

PROPOSTA DE CONSTRUÇÃO DE UM APLICATIVO MÓVEL UTILIZANDO A PLATAFORMA APP INVENTOR PARA O ENSINO DE LANÇAMENTO OBLÍQUO

Gustavo Gewehr Soares – 135310@upf.br

Carlos Ariel Samudio Pérez – samudio@upf.br

Universidade de Passo Fundo, Instituto de Ciências Exatas e Geociências, PPGECM
Passo Fundo – RS

Resumo: A evolução tecnológica e científica, no processo de ensino-aprendizagem, exige a atualização constante por parte dos mediadores. Contudo, trazer a tecnologia para junto do ensino é desafiador para a maioria dos professores. Visando auxiliá-los neste desafio, é apresentado um tutorial de construção de um aplicativo para ser aplicado na disciplina de Física, no ensino médio, para estudar o lançamento oblíquo de objetos. O aplicativo móvel assume a forma de um jogo de tiro ao alvo, construído através da plataforma *APP Inventor*, disponibilizada gratuitamente pelo *Massachusetts Institute of Technology* - MIT. Essas ações podem auxiliar o aluno a tornar-se sujeito ativo no processo de aprendizagem e responsável pela construção do conhecimento. Esse trabalho mostra, para o professor, como pode ser feita a construção do jogo e quais os problemas que possivelmente irá enfrentar.

Palavras-chave: Produto Educacional. APP Inventor. Lançamento Oblíquo.

1 INTRODUÇÃO

Decorrente da evolução científica e tecnológica, a informática é ubíqua e permeia toda a sociedade contemporânea, afetando a vida de todos. Aliar esse fato à forma de ensinar chamada de tradicional favorece a evasão escolar, a falta de interesse do aluno e a aprendizagem mecânica. Evitar isso e despertar o interesse do discente com o intuito de favorecer a aprendizagem é um problema comum para os professores segundo Moreira (1999). Partindo dessa premissa, o objetivo desse produto educacional é ampliar as alternativas de atividades disponíveis aos professores para o ensino de lançamento oblíquo, através da criação de um tutorial de construção.

Uma das premissas básicas do processo de ensino é “tornar o aluno um agente ativo, (co)responsável pelo processo de ensino e aprendizagem” (ARAÚJO; MAZUR, 2013, p. 364). Isso é possibilitado por um ambiente construcionista, contextualizado e significativo, definido por Schlúzen (2000, p. 82) como “um ambiente favorável que desperta o interesse do aluno e o motiva a explorar, a pesquisar”. Tendo isso em mente, sugere-se que o professor insira a atividade de maneira construcionista dentro de sua sequência didática, propondo que o aluno construa o próprio aplicativo. Vale ressaltar que este produto é uma proposição, ou seja, ainda não foi aplicado.

2 PROPOSTA DE ATIVIDADE DIDÁTICA

Durante as aulas de Física no ensino médio, há um momento em que o professor precisa mostrar o conteúdo de lançamento oblíquo. Este, por sua vez, é a junção de diversos outros conhecimentos desenvolvidos anteriormente, desde decomposição de vetores até movimento uniformemente variado, o que o torna chave para compreender diversos conceitos pertencentes à mecânica. Portanto, após o professor ter discutido o conteúdo com a turma, é necessário que o aluno construa um significado para o que ele acabou de ver. Nesta direção é sugerido que o professor proponha a construção de um projeto pelo aluno, um jogo de tiro ao alvo.

Para construir o aplicativo é necessário saber decompor um vetor velocidade em relação ao ângulo, calcular o tempo necessário para atingir a altura máxima e descobrir a distância percorrida no eixo horizontal. Porém, para perceber quais são os passos a serem seguidos, precisa solucionar alguns problemas como: descobrir quais informações devem ser inseridas pelo jogador, entender a ordem em que os cálculos devem ser feitos e perceber que a posição é medida a partir de um referencial.

