Projeto de um Console de Jogos Multiplataforma Usando o Hardware *Raspberry PI* Implementado por Alunos da Primeira Série do Curso de Engenharia de Computação

Marco Antonio Furlan de Souza[1], Everson Denis[1], João Carlos Lopes Fernandes[1] Eric Muszalska Claro Gomes[1], Guilherme Heiti Ito Fukano[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,
ericmcgomes@hotmail.com,guilhermefukano_011@hotmail.com.

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 versus each other as well versus a machine simulating a player. 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students of the first year of an undergraduate degree in Computer Engineering modified the game's structure for make a newer version of the game, more interactive and dynamic.*

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 dois participantes do jogo em uma mesma máquina, como a interação de um jogador com um simulador de jogador. 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 uma versão mais dinâmica, interativa e atual do jogo.*

1. Introdução

A motivação para o desenvolvimento do console de jogos multiplataforma, descrito neste artigo, foi o engajamento dos três responsáveis 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 é a orientação do projeto para os alunos, de modo que eles possam vivenciar situações próximas a da realidade 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 o de criar um console de jogos multiplataforma. 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 serem propostos temas de escolha livre pelos alunos, foi definido um tema de jogo multiplataforma, 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 pudessem jogar remotamente.

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 PIé 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 abrange cada versão; na seção 5 descreve-se a experiência da aplicação do jogo quanto as aulas ministradas e as modificações realizadas pelos alunos; na seção 5 são pontuadas as alterações gráficas e os problemas encontrados durante a fase de desenvolvimento; e, por fim, na seção 8 são apresentadas as conclusões deste trabalho.

2. A placa *Raspberry Pi*

A placa Raspberry PIé 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. O hardware 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 Vídeo Core IV, Open GL 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 PIdeve 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 a outras finalidades, tais como o Libre Office, 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.

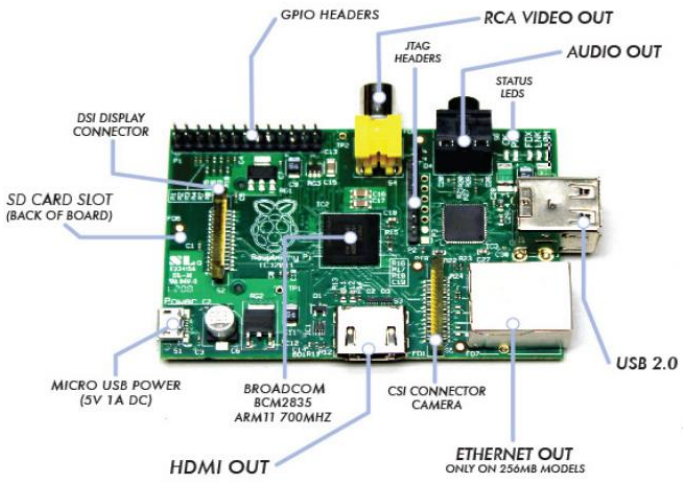

Figura 1. Componentes do *Raspberry PI*[RaspberryPi 2013]

3. A linguagem de programação *Python*

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 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, 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

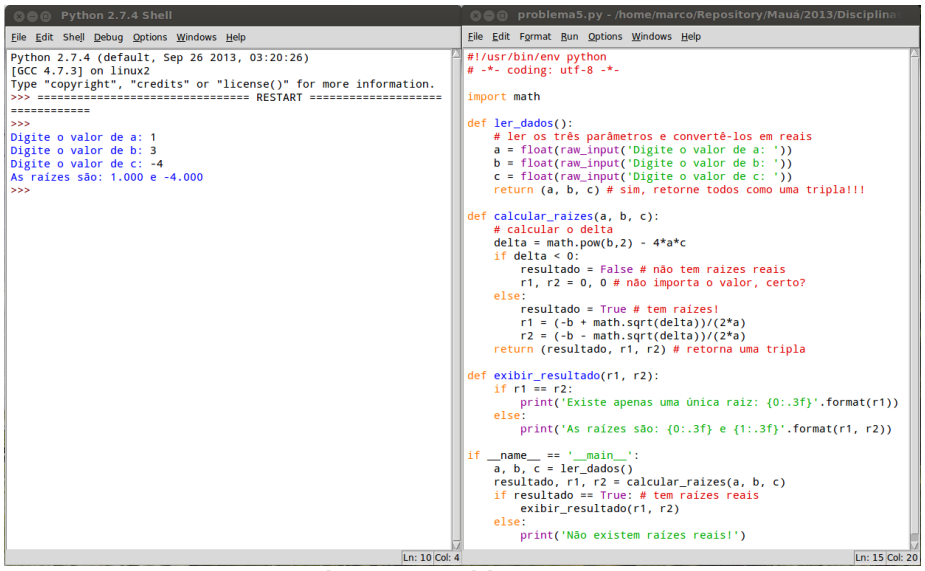

Figura 2. Ambiente IDLE

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.

