

Programação motivacional: uma abordagem para o ensino da lógica de programação.

Marcos Laureano¹

¹Instituto Federal de Educação, Ciência e Tecnologia do Paraná (IFPR) - Campus Curitiba

marcos.laureano@ifpr.edu.br

Abstract. *This article describes the author's experience in teaching programming logic. It presents traditional forms of teaching programming and the approach used by the author to solve some problems, such as student motivation and the use of examples aimed at the game programming.*

Resumo. *Este artigo descreve a experiência do autor no ensino da disciplina de lógica de programação. Apresenta as formas tradicionais de ensino de programação e a abordagem utilizada pelo autor para resolver alguns problemas, como motivação do aluno e a utilização de exemplos voltados para a área de programação de jogos.*

1. Introdução

O Instituto Federal do Paraná - campus Curitiba mantém o curso Técnico em Programação de Jogos Digitais e Técnico em Informática na modalidade ensino-médio integrado. Mas como em todos os cursos relacionados a área de Informação e Comunicação, tem-se o desafio de ensinar a disciplina *Lógica de Programação*. Nas próximas sessões serão descritas como este desafio está sendo superado.

2. Dificuldades Iniciais

Vários autores têm sugerido diferentes abordagens para o ensino da programação. Alguns professores utilizam a abordagem inicial com português e fluxogramas. Outros, utilizam algoritmos e vários partem direto para uma abordagem com ensino de uma linguagem de programação real (Pascal e C são os exemplos mais comuns) [Laureano 2009].

Neste contexto, algumas ferramentas são utilizadas para facilitar o ensino da lógica de programação, dentre elas podemos citar: Webportugol [UNIVALI 2008], Alice [Conway 2008] ou Visualg [Apoio 2005]. Estas ferramentas são indicadas para o ensino da lógica de programação *tradicional*, embora possam ser utilizadas na abordagem aqui proposta.

3. A escolha

Após a realização de pesquisas (consultas a profissionais na área de jogos) sobre qual linguagem de programação seria lecionada no curso de Programação de Jogos Digitais, chegou-se a linguagem C++. Fluxogramas também são utilizados para ilustrar de forma gráfica, porém não rígida, as representações dos exemplos apresentados.

Mas não bastava apenas escolher a linguagem de programação, deveria ser escolhida uma nova abordagem para o ensino da lógica de programação neste curso. O próprio Bjarne Stroustrup [Stroustrup 1999] sugere o aprendizado da linguagem C++, *omitindo* alguns detalhes técnicos no início da aprendizagem. A *omissão*, sugerida por Bjarne Stroustrup, tem

como objetivo não focar em detalhes de implementação que possam desviar o foco do aprendizado da lógica.

Como o foco é desenvolver jogos educacionais livres, desde o início do curso os alunos trabalham com plataformas livres. Como sistema operacional é utilizada a distribuição Ubuntu. O compilador C++ escolhido foi o g++ [GNU 2010] e como IDE de desenvolvimento o *Code::Blocks* [The Code::Blocks Team 2010].

4. A abordagem

Historicamente, o ensino da lógica de programação normalmente está atrelada a exercícios que envolvam cálculos de sequências numéricas (*Fibonacci*, por exemplo), descobertas ou verificações de números (números primos, números perfeitos ou fatorial). Esses exercícios são válidos, mas não são motivacionais ou possuem pouca utilidade prática para um aluno iniciante. Logo, não faz sentido utilizar exemplos tradicionais para o ensino da programação de jogos, isto significa que os exemplos e exercícios não poderiam ser os sugeridos em vários livros. Desta forma, vários exemplos *tradicionais* foram *re-escritos*.

Com o decorrer da disciplina, o aluno aprende as noções de lógica (operadores, condições, repetições) através de exemplos e exercícios relacionados de alguma forma à área de jogos. Por exemplo, é comum solicitar ao aluno para montar um programa de calculadora (pedindo 2 números, o operador e apresentando o resultado). Neste caso, o aluno é incentivado a montar a operação (escolhendo aleatoriamente os números e operandos) e solicitar que o usuário digite a resposta correta (figura 1).

```
Bem vindo ao professor Corujinha versão 2012.  
Qual o resultado para 17+5 ?
```

Figura 1. Exemplo de saída de uma calculadora.

