



## **DESENVOLVIMENTO DE SIMULAÇÕES EDUCATIVAS APLICADAS A DISPOSITIVOS PORTÁTEIS: UM EXEMPLO VOLTADO AO ENSINO DE FÍSICA**

**Jhonatan da Silva Santos** - jhonatansantos296@hotmail.com

Faculdade Municipal “Professor Franco Montoro”, FMPFM  
Rua dos Estudantes, s/n, Cachoeira de Cima, Caixa Postal: 293  
13845-971 – Mogi Guaçu – SP

**José Tarcísio Franco de Camargo** – jtfc@bol.com.br

Faculdade Municipal “Professor Franco Montoro”, FMPFM  
Centro Regional Universitário de Espírito Santo do Pinhal, UNIPINHAL  
Centro Universitário Internacional, UNINTER  
Rua dos Estudantes, s/n, Cachoeira de Cima, Caixa Postal: 293  
13845-971 – Mogi Guaçu – SP

**Estéfano Vizconde Veraszto** – estefanovv@cca.ufscar.br

Universidade Federal de São Carlos  
Depto. de Ciências da Natureza, Matemática e Educação, UFSCar, CCA  
Rodovia Anhanguera, Km 174  
13604-900 – Araras – SP

**Jefferson Henrique Rodrigues** - jefferson-\_rodrigues@hotmail.com

Faculdade Municipal “Professor Franco Montoro”, FMPFM  
Rua dos Estudantes, s/n, Cachoeira de Cima, Caixa Postal: 293  
13845-971 – Mogi Guaçu – SP

**Resumo:** *Este artigo apresenta os resultados de um projeto que consistiu em realizar o desenvolvimento de um jogo educativo para a plataforma Android, utilizando a linguagem de programação LUA. Foi também utilizada a ferramenta de desenvolvimento conhecida como SDK (Software Development Kit) Corona Simulator. O desenvolvimento partiu de pressupostos teóricos educacionais para a conclusão do trabalho, considerando que o potencial de aplicativos audiovisuais e interativos podem trazer contribuições significativas para o campo educacional. O Sistema, a partir dessas ferramentas, é capaz de simular um modelo físico-matemático para o lançamento de projéteis, transformando um conceito clássico em um jogo, buscando assim facilitar o aprendizado deste conteúdo.*

**Palavras-chave:** *Desenvolvimento, LUA, Corona, Balístico, Jogo.*

### **1. INTRODUÇÃO**

Os jogos eletrônicos, além de serem híbridos, pois, para a sua elaboração, envolvem-se programação, roteiro de navegação, design de interface, técnicas de



animação e usabilidade, não se deixam agarrar em categorias e classificações fixas (MOITA, 2007).

Embora a experiência humana tenha sido sempre mediada através do processo de socialização e da linguagem, é a partir da modernidade, com o surgimento de suas mídias típicas de massa (o impresso, depois os sinais eletrônicos), que se observa um enorme crescimento da mediação da experiência decorrente dessas formas de comunicação (MOITA, VERASZTO & CANUTO, 2011).

Paul Gee (2004), em seu recente livro, “What video games have to teach us about learning and literacy”, desenvolve a ideia dos jogos eletrônicos enquanto forma de dotar indivíduos com experiências cruciais para o desenvolvimento cognitivo humano.

Além disso, o autor também aponta que os jogos eletrônicos consistem em ferramentas com capacidades efetivas e positivas para promover a educação já que, de acordo com suas afirmações, aqueles artefatos incrementam um potencial de aprendizagem ativo e crítico (AARSETH, 2005; VERASZTO et al, 2009).

Nesse sentido, esse artigo apresenta os resultados da concepção e desenvolvimento de um jogo eletrônico formulado a partir de um campo de batalha onde se encontra um canhão com objetivo de destruir um castelo que se encontra atrás de uma torre. O usuário visualiza a distância entre o canhão e o castelo, porém antes de disparar o canhão, o jogador deve indicar a velocidade da bala e a inclinação do canhão.