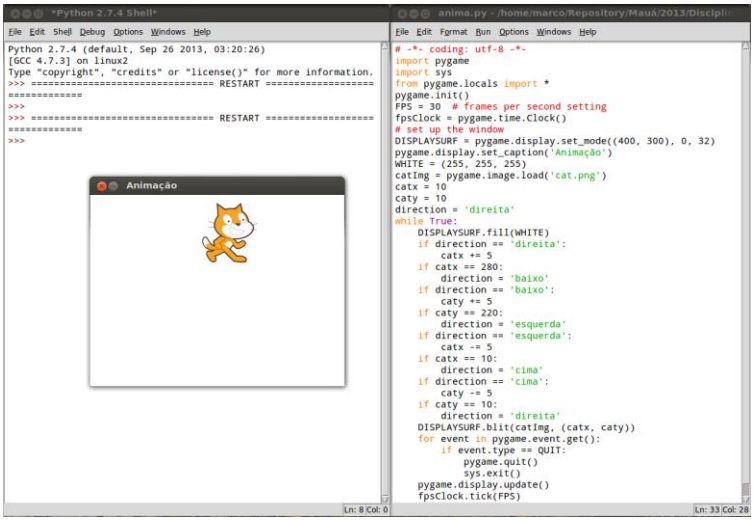

Figura 3. Exemplo de animação com a biblioteca *Pygame*

4. Proposta de desenvolvimento inicial

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.

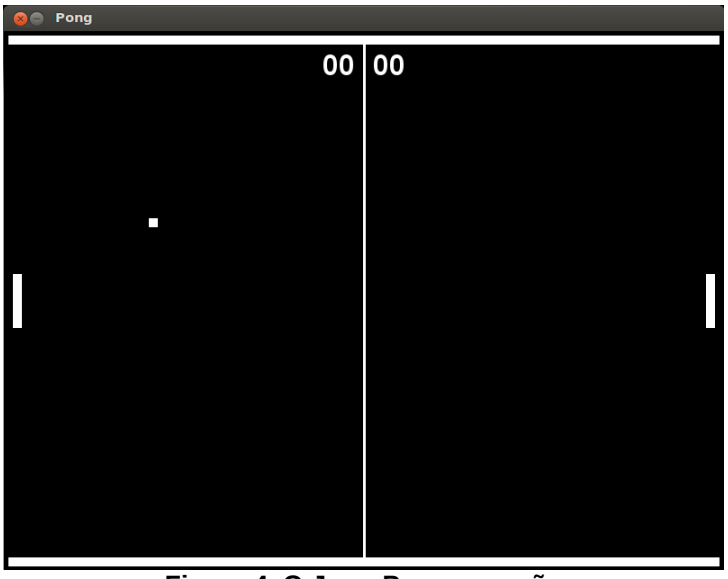

Figura 4. O Jogo Pong em ação

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:

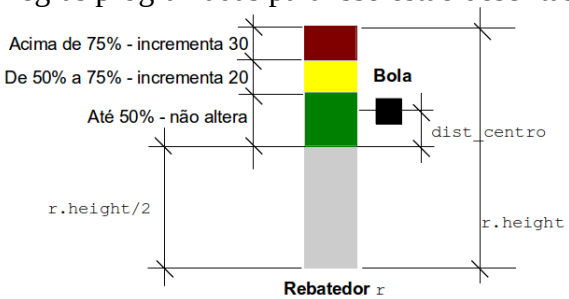

Figura 5. Regras de rebatimento para rebatedores do Pong

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:

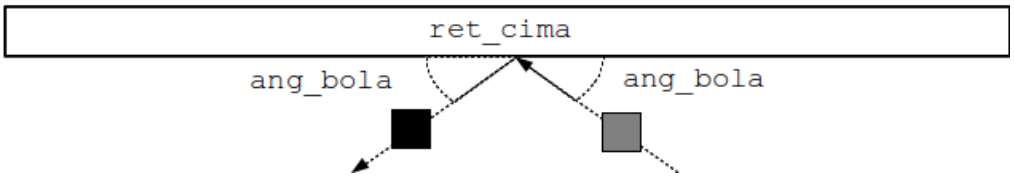

Figura 6. Regras de rebatimento para retângulos superior e inferior

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. Na segunda versão do projeto, foi modelada e confeccionada uma caixa acrílica para acomodar o hardware *Raspberry PI* e projetada uma estrutura de madeira, com o objetivo de se tornar um "totem de jogos", conforme apresentado na Figura 7.

