



TATI - Uma interface textual amigável para o Second Life

Renato P. dos Santos

ULBRA/PPGECIM

e-mail: fisicainteressante@gmail.com

Resumo

As dificuldades de aprendizado dos alunos em Física estão bem discutidas na literatura científica. Já em 1980, Papert denunciava que as escolas ensinam o movimento newtoniano por meio de manipulação de equações, em vez de manipulação dos próprios objetos newtonianos. Por outro lado, o Second Life e sua linguagem de programação apresentam uma notável curva de aprendizado que desestimula a maioria dos professores a usá-lo como ambiente de simulações educacionais. O objetivo deste trabalho é descrever TATI, uma interface textual que, através de TATILogo, uma variante própria da linguagem Logo, permite a geração de vários micromundos de Física no Second Life, contendo diferentes tipos de objetos que seguem distintas leis físicas, proporcionando uma sequência de aprendizagem piagetiana até a Física Newtoniana.

Palavras-chaves: Second Life, micromundos físicos, Ensino de Física, simulações computacionais, interface textual.

TATI - The Amiable Textual Interface for Second Life

Abstract

Student difficulties in learning Physics have been throughout discussed in the scientific literature. Already in 1980, Papert complained that schools teach Newtonian motion by means of manipulation of equations rather than by manipulation of the Newtonian objects themselves. On the other hand, Second Life and its programming language have a remarkable learning curve that discourages most teachers at using it as an environment for educational simulations. The objective of this study is to describe TATI, a textual interface which allows, through an accessible Logo language extension, the generation of various physics microworlds in Second Life, containing different types of objects that follow different physical laws, providing a Piagetian learning path into Newtonian Physics.

Keywords: Second Life, physics microworlds, Physics teaching, computer simulations, textual interface

1. Introdução

As dificuldades de aprendizado dos alunos em Física são bem conhecidas, havendo vasta literatura científica a respeito, acumulada desde os anos 70. Segundo Driver (1989), os estudantes estão pouco inclinados a mudanças conceituais e, no seu dia-a-dia, utilizam mais comumente as suas concepções alternativas do que o conhecimento científico recebido, o qual fica, então, restrito ao contexto escolar.

Por outro lado, já em 1980 Papert apontava que os estudantes quase não têm experiências diretas e físicas do movimento puramente newtoniano e, na sua ausência, as escolas são obrigadas ensinar o movimento newtoniano por meio de manipulação de equações, em vez de manipulação dos próprios objetos newtonianos (Papert, 1985, p. 153). Para esse autor, os estudantes deveriam experimentar com leis de movimento alternativas, num “micromundo da física”, progredindo, numa sequência de aprendizagem piagetiana (1985, p. 152) de Aristóteles até Newton e Einstein e até



mesmo leis que eles mesmos poderiam inventar (1985, p. 154).

É importante compreender que os micromundos de Papert não são meros objetos de aprendizagem, a partir dos quais o aluno aprenda, mas ambientes intelectuais cuja ênfase está no processo, em linguagens expressivas para falar sobre o processo e à reformulação dos velhos conhecimentos em termos dessas novas linguagens (Papert, 1985, p. 218-219). Aqui, esse autor antecipa a ideia de o aluno *pensar-com e aprender-com* o computador de Rosa (2008).

Em (dos Santos, 2009) defendemos a viabilidade do ambiente Second Life (SL) como plataforma para a construção de micromundos físicos (Papert, 1985, p. 154) para o ensino de Física, simulações virtuais imersivas, em que o estudante possa vivenciar descrições físicas alternativas, de forma completamente impossível no mundo real. Tal abordagem, realizando a proposta de Papert com um grau de realismo e similitude incomparáveis aos das tartarugas de Papert (1985) e de diSessa (Abelson; diSessa, 1981), por um lado, promoveria a construção pelo estudante do próprio conhecimento, a explicitação de suas concepções alternativas e sua evolução conceitual (Moreira; Greca, 2003) com maior eficácia e, por outro, constituir-se-ia em ambiente privilegiado para o estudo de processos cognitivos de formação de conceitos físicos durante o aprendizado de Física.

