Projeto de um Console de Jogos Multiplataforma com Raspberry Pi

Marco Antonio Furlan de Souza[1], Everson Denis[1], João Carlos Lopes Fernandes[1.]

[1] Escola de Engenharia Mauá – Instituto Mauá de Tecnologia (IMT)
São Caetano do Sul – São Paulo – Brasil

{marco.furlan}@maua.br, everson@maua.br, jlopesf@maua.br

Abstract. *This paper describes the experience gained from the development of multiplatform games using the Raspberry PI board. The games were developed with the Python programming language and the Pygame library. Versions have been built to support both the interaction between participants of the game on the same machine as well in separate machines, through a data network. Finally, this paper also describes the application of this experience in the classroom, in which students of the first year of an undergraduate degree in Computer Engineering modified the structure of the game to create their own games, with the objective of being introduced to Computer Engineering.*

Resumo. *Este trabalho descreve a experiência obtida com o desenvolvimento de jogos multiplataforma utilizando a placa Raspberry PI. Os jogos foram desenvolvidos com a linguagem de programação Python e a biblioteca Pygame. Foram construídas versões para suportar tanto a interação entre os participantes do jogo em uma mesma máquina quanto em máquinas distintas, em uma rede de dados. Por fim, apresenta-se também neste trabalho a aplicação desta experiência em sala de aula, na qual alunos do primeiro ano de um curso de graduação em Engenharia de Computação modificaram a estrutura do jogo de modo a criar os seus próprios jogos, com objetivo de serem introduzidos no curso de Engenharia de Computação.*

1. Introdução

A motivação para o desenvolvimento do console de jogos multiplataforma, descrito neste artigo, foi o engajamento dos três autores na disciplina de Introdução à Engenharia da primeira série do curso de Engenharia de Computação da Escola de Engenharia Mauá. Esta disciplina tem como objetivo proporcionar ao aluno da primeira série o contato com atividades práticas e problemas de diversas áreas da Engenharia. Uma parte da disciplina deve propor um projeto ao aluno de modo que ele consiga vivenciar situações próximas a da realidade de um engenheiro que, no caso, é de um Engenheiro de Computação.

O primeiro desafio encontrado foi determinar um tema apropriado que capturasse a atenção do aluno interessado em Engenharia da Computação. O tema escolhido foi "jogos para o computador" – criação de um "console de jogos". A justificativa de tal tema é que, além do caráter lúdico que atrai os estudantes da primeira série, o projeto deveria envolver tanto uma parte de *software* quanto de *hardware*, de forma a prover uma visão maior da área. O segundo desafio foi escolher o que utilizar no projeto, em

termos materiais e conceituais, aproveitando os conhecimentos que um aluno típico da primeira série possui para transmitir de modo gradual os conhecimentos de Computação necessários, de forma que o aluno esteja estimulado no desenvolvimento do projeto, sem sofrer uma sobrecarga cognitiva.

Com o surgimento no mercado da placa *Raspberry Pi* em 2012, parte do segundo desafio foi resolvido [RaspberryPi 2013]. O *Raspberry Pi* é um computador completo, apresentando os componentes principais das arquiteturas que os Engenheiros de Computação devem dominar, mas, por seguir um projeto simples e compacto, é mais simples de explicar. Desse modo, quanto ao projeto do console de jogos, restou apenas a decisão de qual sistema operacional utilizar e aplicação do jogo a ser implementada.

O sistema operacional escolhido foi o Linux com a distribuição *Raspbian* [Raspbian, 2013], que é considerada uma distribuição Linux de propósito geral baseada no Debian [Debian 2013] e otimizada para o *Raspberry Pi*. Essa distribuição possui diversos aplicativos de uso geral e de desenvolvimento pré-instalados. Para o desenvolvimento do *software*, optou-se pela linguagem de programação Python [Rossum e Junior 2013], uma linguagem multiplataforma que possui uma sintaxe projetada para ser intuitiva, permitindo que estudantes iniciantes em Computação comecem a escrever de forma rápida os códigos para uma variedade de aplicações. Para a criação de jogos, optou-se pela utilização da biblioteca *Pygame* [para ser utilizada com Python], que permite a criação com relativa facilidade de jogos e aplicações multimídia [Pygame 2013].