Utiliza-se o software *App Inventor*¹, disponibilizado gratuitamente pelo MIT na internet, para que o aluno construa o jogo. O objetivo é acertar o alvo. Para isso o jogador deve inserir a velocidade inicial e o ângulo. Ao clicar no botão começar, o alvo e o botão lançar aparecem. Quando o botão lançar é clicado o computador calcula a posição final do projétil e o arremessa nessa direção, o aplicativo fica testando para saber se a bola colidiu com o alvo. Se a resposta for positiva é adicionado um acerto. Após isso, aparece o botão tentar novamente, que reposiciona o alvo e faz ressurgir o botão lançar. Então é adicionada uma tentativa.

3 CONSTRUÇÃO DA INTERFACE DO APLICATIVO

A construção da interface do aplicativo é o momento no qual todas as funções e ferramentas disponíveis para programar são escolhidas. Ela é composta por: organizadores, *Canvas*; legendas; caixas de texto; botões e o temporizador. A seguir serão apresentados alguns exemplos e soluções para a sua construção.

Os organizadores servem como uma ferramenta para ajustar o posicionamento das outras peças. Portanto é muito importante prestar atenção ao posicionamento e dimensões de cada um. Abaixo uma visualização de um exemplo de distribuição dos organizadores:

1 – *APP Inventor* disponível para o acesso em: <http://appinventor.mit.edu/explore/>. Acesso em: 16 out. 2018.

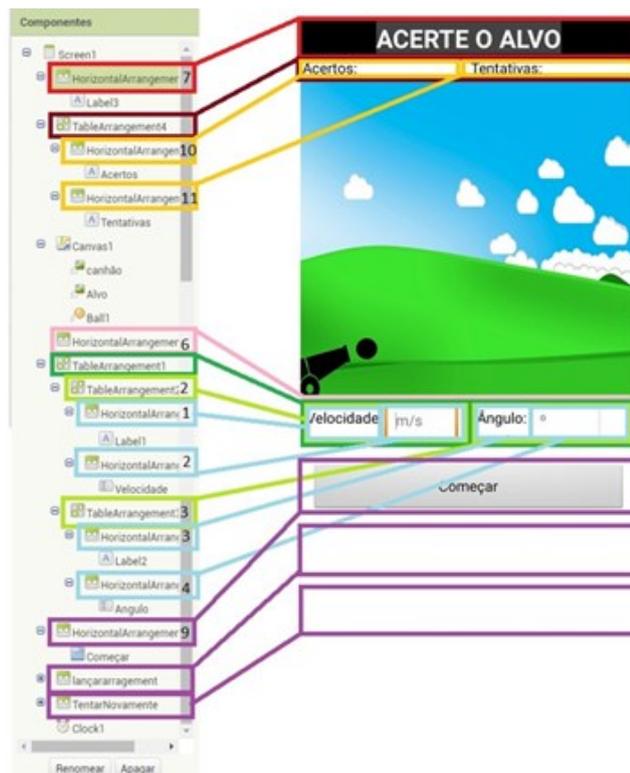


Figura 1 - Exemplo de interface. Fonte: Produzido pelo autor.

O *Canvas* é onde irão acontecer as ações do jogo. Ele toma a proporção de trezentos pixels de altura e preenche a tela horizontalmente, sendo importante adicionar uma imagem de fundo para ficar mais atrativo, que pode ser escolhida livremente pelo estudante. Dentro do *Canvas*, deve ser inserido um *spriteimage*, posicionado no lado inferior esquerdo, que servirá como o canhão. Este leva as dimensões de setenta e cinco pixels de altura e largura. A imagem do canhão provém do repositório de imagens do Google, podendo ser usada qualquer imagem disponível ou até mesmo criada pelo aluno. Por último, a bola tem que ser adicionada, possui um raio de 10 pixels e é posicionada logo à frente do canhão.