Há mais de 50 diferentes Ambientes Virtuais Multiusuários (MUVE, na sigla em inglês) atualmente disponíveis (Taylor, 2007), criados para simular a vida real, em algum sentido, e mais de 700 (MMORG Gamelist, s.d.) mundos de jogo (*game worlds*), tais como o *World of Warcraft*, criados especificamente para entretenimento. Dentre eles, o *Second Life* (SL), seguido pelo *OpenSim* e pelo *Active Worlds*, destaca-se como a plataforma que oferece mais serviços e ferramentas para desenvolvimento de aplicações com qualidade (Reis et al., 2011), em termos de verossimilhança, escalabilidade, interatividade, usabilidade e segurança, mesmo que não seja o com a maior população de usuários (Taylor, 2007). De fato, diferentemente de outros mundos virtuais, onde as leis físicas não são seriamente levadas em conta, os objetos criados no SL são automaticamente controlados pelo poderoso *engine* de física Havok™ (Havok.com, 2008). Como este Grupo está particularmente interessado em simulações de Física, estes pontos levaram-no a ser o escolhido para suas pesquisas.

Conforme demonstramos em (dos Santos, 2012a), o SL é uma grande simulação de todo um mundo semelhante à Terra e, decididamente, não pode mais ser visto como apenas um *game*. Entretanto, alguns pontos importantes devem ser levados em conta quando se pretende construir um simulador ou um micromundo físico no SL (vide (dos Santos, 2012b) para uma discussão mais aprofundada).

Primeiramente, vale lembrar que a Física do SL não é nem uma virtualização da Física do 'mundo real' nem a da Física 'ideal' Galileana/Newtoniana (dos Santos, 2009). Em segundo lugar, o SL não é um simulador 'clássico', como o *Modellus* (Teodoro et al., 1996), sendo necessária alguma criatividade para contornar as diferenças (dos Santos, 2012b).

Por outro lado, embora seja rico em recursos, não se pode dizer que o SL seja uma plataforma de fácil utilização. Uma das principais dificuldades é o aprendizado da *Linden Scripting Language* (LSL) (LSL Portal, s.d.), com estrutura baseada nas linguagens Java e C e disponibilizando quase quatrocentas funções, sem o que, não se pode dar interatividade aos objetos criados, resultando apenas num Lego gigantesco. Há uma apreciável curva de aprendizado (Sanchez, 2009) que faz com que qualquer proposta de utilização do SL para o ensino tenha de reservar várias horas em seu cronograma, apenas para que os estudantes se familiarizem com as operações básicas do

avatar, tais como andar, passar por portas, subir escadas, manipular objetos, etc.

Esses dois fatores certamente desestimulam a maioria dos professores de Física, que provavelmente não estariam dispostos a investir tanto tempo no estudo da LSL apenas para a construção de simulações educacionais simples.

Afortunadamente, há pouco tempo, este Grupo de Pesquisa teve a oportunidade de entrar em contato com o trabalho do Prof. Bogacki com ensino de Matemática no SL, principalmente o seu *VLP Display* (Bogacki, 2009), que permite facilmente construir representações concretas (objetos palpáveis, não apenas gráficos) de vetores e planos, como se vê na Figura 1.

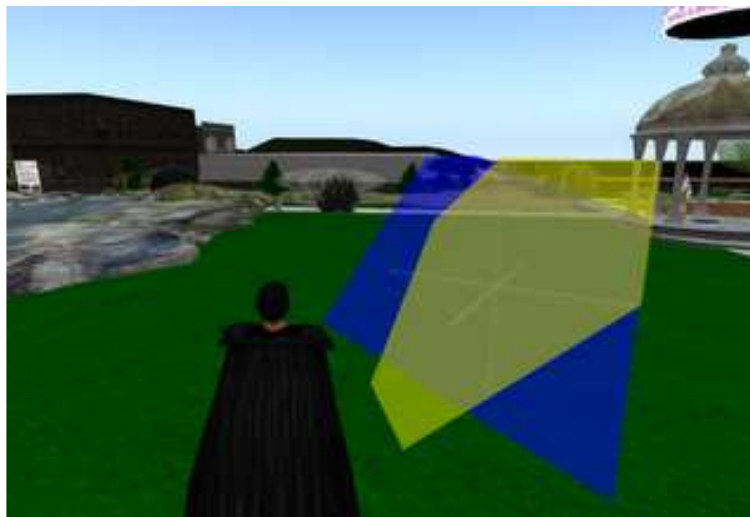


Figura 1 - Dois planos que se intersectam numa reta, gerados pelo *VLP Display* (BOGACKI, 2009)

