

A utilização do Scratch como ferramenta de apoio no ensino da disciplina de Física

-----  
*The use of Scratch as a support tool in the teaching of the discipline of Physics*

-----  
*Utilización del Scratch como herramienta de apoyo en la enseñanza de la disciplina de Física*

João Carlos Lopes Fernandes<sup>1</sup>  
Marco Antônio Furlan de Souza<sup>2</sup>  
Everson Denis<sup>3</sup>

**Resumo:** Os jogos computadorizados utilizados na educação proporcionam, entre outros benefícios, a motivação para o engajamento e a persistência na realização das tarefas, habilitando o aprendizado personalizado no qual o sujeito aprende por si só, através da descoberta de relações e da interação com o software. Este trabalho apresenta uma solução que busca auxiliar a disciplina de Física utilizando-se do ambiente de programação visual Scratch na elaboração de um simulador simples (jogo) no qual os estudantes podem visualizar e testar conceitos relacionados à cinemática de corpos livres. Esta solução foi concebida, para o estudo de situações relacionadas a queda livre e lançamento horizontal, buscando utilizar conceitos de uma situação real. O jogo possui uma esfera que é rebatida por uma plataforma que se movimenta para que o jogador acerte um determinado alvo. O alvo muda de posição aleatoriamente. O número de acertos e erros mostra o desempenho do participante durante o jogo. O jogador pode alterar alguns parâmetros, como aceleração da gravidade, a altura em metros que a esfera será lançada, a visualização da trajetória da bola, entre outros.

**Palavras-chave:** Aprendizagem. Jogos computadorizados. Scratch.

---

**Abstract:** *The computer games used in education provide, among other benefits, motivation for engagement and persistence in the accomplishment of the tasks, enabling the personalized learning in which the subject learns by itself, through the discovery of relationships and interaction with the software. This work presents a solution that seeks to aid the discipline of Physics using the Scratch visual programming environment in the elaboration of a simple simulator (game) in which students can visualize and test concepts related to the kinematics of free bodies. This solution was designed for the study of situations related to free fall and horizontal launch, seeking to use concepts of a real situation. The game has a sphere that is rebounded by a platform that moves for the player to hit a certain target. The target changes position randomly. The number of hits and errors shows the participant's performance during the game. The player can change some parameters, such as acceleration of gravity, height in meters that the ball will be launched, and visualization of the ball trajectory, among others.*

**Keywords:** *Learn. Computer games. Scratch.*

---

**Resumen:** *Los juegos computarizados utilizados en la educación proporcionan, entre otros beneficios, la motivación para el compromiso y la persistencia en la realización de las tareas, habilitando el aprendizaje personalizado en el que el sujeto aprende por sí mismo, a través del descubrimiento de relaciones y de la interacción con el software. Este trabajo presenta una solución que busca auxiliar la disciplina de Física utilizando el ambiente de programación visual Scratch en la elaboración de un simulador simple (Juego) en el cual los estudiantes pueden visualizar y probar conceptos relacionados a la cinemática de cuerpos libres. Esta solución fue concebida para el estudio de situaciones*

---

1 Doutor em Engenharia Biomédica, Pesquisador, Coordenador da Fábrica de Artigos e Iniciação Científica do Centro Universitário ENIAC.

2 Mestre em Engenharia Elétrica, Professor no Centro Universitário do Instituto Mauá de Tecnologia.

3 Mestre em Engenharia Elétrica, Coordenador e Professor no Centro Universitário do Instituto Mauá de Tecnologia.

*relacionadas con caída libre y lanzamiento horizontal, buscando utilizar conceptos de una situación real. El juego posee una esfera que es rebautizada por una plataforma que se mueve para que el jugador ajuste un determinado objetivo. El objetivo cambia de posición aleatoriamente. El número de aciertos y errores muestra el rendimiento del participante durante el juego. El jugador puede cambiar algunos parámetros, como aceleración de la gravedad, la altura en metros que la bola será lanzada, la visualización de la trayectoria de la pelota, entre otros*

**Palabras-chave:** Aprender. Juegos de ordenador. Scratch.

## Introdução

Os jogos computadorizados na educação proporcionam motivação, desenvolvem hábitos de persistência no desenvolvimento de desafios e tarefas e melhoram a flexibilidade cognitiva segundo Tarouco et al. (2004). Eles são criados sobre princípios sólidos de aprendizado pois inserem o estudante em situações de pensar, de entender, de preparar e de executar ações, onde, inclusive, a falha pode servir como experiência de acordo com McClarty et al. (2012). Jogos computadorizados provêm um maior engajamento para o estudante, habilitando o aprendizado personalizado – o sujeito aprende por si só, através da descoberta de relações e da interação com o *software*.

