z: %i"%(x,y,z)
axyz.connect(printout)
app_lock = e32.Ao_lock()
app_lock.wait()
axyz.disconnect()
```

Figura 8: Exemplo de código Python para capturar os valores do acelerômetro.

4.3. PPJoy

PPJoy é um *software* que virtualiza *drivers* de dispositivos Windows para *joysticks* digitais (por exemplo, o antigo Atari, Spectravideo, etc), NES, SNES, Playstation e Sega conectados à porta paralela (WESTHUYSEN, 2009).

4.4. Módulo do Celular

O módulo que é executado no celular, foi desenvolvido utilizando a linguagem de programação Python S60, que é uma linguagem específica para o SymbianOS – Sistema Operacional presente vários celulares, principalmente os da Nokia. Esse módulo é responsável por capturar os eixos *x*, *y* e *z* do acelerômetro e os botões pressionados e enviá-los ao módulo em execução no computador. Para enviar os dados, o programa utiliza o padrão de comunicação que é formado por uma cadeia de caracteres contendo o caractere 'D' que indica o início dos dados, seguido dos valores dos eixos *x*, *y* e *z* separados por vírgula, o nome do botão que foi pressionado no aparelho celular e o caractere '?' que indica o final dos dados, como ilustrado na Figura 9. Para que seja possível estabelecer a comunicação entre esses dois módulos, é utilizado um *socket* por meio de uma conexão Bluetooth.

```
Dx,y,z,botao?
```

Figura 9: Padrão definido para a comunicação entre os módulos.

4.5. Módulo do Computador

O módulo do computador foi desenvolvido utilizando a linguagem Java (subseção 4.1). Esse módulo é responsável por receber os dados do acelerômetro e do botão que foi pressionado, separá-los e enviá-los a uma função que direcionará esses dados, de acordo com as configurações do usuário, a dois tipos diferentes de execução dos comandos: via teclado ou via controlador de jogos. A comunicação estabelecida com o módulo do celular é feita através de um *socket* e de uma conexão Bluetooth (seções 2 e 3).

Nesse módulo é possível configurar as funções que os botões do celular irão executar como demonstra a Figura 10. Essas funções são salvas em um arquivo de configuração para viabilizar a recuperação destas em futuras execuções do módulo. Com isso, o usuário não precisará configurar os botões toda vez que o módulo for inicializado.

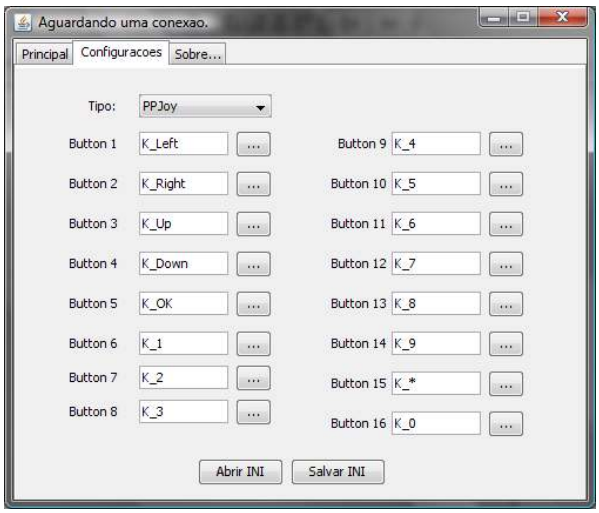

Figura 10: Tela de configuração para os botões do celular.

4.5.1. Via teclado

Essa função recebe os dados dos eixos *x*, *y* e *z* e do botão que foi pressionado no celular. Dos valores dos eixos que são enviados, são apenas utilizados os eixos *y* e *z*; o eixo *y* define o direcional esquerda-direita e o eixo *z* define o direcional cima-baixo correspondentes aos botões das setas do teclado (teclas direcionais). Os valores desses eixos quando estão próximo de 0 (zero), não representam nenhuma direção; quando o valor de *y* se aproxima de -60, representa o pressionamento da tecla direcional da esquerda; e quando se aproxima do valor 60, representa o pressionamento da tecla direcional da direita; quando o valor *z* se aproxima de -60, representa o pressionamento da tecla direcional para cima e quando assume o valor aproximado de 60, representa o pressionamento da tecla direcional para baixo. Se algum botão do celular for pressionado, é realizada uma busca no arquivo de configuração para saber qual função está configurada para ser realizada. Para que seja possível executar a função da tecla, são utilizadas funções disponíveis na classe Robot (subseção 4.1.2).