Por fim, foi decidido o tema do jogo. Ao invés de se propor temas de escolha livre pelos alunos, em que se correria o risco de surgir projetos infactíveis, preferiu-se determinar um tema simples para ser explicado e também interessante, de modo que o aluno pudesse alterá-lo de acordo com suas preferências. Foi escolhido o tema "Pong", um jogo de computador clássico, escolhido pela sua simplicidade e possibilidade de variações [Wikipedia 2013]. A partir deste tema, foram desenvolvidas diversas variações do jogo "Pong" básico que foram explicadas e disponibilizadas aos alunos durante as aulas: uma versão simples para ser jogada com teclado, uma versão com suporte a *joystick* e também uma versão em rede, onde os dois jogadores poderiam ficar separados (em locais diferentes).

Este artigo descreve a experiência obtida no desenvolvimento do console de jogos multiplataforma e também os resultados obtidos na personalização do jogo pelos alunos da disciplina. Para isso, organizou-se este artigo do seguinte modo: na seção 2 a arquitetura da placa *Raspberry Pi* apresentada, bem como é proporcionada uma visão geral do sistema operacional *Raspbian*; na seção 3 apresentam-se os conceitos principais da linguagem Python e da biblioteca *Pygame*, utilizadas na criação do software do jogo; na seção 4, o processo de desenvolvimento do jogo é descrito, bem como o que se contemplou cada versão; na seção 5 descreve-se a experiência da aplicação do jogo quanto as aulas ministradas e as modificações realizadas pelos alunos e, por fim, na seção 6 são tecidas as conclusões deste trabalho.

2. A placa *Raspberry Pi*

A placa *Raspberry Pi* é um microcomputador completo presente em uma pequena placa do tamanho de um cartão de crédito (85,60 mm × 53,98 mm) com peso de 45 gramas. Ele foi desenvolvido em 2006 no Reino Unido pela "*Raspberry Pi Foundation*" com a

intenção de estimular as escolas no ensino básico da Ciência de Computação [Matt e Shawn 2013; RaspberryPi 2013; Monk 2013]. Seu modelo "B", lançado em 2012, possui um SoC ["System on a Chip"] da Broadcom, modelo BCM2835, que embute uma CPU ARM1176JZF-S de 700 MHz e GPU VideoCore IV, OpenGL ES 2.0, decodificador H-264/MPEG-4 e 512 MB de RAM, duas portas USB 2.0, conector para cartão SD, conector RJ-45 para rede 10/100 Mbps, saída HDMI com suporte para vídeo de até 1080p, áudio digital, saída de vídeo composto por conector RCA, saída de áudio analógico por conector plugue P2, dentre outros.

O sistema operacional do *Raspberry Pi* deve ser instalado em um cartão SD (tamanho desejável de no mínimo 4GB) e, embora existam à venda cartões com o sistema operacional pré-instalado, o processo de preparação e instalação é bem simples. Neste projeto, optou-se pelo sistema operacional *Raspbian*, que é um sistema estável e de uso geral, atualizado com frequência, e que possui amplo suporte pela comunidade online [Raspbian 2013]. Ele é uma distribuição baseada no Debian, uma das distribuições Linux mais utilizadas no momento e utilizada como base para muitas outras [Debian 2013]. O sistema tem como padrão de ambiente gráfico o LXDE, adequado a sistemas com capacidades computacionais modestas. A partir do menu do LXDE, pode-se acessar um conjunto significativo de programas, organizados por tipo de aplicação (acessórios, educação, Internet, programação, ferramentas do sistema, entre outros), o que o torna atraente para ser utilizado em especial, em salas de aula.

Dentre as aplicações disponíveis, destacam-se o navegador *Midori*, ferramentas de programação como interpretador da linguagem Python, ambiente de programação com blocos *Scratch* e o ambiente de programação *Squeak*, que possibilitam o pronto ensino de algoritmos e desenvolvimento de projetos de computação. Além disso, por meio de seu gerenciador de pacotes, pode-se ainda instalar inúmeras ferramentas gratuitas e de código aberto disponibilizadas nos repositórios do *Raspbian*, tanto voltadas à programação quanto outras, tais como o *LibreOffice*, que se assemelha ao pacote Office® da Microsoft. Outra fonte para esses aplicativos é a *Pi Store* (http://store.raspberrypi.com/projects), que é um site de onde se pode fazer a transferência de diversos jogos e aplicações disponíveis para o *Raspberry Pi*.

