

Capítulo

5

Uso de robótica e jogos digitais como sistema de apoio ao aprendizado

Humberto A. P. Zanetti, Ana L. S. de Souza,
João V. V. d'Abreu e Marcos A. F. Borges

Abstract

This mini-course will present the context and the use of robotics pedagogical practices and development of digital games as tools for teaching and learning in dynamic methodology based on PBL (Problem Based Learning). This mini-course will present several computational tools and platforms for interdisciplinary practices, providing and contextualizing the school environment, and motivating learning situations involving multidisciplinary. Finally, will illustrate the discussion with examples of application of computational devices to support teaching and learning.

Resumo

Este minicurso apresenta o contexto e a utilização de robótica pedagógica e práticas de desenvolvimento de jogos digitais como ferramentas de ensino-aprendizagem em dinâmicas baseadas na metodologia PBL (Problem Based Learning). Apresenta diversos instrumentos e plataformas computacionais para práticas interdisciplinares, provendo contextualizando no ambiente escolar, situações de aprendizagem motivadoras e multidisciplinares. Ao final, ilustra a discussão com exemplos de aplicação de dispositivos computacionais para apoio ao ensino-aprendizagem

1.1. Tópicos de interesses relacionados

Este mini-curso está relacionado com:

- aprendizado baseado em problemas (PBL -Problem Based Learning);
- métodos e práticas de ensino e aprendizagem utilizando robótica;
- estética construcionista de aprendizado;

- uso de sistemas computacionais para apoio ao ensino-aprendizagem;

1.2. Objetivos

A escola hoje está muito descontextualizada. Alunos que são nativos digitais, ou seja, que desde a sua mais tenra infância interagem com computadores, tablets, celulares, sentem-se desconfortáveis em um ambiente que replica o que ocorria há mais de um século atrás. As poucas iniciativas de levar o computador para a escola, normalmente, restringem-se ao uso de “salas de informática” em atividades “extras”, não integradas com as aulas de conteúdos. Segundo Wagner (2011) “ensinar computação é diferente do que é feito hoje em boa parte das escolas com laboratórios disponíveis”. Ainda para o mesmo autor, o ideal seria diferenciarmos os termos: “computação (estuda principalmente processos que podem ser matematizados) e informática (usa a área de processamento de dados no tratamento de informações)”. E então sugere que “a aplicação do *Computational Thinking* é uma maneira computacional de se encarar, modelar e propor soluções para problemas de qualquer área, beneficiando assim, profissionais das ciências exatas às sociais”.

O objetivo deste minicurso é mostrar as possibilidades de práticas pedagógicas inspiradas na metodologia de aprendizado baseado em problemas (PBL -*Problem Based Learning*) associadas ao uso de robótica e linguagens de programação. Busca-se capacitar e estimular educadores para a adoção de meios motivadores e de forte presença experimental e prática, para um maior envolvimento de alunos com o aprendizado.

O objetivo da abordagem proposta é prover uma melhora na qualidade de ensino, motivar e desenvolver diversas habilidades em atividades pedagógicas, trazer um ambiente de exploração e experimentação em um ambiente educativo, promover a aproximação de atividade tecnológica com o conteúdo disciplinar estudado e estimular o trabalho em grupo, conseqüentemente, o compartilhamento de conhecimentos.

Além do objetivo de se repensar as práticas de ensino/aprendizagem, existe a possibilidade de usar as práticas discutidas neste minicurso como ferramentas de motivação para jovens seguir estudando na área de tecnologia. O país se recente da formação de mais jovens na área, e muitas possibilidades de desenvolvimento da nação deixam de ocorrer devido a falta de profissionais de tecnologia.

O minicurso está estruturado em módulos, que serão discutidos em maiores detalhes nas sub-seções a seguir. O primeiro módulo discute o embasamento teórico do que é proposto. O segundo módulo apresenta um conjunto de ferramentas e práticas que exemplificam a proposta. Ao final, é feita a conclusão.

1.3. Embasamento teórico

No cenário atual das escolas, os alunos cresceram na denominada “sociedade da informação”, vivenciando o surgimento acelerado de novas tecnologias digitais e uso frequente das mesmas. Tais alunos encontram nas escolas modelos de ensino e aprendizagem que não evoluíram com o processo informacional que sua vida fora da escola percebe, causando desmotivação e desinteresse pelas aulas e conteúdos ministrados. A proposta de incluir Robótica e novas tecnologias como elemento motivador e facilitador possibilita trazer ao alunado novas experiências, vivência real e

continua com tecnologias, e principalmente, desafios e participação no processo de aprendizagem.

Segundo Papert (2008), a escola deve estar preparada para acompanhar as transformações tecnológicas da sociedade e conseqüentemente, preparar os jovens para que possam ser inseridos nessa sociedade. Papert propõe uma exploração “construcionista” da tecnologia na prática educativa, ajudando o indivíduo a criar suas próprias ideias e construir seus conhecimentos. Alunos passam a maior parte de seu tempo de seu tempo na escola estudando conteúdos de matemática e física, sem que possam aplicar tais conceitos no mundo real em experimentações. Essas experimentações poderiam ser facilmente amparadas por tecnologias, como softwares de simulação ou a prática de robótica.

Este minicurso apresenta o contexto e a utilização de robótica pedagógica e práticas de desenvolvimento de jogos digitais com ferramentas gratuitas ou de baixo custo em dinâmicas baseadas na metodologia PBL no processo de ensino-aprendizagem de nativos digitais, na qual o professor apresenta-se como um facilitador e o aluno como um agente ativo no seu aprendizado quando enfrenta um problema proposto e trabalha para resolvê-lo.

1.3.1. Robótica pedagógica

A Robótica Pedagógica (RP) tem sido utilizada há várias décadas, por instituições de ensino e pesquisa como ferramenta para enriquecimento do ambiente de ensino/aprendizagem. Ela pode ser entendida como um processo de interação com um dispositivo robótico (mecânico/eletromecânico), como forma de favorecer os processos cognitivos (d'ABREU e GARCIA, 2010). Ou ainda, um conjunto de recursos que visa o aprendizado científico e tecnológico integrado às demais áreas do conhecimento, utilizando-se de atividades como design, construção e programação de robô (LOPES, 2010, p.46).

A RP trata-se de um processo interativo, conciliatório, entre o concreto e o abstrato, na resolução de um problema que envolve etapas como: concepção, implementação, construção, automação e controle de um mecanismo. Em todas estas etapas pode e deve ocorrer a construção de conhecimento, advindo das mais diferentes áreas científicas. Um ambiente de RP deve fazer emergir ideias tecnológicas para permitir que as crianças se apropriem delas. Entendem-se como ideias tecnológicas as possibilidades de conexão de peças mecânicas e de componentes eletrônicos, para realização de uma determinada tarefa (PAPERT, 2008).

Um ambiente de RP pressupõe a existência de professores e alunos interagindo entre si. Ao longo dessa interação, eles produzem novos conhecimentos. É um ambiente de aprendizagem que não existe a priori. Fazem parte deste ambiente, componentes que permitem construir mecanismos que se movimentam e ferramentas de *hardware* e *software* para automação destes mecanismos. Além disso, existe a possibilidade de se desenvolver uma metodologia de ensino rica e diversificada, com base no currículo (d'ABREU e GARCIA, 2010).

A Robótica proporciona, além das questões de motivação e compreensão de conhecimento, um ambiente de trabalho colaborativo e atividades interpessoais que muitas vezes não é possível com instrumentos educacionais convencionais. A prática

pode explorar trabalho em equipe, cooperação, criatividade, expressão escrita e oral, organização e habilidades relacionadas à execução de projetos. A RP trabalha competências além daquelas que os currículos escolares trabalham, oferecendo aos alunos experiências reais de áreas da informática, eletrônica, mecânica e design.

