

Implementação dos micromundos físicos propostos por Seymour Papert para o ensino de física

RESUMO

O objetivo deste trabalho é implementar um micromundo físico desenvolvido com tecnologia WebGL para a aprendizagem de Física. Amparado pela teoria construcionista de Seymour Papert, o trabalho propõe a implementação de um micromundo assim como proposto por Papert, ou seja, de maneira que o próprio aluno defina as condições a serem analisadas no micromundo físico, alterando as variáveis, observando e compreendendo os resultados. Com a proposta, espera-se que o micromundo desenvolvido seja condizente com a proposta inicial de Papert e possibilite a compreensão de conteúdos físicos que exigem maior nível de abstração, através das interações com o ambiente simulado, conteúdos estes que muitas vezes são demasiadamente abstratos para serem relacionados espontaneamente com situações cotidianas.

PALAVRAS-CHAVE: Micromundo físico. Ensino de Física. Construcionismo.

Fabiana Aparecida Flor
fabiana.flor1@outlook.com
[0000-0002-2081-973X](tel:0000-0002-2081-973X)
Instituto Federal do Paraná, Telêmaco
Borba, Paraná, Brasil.

Rafael João Ribeiro
rafael.ribeiro@ifpr.edu.br
[0000-0002-9076-3663](tel:0000-0002-9076-3663)
Instituto Federal do Paraná, Telêmaco
Borba, Paraná, Brasil.

INTRODUÇÃO

É evidente a presença da tecnologia nos mais diversos setores da sociedade atual. E no setor educacional, embora essas mudanças sejam tardias, os recursos tecnológicos vêm ganhando grande espaço como integrantes no processo de ensino.

Entretanto, a inserção das tecnologias educacionais ainda é desafiadora e enfrenta diversos percalços. O primeiro deles é o fato de que a tecnologia ainda está atrelada às condições socioeconômicas, tornando o acesso a ela pouco igualitário. Além disso, é notória a dificuldade encontrada na utilização destes recursos na sala de aula, especialmente por parte dos professores, que em sua maioria não recebem formação adequada quanto à manipulação da tecnologia, seus métodos e seus objetivos pedagógicos. Segundo Bielschowsky (2009), muitas escolas públicas, inclusive, já têm à sua disposição equipamentos como TVs e computadores com o intuito de aprimorar o processo de ensino aprendizagem, porém, a presença desses equipamentos não significa necessariamente que estes serão utilizados de maneira adequada.

No âmbito do ensino das ciências, essas dificuldades se multiplicam. Além da falta de equipamentos e formação adequada, o próprio sistema de ensino do conhecimento científico vem se restringindo cada vez mais à sala de aula, desvinculando-se da percepção na realidade. Os conteúdos puramente teóricos, matematizados e os exercícios repetitivos sem contextualização têm contribuído para essa ruptura entre o mundo real e a teoria aprendida. Contrastando com esta situação, Seymour Papert já propunha, em 1980, a construção de micromundos físicos de realidade simulada, com a intenção de aproximar o mundo real da Geometria na Matemática e, posteriormente, das Leis de Newton do movimento na Física, mesclando-os no mundo virtual (SANTOS, 2008).

Além do campo educacional, condições complicadas e abstratas começaram a ser projetadas virtualmente desde o surgimento dos primeiros computadores, como é o caso da simulação de trajetórias de foguetes, por exemplo (SWAINE; HEMMENDINGER, 2018). Aos poucos, foi sendo percebida a potencialidade das simulações também para o setor educacional. Seymour Papert foi um dos primeiros a perceber e utilizar esse potencial dos computadores na educação. Ele desenvolveu uma sequência de aprendizagem piagetiana baseada na construção de micromundos virtuais, que tem sido difundida desde então, evoluindo desde as primitivas Tartarugas em linguagem LOGO¹, desenvolvidas pelo próprio Papert, para a aprendizagem da Matemática, passando pela sua proposta de micromundos para o ensino da Física, até os ambientes educacionais mais sofisticados disponíveis atualmente na Internet.

Além da LOGO, outras linguagens de programação foram sendo desenvolvidas, desde as mais intuitivas e educativas como a *Scratch*², baseada em blocos lógicos que não exigem conhecimento prévio em programação, até outras como *C#*, *Pascal* e *JavaScript*, por exemplo.