5. Biblioteca de funções

Mesmo com a reformulação da disciplina para facilitar o aprendizado da lógica de programação, alguns outros cuidados foram necessários. Foi desenvolvida uma biblioteca de funções e classes que permitem aos alunos realizarem várias atividades, que no dia-dia seriam complexas, de forma simples e sem perder o objetivo principal que é desenvolver a lógica.

O objetivo também é não perder tempo com possíveis problemas de compatibilidade de sistemas operacionais e diferentes compiladores C++. Muitas funcionalidades foram inseridas conforme demandas dos próprios alunos, tais como o computador gerar números aleatórios, navegação por teclas, resolver problemas de buferização de teclado, manipular *sprites* em ASCII efeitos sonoros ou cores. A biblioteca de funções e seu manual de uso e instalação podem ser encontrados em <http://www.jogos.digitais.ifpr.pro.br>. Esta biblioteca está disponível para o sistema operacional Linux e atualmente quase que totalmente portada para o ambiente Windows.

6. Uso de fluxogramas

Como comentado anteriormente, a utilização de fluxogramas facilita a compreensão dos códigos de exemplo. Neste caso, a estrutura de fluxogramas foi reduzida para alguns poucos elementos (processamento, entrada, saída e condicional). A figura 2 exemplifica o entendimento do comando *do..while* em C++.

```

1 // Introduzindo o play again
2 #include "biblaureano.h"
3 int main()
4 {
5     char again;
6     do
7     {
8         cout << "\n**Este jogo é muito excitante**" << endl;
9         again = readChar("Você quer jogar novamente ? (s/n): ");
10    } while( again == 's' );
11    cout << "Game over!" << endl;
12    return 0;
13 }

```

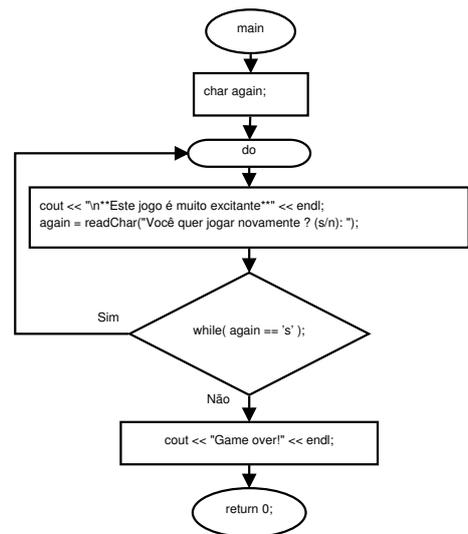


Figura 2. Introduzindo o *play-again*.

Inclusive, utilizando a abordagem sugerida por [Stroustrup 1999], foi possível iniciar o aprendizado de noções de programação orientada a objetos. Por exemplo, uma espaçonave é um objeto, pois combina qualidades (nível de energia ou quantidade de munição restantes) e habilidades (disparar tiros ou levantar o escudo, por exemplo). Seguindo esta linha, o aluno vai associando o conceito de *qualidades* como características de um objeto, ou seja, descobre que os objetos têm *atributos*. E que as *habilidades* são as ações que um objeto pode realizar, ou seja, descobre que os objetos trabalham com *métodos*.

7. Resultados

Este modelo foi utilizado em 6 turmas (3 de jogos e 3 de informática) e os resultados obtidos, por observação e comparação de desempenho, comprovaram que a abordagem está colhendo os seguintes resultados: Os alunos demonstraram habilidades acima da média em programação, motivação e criatividade para criar jogos simples, onde são aplicados todos os conceitos de programação vistos até o momento (repetição, condições, declaração de variáveis, etc). A figura 3 demonstra alguns dos jogos criados pelos alunos.

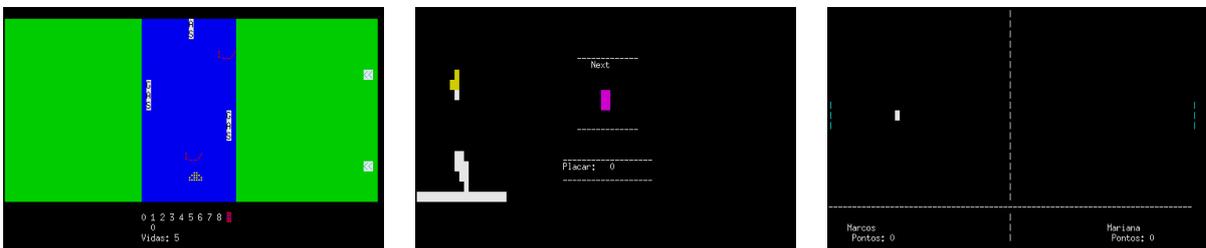


Figura 3. River Raid, Tetris e Pong.