O interesse como ferramenta de ensino vem deixando de ser restrito ao contexto de ensino superior. Há exemplos de projetos e competições na área que são destinados a alunos de ensino fundamental e médio como a “Olimpíada Brasileira de Robótica” (OBR)¹. A OBR promove competições que envolvem práticas e conhecimento fundamentais teóricos com alunos de ensino médio e técnico. Há também iniciativas de pesquisadores em inovar e trazer para o contexto da disciplina ministrada experimentos que possam contribuir com o aprendizado, com projetos isolados e com o objetivo de demonstração de fenômenos matemáticos ou físicos (d’ABREU, 2007).

A RP deve ser aplicada como atividade de experimentação e de construção de instrumentos, visando o aprendizado científico e tecnológico, integrando as demais áreas de conhecimentos, como define Lopes (2008). O principal objetivo não é priorizar o domínio técnico em robótica, mas utilizar ferramentas para construir e manipular artefatos que venham reproduzir, experimentar e analisar conceitos teóricos de maneira simples e objetiva. Existem diversos recursos tecnológicos para construções de modelos reais, como kit de montagem de protótipos Lego® Mindstorms², a plataforma aberta de desenvolvimento e controle de componentes automatizados Arduino³ e o ambiente de controle de robôs virtuais RoboMind⁴.

1.3.2. PBL

A PBL foi utilizada inicialmente em cursos de medicina no início da década de 1970 na Faculdade de Medicina da Universidade McMaster, no Canadá (Rhem, 1998). Esta metodologia propõe que o responsável pela criação do aprendizado são os próprios alunos. Na PBL, o professor age como um facilitador e também seleciona o problema a ser resolvido. Este problema precisa proporcionar a necessidade de manipular diversos conceitos e trabalhar diretamente a motivação dos alunos. O aluno deve analisar as diversas vias de acesso até a solução. Os alunos são incitados a encontrar soluções por si e responsáveis pelas fontes de informação para o seu aprendizado. Qualquer fonte pode ser utilizada. Nesta metodologia, o professor não precisa dominar o assunto, pois age apenas como facilitador, criando questionamentos aos alunos, de uma forma onde eles mesmos criam o seu conhecimento. O processo gera um aprendizado tanto para o aluno como para o professor (Savery & Duffy, 1998).

Para trabalhar com o PBL, em princípio é necessário trabalhar com grupos de alunos. Uma característica do PBL é aperfeiçoar as habilidades de trabalhar em grupo. Os alunos também precisam possuir um conhecimento básico do assunto abordado (Savery & Duffy, 1998).

Segundo Soares & Borges (2011), a metodologia PBL propõe uma estratégia onde o aprender a aprender vale para todos os envolvidos. Sendo assim, um facilitador

¹ Olimpíada Brasileira de Robótica <<http://obr.org.br/>>

² Lego Mindstorms <<http://mindstorms.lego.com>>

³ Arduino <<http://www.arduino.cc>>

⁴ RoboMind <<http://www.robomind.net>>

será responsável por escolher um problema e desenvolver uma atividade. Ao grupo de alunos caberá encontrar uma solução, utilizando para isso as fontes de informação que considerarem mais apropriadas. O trabalho em equipe nessa metodologia é essencial, uma vez que conhecimentos e experiências diferentes podem ajudar na solução. Outro ponto a considerar é que o professor ou facilitador adquire novos conhecimentos ao elaborar e desenvolver essas atividades.

A utilização de robótica ou jogos digitais no desenvolvimento de atividades cativam a criatividade e o processo de criação, devido à necessidade de solucionar os problemas propostos com ferramentas informatizadas. Para que a exploração da atividade possa ser efetiva e cumprir com seu papel de formação e desenvolvimento cognitivo, torna-se preciso métodos de coordenação e delineamento desses projetos com a participação de alunos. Independente de qual ferramenta usada ou propósito da atividade, pode-se caracterizar as fases do desenvolvimento dessas práticas educacionais da seguinte forma: planejamento, implementação, depuração e acompanhamento.

Na fase de planejamento, é essencial o desenho (*design*) do projeto, a definição dos objetivos principais e de quais serão os principais elementos, apontamento de quais questões ou problemas que deverão ser respondidos, e pesquisa de modelos reais representáveis e demais referências para suporte teórico e prático da atividade. Após o planejamento, o trabalho tem como objetivo definir quais serão os objetos e ferramentas necessárias para a montagem dos modelos. Em projetos utilizando robótica, definir quais elementos eletrônicos serão necessários, qual será o formato do protótipo, qual será a rotina de testes e elaboração de toda lógica de programação.

A partir do modelo iniciado, ocorre continuamente a depuração. A depuração consiste em identificar e corrigir problemas e evoluir a atividade através dessa experiência. O aperfeiçoamento dos protótipos torna a atividade extremamente rica na questão do aprendizado, pois estimula a solução de problemas e abre discussões entre os participantes. Atividades de acompanhamento do desenvolvimento do projeto podem ocorrer periodicamente, para uma avaliação por parte do professor. Isto trás para a experiência um contexto similar ao encontrado em projetos reais, trazendo aos alunos, compreensão da importância do seu trabalho. Na conclusão do projeto, também é interessante a criação de um relatório final, relatando todos os problemas encontrados pelos alunos, suas resoluções e experiências vividas.

1.4. Ferramentas e práticas educacionais

Esta seção apresenta exemplos de práticas de aprendizado baseadas em PBL. Na subseção 4.1 são apresentadas sugestões para uso de ambientes robóticos de baixo custo como os *framework* Gogo Board e Arduino.

Os ambientes robóticos são muito importantes na motivação dos aprendizes e na viabilização de experiências multi-ambientes, muito ricas em oportunidades de aprendizado. Soares e Borges (2011) mostraram exemplos de como a robótica pode ser efetiva no apoio a aprendizagem.

Por outro lado, ambientes robóticos, podem também levar a desmotivação, especialmente para alunos que não tenham nenhuma experiência com a manipulação de componentes eletrônicos e mecânicos. Ambientes robóticos sempre demandam algum

custo, por menor que seja, na compra de componentes e ferramentas e na demanda de laboratórios. Apresentam-se assim como opções interessantes propostas que tentam trazer para o ambiente virtual as vantagens da robótica, mantendo a proposta de PBL. Na subseção 4.2 são discutidos os ambientes Scratch, Robomind e Robocode.

1.4.1. Ambientes de robótica de baixo custo

Existem iniciativas de criação de ambientes de robótica de baixo custo de apoio ao ensino. Duas iniciativas que merecem destaque são o *framework Gogo Board* (GGB) (Morelato & Borges, 2008) e o ambiente Arduino (Brock, Bruce, & Reiser, 2009). Ambos tem características muito similares, sendo a principal diferença o fato da Arduino ter um grande conjunto de modelos e de fornecedores de placas previamente montadas e o GGB ter como maior apelo a facilidade para os próprios usuários montarem suas placas, sem a necessidade de um grande ferramental.

O projeto Arduino foi desenvolvido com o objetivo de criar uma plataforma de prototipagem de baixo custo, simples aprendizagem e com arquitetura de *hardware* e *software* de código aberto (ARDUINO, 2012). A plataforma é formada por dois componentes: um elemento de *hardware* que consiste de uma placa eletrônica microcontroladora; e um elemento de *software* composto por um ambiente de desenvolvimento através de uma linguagem de programação. Tanto o *hardware* quanto o *software* tem a documentação aberta e disponível para qualquer usuário utilizar ou alterar, para qualquer projeto ou propósito (BANZI, 2011).