As legendas compõem textos que não podem ser editados pelo jogador, mas que o aluno pode escrever o que quiser. Foram adicionadas legendas para: o título; apontar os acertos e as tentativas; definir as caixas de texto da velocidade e do ângulo.

As caixas de textos são locais onde o jogador pode inserir uma informação para ser lida pelo jogo. Elas devem ser utilizadas para receber a informação da velocidade e do ângulo. Na imagem abaixo, elas estão configuradas para ter o tamanho de 35 pixels de altura e 70 pixels de largura, com a fonte de tamanho 14. É importante lembrar de marcar a opção “*somente números*” pois se uma letra for inserida ocasionará erro.

Os botões servem para o jogador ativar efeitos pré-definidos por quem construiu o jogo. Aqui deverão ser utilizados três botões, um para começar, outro para lançar e o último para

tentar novamente. Eles possuem as mesmas dimensões, sendo 45 pixels de altura e 300 pixels de largura e os dois últimos são marcados como invisíveis. É importante lembrar que aqui os alunos podem adicionar a imagem que quiserem.

O temporizador possui a função de medir o tempo. Aqui ele deve ser utilizado para definir a frequência que o computador realizara os testes de contato entre a bola e o alvo. É importante colocar um intervalo de tempo pequeno, o exemplo é de 150ms.

4 PROGRAMAÇÃO DOS BLOCOS DO APLICATIVO

Após construir toda a interface resta programar as ações do jogo. Nessa etapa, o aluno começa a realizar contas e passa a ter a necessidade de abstrair o conteúdo para entender como os blocos funcionam e como fazê-los interagir para atingir o resultado esperado. Abaixo está disponível para o professor um exemplo de resolução.

4.1 Anunciando as variáveis

As variáveis servem para guardar algum valor. Neste caso, usaremos para guardar o valor da velocidade decomposta nos dois eixos, o tempo, a posição e a quantidade de acertos e tentativas.

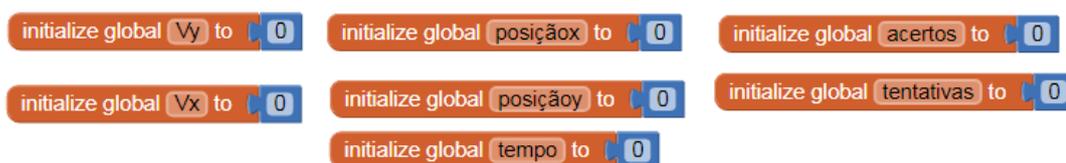


Figura 2 - Exemplo de variáveis utilizadas. Fonte: Produzido pelo autor.

4.2 Descrevendo as rotinas de procedimentos

Procedimentos servem para salvar uma determinada ação que irá se repetir muitas vezes. Nesse caso, foram utilizadas para: mudar a posição do alvo; decompor o vetor velocidade; calcular o tempo; descobrir a posição final da bola; e atualizar a quantidade de acertos e tentativas.

Com o intuito de alterar a posição do alvo a cada tentativa, pode ser criado procedimento que altera os valores do alvo em X e em Y para números inteiros. Existe um limite para esses números para evitar que o alvo fique para fora da área visível da tela.

```

to mudarposicaoalvo
do
  set Alvo . X to random integer from 85 to 320
  set Alvo . Y to random integer from 30 to 270

```

Figura 3 – Procedimento de mudança de posição do alvo. Fonte: Produzido pelo autor.

Para decompor o vetor velocidade basta realizar o cálculo normalmente. O procedimento para isso consiste em multiplicar a velocidade informada pelo cosseno ou seno do ângulo inserido. E então definir as respectivas variáveis para a resposta do cálculo.

```

to Vy
do
  set global Vy to Velocidade . Text x sin . Angulo . Text

```

```

to Vx
do
  set global Vx to Velocidade . Text x cos . Angulo . Text

```

Figura 4 – Procedimento de decomposição do vetor velocidade. Fonte: Produzido pelo autor.