4.5.2. Via Controlador de Jogos

Essa função recebe os dados dos eixos *x*, *y* e *z* e do botão que foi pressionado no celular. Se algum botão do celular for pressionado, é realizada uma busca no arquivo de configuração para saber qual função está configurada para ser realizada. A partir desses dados é formado um vetor contendo 16 *bytes*, os quatro primeiros *bytes* representam o valor do eixo *x*, os quatro segundos o eixo *y*, os próximos quatro o eixo *z*. Os últimos quatro *bytes* representam os 32 botões possíveis onde cada *bit* equivale a um botão, quando o *bit* assume valor 1 indica que o botão está pressionado, quando assume o valor 0 indica que o botão não está pressionado. A Figura 11 mostra o padrão do protocolo de comunicação com o PPJoy formado por um vetor de bytes. Esse vetor é enviado ao PPJoy via *socket*.

```
|4B|4B|4B|32b|
|x|y|z|botões|
```

Figura 11: Protocolo de comunicação com o PPJoy.

4.6. Discussão dos resultados obtidos

Foram realizados diversos testes com jogos de corrida, simulador de aeromodelo e jogos mais simples que podem ser controlados apenas com o teclado do computador. Observou-se um bom comportamento de controle dos jogos mais simples utilizando as teclas do celular e o acelerômetro (movimento do mesmo em torno dos eixos do aparelho). Pode-se comparar o bom desempenho da aplicação (*joystick* com o uso do celular) com *joysticks* reais que não sejam específicos para determinados jogos. Estes *joysticks* seriam, por exemplo, sistemas compostos por volante e acelerador para jogos de corrida e rádio transmissor para simuladores de aeromodelos. Estes demandam um maior número de controles e uma precisão de controle.

CONSIDERAÇÕES FINAIS

Este trabalho foi desenvolvido com o intuito de demonstrar que os aparelhos celulares estão evoluindo tecnologicamente, e uso destes está cada vez mais amplo, não se concentrando apenas na sua característica original a qual foi concebido. Cada vez mais estão sendo agregadas novas funcionalidades, podendo ser utilizados não apenas para efetuar e receber chamadas telefônicas, enviar e receber mensagens, mas também para substituir outros aparelhos eletrônicos como, por exemplo, computadores, *joysticks*, entre outros.

Algumas modificações podem ser desenvolvidas para aprimorar este projeto. Uma delas seria construir uma interface de configuração das teclas do aparelho o celular nele próprio em substituição a interface atual que está no módulo do computador. Outra melhoria seria não utilizar o PPJoy (subseção 4.3) para o envio dos dados provenientes do aparelho celular para o controlador de jogos do Sistema Operacional, e sim utilizar diretamente as API's do Windows.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, Jucelia. **Tecnologia celular: uma convergência de mídias para a aproximação de públicos**. Intercom, Santos, Setembro 2007.

BlueCove Team. **BlueCove**. Disponível em: <http://bluecove.org/>. Acessado em: 12 ago. 2009.

Bluetooth.org. **The Official Bluetooth® Technology Info Site**. Disponível em: <http://www.bluetooth.org/>. Acessado em: 05 set. 2009.

Class Robot. Disponível em: <http://java.sun.com/j2se/1.3/docs/api/java/awt/Robot.html>. Acessado em: 22 fev. 2010.

EDERLI, Leila Lie. **Bluetooth.** Monografia, FATEC, Ourinhos, 2003.

FERNANDES, Marcos Júnior. **Sistema de captura de movimentos baseado em acelerômetro.** Monografia, Universidade Luterana do Brasil, Julho 2008.

FIGUEIREDO, Ligia J.; GAFANIZ, Ana R.; LOPES, Gustavo S.; PEREIRA, Rúben. **Aplicações de Acelerômetro**. Monografia, Lisboa, Dezembro 2007.

GOMES, Otávio Basso. **Programação com o sensor de movimento do celular Nokia N95.** PUCRS, 2008.

KUROSE, James F; ROSS, Keith W. **Redes de Computadores e a Internet: Uma abordagem Top-down.** – 3. ed. – São Paulo: Pearson Addison Wesley, 2006.

MÄHLMANN, Luiz Gustavo. **Java, XML e RSS**. Disponível em: <http://guaiba.ulbra.tche.br/mahlmann/tni2009_2/aula11.ppt>. Acessado em: 12 ago. 2009.

MILLER, Michael. **Descobrindo Bluetooth.** Rio de Janeiro: Ed. Campus, 2001.

Python for S60. Disponível em: <http://opensource.nokia.com/projects/pythonfors60/>. Acessado em: 12 ago. 2009.

VUN ROSSUM, Guido; DRAKE, Fred L. **An Introduction to Python.** Disponível em: <http://www.network-theory.co.uk/docs/pytut/>. Acessado em: 12 ago. 2009.

WESTHUYSEN, Deon van der. **PPJoy**. Disponível em: <http://ppjoy.bossstation.dnsalias.org/>. Acessado em: 05 set. 2009.