Fonte: (dos Santos, 2012c)

O utilizador pode realizar construções interessantes no SL simplesmente digitando comandos no canal de chat do SL, sem ter que conectar blocos primitivos (*prims*) (Primitive, 2010) do SL ou aprender a LSL. Esse dispositivo funciona como um interpretador que traduz uma linguagem de comandos simples, de alto nível, próxima ao jargão usual do contexto dos utilizadores, para os comandos LSL necessários para a construção do objeto desejado. Digitando-se $pl=pl\ 2\ 1\ 3\ 7$ no canal de chat, o VLP Display 'entende' o comando como significando " pl é o plano $2x + y + 3z = 7$ " e o gera (vide mais exemplos em (dos Santos, 2012c)).

Inspirado por essa experiência, este Grupo decidiu construir a TATI (*The Amiable Textual Interface for Second Life*), uma interface textual amigável para o Second Life que traduzisse comandos simples, semelhantes aos do Logo em comandos da LSL que gerassem objetos imbuídos de físicas alternativas, semelhantes às tartarugas de Papert (1985) ou, melhor ainda, às '*Dinatarts*' de diSessa (Abelson; diSessa, 1981). Em vez de 'tartarugas' que obedecessem apenas a comandos geométricos tais como *FORWARD*, *BACKWARD*, *RIGHT* ou *LEFT*, TATI criaria objetos de tipos distintos que entendessem comandos, tais como *SETVELOCITY* ou *CHANGE VELOCITY*. (Papert, 1985, p. 158) e vários outros correspondentes às suas várias *mudanças de estados* físicos (Papert, 1985, p. 156).

No restante deste trabalho, descreveremos a construção de TATI, seus objetos e sua linguagem de alto-nível, TATILogo¹.

2. Materiais e Métodos

Conforme discutido adiante, a implementação dos seis tipos de objeto que TATI oferece para o utilizador baseou-se em Papert (1985, pp. 157-158) para quatro deles, e na descrição em (Physical, 2007) e (Non-Physical, 2006) para os outros dois, que são tipos básicos do Second Life. Para a construção de TATILogo, baseou-se nas metodologias usuais para a construção de softwares interpretadores de comandos, tais como gramáticas livres de contexto, especialmente sua expressão em uma variante da EBNF (vide nota ²). O analisador léxico online do site *RPA Toolkit*³ foi usado para a validação da sintaxe de TATILogo, a análise gramatical preditiva (*top-down*) (Aho et al., 1995) para o analisador gramatical, além de livros e sites de referência sobre a LSL e construção de objetos no SL (LSL Wiki; Moore et al., 2008) para o tradutor de TATILogo para os comandos finais de LSL que efetivamente gerarão os objetos no ambiente SL.

3. TATI (*The Amiable Textual Interface for Second Life*)

Considerando-se a discussão anterior, inicialmente, tratou-se de caracterizar os tipos de objetos que TATI deveria criar. Papert (1985, pp. 157-158) propõe um total de quatro tipos de objetos: tartaruga geométrica, de velocidade, de aceleração e newtoniana. Por compatibilidade, TATI deveria poder gerar também os dois tipos básicos de primitivas do Second Life: objetos físicos e não-físicos (Non-Physical, 2006, Physical, 2007). Resulta assim, que TATI oferece seis tipos de objeto para o utilizador, os quais estão descritos na Tabela 1, onde também é feita sua correspondência com as tartarugas de Papert e os objetos do SL.