A placa eletrônica microcontroladora do Arduino é constituída por um pequeno circuito (a placa) com um microcontrolador⁵ (“chip”) e várias entradas e saídas, analógicas e digitais, além de uma interface via USB para comunicação com o computador (BANZI, 2011). Atualmente encontramos diversos modelos oficiais e não oficiais, que podem variar para atender projetos específicos. Na Figura 1.1, vemos o modelo Uno, um dos modelos mais utilizados atualmente, pois atende a maior parte das funcionalidades da plataforma. A placa possui um processador ATmega328 (Fig. 1.1 – item 1), responsável pelo processamento de entrada, saída e comandos executados na placa. As entradas digitais e entradas analógicas (respectivamente Fig. 1.1 – itens 2 e 3) para o controle de dispositivos. Para o fornecimento de energia para os dispositivos, há pinos de alimentação (Fig. 1.1 – item 4). A placa pode receber alimentada pelo conector de energia (Fig. 1.1 – item 5) ou pelo conector USB (Fig. 1.1 – item 6), conector esse responsável pela interface de conexão com o ambiente de desenvolvimento integrado (*Integrated Development Environment - IDE*) do Arduino.

⁵ Microcontrolador – circuito integrado que contém todas as funções básicas de controle, como entrada, saídas, gerenciamento de memória, etc.

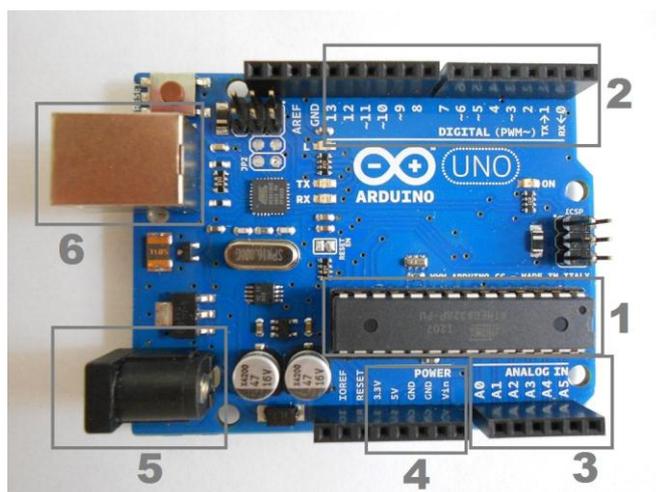


Figure 1.1. Arduino Uno. Processador (1), pinos digitais (2), pinos analógicos (3), pinos de alimentação de dispositivos (4), conector de energia (5) e conector USB (6)

O IDE foi desenvolvido em linguagem Java e a linguagem de programação usada para a programação da placa foi baseada em Linguagem C (BROCK, BRUCE & REISER, 2009). Com o IDE é possível criar os programas (*sketches*) com o conjunto de instruções passo a passo, que posteriormente será transferido ao microcontrolador da placa, para que a mesma interaja com os componentes ligados a ela (BANZI, 2011). A Figura 1.2 mostra a interface principal do IDE do Arduino.

A screenshot of the Arduino IDE software interface. The window title is "File Edit Sketch Tools Help". The main text area contains C++ code for a sketch named "Ping". The code includes a setup function for serial communication and a loop function that sends a pulse to a ping sensor and reads the result. The status bar at the bottom indicates "1" and "Arduino Uno on COM1".

```
File Edit Sketch Tools Help

Ping

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}

void loop() {
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters:
  long duration, inches, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);
}
```

1 Arduino Uno on COM1

Figure 1.2. Tela do IDE Arduino

A utilização da plataforma Arduino em atividades pedagógicas pode ser direcionada para criação de protótipos, automação de maquetes e experimentação de conceitos físicos. Devido a sua interoperabilidade com vários sensores, atuadores e motores, há uma possibilidade bastante grande de utilização da placa. O uso, contudo, demanda uma prática contínua e um estudo prévio de conceitos de eletrônica e programação de computadores. A presença do professor como elemento facilitador se faz necessário, principalmente em projetos iniciais, podendo envolver utilização da plataforma inicialmente sem dar maior detalhamento técnico sobre os componentes eletrônicos. Atividades com as plataformas demandam a concepção do projeto e adequação da atividade ao contexto pedagógico, implementação do modelo com a definição dos componentes necessários e finalmente a construção do dispositivo robótico, em seus elementos de *hardware* e *software*.

Dentre os contextos pedagógicos possíveis, podemos citar criação de modelos para representação de fenômenos relacionados a conceitos de Física e Matemática, como mecânica, termodinâmica, eletricidade, entre outros. Com o uso adequado de motores e sensores, é possível explorar maquetes que tragam conceitos de demais áreas de conhecimentos, oferecendo um ambiente automatizado para experiências e aplicações de conceitos teóricos.

A montagem de maquetes e modelos pode ser resultado de problemas propostos pelo professor pela metodologia PBL. Um exemplo é a proposta da verificação dos conceitos relacionados à natureza da luz e o uso de espelhos côncavos para o experimento “Espelho de Arquimedes”. Tal experimento se baseia na lenda descrita por Alvarenga & Maximo (1997), relatando que no século III a.C. o rei de Siracusa (sul da Itália) pediu ao inventor e filósofo Arquimedes para que projetasse uma arma que pudesse defender a cidade da invasão por navios inimigos. Para isso, Arquimedes projetou uma estrutura com espelhos côncavos para convergir raios solares nos navios de madeira que se aproximassem da costa marítima. Assim, eles seriam incendiados pelo calor gerado no direcionamento desses raios solares.

Como desafio o professor pode propor a criação de um modelo que consiga comprovar a efetividade do projeto de Arquimedes, constatando sua veracidade. Utilizando o recurso de sensores controlados pelo Arduino, o projeto pode criar um modelo esquemático como o proposto na Figura 1.3. Está representado o uso de uma fonte de luz, simulando uma fonte de luz solar, um espelho côncavo para o direcionamento de convergência da luz e um sensor de temperatura LM35. Esse sensor estará ligado à placa Arduino, que fará a leitura da variação de temperatura ocasionada pelo espelho apontado para ele. Ligado ao modelo também poderá existir um display LCD (*Liquid Crystal Display*), controlada pela placa Arduino com o objetivo de informar a variação de temperatura obtida pelo sensor de temperatura.

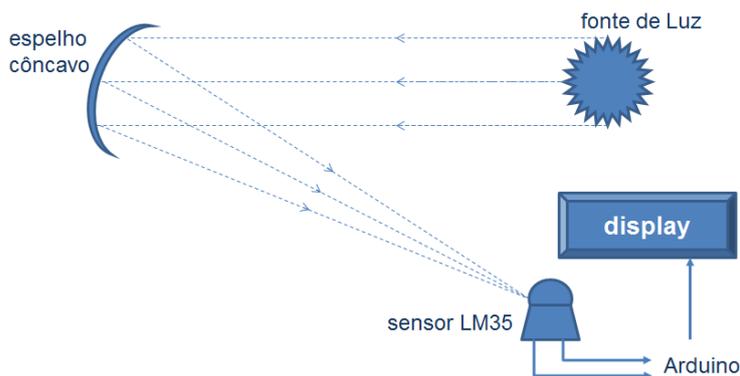


Figure 1.3. Modelo do experimento “Espelho de Arquimedes”

Com o uso do modelo os alunos envolvidos no projeto poderão além verificar conceitos estudados sobre reflexão de luz, analisar aspectos ambientais e de engenharia que podem provar ou não a veracidade, e também quais eram as dificuldades encontradas no projeto de Arquimedes. Com a prática sobre o modelo, discussões sobre posicionamento e tamanho do espelho, intensidade da fonte luminosa e variação de temperatura podem ser criadas e testadas em um ambiente real.