O usuário estará limitado a um tempo de sessenta segundos e a três chances para destruir o alvo. Se errar o lançamento e acertar a torre, o lançamento poderá voltar e destruir o canhão. A cada canhão destruído o jogador será punido com menos mil pontos na sua pontuação. O jogador pode usar até três canhões, um de cada vez. Quando um canhão for destruído outro aparece, até se esgotarem seus canhões. Se os três canhões forem destruídos ou o tempo se esgotar, uma mensagem será exibida: "Você Perdeu".

A cada acerto no castelo o usuário ganhará cem pontos. Por sua vez, quando o castelo for destruído, o jogador receberá quinhentos pontos e um novo castelo surgirá, em uma nova localização.

## **2. CORONA SDK**

Corona SDK (Software Development Kit) é um kit de desenvolvimento de software que utiliza a linguagem de programação LUA para o desenvolvimento de jogos e aplicativos para dispositivos móveis. Este pacote pode ser executado tanto em um ambiente Windows quanto em Mac, porém, na versão para Windows, o mesmo dispõe apenas do dispositivo associado ao sistema Android. Na versão para Mac, o Corona SDK possui os dispositivos da Apple (iPhone, iPhone4 e iPad) e alguns dispositivos que utilizam sistema operacional Android (Driod, Nexus One, myTouch e Galaxy Tab). Este simulador tem se destacado no mercado para desenvolvimento de jogos em plataformas Android e iPhone.

O simulador Corona SDK (Software Development Kit), foi criado pela a empresa Corona Labs, fundada em 2008, com sua sede instalada em Palo Alto, Califórnia (PEREIRA, 2011).



### 3. LINGUAGEM DE PROGRAMAÇÃO

É um método utilizado para expressar instruções para um computador, ou seja, é um conjunto de regras sintáticas e semânticas que são usadas para definir um programa de computador.

A linguagem de programação permite que um programa de computador seja capaz de especificar precisamente sobre quais dados um computador vai atuar, como esses dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.

O computador só é capaz de trabalhar com sequências binárias de 0 e 1, ou seja, todas as instruções dadas a um computador são sequências numéricas compostas por 0 e 1 (ex: 11001100).

Escrever um programa inteiro usando instruções compostas por 0 e 1 seria inviável. Por isso existem as linguagens de programação, para facilitar a comunicação entre o programador e o hardware, que constitui um conjunto de componentes eletrônicos do computador. O programador escreve instruções bem próximas da linguagem que as pessoas usam para se comunicar; posteriormente, um segundo programa é usado para traduzir o que o programador escreveu para a sequência composta por 0 e 1. Esses programas são denominados “compiladores” ou “interpretadores”, os quais são usados para interpretar as instruções escritas pelo o programador e depois as executa (DIGITALDEV, 2012).

#### 3.1. A linguagem de programação LUA

LUA é uma linguagem de programação poderosa, rápida e leve, projetada para estender aplicações.

É uma linguagem de programação desenvolvida por uma parceria entre a PUC – Rio (Pontifícia Universidade Católica do Rio de Janeiro) e o grupo TECGRAF (Grupo de Tecnologia em Computação Gráfica), com a participação da empresa Petrobras, tendo sido desenvolvida pelos professores Roberto Ierusalimschy, Waldemar Celes e Luiz Henrique de Figueiredo, que por sua vez realizam trabalhos na TECGRAF (laboratório de computação gráfica da Universidade).

Sua criação teve como foco principal desenvolver uma linguagem onde houvesse facilidade no desenvolvimento de aplicações em outras linguagens como C e C++, tentando incorporá-la, com a intenção de estender sua plataforma, pois essas linguagens são pesadas, ao contrario de LUA, que é bastante flexível.

A linguagem já teve várias vitórias e conquistas desde seu desenvolvimento. LUA conquistou o mercado nacional e americano através do crescimento de sua plataforma, que se tornou visível por todo o mundo. O mercado americano se rendeu à sua mobilidade e facilidade para desenvolvimento de games, sendo que até mesmo o diretor de “Guerras nas Estrelas”, George Lucas, fundador da LucasArts Entertainment (empresa produtora de games), se voltou para a linguagem efetuando pesquisas e trabalhos em cima de sua plataforma.