Tabela 1 – Correspondências entre os tipos de objetos disponíveis e as tartarugas de Papert e objetos do Second Life

Objeto	Tartaruga ou objeto do Second Life	Características
NOROBJECT	Objeto básico <i>não-físico</i> do SL	Insensível à gravidade, aceita funções <i>cinemáticas</i> do SL (<i>llSetPos</i> , <i>llSetRot</i> , etc.).
GEOBJECT	Tartaruga geométrica	Insensível à gravidade, possui apenas dois componentes geométricos: posição e orientação.
VELOBJECT	Tartaruga de velocidade	Insensível à gravidade, não há comandos para mudar sua posição, mas apenas sua velocidade através do comando <i>SETVEL</i> ; sua posição se alterará como consequência da sua velocidade.
ACCOBJECT	Tartaruga de aceleração	Insensível à gravidade, aceita apenas comandos <i>SPEEDUP</i> , no sentido de “ <i>Mude sua velocidade de x, qualquer que seja ela</i> ”.
NEWOBJECT	Tartaruga newtoniana	Insensível à gravidade, aceita apenas comandos que alterem seu momento (forças).
PHYOBJECT	Objeto básico <i>físico</i> do SL	Sujeito à gravidade e à dinâmica do SL, aceita funções cinéticas do SL (<i>llSetForce</i> , <i>llGetAccel</i> , <i>llGetOmega</i> , etc.).

Definidos os objetos, passou-se à definição dos comandos que constituiriam TATILogo, bem como os tipos de dados necessários, tais como '*vector*', '*float*' e '*string*'. Da mesma forma que Papert estendeu a linguagem básica do Logo para incluir novos comandos para os novos tipos de tartarugas, foram incluídos em TATILogo comandos que atendessem as características de cada um dos objetos acima.

Por coerência com as tartarugas geométricas de Papert, foram mantidos, praticamente sem alteração, os comandos *FORWARD*, *BACKWARD*, *RIGHT* e *LEFT* da linguagem Logo (Harvey, 1993). No entanto, considerou-se que os comandos *SETXY*, *SETX* e *SETY* não tinham interesse suficiente que justificasse sua implementação. Os comandos *SETPOS* e *SETHEADING*, este renomeado como *SETROT*, foram mantidos apenas com o objetivo de facilitar o posicionamento inicial dos objetos para uma simulação dinâmica, especialmente em conjunto com a opção *ONGO*, que deixa a execução de comandos em suspenso até o momento apropriado.

Como o SL é um ambiente 3D, onde rotações em torno dos três eixos são possíveis, foram incluídos em TATILogo os comandos análogos para rotações nos outros eixos *UP*, *DOWN*, *CLOCK* e *ACLOCK*.

Por outro lado, decidiu-se incluir, para os outros objetos, comandos mais semelhantes aos *operadores de mudança de estado* geométricos (Papert, 1985, p. 156) vistos acima. Assim, para os objetos do tipo VELOBJECT, além de *SETVEL*, que corresponde a *assuma a velocidade* $\vec{v} = (v_x, v_y, v_z)$, que pode estar, agora, em qualquer direção, incluíram-se os comandos *SPEEDUP* e *SLOWDOWN*, ambos com parâmetros escalares (float), no sentido de *umente* (ou *diminua*) *sua velocidade de v* (escalar, em inglês, *speed*) na mesma direção da orientação do objeto. Por analogia, incluíram-se, também, em TATILogo, os comandos para variação da velocidade angular *SETANGVEL* (vetorial) e *SPINUP* e *SPINDOWN* (escalares).

Para os objetos do tipo ACCOBJECT, em vez do comando *CHANGE VELOCITY* (Papert, 1985, p. 157-158), definiram-se os comandos *SETACCEL* e *SETANGACCEL* (vetoriais) que, respectivamente, atribuem aceleração e aceleração angular ao objeto. Já, para os objetos do tipo NEWOBJECT e PHYOBJECT, criaram-se os comandos dinâmicos tais como *APPFORCE*, *APPIMPULSE*, *APPTORQUE*, etc., que aplicam forças, impulsos e torques.

Tendo em conta as propriedades de cada objeto, a Tabela 2 apresenta a aplicação de cada comando de TATILogo aos vários tipos.