Figura 7. Totem de jogos com o Pong

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

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.

Em continuidade ao projeto, foi criada uma terceira versão (beta) permitindo jogar o Pong em Rede (Cliente/Servidor), ou seja, um servidor escrito em Python concentrava a lógica do jogo gerenciando a comunicação entre dois jogadores por meio de “threads” e “sockets” TCP/IP. O programa cliente apenas controla as posições dos rebatedores, da bola e apresenta a pontuação. Esta versão ainda não foi finalizada.

5. Aulas Iniciais e Melhorias Implementadas no Projeto

A primeira versão do jogo, mais básica, foi apresentada em duas aulas aos alunos, onde foram discutidos os principais pontos: a criação da tela, os objetos do jogo e conceitos de movimento, aceleração e colisão.

Na sequencia 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, foram solicitados aos alunos que estudassem o jogo e propusessem modificações e melhorias mantendo o tema “Pong”.

A partir de um jogo básico, contendo apenas elementos geométricos simples, foram realizadas várias modificações, como a inclusão de novos efeitos adicionais e uma lógica de “energia”, com uma tecla de “recarga” para dar mais emoção ao jogo. Durante a partida, uma variável que representa a quantidade de energia para cada jogador, representada graficamente por duas barras nos cantos inferiores da tela foi adicionada e caso um jogador possua uma quantidade suficiente de energia, poderá utilizar um poder para se favorecer ou atrapalhar o oponente.

Dentre as características criadas para o jogador, podem-se destacar: movimento com velocidade acelerada, permitindo assim alcançar toda a região que deve proteger a qualquer momento do jogo; aumentar o tamanho vertical de seu rebatedor, a fim de facilitar seus rebatimentos; reduzir o tamanho do rebatedor do oponente, atrapalhando-o; e aumentar a velocidade da bola, simulando assim o movimento conhecido como "corte" em tênis de mesa;

Todos esses efeitos foram testados por um grupo de vinte jogadores, de diferentes níveis de experiência com jogos, a fim de criar a forma mais equilibrada entre utilidade e custo de energia, além de certificar que o jogo permaneceria com uma complexidade mínima suficiente para entreter jogadores mais experientes e manter-se simples o suficiente para que qualquer pessoa possa jogar.

Além dos efeitos ativos adicionados, foi adicionado um fator de curvatura na bola, com o intuito de adicionar mais dinâmica ao jogo e criar um ambiente mais semelhante ao proposto, no caso um ambiente espacial, com distorção gravitacional e aumento na curvatura seguindo o princípio da figura 5. Quanto mais próximo à borda do rebatedor, maior será o acréscimo do fator de curvatura.

Para a criação de uma fórmula de curvatura que agradasse aos jogadores, foi estudado o comportamento de inúmeras funções teoricamente e experimentalmente, de forma que a função que melhor representa um desvio grande o suficiente para criar dinâmica ao jogo, mas que não atrapalhasse os jogadores, confundindo-os foi:

$$PosicaoY = PosicaoY + ds * sen(ang\_bola) * sen(dt * curva)$$

A função mostra o deslocamento da bola momentâneo, sendo PosicaoY a coordenada y do centro da bola; ds o fator de deslocamento padrão, definido no começo do jogo; ang_bola o ângulo inicial, gerado randomicamente entre 0 e 45 graus; dt uma variável que marca o tempo a partir do começo da rodada, até que um dos jogadores pontue; e curva o fator de curvatura que é aumentado conforme a posição de rebatimento.

Durante a fase de testes, foi notado que a bola poderia fixar-se em uma das barras laterais, então foi necessário alterar a função que mudava o sentido, de forma que também pudesse reduzir a curvatura.

Outra adição realizada no jogo foi às telas de menus, onde o jogador pode escolher entre jogar com outro jogador ou jogar com “bots” (simuladores de jogadores) e opção de pausar o jogo, mostrando todos os comandos caso exista alguma dúvida.

A inteligência dos “bots” consiste em uma série de condições, tais como a posição da bola em relação ao rebatedor e a quantidade de energia disponível, dispostas diferentemente em algoritmos definidos de acordo com os cinco níveis de dificuldade.

Outra ferramenta que permitiu aos jogadores uma imersão ao universo criado foi a inclusão de músicas no formato MP3 executadas durante o jogo, tais como a música ambiente e efeitos sonoros. Cada evento diferente possui um efeito sonoro distinto, demonstrando a singularidade de cada ponto do jogo.

6. Aperfeiçoamento gráfico

As primeiras alterações gráficas podem ser vistas logo que o jogo inicia, uma tela de início foi criada com o objetivo de ser uma “sala de espera”, aguardando a entrada de um jogador, nela foi adicionado um fundo agradável e um logo personalizado sobre o jogo como pode ser visto na figura 8 - Tela de abertura do jogo “Space Pong”.