Na Física, modelos computacionais de sistemas e fenômenos físicos são utilizados para investigar e visualizar conceitos complexos e também podem ser integrados às salas de aula para auxiliar os alunos a aprender e compreender temas científicos complexos como descrito por Anderson e Barnett(2013). Percebeu-se que o raciocínio baseado em modelo pode facilitar o desenvolvimento do entendimento matemático-científico do mundo natural, e alguns educadores defendem que a Física é melhor aprendida não por fórmulas matemáticas, mas sim por meio da experimentação, demonstrações e visualizações, permitindo aos alunos desenvolverem uma compreensão conceitual dos fenômenos físicos, segundo Anderson e Barnett(2013). Assim, por meio de visualizações, simulações e jogos, os alunos podem começar a integrar ideias que inicialmente estão desconectadas, tais como aceleração, velocidade e trajetória, permitindo que eles comecem a fazer conexões a partir de suas experiências.

Dentre os tipos de *softwares* educacionais que podem ser empregados no ensino da Física, simuladores são de especial importância para Jimoyiannis e Komis(2001). Eles proporcionam novos ambientes que visam melhorar potencialidades educacionais e facilitam os alunos em um engajamento ativo. A utilização de simulações no aprendizado permite ainda estabelecer uma ponte entre os conhecimentos prévios dos alunos e a aprendizagem de novos conceitos físicos, ajudando, inclusive, no desenvolvimento do conhecimento científico por meio de uma reformulação ativa de eventuais conceitos que não são efetivamente absorvidos pelos alunos.

Existem diversas características dos simuladores de Física que podem ser destacadas por Jimoyiannis e Komis(2001), tais como:

- Auxiliam os alunos no entendimento de fenômenos e leis físicas por um processo de elaboração de hipóteses e teste de ideias;

- Permitem isolar e manipular parâmetros, auxiliando os alunos na compreensão das relações entre conceitos físicos, variáveis e fenômenos;

- Empregam uma variedade de representações (imagens, animações, gráficos) úteis para o entendimento dos conceitos, relações e processos subjacentes;

- Explicitam representações e modelos mentais sobre o mundo físico;

- Facilitam a investigação de fenômenos que são difíceis de experimentar em sala de aula por causa da complexidade, dificuldade, periculosidade, demora ou custo.

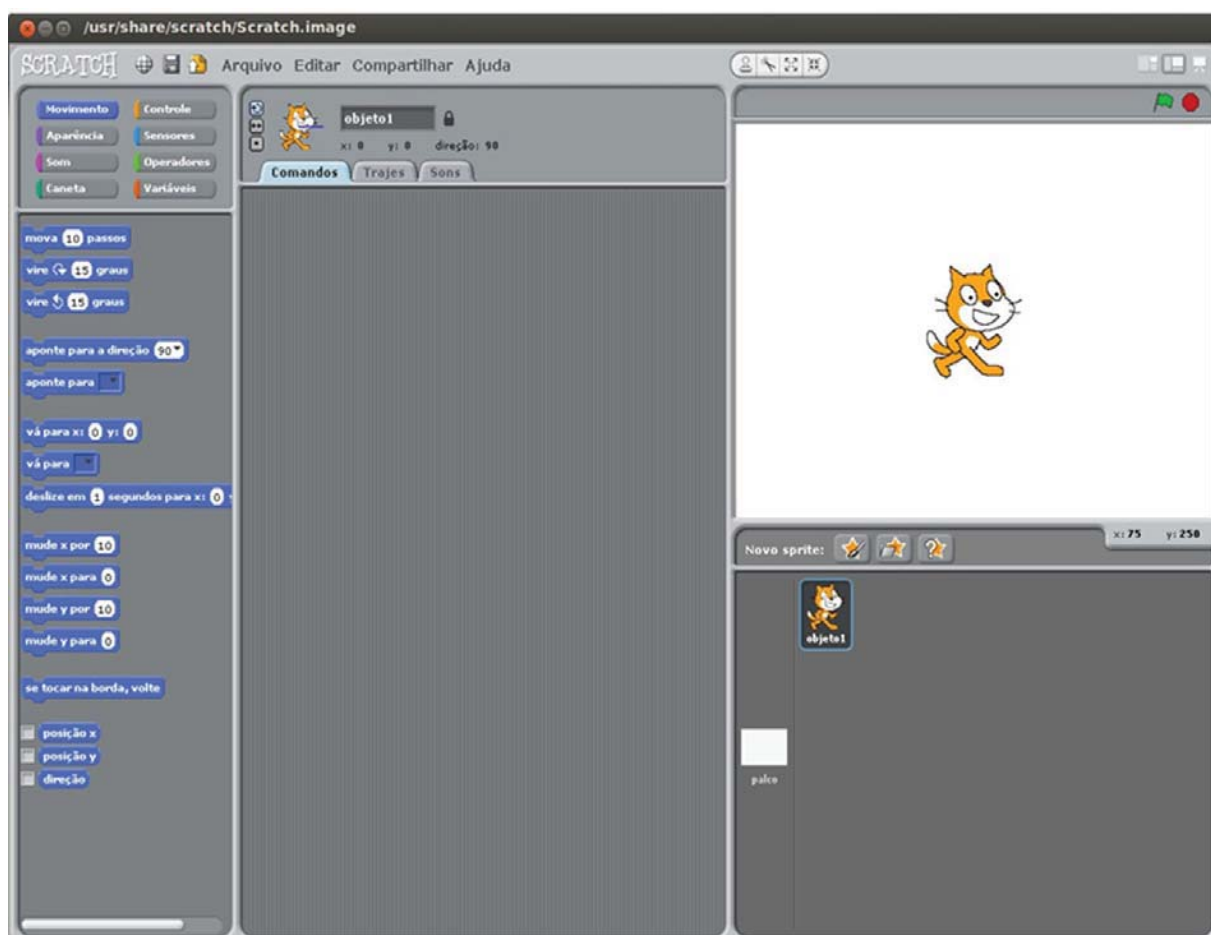
Esse trabalho apresenta uma solução de apoio para a disciplina de Física, que faz parte do ciclo básico dos cursos de engenharia da instituição. Ela utiliza um ambiente de programação visual denominado Scratch que possibilita aos estudantes a aplicação de conceitos relacionados à cinemática de corpos livres.

