

Incentivo ao aprendizado de tecnologia para jovens por meio de competições de robótica

Using robotics competitions as motivation for young students to learn technology

RESUMO

Como uma continuação ao projeto de extensão “Inclusão digital de jovens pelo estudo da robótica”, este projeto tem como objetivo capacitar estudantes do ensino de nível básico a entender os fundamentos principais de programação, por meio da interação com dispositivos robóticos programáveis, sejam eles físicos ou simulados. Com esse método, incentiva-se a participação dos estudantes em eventos e competições de robótica. Após testes, o uso dessa metodologia mostrou-se muito útil e capaz de despertar o interesse dos alunos para a área da robótica. Além disso o uso da ferramenta desenvolvida no projeto proporciona uma gama de opções para o ensino de robótica nas escolas, atraindo também maior interesse para toda a área tecnológica.

PALAVRAS-CHAVE: Robótica. Educação. Fluxograma.

ABSTRACT

This work is a continuation to the extension project "Digital inclusion of the young by the study of robotics". This project has the objective of enabling primary to high-school students to understand the fundamentals of computer programming, through the interaction with programmable mobile robots, whether physical or simulated. For this, students are encouraged to participate in robotics competitions. After testing, this methodology was shown to be very useful and capable of awakening the students' interest in the field of robotics. Furthermore, the use of the developed tool in this project provides a myriad of options for teaching robotics in schools, thus also attracting greater interest in the whole technological field.

KEYWORDS: Robotics. Education. Flowcharts.

Misael Rosa da Costa
misael@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

João Alberto Fabro
fabro@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Álefe Felipe Gonçalves Pereira Dias
alefe@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Eli Teixeira Paula Junior
elijunior@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Gabriel Rauta Buiar
gabrielbuiar@gmail.com
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Recebido: 19 ago. 2019.

Aprovado: 01 out. 2019.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

Atualmente, cada vez mais a robótica educacional tem ganho espaço na educação brasileira. Pesquisas mostram que tais iniciativas são facilmente aceitas por grande parte dos alunos, além de serem de grande ajuda no ensino das disciplinas do ensino básico, como física e matemática. Esses métodos têm frequentemente sucesso pelo fato de proporcionar ao aluno uma visualização do resultado do que foi feito, despertando assim o interesse dos estudantes. Além disso, a robótica é um meio eficiente de apresentar as abstrações necessárias para a compreensão de diversos conceitos abstratos relacionados à programação de computadores (VARGAS et al, 2012; OLIVEIRA et al, 2012).

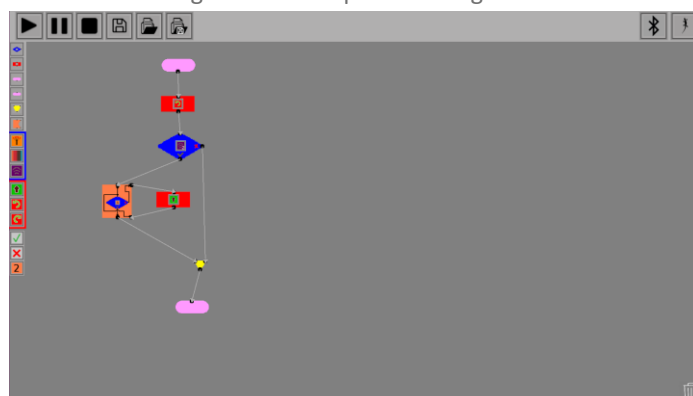
Essas estratégias também facilitam a identificação e correção de erros, visto que estes são fisicamente visíveis. Elas também ajudam a melhorar a autoestima dos alunos, as relações entre estes na turma e no desenvolvimento de trabalho em grupo. No entanto, muitas dessas propostas envolvem alunos com experiência em programação (SANTOS; NASCIMENTO; BEZERRA, 2010) e/ou utilizam kits de desenvolvimento da empresa Lego (ARANIBAR et al, 2006) que tem custo relativamente alto. Tendo isso em vista, optou-se por manufaturar um kit de desenvolvimento que tivesse baixo custo, usando componentes eletrônicos fáceis de se encontrar e usando fluxogramas como programação, já que esta se mostra como a maneira mais simples de apresentar os alunos à programação (MASCHIO; DIRENE, 2007).