A NASA (Administração Nacional da Aeronáutica e do Espaço), até 2003 utilizava LUA para controlar o armazenamento de gases explosivos nos ônibus espaciais (LUA, 2012).



### **3.2. Aplicações para LUA**

Empresas mundialmente conhecidas por seus projetos também empregam e apoiam LUA. A Microsoft é um exemplo, depois que games da LucasArts, como “Fuga da Ilha dos Macacos” e “Grim Fandango” utilizaram a linguagem.

A Intel, empresa de tecnologia mundialmente presente, e o InCor (Instituto do Coração) também empregam LUA, embora os “games” sejam o principal alvo da linguagem. Outro jogo que utilizou a linguagem de programação LUA foi “FarCry”. Neste software, LUA foi utilizada para desenvolver os gráficos do jogo, inserir e ajustar sons, configurar o uso de armas, além de todos os comportamentos dos adversários do jogo (LUA, 2012).

### **3.3. Escolha da linguagem LUA**

Essa linguagem foi escolhida por ser uma ferramenta usada em muitas aplicações gráficas, com ênfase em sistemas embutidos e jogos.

LUA atualmente é a linguagem de scripts (sequência de passos que o computador vai interpretar para realizar tarefas) mais usada em jogos. Além de ser flexível, é rápida, simples e portátil. Outros fatores como a sua mobilidade, sua facilidade para desenvolvimento de games e por ser uma das mais completas que estão no mercado, contribuem para a sua popularidade entre programadores de jogos (LUA, 2012).

## **4. IMPLEMENTAÇÃO DO PROJETO**

O tema “desenvolvimento de simulações educativas aplicadas a dispositivos portáteis: um exemplo voltado ao ensino de física” foi escolhido por se tratar de um tema que, além de estar em crescimento no mercado de tecnologia, é um enorme passo para o futuro de jogos em dispositivos móveis.

A escolha do projeto se baseia em um jogo educativo, desenvolvido para que os jogadores consigam entender um pouco da física do lançamento balístico. No entanto, a escolha da linguagem de programação LUA se estabeleceu por seu conceito, marcado por ser um software totalmente livre, com toda licença e código aberto para o desenvolvimento, oferecendo uma grande flexibilidade e suporte à programação orientada a objeto.

### **4.1. Implicações educacionais**

Santaella (2003) aponta que qualquer mídia ou meio de comunicação são inseparáveis das formas de socialização e cultura que criam, de modo que o advento de cada novo meio de comunicação traz consigo um ciclo cultural que lhe é próprio. Assim, jogar um jogo eletrônico é como ser alfabetizado de uma nova forma (GEE, 2004). A interação entre o indivíduo e o artefato eletrônico ocorre diretamente no espaço e no tempo, proporcionando motivação e interação, fatores indispensáveis para prender a atenção para suas imagens-sons.

Essa realidade é possível, tanto através de consoles, como de celulares, dispositivos móveis, televisão interativa, internet, etc. Pode-se imaginar que, no caso dos jogos eletrônicos, devido à redução dos custos dos aparelhos eletrônicos, esses artefatos serão de uso comum da maioria dos jovens dentro de pouco tempo (MOITA, VERASZTO & CANUTO, 2011).



Muito frequentemente, são justapostas palavras e imagens de vários tipos e integradas em uma variedade de modos. Em jornais e revistas como também em livros didáticos as imagens ocupam cada vez maior espaço ao lado de palavras. No entanto, para que essas aprendizagens possam ser aproveitadas, os jovens têm que aprender a experimentar o mundo de um modo novo (GEE, 2004). Segundo Moita, Veraszto & Canuto (2011), é possível apontar que, numa aprendizagem ativa, estão envolvidas três ações: experimentar o mundo de formas novas, formar afiliações novas e preparar aprendizagens futuras. Para Gee (2004), não basta ser ativo, é necessário ser crítico e, para isso, entender e produzir significados, ser criativo. Pensando nesses pressupostos rapidamente descritos, o projeto foi desenvolvido seguindo as técnicas descritas na sequência.