## O Ambiente de Programação Visual Scratch

O Scratch é um ambiente de programação visual que permite aos usuários criar projetos multimídia interativos segundo Garner et al. (2009) e Maloney et al. (2013). Pode-se criar uma ampla gama de projetos com o Scratch, tais como histórias animadas, jogos e simulações, somente para citar alguns. Seu desenvolvimento começou em 2003 e foi lançado publicamente em 2007, sendo um *software* livre e disponível em cerca de 50 línguas<sup>4</sup>. O objetivo principal do Scratch é introduzir a programação para quem não tem experiência anterior em programação. O Scratch utiliza uma linguagem gráfica, baseada em blocos. A metáfora empregada pelo Scratch para criar programas é baseada em *storytelling* (contar estórias), ou seja, o programa em si é uma estória que é preparada pelo desenvolvedor.

Com o Scratch, segundo Resnick et al. (2009) é possível criar programas que explorem (com o mínimo de dificuldade) os seguintes conceitos de Computação: projeto de programas, laços de repetição, comandos condicionais, lógica booleana, variáveis, arranjos e listas, eventos, sincronização e coordenação, *threads*, interação dinâmica e números aleatórios. Ou seja, quase tudo que é necessário para se construir um programa, usados na grande maioria das linguagens de programação atuais. O ambiente gráfico do Scratch está apresentado na Figura 1.

Figura 1- O ambiente gráfico do Scratch.



Fonte: Elaborado pelos autores.

4 <http://info.scratch.mit.edu/>

Neste ambiente, destacam-se:

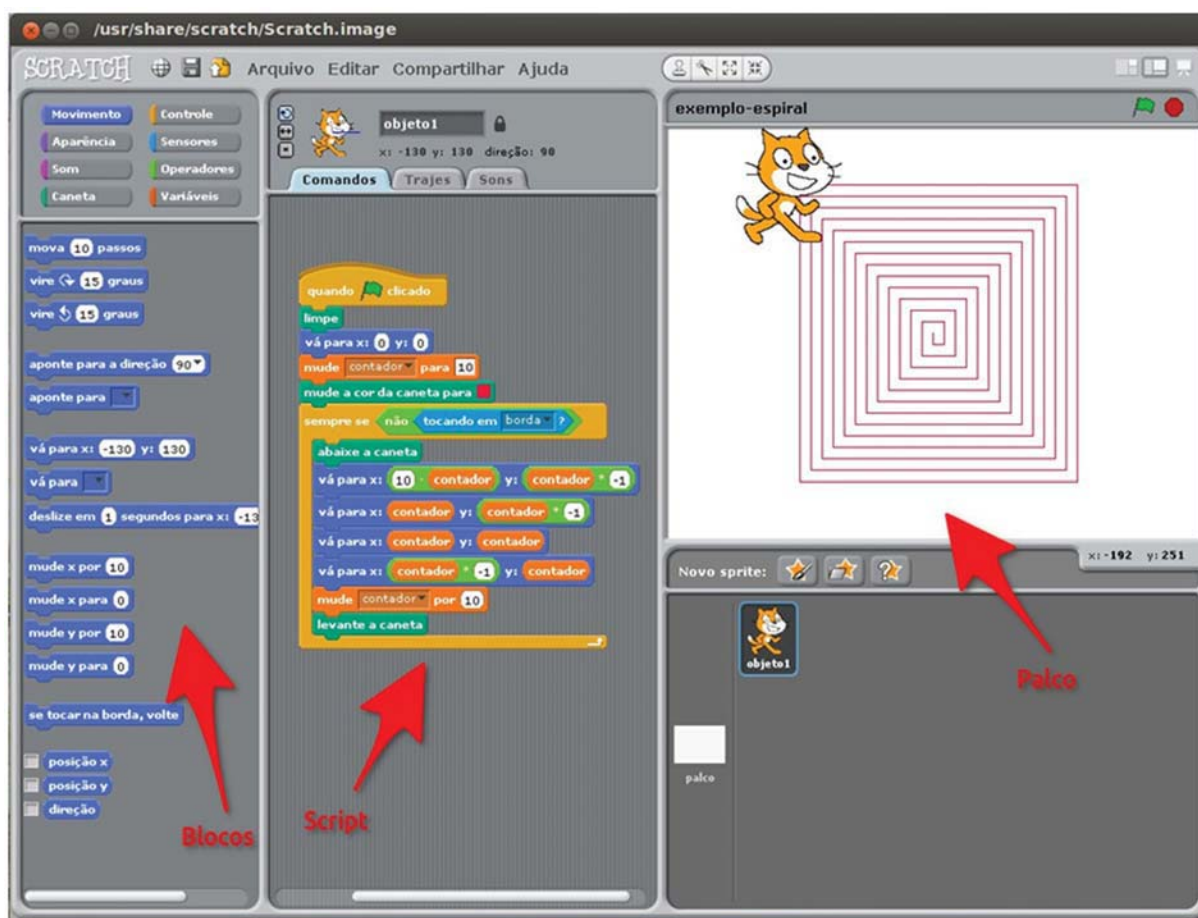
- Blocos: na parte esquerda do ambiente existem as categorias de blocos que podem ser utilizados para criar programas. Existem oito categorias de blocos: movimento (trata do comportamento dinâmico das animações); aparência (apresentação das animações); som (responsável pelo áudio nas animações); caneta (permite escrever linhas com diferentes cores e espessuras); controle (executa comandos necessários ao controle do fluxo dos programas); sensores (permite a leitura e seus valores influenciam na execução de um programa); operadores (servem para realizar operações lógicas matemáticas e operações de textos dentro dos programas); variáveis (permite a criação de variáveis novas ao programa).

- Seção de *script*: é o lugar do ambiente onde um programa (denominado de *script* no Scratch) é elaborado e executado. Nele se encadeiam os blocos disponíveis para o desenvolvimento do programa. Pode-se definir um *script* para cada personagem de um programa e os comandos apresentados alterarão o comportamento deste personagem assim que executados;

- Palco: é a área onde as animações decorrentes de um *script* acontecem. O personagem animado pela execução de um *script* é denominado de *sprite*. Os Sprites podem ser desenhados dentro do próprio ambiente do Scratch ou fora e depois importados. Um mesmo *sprite* pode ter sua forma alterada durante a execução de um *script* e esta nova apresentação é denominada de traje.

A Figura 2 exemplifica um *script* implementado usando o Scratch, que desenha uma espiral quadrada no palco, destacando os componentes do ambiente vistos anteriormente:

Figura 2- Exemplo de um *script* para Scratch.



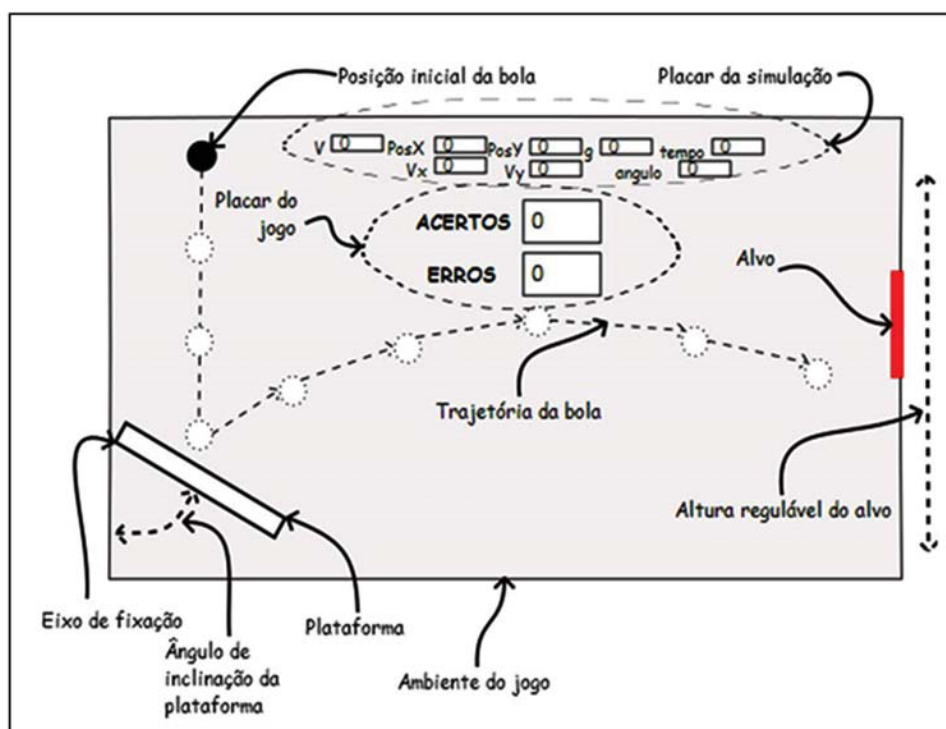
Fonte: Elaborado pelos autores.