Figura 8 - Tela de abertura do jogo “Space Pong”.

Para incrementar no jogo a possibilidade de escolha do número de jogadores e o nível de dificuldade dos “bots”, houve a necessidade de fazer um menu interativo e bem intuitivo, onde o jogador pode facilmente fazer as escolhas sem qualquer dificuldade, para isso foram criados “Sprites” para mostrar a seleção das opções como mostra a figura 9 - Menu de seleção de partida.

Figura 9 - Tela de seleção de dificuldades.

Na parte visual foram realizadas inúmeras melhorias, como a adição de “sprites” personalizados para os seguintes elementos visuais do jogo: rebatedores (como pode ser visto na figura 10); números dos placares; filtros de textura (são colocadas imagens sobrepostas utilizando transparência para dar o efeito de sombra) para efeitos quando há a colisão de “sprites”, criando assim efeitos luminosos que deixam o jogo mais agradável visualmente como pode ser vista como exemplo na figura 11.

Figura 10 - Rebatedores personalizados especial.

Figura 11 - Rebatedor vermelho utilizando um

Com a adição do controle de energia no jogo, houve a necessidade da criação das barras de energia, porém quando foram colocadas, não agradavam muito visualmente por serem muito robustas, assim foi criada uma função de suavização dos cálculos, possibilitando um melhor efeito visual conforme o aumento ou diminuição das barras de energia, tendo assim um aumento gradual da intensidade das cores e da barra, como mostra a figura 12.

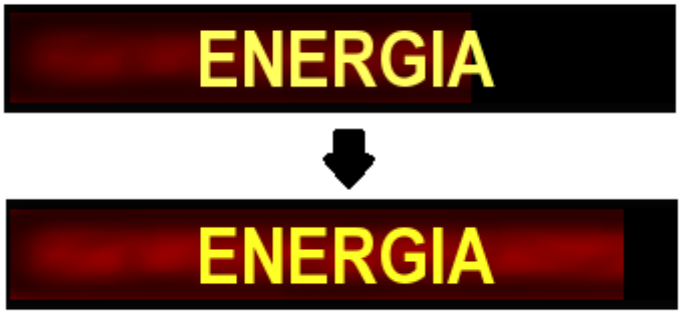

Figura 12 - Aumento da barra de energia.

A proposta foi de criar um jogo multiplataforma que funcionasse tanto para computadores com sistema operacional Windows e Linux com no *Raspberry PI*, inicialmente a programação do código foi realizada em um computador com o sistema Windows e posteriormente testado em Linux onde foram resolvidos os problemas da fase de desenvolvimento.

Quando o jogo foi transportado para o *Raspberry PI*, pode-se notar uma latência significativa que impossibilitada seu funcionamento. Após a realização de vários testes foi observado que o fator era o processamento dos efeitos gráficos que estavam interferindo, enquanto os efeitos sonoros quase não influenciaram.

Para solução deste problema, foram substituídas várias funções por "semelhantes" que possuíam um consumo de memória menor além da retirada de alguns efeitos gráficos.

O maior dilema era manter uma aparência agradável e uma "jogabilidade" aceitável, e depois de inúmeros testes decidiu-se que a melhor solução seria transformar os efeitos gráficos em vetores, já pré-definidos na biblioteca PyGame, abandonando assim diversas texturas, uma vez que os "*sprites"* com transparência consumiam muito processamento.

Com o intuito de não prejudicar os efeitos gráficos dos jogadores em diferentes plataformas, foram criadas duas distribuições, uma exclusiva para o Raspberry Pi, com algumas limitações nos efeitos gráficos, e outra para os demais hardwares, tais como computadores com sistemas operacionais Windows e Linux.

7. Conclusões

Com a experiência no desenvolvimento de um console de jogos multiplataforma foi possível observar os seguintes aspectos:

- O *Python* é de fato uma linguagem simples de aprender 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 interpretado, o *Python* permite "enxergar" e interagir melhor com a lógica de um programa 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, pois ele permite a utilização de muitos softwares de código aberto e gratuitos. Seu custo é baixo (US$ 35,00) 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 durante o projeto, não interferindo na usabilidade da solução;
- A adição de novos efeitos especiais permitiu não somente um aprendizado de lógica e linguagem de programação, mas também adicionou um conhecimento físico e matemático;
- As limitações do hardware *Raspberry PI* possibilitou ao grupo agregar conhecimentos de Arquitetura de Computadores e Sistemas Operacionais, tais como, limitações de CPU, memória, gerenciamento de processos e de hardware, *entre* outros;
- Este projeto possibilita novas soluções e projetos com o Hardware *Raspberry PI*.

8. 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.