8. Trabalho em andamento

Após os resultados na criação de jogos em modo texto o próximo objetivo é permitir que os alunos criem seus jogos em modo gráfico. Logo, estão sendo realizadas adaptações para uso da biblioteca *Allegro* [Allegro 2010] no aprendizado da lógica. Parte dessas adaptações podem ser vistas na figura 4. Nesta figura pode-se observar que algumas características técnicas da biblioteca estão sendo *omitidas* aos alunos.

```

1#include "jogos.h"
2INICIO(800,600)
3 int x=1;
4 int y=1;
5 int bg=WHITE;
6 for(int cor = BLACK; cor<=WHITE; ++cor)
7 {
8     mostraMensagem("Brincando com Allegro", x,y, cor, bg );
9     --bg;
10    y+=ALTURA;
11    x+=LARGURA("A");
12 }
13 ESPERA(KEY_ESC)
14 {
15     if (MLEFT)
16         mostraMensagem("LEFT", mouse_x, mouse_y, DARK_CYAN);
17     if (MRIGHT)
18         mostraMensagem("RIGHT", mouse_x, mouse_y, DARK_CYAN);
19     if (MCENTER)
20         mostraMensagem("CENTER", mouse_x, mouse_y, DARK_CYAN);
21 }
22 FIM

1#include <iostream>
2#include <allegro.h>
3using namespace std;
4
5int main(){
6    allegro_init();
7    install_keyboard();
8    install_mouse();
9    set_gfx_mode(GFX_SAFE, 800, 600, 0, 0);
10   show_mouse(screen);
11
12   int y=1;
13   int x=1;
14   int bg=15;
15   for(int i = 0; i<16;++i)
16   {
17       textout_ex(screen,font, "Brincando com Allegro", x, y,i, bg);
18       --bg;
19       y+=text_height(font);
20       x+=text_length(font, "A")
21   }
22
23   while (!key[KEY_ESC])
24   {
25       if (mouse_b & 1)
26           textout_ex(screen,font, "LEFT", mouse_x, mouse_y,3,-1 );
27       if (mouse_b & 2)
28           textout_ex(screen,font, "RIGHT", mouse_x, mouse_y,3,-1 );
29       if (mouse_b & 4)
30           textout_ex(screen,font, "CENTER", mouse_x, mouse_y,3,-1 );
31   }
32   return 0;
33 }
34 END_OF_MAIN()

```

Figura 4. Demonstração com a biblioteca Allegro (direita código simplificado, esquerda código tradicional).

9. Conclusão

Utilizar uma abordagem no ensino da programação, que estimula a criatividade em resolução de problemas de lógica, percebeu-se que os alunos tem respondido mais rapidamente e com mais motivação ao aprendizado de lógica. Em comparação com turmas anteriores, onde o ensino era o tradicional, notou-se um significativo avanço no ensino dos conceitos de condição, repetição, vetores, etc. inclusive permitindo a introdução de conceitos de programação orientada a objetos.

Referências

- Allegro (2010). Allegro - a game programming library. <http://alleg.sourceforge.net/>.
- Apoio (2005). Visualg - editor e interpretador de pseudocódigos. <http://www.apoioinformatica.inf.br/>.
- Conway, M. (2008). Alice - an educational software that teaches students computer programming in a 3d environment. <http://www.alice.org/>.
- GNU (2010). GCC, the GNU compiler collection. <http://gcc.gnu.org/>.
- Laureano, M. A. P. (2009). *Lógica de Programação - Uma Abordagem em Pascal*. Editora Ciência Moderna, Rio de Janeiro, 1 edition.
- Ministério da Educação (2009). Catálogo nacional de cursos técnicos. <http://catalogonct.mec.gov.br/>.
- Stroustrup, B. (1999). Learning standart c++ as a new language. *The C/C++ Users Journal*.
- The Code::Blocks Team (2010). Code::blocks - the open source, cross platform, free c++ ide. <http://www.codeblocks.org/>.
- UNIVALI (2008). WEBportugol. <http://siaiacad17.univali.br/webportugol/>.