Tabela 2 - Comandos de TATILogo permitidos para cada tipo de objeto disponível

	NROBJECT	GEOBJECT	VELOBJECT	ACCOBJECT	NEWOBJECT	PHYOBJECT
GETPOS, GETROT, SETPOS	✓	✓	✓	✓	✓	✓
FORWARD, BACKWARD, RIGHT, LEFT, UP, DOWN, CLOCK, ACLOCK	✓	✓	✗	✗	✗	✗
SETVEL, SPEEDUP, SLOWDOWN	✗	✗	✓	✗	✗	✗
SETANGVEL, SPINUP, SPINDOWN	✓ ¹	✗	✓	✗	✗	✓
SETACCEL, ACCELERATE, DECELERATE, SETANGACCEL	✗	✗	✗	✓	✗	✓
GETVEL, GETANGVEL	✗	✗	✓	✓	✓	✓
GETACCEL,	✗	✗	✗	✓	✓	✓

GETANGACCEL						
APPFORCE, APPIMPULSE, APPTORQUE, APPROTIMPULSE GETFORCE, GETTORQUE	x	x	x	x	✓	✓

¹ apenas como um efeito local do cliente, não do servidor.

Para a construção do interpretador, foi elaborada a sintaxe dos comandos, de que se apresenta uma parte, em EBNF-W3C², na Figura 2.

```
object_id          ::= <identifier>
vector            ::= "(" <S>? <float> <S> <float> <S> <float> <S>? ")" |
                  <zero_vector>
velocity          ::= <vector>
speed            ::= <float>
colour           ::= "BLACK" | "BLUE" | "GREEN" | "ORANGE" | "PINK" |
                  "RED" | "YELLOW" | "WHITE"
object_type      ::= "NOROBJECT" | "GEOBJECT" | "VELOBJECT" |
                  "ACCOBJECT" | "NEWOBJECT" | "PHYOBJECT"
object_shape     ::= "APPLE" | "ARROW" | "CONE" | "CUBE" | "CYLINDER" |
                  "PLANE" | "PRISM" | "SPHERE"
create_statement ::= "CREATE" <S> <object_id> <S> ( <object_type> <S>
                  )? ( <object_shape> <S> )? ( <colour> <S> )?
delete_statement ::= "DELETE" <S> <object_id>
setcol_statement ::= "SETCOL" <S> <object_id> <S> <colour>
setpos_statement ::= "SETPOS" <S> <object_id> <S> ( <position> |
                  "ABOVEME" | "REZPOS" )
getrot_statement ::= "GETROT" <S> <object_id>
forward_statement ::= "FORWARD" <S> <object_id> <S> <distance> ( <S>
                  "ONGO" )?
setvel_statement ::= "SETVEL" <S> <object_id> <S> <velocity> ( <S>
                  "ONGO" )?
speedup_statement ::= "SPEEDUP" <S> <object_id> <S> <speed> ( <S>
                  "ONGO" )?
connect_statement ::= "CONNECT" <S> <object_id1> <S> <object_id2>
repeat_statement ::= "REPEAT" <S> <integer> <S> "(" <S>?
                  <list_of_statements> <S>? ")"
list_of_statements ::= <simple_statement> ( <S>? ; <S>?
                  <list_of_statements> )?
list_statement    ::= "LIST" ( "OBJECTS" | "COLOURS" | "SHAPES" | "TYPES" |
                  "COMMANDS" )?
help_statement    ::= "HELP" | "?"
```

Figura 2 - Parte da sintaxe de TATILogo em EBNF-W3C

Uma vez elaborada, a sintaxe de TATILogo foi verificada e validada através do *custom parse* online do site RPA Toolkit³. A partir dessa sintaxe, o interpretador foi, então, construído.

Inicialmente, os comandos digitados pelo usuário são decompostos em *tokens* por um *parser*. Intencionalmente, a sintaxe foi feita o mais simples possível, não só para facilitar o trabalho do *parser* como também para tornar mais fácil a interação do usuário com TATI. Vale destacar a sintaxe simplificada que definimos para o não-terminal *vector* como $(v_x v_y v_x)$ em vez de (v_x, v_y, v_x) , como usual, ou de $\langle v_x, v_y, v_x \rangle$, como utilizada nas funções do SL.