A Figura 1 apresenta os principais componentes da arquitetura de *hardware Raspberry PI.*

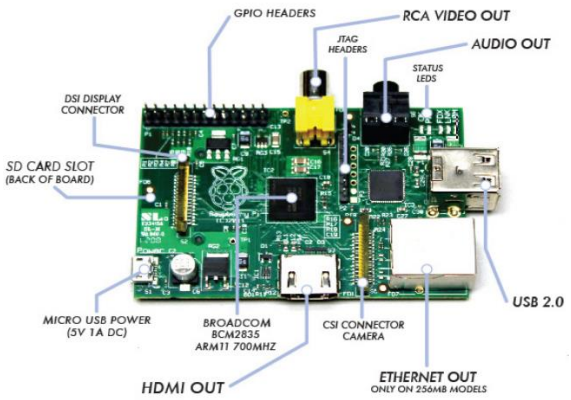

Figura 1. Componentes do *Raspberry Pi* [RaspberryPi 2013]

3. Programação *Python* no *Raspberry Pi*

Python é uma linguagem de alto nível criada em 1989 por Guido Van Rossum e disponibilizada ao público em 1991 [Deitel et al. 2002; Campbell et al. 2009; Göktürk e Kalkan 2012; Lutz 2009]. Interpretadores para esta linguagem e seus códigos-fonte estão disponíveis de forma gratuita[1], assim como uma grande quantidade de tutoriais e guias.

As principais características de Python são [Agarwal e Achla 2005]:

- Multiplataforma;
- Simples, com sintaxe e semântica consistentes;
- Sistema de tipos dinâmico;
- Adequada tanto para *scripts* e programação em grande escala;
- Modular;
- Gerenciamento automático de memória;
- Possui uma grande quantidade de bibliotecas para diversos domínios de aplicação;
- Suporte a vários tipos de bibliotecas de interface homem-máquina;
- Suporta programação orientada a objetos, estruturada e funcional [de forma parcial];
- Bem suportada por uma grande comunidade;
- Indicado para desenvolvimento rápido de aplicações;
- Fornece boa interface com C / C + +, Java e outras linguagens.

O que levou à escolha de *Python* como a linguagem deste projeto foi sua sintaxe simples e semântica consistente, tipagem automática, gerenciamento automático de memória e a farta disponibilidade de bibliotecas para diversos fins. Além disso, possui um ambiente simples de desenvolvimento denominado IDLE, no qual os estudantes podem interagir linha a linha para entender os conceitos básicos de programação antes de partir para escrever programas completos.

Para o projeto do console de jogos, foi utilizada a biblioteca *Pygame*, um módulo do *Python* que simplifica o uso das funções da biblioteca SDL (*Simple Direct Media Layer*), destinada à criação de jogos e aplicações multimídia [Sweigart 2012; Sweigart 2010]. Além das funcionalidades da SDL, *Pygame* acrescenta facilidades para a criação de jogos tais como *sprites*, *render groups*, detecção de colisão básica [retângulos] e também é multiplataforma como o próprio interpretador Python.

A Figura 2 apresenta o ambiente IDLE, utilizado pelos alunos neste projeto, com um exemplo em *Python* da solução das raízes de equação de segundo grau pelo método de *Bhaskara*:

1 www.python.org

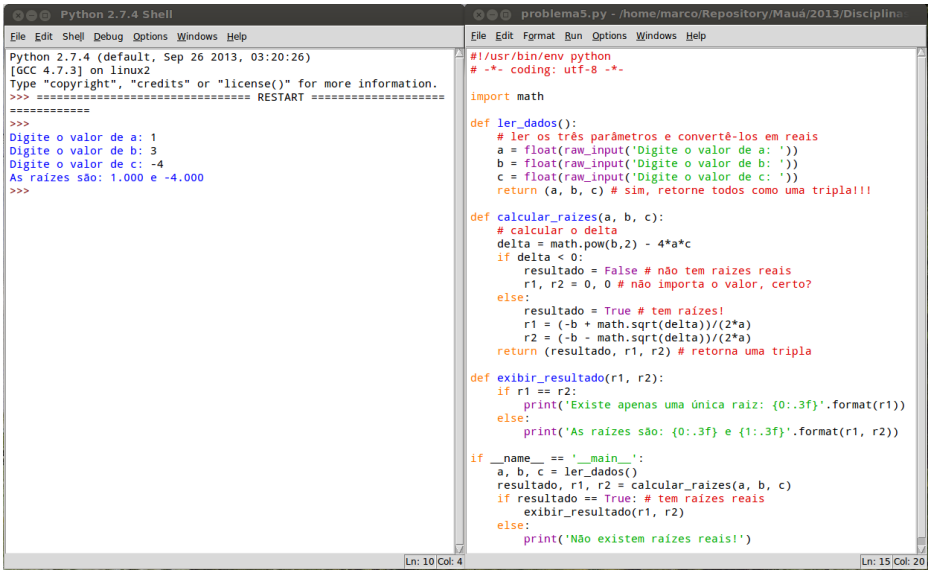

Figura 2. Ambiente IDLE

Um exemplo simples em Python da utilização da biblioteca *Pygame* está apresentado na Figura 3. Nela, uma imagem segue um percurso retangular de forma infinita. A lógica da animação está dentro de um laço de repetição no qual a imagem é desenhada na superfície da tela em uma posição conveniente. Cada quadro de animação é calibrado por um relógio e os eventos são tratados por um simples comando de repetição que inspeciona uma lista de eventos recebidos durante a iteração.

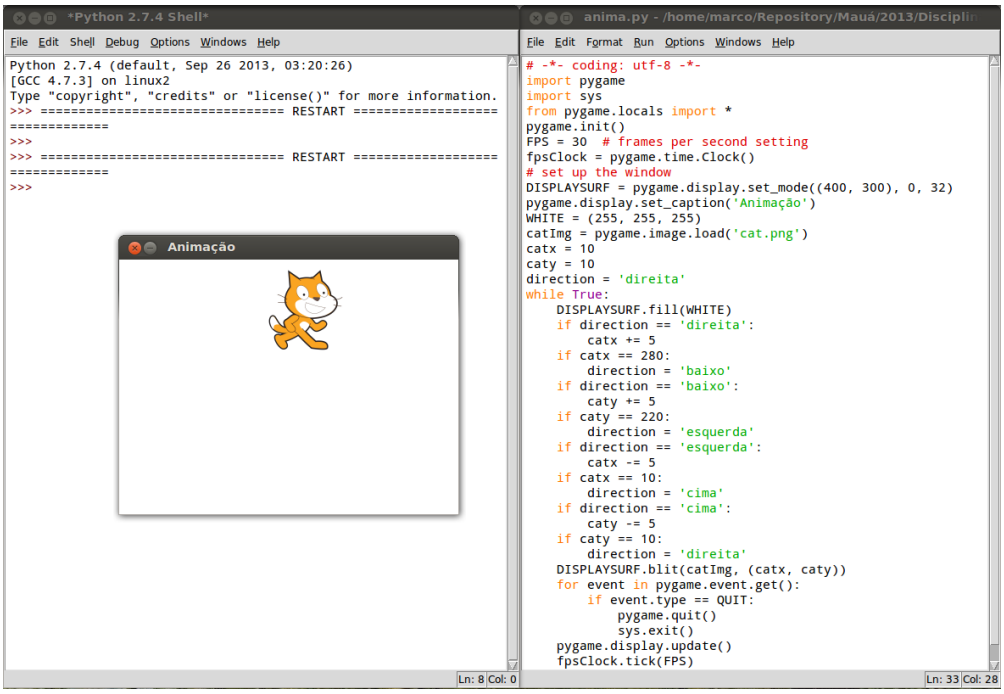

Figura 3. Exemplo de animação com a biblioteca *Pygame*

4. Desenvolvimento do console de jogos

Neste projeto, o jogo desenvolvido foi o “Pong”, que é um jogo eletrônico de esporte em duas dimensões que simula um tênis de mesa [Wikipedia 2013]. Na versão construída, o jogador controla um rebatedor no jogo movendo-o de forma vertical no lado esquerdo da tela, e compete contra outro jogador que controla um segundo rebatedor no lado oposto, conforme ilustrado na Figura 4.