#### 4.2. Desenvolvimento do projeto

O projeto tem seu início com a construção do cenário. Uma imagem foi eleita de acordo com a expectativa de um campo de batalha (Figura 1), sendo que a mesma foi editada através de softwares específicos, para se adequar nas dimensões do Corona (480 x 320). Foram implementadas várias imagens, com as mesmas dimensões, em sequência, uma ao lado da outra, para se fazer o “scroll” (movimentação da tela). Esse scroll é necessário para que o jogador possa arrastar ou movimentar a tela no sentido vertical, para identificar e calcular a distância do canhão até o castelo. Dessa forma foi desenvolvida a construção do cenário.



Figura 1 - Cenário do projeto.

#### 4.3. Criação das personagens

O desenvolvimento dos canhões foi realizado através de ferramentas gráficas adequadas, em um estilo “cartoon” (desenho humorístico), para se tornar algo mais engraçado e amigável (Figura 2). Foi criada uma sequência de “sprites” (sequência de quadros com as ações dos personagens), desenhadas através de software apropriado. Para fazer a animação de cada ação do canhão, que são piscar e atirar, essas sequências de sprites foram pintadas em cores diferentes (preto, azul e vermelho), para se diferenciar os três canhões. Após esta etapa, essas sprites foram editadas para serem ajustados os tamanhos das sequências de sprites da personagem, tal que se adequassem ao simulador Corona SDK.

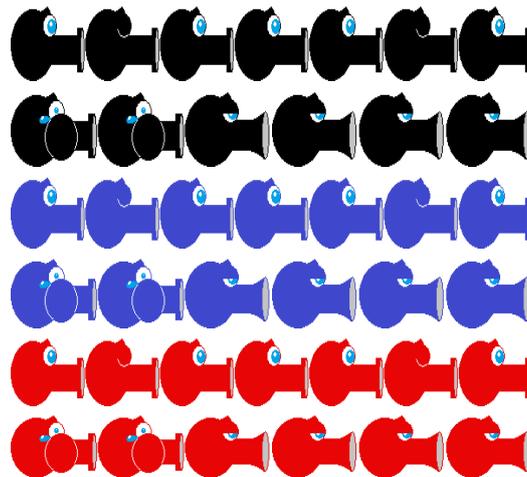


Figura 2 - Canhão gerando animações.

A torre e a bala do canhão (Figura 3) foram desenhadas e editadas pelos autores em um software gráfico apropriado. O castelo utiliza o mesmo método do canhão, com sprites desenhadas e coloridas em software próprio pelos autores. Essas sprites simulam a animação do castelo estar sendo demolido a cada acerto do tiro disparado pelo canhão.

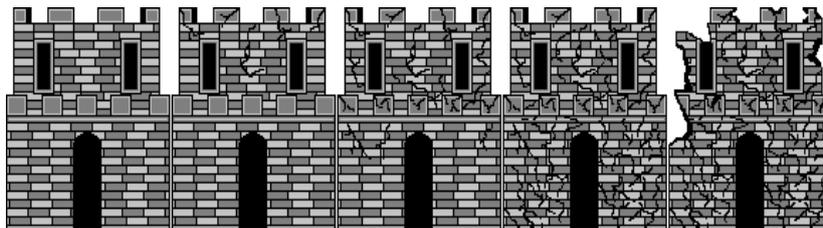


Figura 3 - Castelo gerando animações.

Os personagens canhão e castelo são animados através de um código de transição de imagens. Esse código se encontra abaixo.

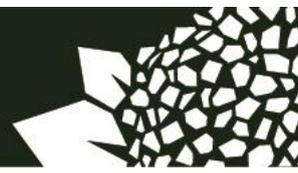
```

local spriteSheetPiscando =
  sprite.newSpriteSheet("canhao1piscando.png", 100, 70)
local spritePiscando = sprite.newSpriteSet(
  spriteSheetPiscando, 1, 7)
sprite.add( spritePiscando, "piscando", 1, 7, 3500, 0).

```

O castelo utiliza o mesmo método do canhão, com sprites desenhadas e coloridas em software próprio pelos autores. Essas sprites simulam a animação do castelo estar sendo demolido a cada acerto do tiro disparado pelo canhão.