Outro exemplo envolvendo o estudo de mecânica em disciplina de Física, mais especificamente força e trabalho, pode ser demonstrado utilizando um modelo construído com Arduino. Para essa proposta o professor pode questionar os alunos sobre o uso de roldanas conjuntas para suspender objetos. O modelo utiliza uma estrutura fixa de apoio para o conjunto de roldanas e posicionamento de um servo motor controlado pela Arduino. Servo motor é um mecanismo que apresenta o movimento de giro proporcional ao um comando e fará o papel do elemento que executará o trabalho de força potente para suspender um objeto. O modelo inicial apresentado na Figura 1.4 demonstrará o a força necessária a ser aplicada para suspender um objeto utilizando uma roldana fixa (R_1). Para a análise de resultados envolvendo a associação de roldanas, o modelo poderá ser acrescido de uma roldana móvel (R_2) presa ao objeto, como mostra a Figura 1.5.

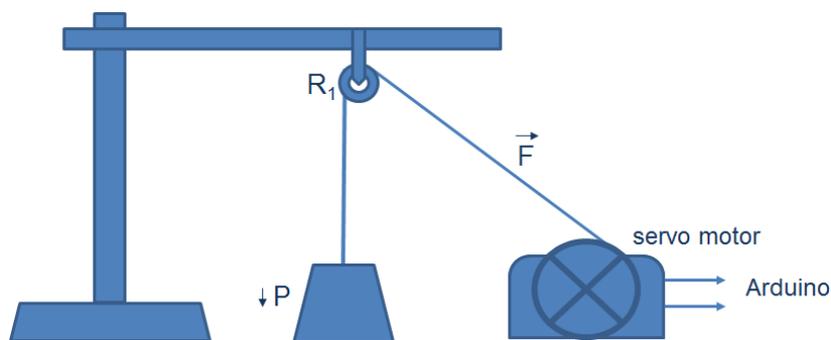


Figure 1.4. Modelo inicial do experimento com roldanas

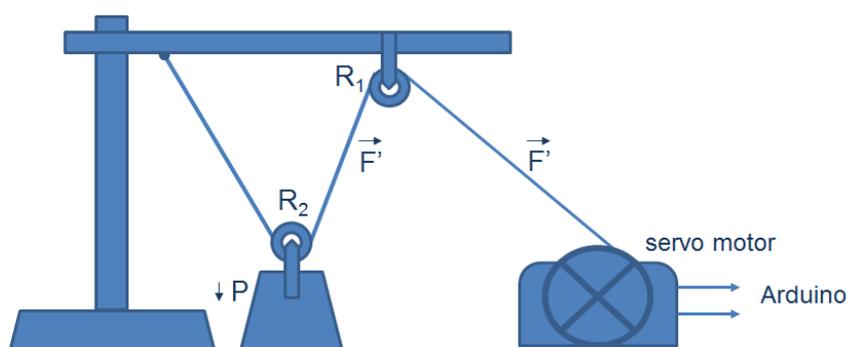


Figure 1.5. Modelo incrementado do experimento com roldanas

Ao final do experimento os alunos terão recurso para constatar o resultado da associação de roldanas em um modelo real, analisando alguns critérios aplicados, como o controle do servo motor para as tarefas (aplicação de força) e medição de tempo gasto para suspender o objeto nas diferentes modelos propostos.

1.4.1.1. Utilização de Robótica Pedagógica no projeto UCA

Esta seção apresenta um exemplo de uso de Arduino em um contexto de uso em escolas públicas de ensino fundamental. A ideia de utilização de computadores de baixo custo do tipo XO e *classmate* (Classmate, 2012). Com base nesses computadores, estão sendo desenvolvidas pesquisas que visam tornar a RP mais acessível, para permitir a sua difusão em escolas públicas (RAMOS et al, 2007).

O projeto “Um Computador por Aluno” (UCA, 2012) teve início quando a Organização Não Governamental “One Laptop Per Child” (OLPC) foi apresentada ao governo brasileiro no Fórum Econômico Mundial em Davos - Suíça, em janeiro de 2005 (Educação, 2012). Em junho daquele ano, Nicholas Negroponte, Seymour Papert e Mary Lou Jepsen vieram ao Brasil especialmente para expor em detalhes esta ideia. Durante o ano de 2007 foram selecionadas, pelo governo brasileiro, escolas públicas de ensino fundamental, para as quais foram distribuídos *laptops* educacionais para alunos e professores. Foi criada uma infraestrutura para acesso à internet. Também foi promovida a capacitação tanto dos gestores quanto dos professores no uso das tecnologias digitais. Em 2010, o UCA entrou na sua fase 2, denominada piloto, na qual se iniciou efetivamente o processo de formação de professores. É no contexto deste processo de formação, em um projeto liderado pelo Núcleo de Informática Aplicada à Educação (NIED-UNICAMP) que o exemplo aqui relatado ocorre.

Em uma das escolas está sendo possível explorar o potencial da robótica pedagógica, como ferramenta capaz de diversificar e enriquecer a forma como se dá a prática de construção de conhecimentos científicos. A partir de atividades práticas de concepção/design, montagem, automação e controle de dispositivos robóticos via computador Classmate, tem sido possível criar situações de aprendizagem nas quais os alunos e professores, no processo de desenvolver atividades robótica pedagógica, se apropriam do uso das Tecnologias de Informação e Comunicação (TIC's) na sala de aula, de forma simples e motivadora. A metodologia utilizada, de cunho qualitativo, tem sido focada no desenvolvimento de atividades de formação de professores para a utilização da robótica como recurso pedagógico. Uma vez formados os professores, a

atividade dever-se-á se concentrar no acompanhamento e avaliação da atuação destes junto aos seus alunos com vistas a identificar indicadores que apontem para uma efetiva contribuição deste recurso no enriquecimento do currículo da escola pública. As atividades de pesquisa envolvendo a formação de professores para o uso da Robótica Pedagógica podem ser destacadas em três etapas distintas descritas a seguir.

A primeira etapa consistiu dos estudos desenvolvidos internamente ao grupo de pesquisas buscando entender como seria implementada a interface entre o Classmate e as placas GGB. Foram propostas dinâmicas e foram adquiridos os materiais necessários para a condução das dinâmicas. Foi coletado também um conjunto de material alternativo de baixo custo (“sucata”): material a base de alumínio, madeira, plástico, papelão, dentre outras peças de padrão não comercial. Com estas peças foi possível construir um dispositivo robótico capaz de ser automatizado utilizando componentes eletro/eletrônicos como motores, sensores e luzes. Este dispositivo robótico, por sua vez, pode ser programado/automatizado, utilizando o programa Scratch. Seguiu-se a montagem e testes experimentais de alguns dispositivos robóticos.

A segunda etapa consistiu do início de atividades na escola, junto aos professores e alunos. Foi feita uma palestra/demonstração sobre o ambiente de Robótica Pedagógica para os professores da escola. Foi definido o grupo de professores que faria parte da equipe. Foram realizadas oficinas piloto na escola. Esta etapa se encerrou com a participação de alunos em um evento no qual puderam expor os projetos desenvolvidos durante as Oficinas Piloto.

O projeto de pesquisa se encontra atualmente na terceira etapa. O foco está na consolidação do processo de formação dos professores com vistas a garantir a sustentabilidade do projeto, envolvendo: a aquisição de material de robótica com recursos da própria escola; práticas de ensino tendo como base metodologias que subsidiem as atividades de RP com as respectivas classes, sob a orientação de pesquisadores do NIED; observações, registros e análise de dados.

1.4.2. Ambientes virtuais de aprendizado

Buscando trazer para o ambiente virtual as vantagens do uso de robótica, em especial a motivação dos alunos, existem ambientes que tentam simular os experimentos robóticos em software. Esta seção irá discutir três exemplos, em suas subseções: o Scratch, o RoboMind e o Robocode.