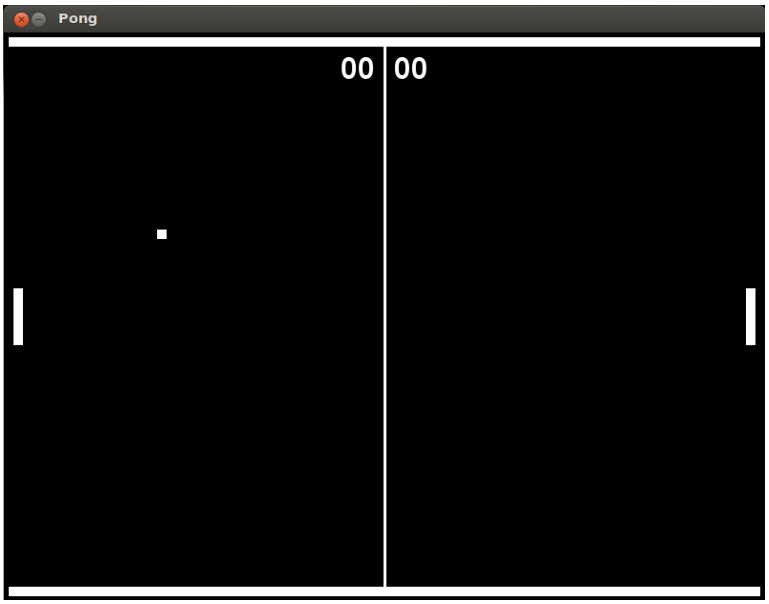

Figura 4. O Pong em ação

Aquele jogador que conseguir lançar uma bola que o outro não consiga rebater recebe um ponto. Para simplificar sua construção, o jogo foi baseado apenas em duas formas geométricas, linha e retângulo e a dinâmica do jogo emprega conceitos básicos da Física que um aluno da primeira série é capaz de dominar:

- Espaço percorrido é igual à velocidade multiplicado pelo tempo transcorrido;
- Choques elásticos.

O movimento da bola é ditado por uma velocidade absoluta [em pixels/segundo], uma direção (inclinação com a horizontal) e um sentido (esquerda ou direita). A velocidade inicial é fixa, mas conforme será descrito, poderá ser alterada de acordo com a dinâmica do jogo. Já os valores iniciais das outras propriedades são determinados com o auxílio de uma função de geração de números pseudoaleatórios.

A bola ao atingir um dos rebatedores terá, além da inversão de movimento, um incremento em pixels/segundo na componente vertical de sua velocidade. Isto foi decidido para tornar o jogo mais “emocionante” e aumentar seu grau de dificuldade com o passar do tempo. As regras programadas para isso estão descritas na Figura 5, a seguir:

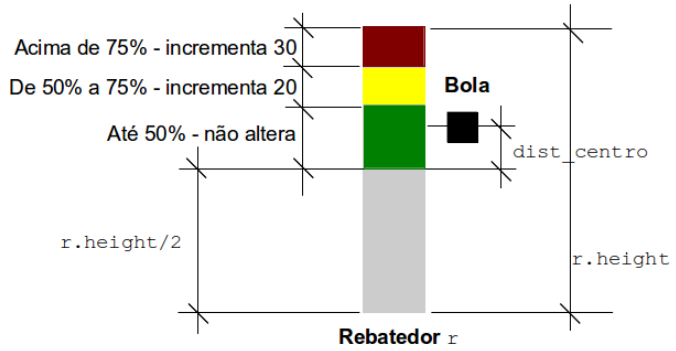

Figura 5. Regras de rebatimento para rebatedores do Pong

A bola também será rebatida se tocar nos retângulos brancos superiores e inferiores do jogo. Neste caso, será realizada apenas a inversão de direção, conforme ilustrado na Figura 6:

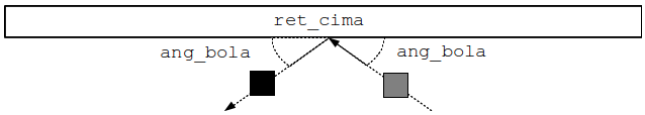

Figura 6. Regras de rebatimento para retângulos superior e inferior

Depois vem a verificação se houve ou não a passagem da bola por um dos fundos, direito ou esquerdo, que configurará a pontuação dos jogadores. O teste é bem simples e consistia apenas na verificação da colisão com retângulos ocultos naquelas posições.

A movimentação da bola segue o princípio da Física no qual a distância percorrida na trajetória linear da bola [aqui, em pixels] é igual ao produto de sua velocidade pelo intervalo de tempo mensurado. Foi utilizada uma função da biblioteca *Pygame* que permitiu a determinação do tempo em segundos transcorrido de um quadro a outro dentro do laço de repetição do jogo, sendo então utilizada para desenhar a próxima posição da bola no próximo quadro.