Entre as demais linguagens de programação, o *JavaScript* destaca-se por sua larga aplicação e praticidade, pois para programar em *JavaScript* não é necessário nada além de um navegador *Web*, diferente das demais linguagens que exigem *softwares* pesados e outras ferramentas.

Utilizando as vantagens da linguagem *JavaScript* e seu poder para o desenvolvimento de animações, foi criada, em 2013, a plataforma *Babylon.js*³, que oferece uma gama de possibilidades em interação e imersão virtual. O *Babylon.js* trata-se de uma estrutura completa de *JavaScript* para a criação de jogos e experiências 3D com HTML5 e WebGL, de código aberto. O *Babylon.js* possui uma área para a criação das cenas e animações 3D, chamada de *Playground*. Embora não tenha sido desenvolvido restritamente com intuito educacional, o *Playground* oferece todo o aporte necessário para a prototipação de micromundos físicos condizentes com a aprendizagem construcionista de Papert.

Segundo Valente (1993), Papert utilizou o termo Construcionismo para caracterizar:

[...] um outro nível de construção do conhecimento: a construção do conhecimento que acontece quando o aluno elabora um objeto de seu interesse, como uma obra de arte, um relato de experiência ou um programa de computador (VALENTE, 1993, p. 40).

Nesse sentido, o objetivo deste trabalho, de caráter exploratório, é implementar um micromundo dentro da proposta original de Papert através do *Playground* do *Babylon.js*, voltada à aprendizagem de Física. Para tal, será realizado inicialmente um levantamento das propostas já existentes de uso do computador com base no Construcionismo para o ensino de Física e/ou Ciências.

O micromundo desenvolvido com o *Babylon.js* contempla o caso da queda livre no vácuo em contraponto com a queda com resistência, um exemplo de conteúdo físico que exige alto grau de abstração dos estudantes.

Por fim, pretende-se verificar teoricamente se o micromundo físico WebGL implementado de fato é condizente com a proposta de Papert do micromundo como ferramenta voltada à aprendizagem da Física, usando como parâmetro os estudos e pesquisas anteriores que utilizam outras ferramentas e plataformas de simulação com o mesmo fim.

CONSTRUCIONISMO E MICROMUNDOS

Seymour Papert (1928-2016) foi um destacado educador e matemático sul-africano, pioneiro na área de Inteligência Artificial. É reconhecido internacionalmente pelas suas pesquisas referentes às maneiras pelas quais o uso da tecnologia pode transformar o processo de aprendizagem. Durante o período de 1958 a 1963, trabalhou na *University of Geneva* com Jean Piaget⁴.

Seymour Papert baseou-se, em grande parte, na teoria Construtivista de Piaget para desenvolver sua própria teoria educacional, em 1980. Papert aceitava as ideias centrais do construtivismo de Piaget, como quando este propunha que a criança se trata de um ser pensante e responsável pela construção de seu próprio conhecimento (PIAGET, 1976). Porém, enquanto o Construtivismo se atenta às formas pelas quais o conhecimento se constrói, o Construcionismo de Papert se preocupa em desenvolver ambientes que propiciem a construção desse conhecimento (PEREIRA; PEREIRA, 2006).

Foi justamente sobre essas divergências de ideias que Papert desenvolveu sua teoria, o chamado Construcionismo (PAPERT, 1997). No Construtivismo, o conhecimento é construído pelo indivíduo, ainda que este não seja ensinado.

Nesse sentido, o Construcionismo de Papert se diferencia do Construtivismo piagetiano ao focar seus interesses no ato da aprendizagem e não apenas nas habilidades e interesses do indivíduo em diferentes estágios de seu desenvolvimento cognitivo.

O Construcionismo de Papert concentra-se no processo de ensinar, de modo a desenvolver métodos para otimizar a aprendizagem, com o mínimo de ensino. Não se trata de suplantando o ato de ensinar, mas favorecer os métodos que deem ênfase à construção cognitiva do indivíduo, fortalecendo sua capacidade mental de construir novos conhecimentos, assimilando esses à sua estrutura de conhecimentos anteriores.