Utilizando a componente da velocidade na vertical basta dividir pela gravidade e obtêm-se o tempo. Após isso, a variável tempo é definida para o resultado da conta. Usando essas informações e multiplicando pela velocidade no eixo X é possível descobrir a distância máxima percorrida na horizontal. Para calcular a altura máxima, deve-se multiplicar a velocidade em Y, descoberta anteriormente e definida como a variável “Vy”, pelo variável tempo, calculada num momento anterior. Subtrair do resultado à metade da multiplicação do quadrado do tempo com a aceleração da gravidade. Tomando atenção para subtrair trezentos pelo resultado do cálculo, pois o ponto mais alto do *Canvas* é o zero.

```

to Vy
do
  set global Vy to Velocidade . Text x sin . Angulo . Text

```

```

to Vx
do
  set global Vx to Velocidade . Text x cos . Angulo . Text

```

```

to descobrirtempo
do
  set global tempo to get global Vy / 9.8

```

```

to posicaoX
do
  set global posicaoX to get global Vx x get global tempo

```

```

to posicaoY
do
  set global posicaoY to 300 - (get global Vy x get global tempo) - (9.8 x (get global tempo ^ 2) / 2)

```

Figura 5 – Procedimentos de cálculos do lançamento oblíquo. Fonte: Produzido pelo autor

O último procedimento é atualizar os acertos e as tentativas. Basta colocar a variável adicionada a mais um e definir a legenda como a união do texto mais o valor da variável.

```

to atualizar_acertos
do
  set global acertos to get global acertos + 1
  set Acertos . Text to join " Acertos: " get global acertos

```

```

to atualizar_tentativas
do
  set global tentativas to get global tentativas + 1
  set Tentativas . Text to join " Tentativas: " get global tentativas

```

Figura 6 – Procedimento de acerto e tentativas. Fonte: Produzido pelo autor

4.3 Programando a interação com o jogador

Nesta etapa é definido o que o jogo deve fazer quando o jogador pressionar os botões. Tal interação vai servir como gatilho pra diversas ações por parte do jogo, mas que devem ser programadas. A primeira ação esperada pelo jogador é começar o jogo. Com isso em mente, quando ele clicar no respectivo botão, o jogo deve mostrar o alvo bem como atribuir uma posição aleatória para o mesmo e mostrar o botão de lançar.



Figura 7 – Programação do botão começar. Fonte: Produzido pelo autor

Após o alvo aparecer, o jogador irá inserir a velocidade e o ângulo que ele julgar necessário para atingir o alvo e então clicar em lançar. Para isso, o jogo deve: realizar todas as contas já expostas; acionar o relógio que está atrelado ao sensor de acerto; mirar a bola e arremessá-la; e, por último, irá esconder o botão de lançar e mostrar o botão de tentar novamente.

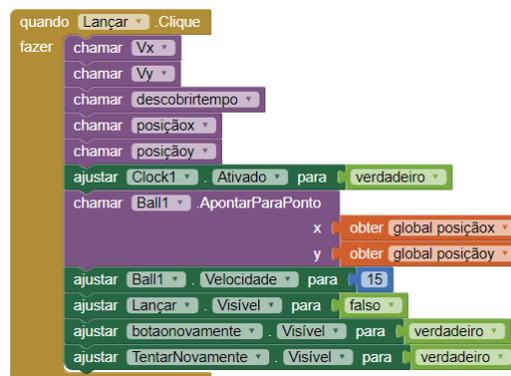


Figura - 8 – Programação do botão lançar. Fonte: Produzido pelo autor

Quando a ação de tornar visível o botão tentar novamente estiver pronta só resta programá-lo. O jogador deve clicar nele para que o jogo: pare a bola; mude a posição do alvo; esconda o botão e mostre novamente o de lançar; e reposicione a bola para o início. Tem de ser notada a necessidade de saber se posicionar no *Canvas* para definir onde é a posição inicial da bola. Neste caso, era no pixel 53 da horizontal e no pixel 246 da vertical.