O jogo começa em um ambiente com os seguintes personagens: canhão, torre e castelo. Esse ambiente possui dois botões que contém os comandos para o ajuste do canhão, tal que o jogador possa ajustar a inclinação do mesmo, para que o lançamento não colida com a torre. A barra de velocidade marca a velocidade do lançamento, tendo sido desenvolvida para ajustar a velocidade do lançamento, para que seja possível



acertar o alvo (castelo). Nessa barra há um marcador, que indica a velocidade do lançamento da bala.

A torre é uma imagem que possui o comando “static” (estático), o qual faz com que ela fique parada naquele determinado ponto, além de um comando “collision” (colisão), que detecta quando ela é atingida pelo tiro do canhão. O objeto bala de canhão pode ser rebatido pela torre. O canhão e o castelo também possuem o comando “collision” e um comando para retirar a sua visibilidade. Dessa forma, quando o canhão ou o castelo são destruídos, esse comando faz com que eles desapareçam, dando assim a impressão de que foram destruídos. O castelo possui outro comando que faz com que ele reapareça após ser destruído em outro local diferente. Para o castelo reaparecer em outro local diferente, foi utilizado o comando “random” (comando de direcionamento de posições). O código abaixo aplica-se à função random do castelo.

```
local randx = math.random(950, 1800).
```

Quando o canhão é atingido, utiliza-se uma sprite para criar a explosão (Figura 4). O castelo também possui sprites para a sua explosão, tal como o canhão. A diferença é que o castelo possui duas sprites de explosão, sendo uma sprite utilizada quando ele é atingindo, causando uma pequena explosão, e a outra sprite sendo usada quando ele é totalmente destruído após os cinco acertos.

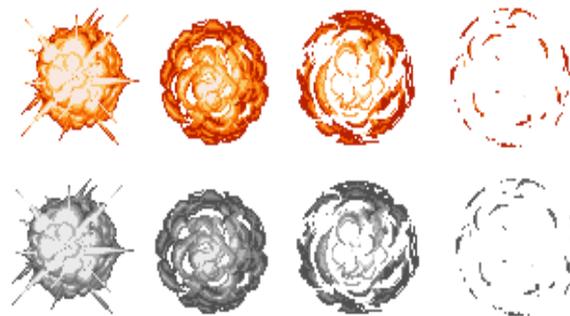


Figura 4 - Sprites de explosão.

A cada acerto no castelo o jogador recebe cem pontos. Sendo o castelo totalmente destruído, o usuário irá receber quinhentos pontos e quinze segundos a mais no tempo. O jogo também possui um limite de tempo de sessenta segundos (Figura 5).



Figura 5 - Jogabilidade.

Quando o tempo acaba ou os três canhões são destruídos, uma tela com o texto “Você Perdeu” é mostrada. Os demais textos “tentar novamente” e “sair”, foram feitos direto no código. Quando o jogador clicar na opção “tentar novamente”, ele será redirecionado a uma nova partida (Figura 6). Se o jogador clicar na opção “sair”, essa opção irá fechar o jogo e o simulador Corona SDK. O tempo, a pontuação e os recordes são escritos no próprio código. O recorde é salvo em um arquivo “highscore”, que se encontra na pasta “documents” do projeto. Isso faz com que o recorde seja preservado quando o jogo é fechado. Caso o jogador consiga um novo recorde, maior que o anterior, esse novo recorde substitui o anterior. O código para a criação desse recorde foi baseado no código encontrado no site *developer.coronalabs* por *Peach Pellen*.



Figura 6 - Ambiente do jogo.

Foi implementada uma capa para o projeto. Uma imagem foi desenvolvida para dar ao jogador uma breve impressão do que estará por vir. Esta imagem apresenta alguns personagens do jogo, como os três canhões, além do título do projeto em destaque: “Angry Physics” (Figura 7). Nesta imagem foi utilizado um “pause”, ou seja, um comando desenvolvido para a capa congelar por um determinado período de tempo na tela, para que o jogador tenha um tempo para poder visualizar a capa do jogo.