## A Concepção do Simulador

O simulador foi concebido, para o estudo de situações de queda livre e lançamento horizontal, procurando utilizar conceitos de uma situação real. O jogo consiste de uma esfera que é rebatida por uma plataforma móvel controlada pelo jogador, de forma que acerte um alvo. Esse alvo muda de posição aleatoriamente a cada jogada. Existe uma pontuação de acertos e erros que mostra o desempenho do participante durante o jogo. Alguns parâmetros podem ser configurados e alterados pelo jogador, tais como aceleração da gravidade, altura em metros em que a esfera será lançada, visualização em tempo real da trajetória da bola, entre outros.

A Figura 3 descreve com detalhes a concepção deste simulador.

Figura 3- Concepção do simulador de Física.



Fonte: Elaborado pelos autores.

## A Implementação do Jogo Usando o Scratch

O processo de criação do simulador no Scratch está descrito nas seções a seguir, nas quais, por restrições de espaço, resume-se o trabalho executado. Os códigos dos *scripts* referenciados em cada seção estão apresentados nas figuras 6 a 11.

### Criação dos personagens

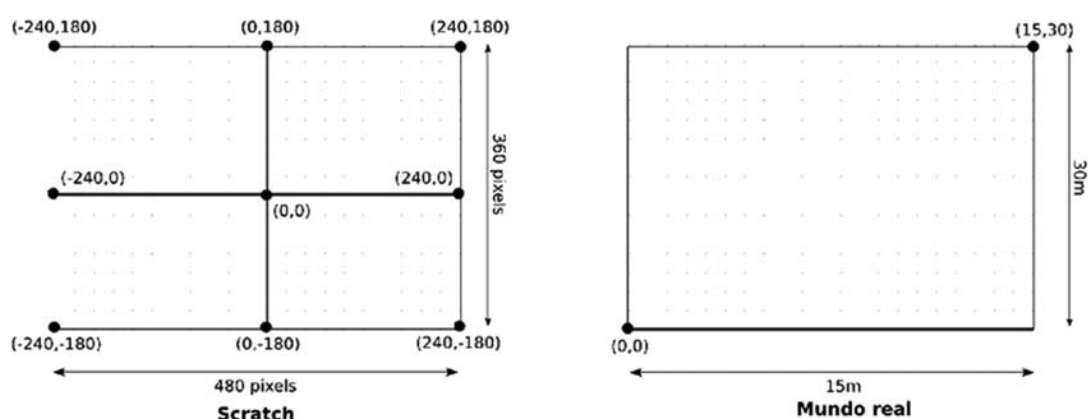
A primeira atividade realizada foi a criação dos personagens. Em um jogo sofisticado, os personagens podem requerer um trabalho artístico muito elaborado, mas, neste jogo os personagens são figuras simples. Com o editor de personagens do próprio Scratch foram definidos os seguintes personagens: bola, plataforma, alvo, rótulos das pontuações. Esses personagens são então posicionados na tela de acordo com a concepção inicial do jogo.

## Inicialização do jogo e conversão de coordenadas

Como este simulador deve refletir uma situação do mundo real que envolve a queda livre de um corpo, foram determinadas algumas premissas: o chão do palco é a referência do ambiente físico, de altura zero; foi definida a posição inicial da bola, para que, partindo do repouso, a sua velocidade seja incrementada de acordo com a aceleração da gravidade,  $g$ , cujo valor adotado foi de  $10\text{m/s}^2$ .

No entanto, o palco do Scratch (onde ocorrem as animações) possui medidas e origens diferentes daquelas do mundo real. No Scratch, as dimensões do palco estão em *pixels*. A largura do palco é de 480 pixels e a altura é de 360 pixels. Além disso, a origem (0,0) está no centro do palco. Por exemplo, se fosse simulada uma situação na qual a área em estudo possui largura de 15m e altura de 30m, seria necessário fazer uma conversão de coordenadas conforme o apresentado na figura 4.

Figura 4- Diferenças de coordenadas entre o Scratch e o mundo real.



Fonte: Elaborado pelos autores.