Figura 9 – Programação do botão tentar novamente. Fonte: Produzido pelo autor

4.4 Programando a interação entre objetos do jogo

Nesta etapa programam-se as consequências das ações desencadeadas pelo jogador. Portanto, o jogo deve perceber se a bola acertou ou não o alvo. Para determinar se houve a colisão basta utilizar o comando do próprio objeto e definir o alvo como o outro objeto. Caso isso ocorra só resta mudar a legenda, atualizar os acertos e parar a bola e o relógio.



Figura 10 – Programação para a bola detectar se acertou. Fonte: Produzido pelo autor

Contudo, não existe um comando para determinar se a bola errou o alvo, por isso é necessário que o jogo fique constantemente rastreando a posição de tal objeto. Com esse intuito é utilizado o comando do relógio que a cada ciclo procura saber se a posição horizontal da bola é maior que a posição do alvo. Pode acontecer da bola não ultrapassar a posição horizontal do alvo. Portanto, é necessário outro meio para determinar se a bola errou o alvo, nesse caso é só usar o comando do próprio objeto para determinar se ele encostou em alguma borda. Caso isso seja verdade, ele para a bola, muda a legenda, torna visível o botão de tentar novamente, atualiza as tentativas e para o relógio.



Figura - 11 – Programação para detectar se a bola errou o alvo. Fonte: Produzido pelo autor

5 CONSIDERAÇÕES FINAIS

Trazer a tecnologia para perto da sala de aula é completamente necessário. A escola não pode ser considerada obsoleta, uma vez que é intrinsecamente ligada ao conhecimento científico. A atividade proposta consegue suprir essa necessidade já que ela utiliza o computador. Considera-se que o trabalho atingiu os objetivos pois o jogo funciona e representa bem os conceitos do lançamento oblíquo. Cabe ao professor decidir qual a melhor maneira de aplica-lo.

REFERÊNCIAS

ARAÚJO, Ives Solano; MAZUR, Eric. Instrução pelos colegas e ensino sob medida: uma proposta para o engajamento dos alunos no processo de ensino-aprendizagem de física. **Caderno Brasileiro de Ensino de Física**, Florianópolis, v. 30, n. 2, p. 362-384, ago. 2013. Disponível em: < <https://bit.ly/2OqBvOt> >. Acesso em: 16 out. 2018.

LIMA, Ivan Shirahama Loureiro de et al. Criando interfaces para objetos de aprendizagem. In: PRATA, Carmem Lúcia; NASCIMENTO, Anna Christina Aun de Azevedo (Org.). **Objetos de aprendizagem: uma proposta de recurso pedagógico**. Brasília: MEC, 2007. p. 39-48. Disponível em: <<https://bit.ly/2CjIC59>>. Acesso em: 16 out. 2018.

MILÉO, Maria Tereza Rodrigues; SILVA, Juliano Tonezer da. **Estatística descritiva e suas aplicações na planilha excel**. 2017. 70 p. Produto Educacional (Mestre em Ensino de Ciências e Matemática) - Instituto de Ciências Exatas e Geociências, Universidade de Passo Fundo, Passo Fundo, 2017. Disponível em: <<https://bit.ly/2PzeUMc>>. Acesso em: 16 out. 2018.

MOREIRA, Marco Antônio. **Teorias de Aprendizagem**. São Paulo: EPU, 1999.

SCHLÜNZEN, Elisa Tomoe Moriya. **Mudanças nas práticas pedagógicas do professor: criando um ambiente construcionista contextualizado e significativo para crianças com necessidades especiais físicas**. 2000. 53 p. Tese (Doutor em Educação) - Pontifícia Universidade Católica de São Paulo, São Paulo, 2000. Disponível em: <<https://bit.ly/2EzBO5l>>. Acesso em: 16 out. 2018.