Este projeto é desenvolvido no LASER, Laboratório Avançado de Sistemas Embarcados e Robótica, na UTFRPR, câmpus Curitiba, sede Centro.

MATERIAL E MÉTODOS

Para o ensino dos fundamentos de programação, foi desenvolvido um ambiente gráfico para criação de fluxogramas. O mesmo é dividido em duas partes: *Fluxprog* e *FluxprogBackEnd*. No *Fluxprog* há blocos que podem ser arrastados pelo programador de modo a formar o programa (Figura 1). Ao final, o fluxograma é interpretado e convertido nos comandos captados pelo robô, são eles: andar um passo para frente, virar à direita e virar à esquerda. Todas as partes do projeto foram desenvolvidas em C++ e a interface gráfica foi desenvolvida com uso da biblioteca *Allegro*.

Figura 1 – Exemplo de fluxograma



Fonte: Autoria Própria.

Pelo fato de a primeira versão do *Fluxprog*, desenvolvida no projeto “Inclusão digital de jovens pelo estudo da robótica”, estar com vários problemas e não estar estruturada em forma de Orientação a Objetos, decidiu-se criar uma nova versão do programa, sendo que novos modos de programação foram adicionados, levando os alunos que usam o programa a um nível de maior aprendizado lógico.

O ambiente simulado é feito no V-REP (*Virtual Robotics Experimentation Platform*). Este ambiente apresenta um robô com duas rodas, 5 sensores de proximidades simulando ultrassom e 5 sensores de cor simulando sensores de fita preta. Esses sensores também podem ser usados na leitura de cor. No caso, a cor escolhida é verde, já que consta em uma das regras da OBR (Olimpíada Brasileira de Robótica) que o robô deve virar ao ver algo verde no chão.

O *FluxprogBackend* capta os comandos enviados pelo *Fluxprog* e os envia para o robô simulado ou físico. Os resultados do controle do robô simulado são bons, já que mesmo com o uso de fatores físicos, eles não são aleatórios o suficiente para gerar um mau desempenho, como há nos robôs físicos.

Para comunicação entre o *FuxprogBackend* e o *Fluxprog* é usado o compartilhamento de memória, por meio da biblioteca *Boost::SharedMemory*. Cabe ao *Fluxprog* abrir o programa intermediário, selecionando através de uma posição de memória específica se deseja utilizar o robô real ou o simulado. Depois ambos leem e escrevem na memória de forma que o controle do robô físico ou simulado seja feito sem problemas intermediários.

Com os atuais modos de operação do programa, o usuário possui 3 níveis de controle do robô: no primeiro nível, o usuário apenas usa comandos de movimentação, não lendo os sensores do robô para tomar decisões. Assim, cabe ao *FluxProgBackend* cuidar para que o robô não colida com um obstáculo caso o usuário cometa algum erro na programação. No segundo, o robô segue linha de forma automática, mas não evita colisões. Com isso o usuário deve ler os sensores de ultrassom para tomar decisões em caso de percepção de obstáculos. No terceiro nível o robô não possui mais nenhuma “inteligência pré-programada”, cabe ao usuário controlar o robô, fazendo a lógica de seguir linha, entre outros casos, a partir da leitura dos sensores de fita e de cor.

Para o robô real, a plataforma de desenvolvimento usada é o Arduino, sendo que o modelo escolhido foi o MEGA, devido ao grande número de sensores e componentes do robô. A programação foi feita em C++, com base no algoritmo já funcionando no simulador, dessa forma o *FluxProg* envia um dos três comandos já citados, o robô identifica e age de formar a executá-lo. O ambiente físico usado é um banner impresso, contendo um caminho semelhante ao do V-REP, porém há também pistas para treinos para a OBR.

Para fazer a comunicação com o robô físico, utilizou-se o módulo HC-05, que usa tecnologia Bluetooth. Esta tecnologia possui a limitação de ter comunicação um para um, mas foi escolhida por se esperar usar smartphones no controle dos robôs posteriormente.