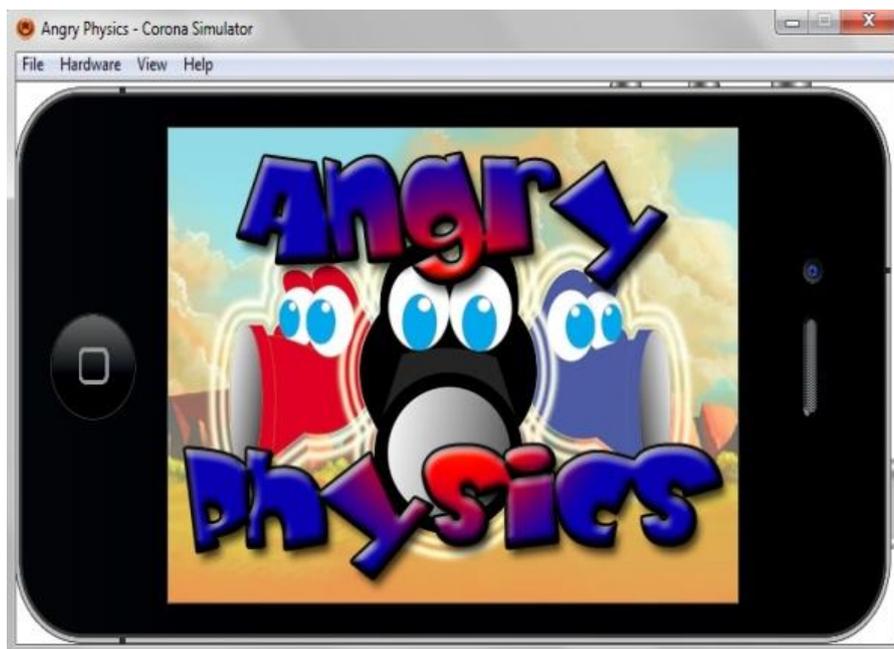


Figura 7 - Capa

O projeto dispõe de uma introdução, onde uma imagem é mostrada logo após a “pausa” da capa, com a finalidade de facilitar a interpretação e o entendimento do jogo. Nessa introdução foi inserido um botão “clique na tela para começar”, com um link fazendo com que o jogador, após ler a breve introdução, clique sobre a tela para iniciar.

Após essa introdução é apresentada outra tela, com um tutorial e instruções de como jogar. Nessa tela encontra-se um botão que deve ser pressionado quando o jogador terminar de ler as instruções e estiver pronto para jogar. Basta clicar sobre este botão para iniciar o jogo.

O projeto dispõe de um áudio para o fundo do jogo. Também possui uma música para a tela “Você Perdeu”, que é exibida quando o jogador perde. Outro áudio foi introduzido no projeto para dar som às explosões. Este som foi direcionado para aparecer no momento em que o tiro colidir com a torre ou castelo, dando o efeito sonoro de uma explosão. Para que isso possa ocorrer foi desenvolvida uma função, a qual aciona o áudio no momento da colisão.

## 5. PROJETOS FUTUROS

Como projetos futuros a serem desenvolvidos em continuidade ao jogo aqui apresentado, podemos vislumbrar:



- i. Implementação de um arquivo de recorde que grave os nomes dos jogadores.
- ii. Aplicação de uma força que simule o vento.
- iii. Utilização de ferramentas gráficas mais sofisticadas para o desenvolvimento das imagens dos personagens com uma melhor qualidade.
- iv. Utilização de ferramentas sonoras mais sofisticadas para o desenvolvimento dos sons do jogo.
- v. Procurar por empresas-parceiras no desenvolvimento de jogos educativos para plataformas móveis.

## 6. CONSIDERAÇÕES FINAIS

Este projeto pretendeu estimular o desenvolvimento de jogos educativos dentro e fora do ambiente acadêmico, sendo a implementação do software aqui apresentado direcionada especialmente para o público do ensino fundamental e médio. O tema escolhido, lançamento balístico, foi selecionado por se tratar de uma questão de relativa dificuldade de absorção nas aulas de física.

Para alcançar este objetivo, LUA mostrou-se uma linguagem de programação muito adequada para a implementação do jogo. O software Corona SDK também mostrou ser muito prático para que este jogo pudesse ser desenvolvido. Infelizmente a versão de “degustação” do Corona SDK, que foi utilizada, não permitiu que o jogo fosse exportado para um dispositivo Android (ou outro compatível com iOS). Assim, o produto gerado foi apenas simulado em um ambiente Windows.