Estas diferenças de coordenadas afetarão somente o movimento da bola, que se deseja ser o mais próximo de uma situação real. Para esse fim, foram definidas fórmulas para converter as coordenadas por meio de fatores de escala horizontal e vertical e também por correções de deslocamento em relação à origem. Com estes ajustes, foram definidas equações que convertem a coordenada da bola do mundo real,  $(PosX, PosY)$  em um ponto  $(x,y)$  no ambiente do Scratch:

$$x = PosX \times EX + DX$$

$$y = PosY \times EY + DY$$

Por fim, ainda na preparação do ambiente, deve-se ajustar o tempo da simulação. O tempo inicial da simulação será zero e a animação será atualizada em incrementos de 0,01 segundos. As variáveis utilizadas neste simulador estão descritas a seguir:

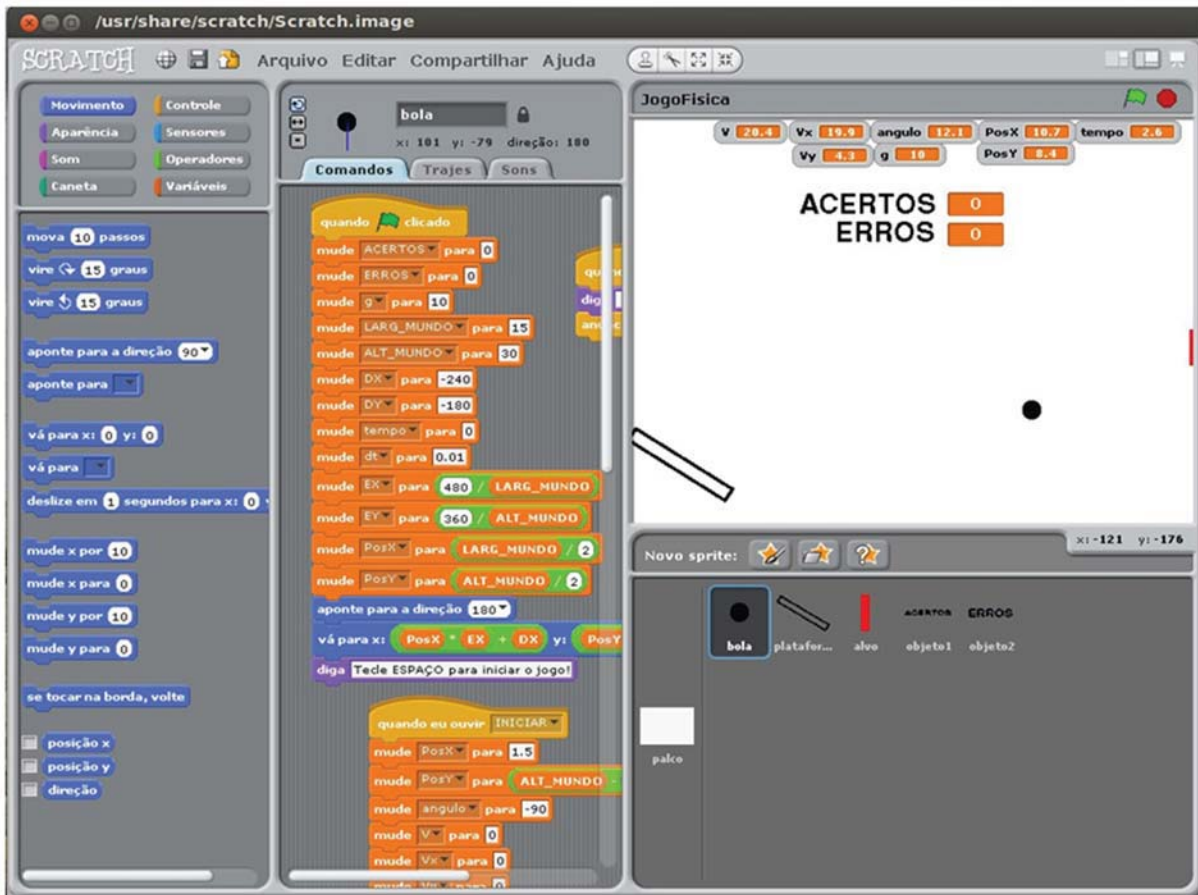
- $g$ : aceleração da gravidade;
- ALT\_MUNDO: altura em metros da situação real que será simulada;
- LARG\_MUNDO: largura em metros da situação real que será simulada;
- DX: valor de ajuste horizontal em relação à origem;
- DY: valor de ajuste vertical em relação à origem;
- EX: escala horizontal que converte metros no mundo real em *pixels* no palco do Scratch;
- EY: escala vertical que converte metros no mundo real em *pixels* no palco do Scratch;

- PosX: coordenada da bola no mundo real, em metros;
- PosY: coordenada da bola no mundo real, em metros;
- Vx: componente vertical da velocidade da bola, em metros por segundo;
- Vy: componente horizontal da velocidade da bola, em metros por segundo;
- V: módulo da velocidade;
- tempo: armazena o tempo decorrido da simulação;
- dt: incremento de tempo;
- angulo: ângulo em graus do vetor velocidade da bola.