Já a movimentação dos rebatedores ficou por conta de uma função da biblioteca *Pygame* que permitiu a verificação do estado do dispositivo de entrada envolvido, que na primeira versão do jogo foi o próprio teclado do computador [jogador 1 com as teclas A – sobe, Z – desce e jogador 2 com as teclas K – sobe e M – desce]. De acordo com a tecla pressionada o rebatedor associado é movido de acordo com um deslocamento fixo em pixels para cima ou para baixo.

A primeira versão do jogo foi criada de modo "plano", isto é, sem a utilização de funções ou estruturas mais sofisticadas de modo a permitir que o aluno pudesse se concentrar-se na lógica da construção do jogo e testá-la linha a linha no ambiente IDLE. Na segunda versão o jogo foi construído de forma modular com a utilização de funções e foi acrescentando a utilização de joysticks, que facilitou o modo de jogar. A segunda versão foi instalada [embutida] em uma placa *Raspberry Pi* que foi embutido em uma estrutura, construída para se tornar um "totem de jogos", conforme apresentado na Figura 7.

Figura 7. Totem de jogos com o Pong

Por fim, foi criada ainda uma versão em rede do jogo, onde um servidor escrito em Python concentrava a lógica do jogo e gerenciava a comunicação entre os dois jogadores por meio de *threads* e *sockets* TCP/IP. Ao programa cliente cabia apenas o desenho das posições dos rebatedores, da bola e da pontuação. O programa servidor permite a realização concorrente de diversas partidas distintas.

Resume-se aqui, as etapas do desenvolvimento deste projeto:

1. Preparação e instalação do Linux [Raspbian] no cartão SD do *hardware Raspberry Pi*;
2. Instalação do *software* Python e da biblioteca de jogos *Pygame*;
3. Codificação do jogo "Pong" em *Python*;
4. Montagem e implementação de um totem com o hardware *Raspberry PI* denominado "COMPGAME" contendo os seguintes componentes: um monitor LCD, dois joysticks. Esse *layout* possibilita que os participantes joguem o "Pong";
5. Testes de usabilidade.

5. Aplicação do jogo nas aulas

A primeira versão do jogo, mais básica, foi apresentada em duas aulas aos alunos, nestas aulas foi discutida: a criação da tela, os objetos que comporiam o jogo e foram passadas noções de movimento, aceleração e colisão.

Logo após foram realizadas mais três aulas de treinamento básico com *Python* e três aulas de ambientação com o *Raspberry Pi* e *Raspbian*. Após as 8 (oito) aulas, foi solicitado aos alunos que estudassem o jogo e propusessem modificações e melhorias, mas mantendo o tema "Pong".

Um dos resultados obtidos está apresentado na Figura 8:

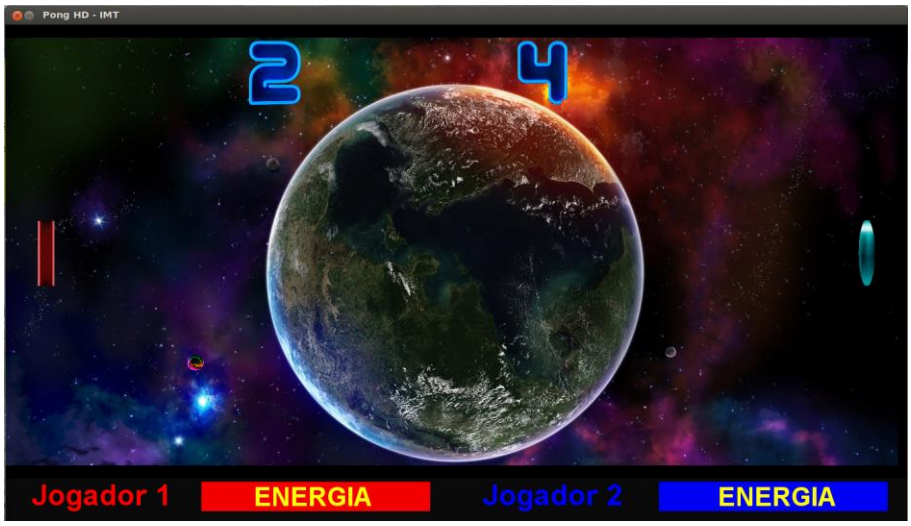

Figura 8. Uma versão do Pong construída por alunos

A partir de um jogo básico, contendo apenas elementos geométricos simples, essa equipe realizou as seguintes alterações:

- Substituição do fundo preto por imagens;
- Substituição da bola quadrada por um *sprite* animado;
- Substituição dos rebatedores retangulares por rebatedores estilizados;
- Substituição do placar com imagens de números;
- Inclusão de uma lógica de "energia", com uma tecla de "recarga" para dar mais emoção ao jogo;
- Inclusão de músicas no formato MP3 executadas durante o jogo.