A lista de *tokens* é, então, passada a um analisador gramatical preditivo (Aho et al., 1995), composto de vários procedimentos, correspondentes aos vários não-terminais da linguagem, conforme definidos pela sintaxe.

Em caso de qualquer desvio da sintaxe da linguagem, seja por má formação do parâmetro, por inadequação de seu tipo ao comando ou mesmo do comando ao tipo de objeto, o analisador gramatical retorna ao usuário mensagens de erro adequadas. Além disso, como se pode observar da Figura 2, há comandos de ajuda ao usuário, tais como *help*, *list* e até mesmo ?.

Verificada a correção sintática do comando, outros procedimentos são invocados para a chamada aos correspondentes comandos necessários de LSL que são, então, passados ao interpretador de scripts do SL para sua execução.

Além dos comandos de alteração dos estados dos objetos, descritos acima, algumas outras funcionalidades de TATI, previstas em TATILogo, merecem destaque.

A função LSL que materializa (*rez*) objetos o faz a partir de cópias de primitivas que já estão no inventário de TATI (IlRezObject, 2011). Através do parâmetro *object_shape* do comando *CREATE* da linguagem, não só o usuário pode criar objetos de tipos diferentes, como também, com formas diferentes, dentre um conjunto de objetos-formas (*shapes*) previamente incluídos no inventário de TATI, tais como cubo, esfera, cilindro, cone, maçã e avião. Outras formas podem ser acrescentadas pelo usuário ao inventário de TATI, sendo reconhecidas pelo script e passando a estar disponível na lista de *object_shape*.

Os objetos-formas tem cor branca e, por isso, suas cópias são materializadas dessa mesma cor. No entanto, o usuário pode, em seguida, alterar sua cor, através do comando *SETCOL*, dentro de um conjunto pré-definido de oito cores, como se vê na definição do parâmetro *colour* na Figura 2.

Há vários comandos de recuperação de informação sobre os objetos, tais como *GETCOL*, *GETTYPE*, *GETPOS*, etc. Cada objeto é criado com um identificador definido pelo usuário através do parâmetro *object_id* do comando *CREATE*. Este é atribuído pelo script como nome da primitiva criada e é feito aparecer como texto visível acima do objeto, facilitando sua referência posterior.

Finalmente, tendo em conta a cultura da *Sociedade da Impressão (Impression Society)* vigente no SL (Au, 2008, p. xix), da valorização do que causa impacto visual, procurou-se dar um design ao objeto que iria abrigar o script de TATI condizente com o simbolismo de sua função. Para tal, obteve-se um ‘chapéu de mágico’, em cujo inventário foram colocados o script de TATI e os objetos-formas. O script foi ajustado, de forma que os objetos fossem gerados logo acima da abertura do chapéu, como que ‘por mágica’.



Figura 3 – Exemplos de objetos criados com a interface TATI

Na Figura 3 veem-se alguns exemplos de objetos criados: um de forma CUBE e cor azul do tipo NOROBJECT, um de forma PLANE e tipo VELOBJECT e um de forma CONE e cor amarela do tipo PHYOBJECT, o qual está caído no chão por ser sujeito à gravidade.

O comando *REPEAT* permite a repetição de um conjunto de comandos, tal como no clássico exemplo Logo (Papert, 1985, p. 81-82) de desenhar uma circunferência (Figura 4). Uma discussão sobre a realização de trajetórias circulares com outros tipos de objeto aparece em (dos Santos, 2012d).

```
UM CÍRCULO REPITA  
(PARAFRENTE 1 PARADIREITA 1)
```

Figura 4 - procedimento em Logo para desenhar uma circunferência (Papert, 1985, p. 81-82)

Está, também, prevista a implementação do comando *CONNECT*, o qual realizará a ideia das ‘tartarugas conectadas’ de Papert (1985, p. 160) para o aprendizado da 3ª Lei de Newton, de ação e reação, ao fazer com que o comando dado a um objeto seja reproduzido, de forma invertida, pelo objeto a ele conectado. Pretende-se, também tornar esta linguagem recursiva, já que Papert considerou que “de todas as ideias que apresentei às crianças, a recursão se destacou como uma ideia capaz de provocar uma resposta entusiástica” (Papert, 1985, p. 97).