### O Jogo em ação

O placar de variáveis serve também como uma forma de analisar as características físicas da simulação, apresentando os valores físicos reais do ambiente (distância em metros, velocidade em metros por segundo). Pode-se alterar os valores das constantes e das variáveis para se perceber por experimentação o efeito de cada parâmetro. Por exemplo, se for definido uma velocidade inicial diferente de zero e for zerada a aceleração da gravidade, seria possível perceber que a bola não seguirá por uma trajetória parabólica. Alterando-se qualquer dimensão do ambiente e o valor da aceleração da gravidade em conjunto, outros fenômenos físicos podem ser observados. A figura 5 representa o jogo em ação.

Figura 5- Instantâneo da tela mostrando o jogo em ação.



Fonte: Elaborado pelos autores.

A inicialização dessas variáveis ocorre: quando o usuário executar o jogo (deve ser pressionado o botão representado com uma bandeira verde) será executado o *script* apresentado na Figura 6.

Figura 6- Script que inicializa as variáveis do jogo.



Fonte: Elaborado pelos autores.

### Movimentação da bola e da plataforma

A movimentação da bola é iniciada assim que o usuário teclar a barra de espaço de seu teclado, representado pelo *script* da figura 7. Esse *script* responderá com um sinal que inicializará o movimento da bola (figura 8), posicionando-a em uma determinada altura e inicializando sua velocidade.

Figura7- Script de inicialização do movimento da bola.



Fonte: Elaborado pelos autores.



Figura 8- Script que inicializará o movimento da bola



Fonte: Elaborado pelos autores.

A figura 9 apresenta o *script que* será responsável pela posição do alvo com uma altura aleatória, ou seja, o movimento vertical. É importante salientar que os *scripts* do Scratch podem ser executados de modo concorrente.

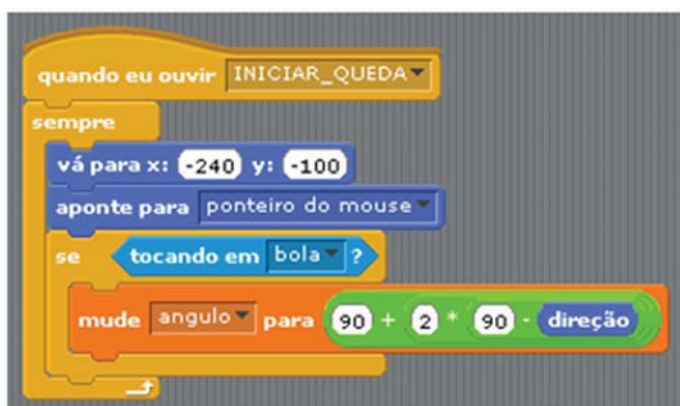
Figura 9-Script responsável pelo posicionamento aleatório do alvo (movimento vertical).



Fonte: Elaborado pelos autores.

O *script* da figura 10 permite que a plataforma acompanhe o movimento do mouse, rotacionando-se de acordo com um pivô na sua extremidade esquerda. Este é o movimento que o jogador realizará de modo a incliná-la para rebater a bola, na tentativa de acertar o alvo. Além disso, existe o teste de colisão com a bola, que também pode ser observado no *script* no bloco de cor laranja com o nome “mude ângulo”, que corrigirá o ângulo que a orienta de acordo com a regra de colisão perfeitamente elástica, ou seja, o ângulo de reflexão deverá ser o mesmo que o ângulo de incidência, em relação à superfície da plataforma.

Figura 10- Script que inicializa o movimento da plataforma.



Fonte: Elaborado pelos autores.

O movimento da bola apresentado no *script* da figura 11 é baseado em intervalos de tempo calculados em incrementos de 0,01 segundo. Antes de calcular a posição e a velocidade da bola, deve-se primeiramente calcular o próximo instante de tempo da simulação:

$$tempo = tempo + dt$$

No *script* da Figura 11, a seguir, enquanto a bola não tocar na plataforma, ela seguirá em uma única direção, respeitando um movimento uniformemente acelerado com aceleração  $g$ . Se houver um choque com a plataforma, haverá uma mudança na direção, e o movimento continuará, possivelmente, acrescentando uma componente horizontal da velocidade. Se, durante sua movimentação, a bola tocar em alguma borda do ambiente, é registrado um erro e se a bola tocar no alvo é registrado um acerto.

Figura11- Script responsável pelo movimento da bola.



Fonte: Elaborado pelos autores.

Com o novo tempo, calcula-se a nova velocidade e a posição da bola, bem como sua posição é atualizada no palco. A velocidade pode ser decomposta na componente velocidade horizontal e vertical de acordo com o ângulo atual que indica a direção do seu movimento.