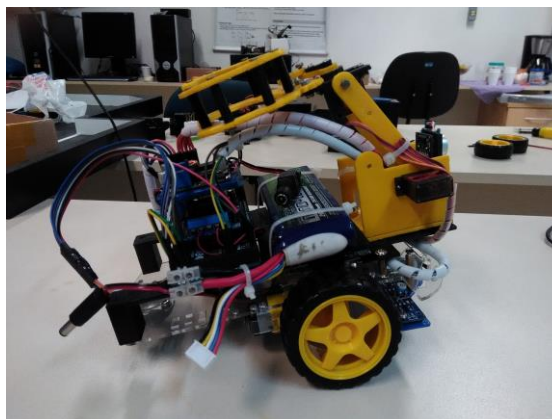
Os sensores de fita preta utilizados no robô fazem parte do módulo BFD-1000. Já o sensor ultrassônico escolhido foi o do módulo HC-SR04. Na movimentação do robô são usados dois motores DC com tensão de operação entre 3V e 6V. Para ligar e controlar a rotação dos motores é usado o *Motor Shield L293D*. No controle da

velocidade e direção do robô, são utilizados dois sensores do tipo *encoder*, módulo MH-Sensor-Series, que simulam um *encoder* analógico usando infravermelho.

A carcaça do robô, os dois motores, as rodas e discos *encoder*, foram adquiridos pelo kit 2WD. Para a confecção das bases dos diversos componentes do robô foi usado modelagem 3D, e gerado o modelo real usando uma impressora 3D.

Com o objetivo de deixar os robôs físicos aptos a participar da OBR, incluiu-se sensores de cor ao robô, módulo TC34725. A grande quantidade de sensores foi um problema, principalmente devido ao ultrassom, que gera um grande atraso ao sistema. Isso foi resolvido usando interrupções e preempções, porém o fator físico de espaço se agravou e com isso 2 sensores ultrassons não fazem mais parte do projeto. Além disso, uma garra está sendo desenvolvida para captação de uma bolinha, simulando um resgate, que é um dos objetivos da OBR (Figura 2).

Figura 2 – Robô físico com garra



Fonte: Autoria própria.

Testes de bateria foram realizados e percebeu-se que pilhas recarregáveis não podem ser usadas no robô físico devido à baixa quantidade de corrente elétrica suprida. Com isso baterias de lítio de alto desempenho estão sendo usadas no projeto.

Para avaliação do método proposto, desenvolveu-se um questionário, que é aplicado aos alunos ou professores que utilizam o *FluxProg*. Esse questionário é útil para que se possa fazer melhorias na metodologia, e entender a eficácia da mesma.

A metodologia completa proposta neste projeto ocorre da seguinte maneira: os usuários primeiro aprendem a lógica da programação, usando o *FluxProg*. Após isso, começam os testes básicos no Arduino, como acender um LED. Depois os usuários começam a montar seu próprio robô, e a programação para as atividades da OBR começam. O *FluxProg* apenas envia “mensagens” para o robô executar um dos comandos predeterminados, posteriormente é necessário que os alunos aprendam programar diretamente no Arduino. Para programação do Arduino, inicialmente se usa o *ArduBlock*, uma IDE que permite a programação do Arduino com blocos. Conforme a familiaridade com o microcontrolador aumenta, se passa para a IDE padrão do Arduino, com programação de forma textual.

RESULTADO E DISCUSSÃO

Desde que o *Fluxprog* passou pela repaginação, houveram três exposições nas quais foi apresentada a metodologia. A primeira foi realizada em um evento de premiação da Olimpíada de Astronomia, realizada no Parque Barigui, em Curitiba. Nesse dia, houve apenas a apresentação do projeto.

A segunda consistiu em apresentar e ensinar o uso do *FluxProg* para alunos do primeiro período dos cursos de Engenharia de Computação e Sistemas de Informação da UTFPR Campus Curitiba. Esta se deu em dois dias, um para cada curso, sendo que ao final da apresentação foi aplicado um questionário. Infelizmente para a primeira turma não foi possível apresentar uma boa usabilidade, já que o *FluxProg* estava travando devido a vazamento de memória, problema que foi corrigido para a segunda turma. Após a aplicação do questionário, percebeu-se que o método, mesmo para alunos da graduação, gera bons resultados, já que incentiva mais ainda a busca pelo aprendizado de programação, além de auxiliar na disciplina e motivar os alunos.