1.4.2.1. Ambientes virtuais de aprendizado

O RoboMind (Benitti et al., 2009) é um projeto desenvolvido pela Universidade de Amsterdam e dirigido por Arvid Halma com o propósito de criar uma ambiente que oferece uma linguagem de programação simples para movimentação de um robô virtual em uma ambiente bidimensional e que incentive a prática de experimentação com uso de conceitos de lógica de programação e robótica. O ambiente propõe o controle de um robô em uma arena, sendo a arena composta por obstáculos, áreas demarcadas e objetos, na qual é possível simular tarefas de deslocamento de objetos e reconhecimento do ambiente. O robô possui, além de sua movimentação pelo plano bidimensional, funções de sensoriamento (obstáculos, cores, linhas, etc.), demarcação do solo através de um pincel e uma garra acoplada para deslocamento de objetos (HALMA, 2009).

Como podemos observar na Figura 1.6, toda a interface se encontra traduzida em português e é dividida em edição do programa à esquerda; à direita a arena bidimensional de ação do robô; e na parte inferior um console de apresentação de erros e mensagens de execução da rotina do robô. A linguagem de programação possui as instruções traduzidas para português, e constituem o controle dos atuadores e sensores responsáveis pela movimentação do robô, como a função de movimentar para frente (`andarFrente()`), coletar objetos (`pegar()`) e percepção de obstáculos (`temObstáculoDireita()`) (BENITTI et al., 2009).

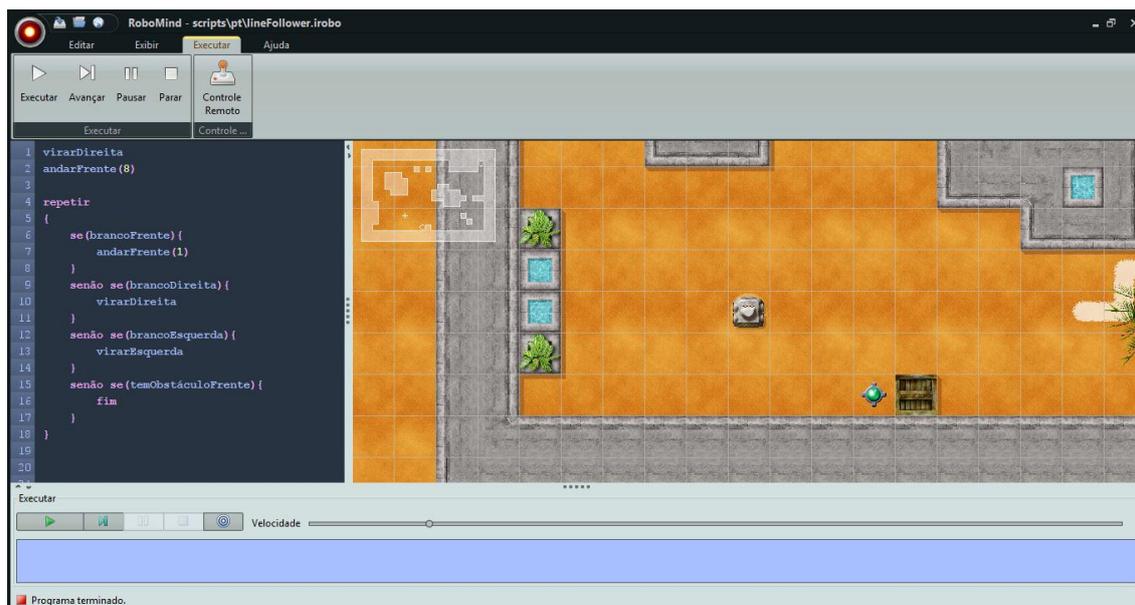


Figure 1.6. Tela do ambiente RoboMind

O ambiente do RoboMind possibilita desenvolver práticas que podem relacionar conceitos de Física, Matemática, Computação, entre outros. Pela possibilidade de definir a arena com planos (*skins*) que representam mapas e áreas demarcadas, pode-se também desenvolver conceitos fundamentais de Geografia, com atividades de posicionamento geográfico e deslocamento por coordenadas, utilizando funções de movimentação específicas. Devido à linguagem de programação e interface estar adequada e totalmente traduzida para língua portuguesa, a curva de aprendizagem da utilização da ferramenta se torna menor. A ferramenta também possibilita o rápido desenvolvimento de ações com o robô virtual, que em comparação a um mecanismo similar real, abstrai vários conceitos que poderiam se tornar empecilhos no momento da aprendizagem e uso do ambiente.

A abstração de vários conceitos técnicos é muito bem vinda quando estamos introduzindo a prática de robótica. O RoboMind permite expandir seu uso em tarefas que exigem uma percepção mais complexa da robótica, pois simula conceitos mais fiéis aos componentes reais robóticos, inclusive com interface com dispositivos reais, como o kit Lego® Mindstorms.

1.4.2.2. Scratch

O Scratch é um software gratuito, criado pelo M.I.T. (Massachusetts Institute of Technology, 2012; Cristovão, 2008). A versão 1.4 pode ser instalada nas plataformas

Windows, Mac e Linux. Scratch foi escrita em Squeak e possui duas versões do código-fonte disponíveis (“Gnu General Public License V 2.0” e “Scratch Source Code License”).

Scratch pode ser utilizado para o aprendizado de linguagem de programação através da criação de histórias em quadrinhos, jogos, animações, além de outros projetos multimídia. Toda produção pode ser compartilhada no site do M.I.T., sendo para isso necessário um cadastro que possibilita conhecer novas ideias, tendo acesso aos códigos dos projetos compartilhados. Também é possível inserir arquivos do Scratch em qualquer outro site. A visualização dos jogos ou histórias em quadrinhos desenvolvidos e enviados para a web não necessitam da instalação do Scratch.

Programar com Scratch é simples. Para iniciar um projeto, não é necessário ter conhecimento prévio de uma linguagem de programação. Para programar, basta arrastar e encaixar os blocos, como se estivesse montando um quebra-cabeça. A lógica é construída a partir de cada proposta de projeto e criatividade do desenvolvedor.

A figura 1.7 ilustra a tela principal do Scratch dividida em 4 áreas que são a base para iniciar um projeto:

- A parte 1 é onde ficam os blocos de comandos utilizados para programação. Eles são divididos em oito grupos separados, com cores específicas: movimento, controle, aparência, sensores, som, operadores, caneta, variáveis são categorias que possuem comandos específicos;
- A parte 2 é a área onde os comandos são arrastados e a estrutura da programação é formada. Ao mover os comandos para a área 2, é possível identificar o grupo de cada comando pela sua cor;
- A parte 3 é onde é possível visualizar o que está sendo programado. Sempre que o Scratch é aberto, o gato aparece;
- A parte 4 mostra os objetos utilizados no projeto que está aberto. O objeto que está sendo editado fica selecionado.

Um exemplo de uso de Scratch, baseado numa apresentação de Brennan (2010), alinhado com o uso de PBL, é descrito a seguir. É uma atividade que deve ser proposta para um grupo de estudantes ou profissionais que não são da área de tecnologia e nem conhecem o Scratch. O facilitador deve fazer uma apresentação sobre informando o que pode ser criado e mostrando um exemplo de um jogo ou animação desenvolvido com o Scratch. Em seguida, fará uma rápida explicação sobre as áreas de construção, conforme explicado anteriormente. Segue-se um projeto com 15 comandos espalhados pela área de programação (figura 1.8).