A posição da bola, que também pode ser decomposta em duas direções, deve ser incrementada pela distância percorrida no intervalo de tempo. Como já se conhece a velocidade atual, então calcula-se:

$$PosX = PosX + Vx \times dt$$

$$PosY = PosY + Vy \times dt$$

Para o próximo intervalo, é necessário corrigir a componente vertical da velocidade, pois o movimento vertical é uniformemente acelerado por  $g$ . Trata-se de um decréscimo da componente vertical da velocidade por unidade de tempo – quando a bola está caindo, aumentando seu módulo (pois a velocidade vertical já é negativa):

$$Vy = Vy - g \times dt$$

E então o novo módulo da velocidade é calculado:

$$V = \sqrt{(Vx)^2 + (Vy)^2}$$

Com o qual, calcula-se a nova inclinação:

$$angulo = \arctan \frac{Vy}{Vx}, \text{ se } Vx > 0$$

$$angulo = \frac{\pi}{2} - \arctan \frac{Vy}{Vx}, \text{ se } Vx < 0$$

Por fim, é importante determinar se a bola se chocou contra o alvo ou não – que então contará um acerto ou um erro.

$$Vx = V \times \cos(angulo)$$

$$Vy = V \times \sin(angulo)$$

## Conclusão

Atualmente os jogos computadorizados na educação, proporcionam motivação aos alunos, em sua aprendizagem. Neste trabalho, apresentamos uma solução que buscou auxiliar a disciplina de Física, desenvolvendo um simulador (jogo) construído com auxílio do ambiente de programação visual Scratch. O intuito do jogo foi permitir que os alunos compreendessem situações relacionadas a queda livre e lançamento horizontal abordados na disciplina, através de situações que simulavam a realidade. O jogo tentou elucidar de forma gráfica a cinemática de corpos livres que foi explanada de forma teórica pelos professores. Com a interação com o ambiente (jogo), percebeu-se uma maior aproximação do aluno com a aprendizagem, tornando-se mais ativo e participativo.

Pode-se observar que o jogo possibilitou ao aluno estudar, visualizar e testar conceitos relacionados ao conteúdo recebido em sala de aula.

Pelos resultados preliminares das simulações, ficou evidente o potencial do Scratch; ele poderá ser utilizado para outras disciplinas do ciclo básico da engenharia, através da criação de novos *scripts* que ficaram como trabalhos futuros.

## Referências

ANDERSON, Janice L.; BARNETT, Melissa. **Learning Physics with Digital Game Simulations in Middle School Science**: Journal of Science Education and Technology. v. 22, n. 6, pp 914–926, dez 2013. Disponível em: <<http://link.springer.com/10.1007/s10956-013-9438-8>>. Acesso em: 17 mar. 2016.

GARNER, Stuart. **Learning to Program from Scratch**. IEEE. Ago 2009. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5194273>>. Acesso em: 25 nov. 2016.

JIMOYIANNIS, Athanassios; KOMIS, Vassilis. Computer simulations in physics teaching and learning: **a case study on students' understanding of trajectory motion**. Computers & Education. pp. 184-203, nov. 2000. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.455.1401&rep=rep1&type=pdf>>. Acesso em: 23 set. 2016.

MALONEY, John; MITCHEL, Resnick; NATALIE, Rusk; BRIAN, Silverman; EVELYN, Eastmond. **The Scratch Programming Language and Environment**. ACM. v. 10, n. 5, pp 313–338, nov. 2013. Disponível em: <<http://dl.acm.org/citation.cfm?doid=1868358.1868363>>. Acesso em: 17 nov. 2016.

MCCLARTY, Katie Larsen; ORR, Peter M.; FREY, Robert P.; DOLAN, Victoria Vassileva; ARON, McVay. **A Literature Review of Gaming in Education**. Gaming In Education. Pearson's research report. Jun. 2012. Disponível em: <[http://formative.pearsonassessments.com/hai/Images/tmrs/Lit\\_Review\\_of\\_Gaming\\_in\\_Education.pdf](http://formative.pearsonassessments.com/hai/Images/tmrs/Lit_Review_of_Gaming_in_Education.pdf)>. Acesso em: 15 nov. 2016.

RESNICK, Mitchel; JOHN, Mayer; ANDRÉS, ANDRÉS, Monroy Hernández; NATALIE, Rusk; EVELYN, Eastmond; KAREN, Brennan; AMON, Moyo. **Scratch: programming for all**. ACM. v. 52, n. 11, pp 60-97, nov. 2009. Disponível em: <<http://dl.acm.org/citation.cfm?id=1592779>>. Acesso em: 27 out. 2016.

TAROUCO, Liane; FABRE, MARIE Christine; KONRATH, Mary; ROLAND, Leticia. **Jogos educacionais**. III Ciclo de Palestras sobre Novas Tecnologias na Educação. CINTED-UFRGS. Mar. 2004. Disponível em: <<http://www.cinted.ufrgs.br/ciclo3/af/30-jogoseducacionais.pdf>>. Acesso em: 30 jan. 2017.

Recebido em 11 de setembro de 2016  
Aceito em 24 de agosto de 2017