A terceira foi um encontro com professores da Rede de Ensino Municipal de Curitiba. Este foi realizado com o objetivo de montar os robôs físicos e fazê-los se movimentar. Foi apresentado o *FluxProg*, o *ArduBlock* e realizado alguns testes com o Arduino, e ao final os robôs estavam se movimentando. O objetivo dos encontros com professores é ensiná-los a metodologia e o desenvolvimento de robôs físicos, de modo que eles possam repassar para os seus alunos em sala de aula.

CONCLUSÃO

Mesmo ainda em estado de desenvolvimento, o ambiente de programação já está funcional, estando disponibilizado para download no repositório <https://github.com/laser-utfpr/FluxProg/>. Neste link podem ser encontradas as instruções de download, compilação e execução do ambiente de programação desenvolvido, bem como instruções de execução do simulador, permitindo o uso completo em atividades introdutórias de programação.

Entretanto, mesmo sem a construção física deste robô, deve ser possível utilizar o ambiente de programação para prover o aprendizado inicial de programação de uma forma direta, lúdica e interessante.

O valor final do robô ficou na média de 350 reais. Apesar de parecer um valor relativamente alto para pessoas que não tem um conhecimento prático na área da computação, esse valor está bem abaixo dos kits de desenvolvimento da LEGO, por exemplo. O valor do robô é correspondente apenas aos componentes físicos, já que o software desenvolvido pela equipe, o *ArduBlock* e a IDE do Arduino são gratuitos.

Posteriormente, é necessário aplicar mais vezes a metodologia (aproveitando que a mesma já está praticamente resolvida) para poder tomar decisões sobre as mudanças necessárias e gerar todos os resultados esperados.

AGRADECIMENTOS

Os autores agradecem à UTFPR pela concessão da bolsa de extensão. Agradecem também a todos que auxiliaram no desenvolvimento do projeto

FluxProg, no projeto e construção do protótipo do robô físico, e com a ajuda em diversos outros pontos do projeto.

REFERÊNCIAS

VARGAS, M. N.; MENEZES, A. G. C.; MASSARO C. M.; GONÇALVES T. M. Utilização da robótica educacional como ferramenta lúdica de aprendizagem na engenharia de produção: introdução à produção automatizada. *In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA*, 40., 2012, Belém. **Anais...** Belém: ABENGE, 2012.

OLIVEIRA, D.; FERREIRA, S.; CELESTINO, H.; FERREIRA, S.; ABRANTES, P. Uma proposta de ensino e aprendizagem de programação utilizando a robótica educativa e *storytelling*. *In: CONGRESSO INTERNACIONAL TIC E EDUCAÇÃO*, 2., 2012, Lisboa. **Atas...** Lisboa: Instituto de Educação da Universidade de Lisboa, 2012.

SANTOS, F. L.; NASCIMENTO, F. M. S.; BEZERRA, R. M. S. REDUC: A robótica educacional como abordagem de baixo custo para o ensino de computação em cursos técnicos e tecnológicos. *In: WORKSHOP DE INFORMÁTICA NA ESCOLA*, 16., 2010, Belo Horizonte. **Anais...** Belo Horizonte: Sociedade Brasileira de Computação, 2010.

ARANIBAR, D. B.; GURGEL, V.; SANTOS, M.; ARAUJO, G. R.; ROZA, V. X.; NASCIMENTO, R. A.; SILVA, A. F.; SILVA, A.; NASCIMENTO, L. M. G. Roboeduc: A software for teaching robotics to technological excluded children using lego prototypes. *In: Latin American Robotic Symposium*, 3., 2006, Santiago. **Anais...** Santiago: IEEE, 2006.

MASCHIO, E.; DIRENE, A. I. Múltiplas representações externas para o ensino de programação de computadores. *In: WORKSHOP DE AMBIENTES DE APOIO À APRENDIZAGEM DE ALGORITMOS E PROGRAMAÇÃO*, 1., 2007, São Paulo. **Anais...** São Paulo: Sociedade Brasileira de Computação, 2007.