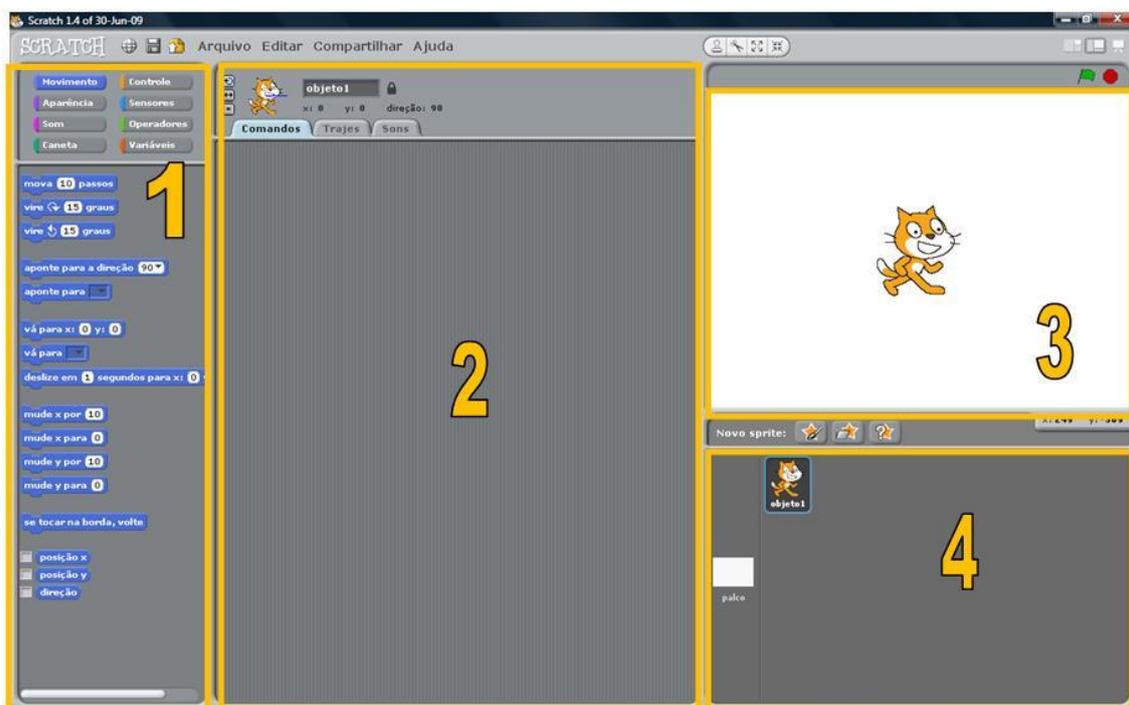


Figure 1.7. Tela inicial do Scratch dividida em quatro áreas

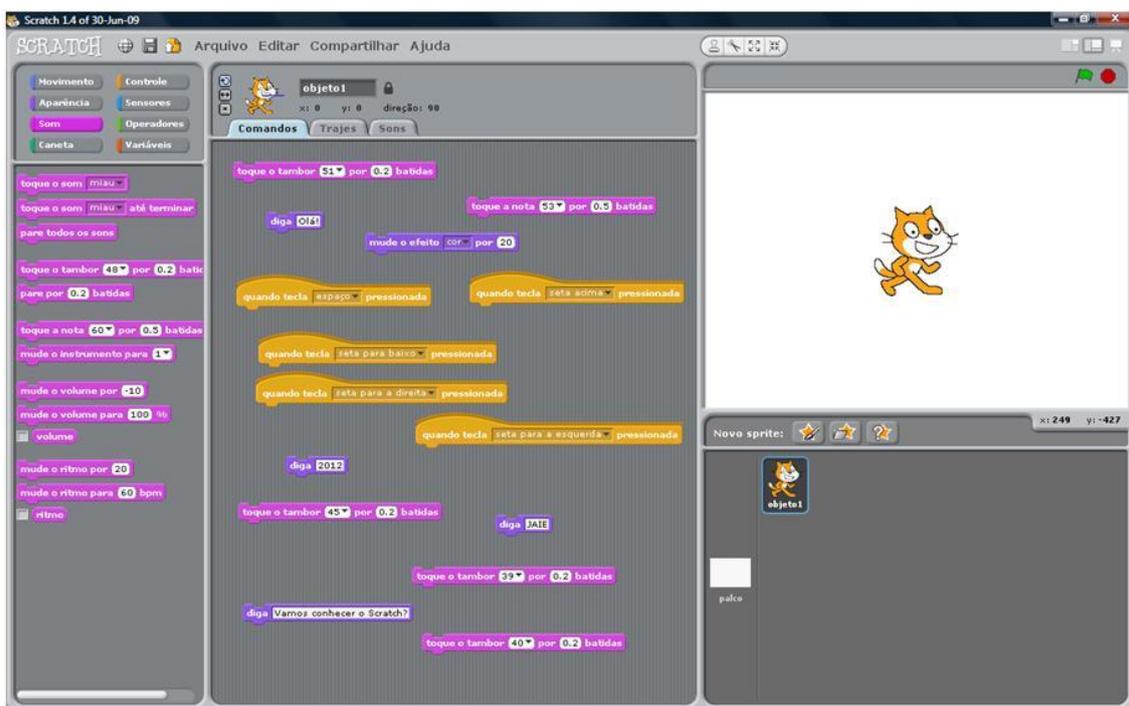


Figure 1.8. Desafio utilizando o Scratch

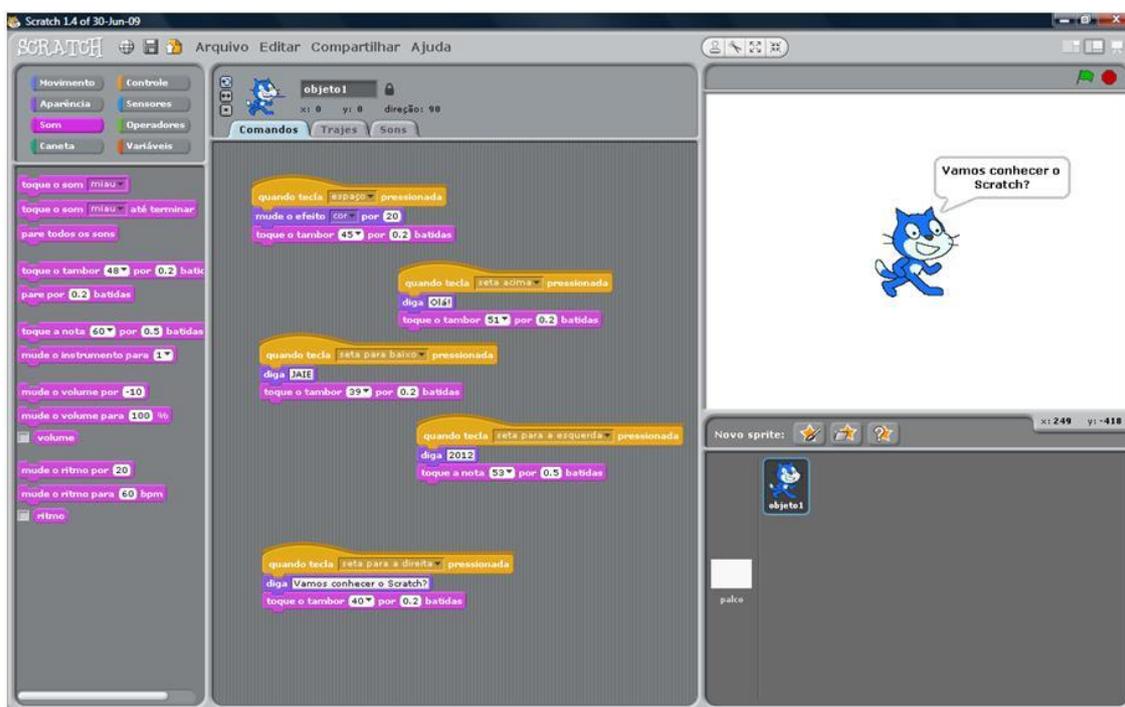


Figura 1.9. Possível solução para o desafio proposto



Figura 1.10. Tela de ajuda do Scratch

O facilitador deve solicitar que os alunos descubram o que esses comandos executarão e como poderão utilizar esses comandos para criar uma animação. Cada grupo terá um tempo para organizar o trabalho e solucionar o problema. O trabalho em equipe é fomentado, potencializando a troca de informações e novas descobertas. Além de encaixar os blocos, os alunos deverão executar a animação e interagir, tendo

alcançado o objetivo quem descobrir que o gato muda de cor, exibe algumas mensagens e sons conforme a tecla pressionada. Um exemplo de solução é apresentado na figura 1.9.