Apesar de toda essa sofisticação, o código desenvolvido pelos alunos apresenta a mesma base que foi transmitida a eles durante as aulas de projeto. Com a facilidade que *Python* e *Pygame* apresentado em termos de programação e bibliotecas, a equipe conseguiu um resultado bem impressionante, se comparado com outros tipos de tecnologias mais complexas utilizadas em jogos profissionais, tais como utilização de C++ e *OpenGL*.

6. Conclusões

Algumas conclusões obtidas com esta experiência no desenvolvimento do console de jogos e de sua aplicação em sala de aula:

- Python é de fato uma linguagem enxuta e simples de aprender, de ensinar e rápida para criar aplicações. Ela pode ser utilizada de modo incremental, a partir de um protótipo e com o tempo acrescentar novos elementos para tornar o projeto mais robusto;
- Por ser interpretada, Python permite que os alunos consigam "enxergar" e interagir melhor a lógica de um programa, pois podem executar as linhas de códigos que estão testando e receber uma resposta, ao invés do tradicional processo "edita-compila-executa". Assim, muitas dúvidas sobre o que acontece na memória, por exemplo, são resolvidas com a inspeção das variáveis a qualquer instante, na linha de comando;

- O *Raspberry Pi* é uma solução muito interessante para a Educação e permite a utilização de muitos softwares de código aberto e gratuitos. Seu custo é baixo (US$ 35,00) e é de simples manipulação;
- Apesar do Linux não ser um sistema operacional muito utilizado pelos usuários finais, a combinação do Linux com a distribuição *Raspbian* e interface LXDE não ofereceu barreiras aos alunos durante seu aprendizado como foi suposto no início do projeto, não interferindo na usabilidade da solução;
- Quando os alunos de primeira séria possuem ferramentas de programação mais simples de entender (diminuindo assim o nível de abstração da área de programação), onde os conceitos envolvidos são mais acessíveis e interessantes (como as regras do jogo Pong), eles se tornam mais motivados e envolvidos nos projetos.

Referências

Agarwal, K. e Achla A. (2005) "Python for CS1, CS2 and beyond". *Journal of Computing Sciences in Colleges* 20 [4]: 262–270.

Campbell, J., Gries P., Montojo, J., Greg, W. (2009) *Practical Programming : an Introduction to Computer Science Using Python*. Raleigh, N.C.: Pragmatic Bookshelf.

Debian. (2013) "Debian - The Universal Operating System". Acessado outubro 28. http://www.debian.org/.

Deitel, H. M., Deitel, P. J., Lipieri, J. P., Ben W. (2002) *Python : How to Program*. Upper Saddle River, N.J.: Prentice Hall.

Göktürk, U. e Kalkan, S. (2012) "Introduction to Programming Concepts with Case Studies in Python". http://site.ebrary.com/id/10653536.

Lutz, M. (2009) *Learning Python*. 4º ed. Sebastopol, CA: O'Reilly.

Matt, R. e Shawn, W. (2013) *Getting Started with Raspberry Pi.* New York: O'Reilly.

Monk, S. (2013) *Programming the Raspberry Pi : Getting Started with Python*. New York: McGraw-Hill.

Pygame. (2013) "Pygame Documentation". Acessado outubro 28. http://www.pygame.org/docs/.

RaspberryPi. (2013) "Raspberry Pi". http://www.raspberrypi.org/.

Raspbian. (2013) "Raspbian". Acessado outubro 28. http://www.raspbian.org/.

Rossum, V. G. e Junior, F. L. D. (2013) "The Python Language Reference". http://docs.python.org/2/reference/.

Sweigart, Al. (2010) *Invent Your Own Computer Games with Python*. King of Prussia, PA: Al Sweigart.

———. (2012) *Making Games with Python & Pygame a Guide to Programming with Graphics, Animation, and Sound*. Charleston, S. C.: Creative Commons.

Wikipedia. (2013) "Pong – Wikipédia, a enciclopédia livre". Acessado outubro 28. http://pt.wikipedia.org/wiki/Pong.