Uma dificuldade enfrentada foi que, no SL, os scripts são restritos a rodar em seções de apenas 64KB de memória do simulador (“LSL Script Memory”, 2011). Este fato restringiu em muito as possibilidades de TATILogo e exigiu criatividade para implementar as presentes funções.

Conclusões

Conforme discutido anteriormente, Papert propôs uma sequência de aprendizagem piagetiana (1985, p. 152) para a Física Newtoniana. Passados trinta anos da proposta de Papert, muito já se discutiu sobre a linguagem Logo, mas, com exceção das primitivas ‘*Dinartarts*’ (tartarugas dinâmicas) de diSessa (Abelson; diSessa, 1981), não se tem conhecimento de outras implementações de micromundos que efetivamente disponibilizem essa experimentação com leis físicas diferentes das newtonianas, tal



como concebida por Papert e realizada por TATI. Acreditamos que a sequência de tipos de objetos acima realiza essa proposta.

Espera-se que TATI e TATILogo representem uma contribuição significativa para o aprendizado de Física, ao permitir ao usuário por em prática dois princípios matemáticos: primeiro, relacionar a novidade a ser aprendida com alguma coisa que já se sabe e, segundo, tomar a coisa nova e apropriar-se dela, fazer alguma coisa nova com ela, brincar com ela, construir com ela (Papert, 1985, p. 148). Ao mesmo tempo, espera-se que elas reduzam a curva de aprendizado do SL (Sanchez, 2009), possibilitando aos utilizadores construir suas simulações sem ter que entrar tão a fundo na programação LSL e facilitando a um grande público de professores seu acesso a uma ferramenta tão interessante como o SL.

Referencias Bibliográficas

ABELSON, H.; DISESSA, A. A. **Turtle Geometry: Computations as a Medium for Exploring Mathematics**. Cambridge, MA: MIT Press, 1981.

AHO, A. V.; SETHI, R.; ULLMAN, J. D. **Compiladores: Princípios, Técnicas e Ferramentas**. Rio de Janeiro: LTC - Livros Técnicos e Científicos, 1995.

AU, W. J. **The Making of Second Life: notes from the new world**. New York: Harper Collins, 2008.

BOGACKI, P. Multivariable Calculus Virtual Office Hours in a Metaverse. In: J. Foster (Ed.); In: **CD-ROM PROCEEDINGS OF THE TWENTIETH ANNUAL INTERNATIONAL CONFERENCE ON TECHNOLOGY IN COLLEGIATE MATHEMATICS**, San Antonio, Texas, March 6-9, 2008. p. C015, 2009. Upper Saddle River, NJ: Pearson Education. Disponível em: <<http://archives.math.utk.edu/ICTCM/i/20/C015.html>>. Acesso em: 10/3/2012.

CASTRONOVA, E. The Right to Play. **New York Law School Law Review**, v. 49, n. 1, p. 185-210, 2004.

dos SANTOS, R. P. Second Life Physics: Virtual, real or surreal? **Journal of Virtual Worlds Research**, v. 2, n. 1, p. 1-21, 2009.

dos SANTOS, R. P. Simulações de Física no Second Life: Uma análise de viabilidade. **Acta Scientiae**, v. 14, n. 2, 2012a.

dos SANTOS. Second Life as a Platform for Physics Simulations and Microworlds: An Evaluation. CBLIS 2012 - 10th Computer-Based Learning in Science, Barcelona, 26th to 29th June, 2012. **Proceedings...**, 2012b.

dos SANTOS, R. P. **Immersive Geometry with Bogacki in Second Life**. 2012c. Disponível em: <<http://www.secondlifephysics.com/2012/03/immersive-geometry-with-bogacki-in.html>>.

dos SANTOS, R. P. **Running in Circles with TATILogo**. 2012d. Disponível em: <<http://www.secondlifephysics.com/2012/06/running-in-circles-with-tatilo.html>>.