De qualquer forma, independente desta limitação, os resultados obtidos foram muito promissores. O jogo foi testado em um evento acadêmico realizado na Instituição de origem dos autores, gerando uma repercussão positiva.

Assim, constatamos que o desenvolvimento de jogos educativos pode gerar grande repercussão dentro e fora do ambiente acadêmico. Espera-se, portanto, que novas iniciativas de desenvolvimento de jogos para plataformas móveis possam vir a ser desenvolvidas não apenas dentro do ambiente acadêmico tradicional, mas também envolvendo parcerias com empresas de software e aficionados em geral.

## REFERÊNCIAS BIBLIOGRÁFICAS

AARSETH, E. J. Cibertexto: perspectivas sobre a literatura ergótica. Lisboa: pedra da roseta, 2005.

DIGITALDEV. **O que é uma linguagem de programação?** Disponível em: <<http://www.digitaldev.com.br/linguagens/>> Acesso em: 25 out. 2012.

GEE, J. P. Lo que Nos Ensañan los Videojuegos Sobre el Aprendizaje y el Alfabetismo, Colección aule, Ediciones Aljibe, Enseña Abierta de Andalucía: Consorcio Fernando de Los Rios, 2004.

LUA. **Linguagem de Programação Lua, Para quem Precisa do Mundo da Lua.** Disponível em: <<http://www.lua.org/press.html#gdm>> Acesso em: 25 out. 2012.

LUA. **Aplicações Lua, Brilho Lunar.** Disponível em: <<http://www.lua.org/press.html#gdm>> Acesso em: 25 out. 2012.

LUA. **Escolha da Linguagem, Por que Escolher Lua.** Disponível em: <<http://www.lua.org/portugues.html>> Acesso em: 29 out. 2012.



MOITA, F. M. G. S. C. Jogos eletrônicos na escola e na vida da geração @. São Paulo: Atomoealinea, 2007.

MOITA, F. M. G. S. C., VERASZTO, E. V., CANUTO, E. C. A. Jogos Eletrônicos e Estilos de Aprendizagem: uma relação possível - breve análise do perfil de alunos do Ensino Médio In: Estilos de Aprendizagem na atualidade, 2011, v.1, p. 1-14.

PEREIRA, R. R. **Corona SDK**. Disponível em: <[http://www.ulbra.inf.br/joomla/images/documentos/TCCs/2011\\_01/TCCI\\_SI\\_RicardoRauber.pdf](http://www.ulbra.inf.br/joomla/images/documentos/TCCs/2011_01/TCCI_SI_RicardoRauber.pdf)> Acesso em: 24 out. 2012.

PELLEN, P. **Developer.coronalabs**. Disponível em: <<http://developer.coronalabs.com/forum/2012/01/05/saving-files-using-ego>> Acesso em: 17 jun. 2012.

SANTAELLA, L. Cultura e artes do pós-humano: da cultura das mídias à cibercultura. São Paulo: Paulus, 2003.

VERASZTO, E. V. et al. El lenguaje audiovisual interactivo en el contexto educativo. In: Medina, A. R.. (Org.). Investigación e Innovación de la docencia universitaria en el EEES. 1 ed. Madrid: Ramón Areces, 2009, v. 1, p. 209-218.

## **DEVELOPMENT OF EDUCATIONAL SIMULATIONS APPLIED TO PORTABLE DEVICES: AN EXAMPLE FOCUSED TO THE TEACHING OF PHYSICS**

**Abstract:** *This article presents the results of a project that consisted in making the development of an educational game for the Android platform, using the programming language LUA. It was also used the development tool known as SDK (Software Development Kit) Corona Simulator. The development started from educational theoretical assumptions for the conclusion of the work, considering that the potential of audiovisual and interactive applications can bring significant contributions to the field of education. The system, based on these tools, is able to simulate a physical-mathematical model for the launching of projectiles, transforming a classic concept in a game, this way seeking to facilitate the learning of this content.*

**Key-words:** *Development, LUA, Corona, Ballistic, Game.*