Um grupo curioso poderá descobrir que ao se selecionar o menu “Ajuda” (Figura 1.10) receberá informações completas sobre como iniciar um projeto no Scratch e a descrição de todos os blocos com alguns exemplos e um guia de referência. Se os alunos não explorarem essa possibilidade, cabe ao facilitador incentivá-los.

O Scratch também pode ser utilizado nas aulas de informática, como complemento de estudos para as disciplinas da grade curricular como: português e matemática. No exemplo da figura 1.11, através de uma animação, será possível programar uma atividade para crianças aprenderem a tabuada do 2. Ao clicar na bandeira verde, o gato realizará o cálculo na sequência apresentada na tela. Por envolver conceitos de programação mais avançados envolvendo variáveis, a atividade poderá ser realizada por alunos do ensino fundamental 2 ou médio, tendo como motivação a informação de que o jogo será usado por crianças como uma ferramenta de apoio para as aulas de tabuada.

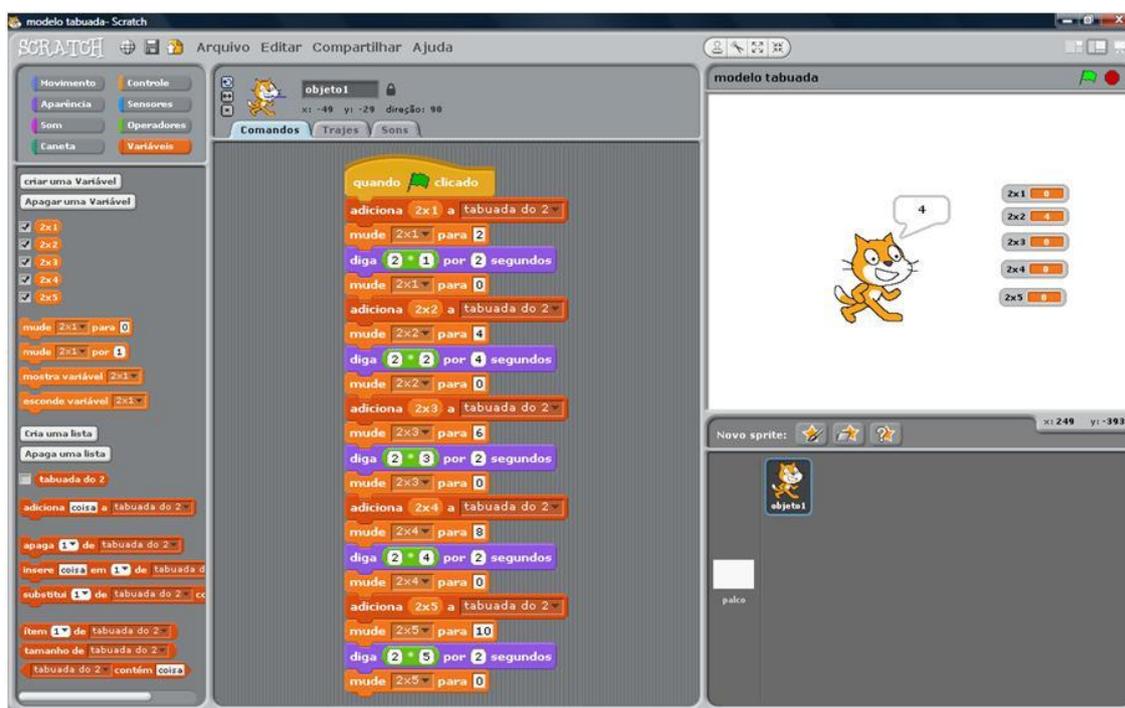


Figura 1.11. Exemplo de programação da tabuada do 2

Brennan & Resnick (2012), definiram sete conceitos computacionais (sequências, loops, eventos, paralelismos, condicionais, operadores e dados) utilizados no Scratch através dos blocos de comandos que são comuns em outras linguagens de programação. Pode-se pensar em exemplos que trabalhem com os sete conceitos.

1.4.2.2. Robocode

Um problema utilizado que pode ser proposto pelo PBL para informática é o Robocode. O Robocode é um ambiente de programação, onde se desenvolve tanques de guerra que batalham entre si, usando algoritmos criados nas plataformas Java ou .NET.

Essas batalhas (LOADHOLTES, 2003) são executadas em tempo real. No Robocode robôs lutam em um campo de batalha. É um problema interessante para ser utilizado pelo PBL, pois traz o aprendizado em Java de uma forma lúdica, a partir de um jogo de computador (O'Kelly & Gibson, 2006).

Robocode foi criado para apoiar o aprendizado a programar, principalmente na linguagem Java, de forma simples e altamente motivadora, por se assemelhar a um jogo de computador tradicional. O jogo foi originalmente iniciado por Matthew A. Nelson, como um trabalho pessoal no final de 2000 e se tornou oficial quando foi levado à IBM, na forma de um *download* no AlphaWorks, em julho de 2001. Ele foi construído e compilado em Java e só requer um JVM (Java Virtual Machine): isso torna possível executá-lo em qualquer sistema operacional, seja Windows, Linux, Mac OS e/ou variações de Unix com Java pré-instalado. Ele também é Open Source, isto é, todo o código está aberto e disponível para qualquer um que tenha o interesse em aprofundar-se no assunto. O ambiente é fornecido sob os termos PL (Eclipse PublicLicense), que são termos de contrato para desenvolvimento de software livre. Recentemente (a partir da versão 1.7.2.0), o Robocode passou a ser compatível com a plataforma. A Figura 1.12 ilustra uma batalha no ambiente (Nicoletto, 2012).

Várias escolas e universidades do mundo todo estão usando o Robocode para ensinar técnicas de programação, inclusive integrando com o desenvolvimento de Inteligência Artificial – IA (LOADHOLTES, 2003). Diversas instituições de estudo e tecnologia do mundo inteiro estão criando torneios de Robocode para incentivar o aprendizado de Java e boas práticas de programação. Esta tendência de torneios se iniciou na Europa (onde o Robocode é mais conhecido), tendo a Irlanda como o grande destaque em termos de sucesso do torneio. Mas hoje há torneios em: Tailândia, Bélgica, Austrália, Nova Zelândia, entre outros países. Desde 2010, o LIAG organiza torneios Robocode⁶, colhendo bons resultados associados à motivação de jovens a se interessar por tecnologia e programação. Um dos resultados interessantes foi o ingresso no curso de tecnologia de informática da FT/Unicamp, onde o LIAG está localizado, de vários alunos que participavam, ainda como alunos de uma escola técnica de uma cidade que fica há mais de cem quilômetros de distância, de um torneio. Isto pode indicar que o envolvimento destes no torneio aumentou bastante a motivação deles para ingressar em um curso superior de informática (Nicoletto, 2012).