DRIVER, R. Changing Conceptions. In: P. Adley; J. Blis; J. Head; M. Shayer (Eds.); **Adolescent Development and School Science**. London: The Falmer Press, 1989.

HARVEY, B. **Berkeley Logo User Manual**. Berkeley, CA: University of California, 1993.

HAVOK.COM. **Havok Physics Animation 6.0.0 PC XS User Guide**. Dublin: Havok.Com, Inc., 2008.

LLREZOBJECT. Second Life Wiki. 2011. Disponível em: <<http://wiki.secondlife.com/wiki/LlRezObject>>. Acesso em: 12/1/2012.

LSL PORTAL. s.d. Disponível em: <http://wiki.secondlife.com/wiki/LSL_Portal>. Acesso em: 29/10/2008.

LSL SCRIPT MEMORY. Disponível em <https://wiki.secondlife.com/wiki/LSL_Script_Memory>. Acesso em: 07/05/2012.

LSL WIKI. Disponível em: <<http://lslwiki.net/lslwiki/>>. Acesso em: 29/10/2008.

MMORG Gamelist-All Listed Games. MMORPG site. s.d. Disponível em: <<http://www.mmorpg.com/gamelist.cfm>>. Acesso em: 16/2/2012.

MOORE, D.; THOME, M.; KAREN, H. **Scripting Your World: The Official Guide to Second Life Scripting**. Berkeley, CA: Sybex, 2008.

MOREIRA, M. A.; GRECA, I. M. Cambio Conceptual: Análisis Crítico y Propuestas a La Luz de la Teoría del Aprendizaje Significativo. **Ciência & Educação**, v. 9, n. 2, p. 301-315, 2003.

NON-PHYSICAL. LSL Wiki, 2006. Disponível em: <<http://lslwiki.net/lslwiki/wakka.php?wakka=nonphysical>>. Acesso em: 17/5/2010.

PAPERT, S. A. **Logo: Computadores e Educação**. São Paulo: Brasiliense, 1985.

PHYSICAL. LSL Wiki, 2007. Disponível em: <<http://lslwiki.net/lslwiki/wakka.php?wakka=physical>>. Acesso em: 17/5/2010.

PRIMITIVE. Second Life Wiki. Linden Research, Inc., 2010. Disponível em: <<http://wiki.secondlife.com/wiki/Primitive>>. Acesso em: 17/5/2010.

REIS, R.; FONSECA, B.; ESCUDEIRO, P. Comparative analysis of virtual worlds. In: **Proceedings of the 6th Iberian Conference on Information Systems and Technologies (CISTI 2011)**, Chaves, Portugal, 15-18 June 2011. p. 1-7, 2011.

ROSA, M. **A Construção de Identidades online por meio do Role Playing Game: relações com o ensino e aprendizagem de matemática em um curso à distância**. Tese (Doutorado em Educação Matemática). Universidade Estadual Paulista, Rio Claro, 2008.



SANCHEZ, J. Barriers to Student Learning in Second Life. **Library Technology Reports**, v. 45, n. 2, p. 29-34, 2009.

TAYLOR, D. Second Life in perspective: A round-up of 50 virtual worlds. 2007. Disponível em: <<http://www.fabricoffolly.com/2007/10/second-life-in-perspective-round-up-of.html>>. Acesso em: 13/2/2012.

TEODORO, V. D.; DUQUE VIEIRA, J. P.; COSTA CLÉRIGO, F. **Introdução ao Modellus - Experiências com Modelos Matemáticos em Física-Química e Matemática - Versão 1.11**. Lisboa: Didática, 1996.

¹ <http://www.tatilogo.com>

² Utilizamos aqui a variação da EBNF recomendada pelo W3C (Consórcio World Wide Web) (<http://www.w3.org/TR/REC-xml/#sec-notation>) para compatibilidade com o *custom parse* online do site RPA Toolkit (vide próxima nota de rodapé)

³ Disponível em <http://www.rpatk.net/web/en/parsecustom.php>