Criar um torneio Robocode que seja motivador demanda um planejamento não trivial. É necessário planejar a duração do torneio e criar regras claras, buscando maximizar as oportunidades de aprendizado. Conseguir associar o torneio ao desempenho em uma disciplina pode aumentar o interesse de alunos. Outra possibilidade, com grande resultado até o momento na experiência do LIAG, é o oferecimento de prêmios oferecidos por empresas patrocinadoras que estejam em sintonia com os interesses dos participantes. Buscando viabilizar a criação de muitos torneios seguindo um mesmo arcabouço de regras, o LIAG criou a Liga Robocode. Para criar um torneio Robocode, basta associar-se a liga e seguir as regras pré-estabelecidas, validadas com a experiência do laboratório de três anos no tema.

⁶ <http://www.ft.unicamp.br/liag/wp/robocode/>



Figura 1.12. Uma partida entre tanques programados em Robocode

1.5. Conclusão

Este minicurso apresentou um arcabouço teórico para o uso da robótica no apoio ao aprendizado. O trabalho sugere a integração entre a Robótica Pedagógica e o Aprendizado Baseado em Problemas em uma proposta que seja ao mesmo tempo motivadora e fácil de ser implantada.

Foram apresentados exemplos de uso desse arcabouço em duas frentes, uma de robótica real, utilizando a integração entre computadores e plataformas robóticas de baixo custo como a Arduino. A outra frente busca explorar o potencial da robótica, sem a necessidade de laboratórios, componentes eletrônicos e outras características que podem inviabilizar o uso da primeira. A segunda frente, de robótica “virtual”, foi ilustrada neste trabalho com Robomind, Scratch e Robocode.

O uso da Robótica Pedagógica associada ao Aprendizado Baseado em Problemas mostra-se, pelos exemplos citados, como positiva. Ela não só aumenta a motivação dos alunos para aprender a programação, como os coloca em situações onde eles aprendem a resolver problemas não triviais e os motiva a continuar trabalhando com tecnologia. Em um país onde a demanda por profissionais nessa área pelo mercado é muito grande e não há uma procura por cursos superiores pelos alunos em número compatível com a demanda, ações que motivem os jovens a ingressar na área de tecnologia são importantes.

Agradecimento: FAPESP – apoio em projeto de pesquisa e na participação em eventos

Referências

- Alvarenga, B. Maximo, A., Física: volume único. São Paulo: Editora Scipione, 1997.
- ARDUINO. Arduino. Disponível em: <http://www.arduino.cc/>. Acesso em: set 2012.
- BANZI, M. Primeiros passos com o Arduino. São Paulo: Novatec Editora, 2011. 151 p.
- Benitti, F.B.V., Vahldick, A., Urban, D.L., Krueger M.L., Halma, A.: Experimentação com Robótica Educativa no Ensino Médio: ambiente, atividades e resultados. In: WIE – XV Workshop sobre Informática na Escola 2009, Bento Gonçalves. XXIX Congresso da Sociedade Brasileira de Computação, Bento Gonçalves: SBC, 2009
- Brennan, K. Helping People Get Started with Scratch: Approaches and Trade-offs. MIT Media Labs, 2010. Disponível em: <http://scratched.media.mit.edu/sites/default/files/HelpingPeopleGetStarted.pdf>. Acesso em: 15 de setembro de 2012.
- Brennan, K.; Resnick, M. New frameworks for studying and assessing the development of computational thinking. AERA 2012. Disponível em: http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf. Acesso em: 12 Out. 2012.
- Brock, J. D., Bruce, R. F., & Reiser, S. L. (2009). Using Arduino for introductory programming courses. *Journal of Computing Sciences in Colleges*, 129-130.
- Classmate PC. Disponível em <http://pt.wikipedia.org/wiki/Classmate_PC>. Acesso em 15 Fev. 2012.
- UCA, Conheça o Disponível em <<http://www.uca.gov.br/institucional/projeto.jsp>> Acesso em 15 Fev. 2012.
- Cristovão, H. M. (2008). Aprendizagem de Algoritmos num Contexto Significativo e Motivador: Um Relato de Experiência. In: WEI – XVI Workshop sobre Educação em Computação 2008, Belém do Pará. XXVII Congresso da Sociedade Brasileira de Computação. Belém do Pará : SBC, 2008.
- D'Abreu, J. V. V. (2007). Ambiente de aprendizagem baseado no uso de dispositivos robóticos automatizados. In: Aprendizagem na era das tecnologias digitais. Ed. Valente, J.A., Mazzone, J., Baranauskas, M. C. C., Cortez/FAPESP, São Paulo.
- d'ABREU, J. V. V., GARCIA, M. F. Robótica Pedagógica e Currículo In: Workshop de Robótica Educacional WRE - 2010, Anais de Joint Conference 2010 - SBIA-SBRN-JRI, Workshops, São Bernardo do Campo - SP, 2010. p.01 – 06.
- Educação, Ministério. <http://www.uca.gov.br/institucional/projeto.jsp>. Última visita: outubro de 2012.
- Halma, A. Robomind.net – Welcome to Robomind.net, the new way to learn programming. <http://www.robomind.net>. Acesso em: jul. 2009.
- LOADHOLTES, Nick. IBM's Robocode: A Platform for Learning AI. January 5, 2003.

- LOPES, D. Q. A exploração de modelos e os níveis de abstração nas construções criativas com robótica educacional. 2008. 326f. Tese (Doutor em Informática na Educação) - Universidade Federal do Rio Grande do Sul- URGs, Porto Alegre, 2008.
- LOPES, D. Q, Brincando com Robôs: desenhando problemas e inventando porquês, Edunisc, Santa Cruz do Sul, RS, 2010. 115p.
- M.I.T. Scratch. Disponível em: <http://scratch.mit.edu> . Acesso em: 15 de setembro de 2012.
- Morelato, L. A., Borges, M. A. F. (2008). Alternativas de Baixo Custo para o uso da Robótica Educacional: Construção e Avaliação do Framework GoGo Board In: V Seminário Municipal de Informática Educativa, Anais do XIX Simpósio Brasileiro de Informática na Educação. Porto Alegre - RS: SBC, 2008.
- Nicoletto, L. Estudo sobre Robocode. Trabalho de conclusão de curso. Tecnologia em Informática. Faculdade de Tecnologia. UNICAMP. 2012.
- O'Kelly, J., Gibson, J. P.. RoboCode & problem-based learning: a non-prescriptive approach to teaching programming. SIGCSE Bull. Vol. 38, Nº 3, pag. 217-221, Set. 2006.
- PAPERT, Seymour. A máquina das crianças: repensando a escola na era da informática. Edição Revisada. Porto Alegre: Artmed, 2008. 224 p.
- RAMOS, J. J. G ET al, Iniciativa Para Robótica Pedagógica Aberta e de Baixo Custo para Inclusão Social e Digital no Brasil. In VIII Simpósio Brasileiro de Automação Inteligente (SBAI 2007) Anais, Florianópolis, SC, 2007.
- Rhem, J. (1998). Problem-Based Learning An Introduction. The National Teaching & Learning Forum , pp. 1-4.
- Savery, J. R., Duffy, T. M. (1998). Problem Based Learning: An Instructional Model and Its Constructivist Framework. In: B. G. Wilson, Constructivist learning environments: case studies in instructional design (pp. 135-148). Englewood Cliffs, N.J : Educational Technology Publications.
- Soares, R. F.; Borges, M. A. F. Robótica: aprendizado em informática de forma lúdica. WEI – XXXI Congresso da Sociedade Brasileira de Computação, Natal-RN: SBC, 2011. Disponível em: http://www.dimap.ufrn.br/csbc2011/anais/eventos/contents/WEI/Wei_Secao_1_Artigo_4_Soares.pdf . Acesso em: 15 de setembro de 2012.
- Wagner, Flávio. A Computação como Ciência Básica. Revista da Sociedade Brasileira de Computação. Computação Brasil. Porto Alegre, n. 15. P. 8-11, Abr/mai/jun 2011.