Quanto aos instrumentos de aprendizagem, Papert salienta que o processo de aprendizagem se otimiza quando esses instrumentos dão à criança certo nível de poder e atuação sobre um objeto. Essa abordagem elucida porque algumas crianças têm mais dificuldade em desenvolver determinadas atividades que outras. Nesses casos a criança não teve uma oportunidade anterior de contato com tais atividades em seu dia-a-dia. Sobre isso, Papert destaca que:

[...] qualquer coisa é simples se a pessoa consegue incorporá-la ao seu arsenal de modelos; caso contrário tudo pode ser extremamente difícil. [...] O que um indivíduo pode aprender e como ele aprende isso depende dos modelos que tem disponíveis (PAPERT, 1985, p. 13).

De acordo com Maltempi (2005), no Construcionismo o aluno constrói novos conhecimentos sobre as estruturas intelectuais anteriores, apropriando-se, além disso, dos significados da cultura, do meio que o rodeia e de instrumentos que ele tem disponíveis. Nesse processo, o professor não é, de forma alguma dispensável. Mas ele passa a atuar a partir de outra perspectiva, organizando um ambiente adequado que favoreça o desenvolvimento ativo do aluno, permitindo que ele interaja e faça suas próprias conexões mentais com o seu meio, com os materiais e com os demais colegas.

Além de fornecer as condições para a construção do conhecimento, o Construcionismo de Papert destaca que o aprendizado se efetiva quando o aluno está empenhado em produzir algo com um significado especial para ele, algo que esteja de fato dentro de sua esfera de interesses, relacionado à sua realidade e aos seus gostos (MALTEMPI, 2005).

Por isso é importante também buscar problematizar as situações cotidianas, vinculando-as às diversas áreas do conhecimento estudadas, permitindo que o aluno consiga construir seu conhecimento sobre suas próprias estruturas cognitivas. Isso sem supervalorizar as rápidas transições entre as etapas desse processo, afinal, fica evidente que o conhecimento é uma construção sucessiva, como uma parede de tijolos que se inicia desde a fase sensorial até ir aprimorando-se, reconstruindo-se sobre as bases mais rudimentares, se tornando um conhecimento mais organizado, formal. Papert chama a atenção para a importância de se respeitar os processos de cada uma dessas fases, permitindo a maturação das estruturas cognitivas de cada uma, viabilizando assim a assimilação de novos conhecimentos. Dessa maneira ocorre a aprendizagem de forma mais espontânea e agradável (PAPERT, 1986).

2.1 CONSTRUCIONISMO E A EDUCAÇÃO DIGITAL

Embora o Construcionismo anteceda a era dos computadores, a difusão da informática no final do século XX viabilizou as propostas dessa teoria. Papert percebeu as possibilidades e o poder do uso do computador na educação e se utilizou dessa ferramenta para aplicar as ideias construcionistas amplamente (VALENTE, 1999).

Papert propôs que, através do uso do computador, o aluno consegue relacionar seus pensamentos concretos e abstratos ao interagir com ele. Essa interação colabora com a construção das estruturas cognitivas, desde que essa ferramenta seja eficiente para a aprendizagem partindo dos conteúdos mais compreensíveis até os que exigem um alto nível de abstração, como a Matemática ou a Física. Papert compreendeu que seria necessário investir em mídias poderosas que dessem ênfase a interação da criança com a ferramenta e, além disso, que incitassem o seu interesse e curiosidade, fazendo com que ela própria explorasse seu potencial de aprendizagem, construindo seus conhecimentos (PAPERT, 1994).

Como um dos pilares da teoria Construcionista, Papert propõe a criação de ambientes mais ativos, onde o aluno tenha a possibilidade de experimentar, simular e testar suas ideias, concretas ou abstratas. Esses ambientes dariam ao aluno o poder de criar, modificar e interagir com objetos educativos. Papert percebeu que os computadores, recém desenvolvidos, poderiam viabilizar suas idealizações, proporcionando às pessoas um aprendizado muito mais eficiente, significativo e dinâmico.

2.2 O COMPUTADOR NA EDUCAÇÃO

Mesmo com toda a dimensão de possibilidades abertas pelos computadores, percebeu-se a importância de se definir como essa ferramenta seria utilizada na educação. Pois, embora o computador seja uma inovação tecnológica, ele pode fazer o mesmo papel que quadro e giz já fazem há séculos: transmitir conteúdos, estimulando a memorização e avaliando tais conteúdos em testes. Dentro do uso do computador na educação, há basicamente duas opções, já muito disseminadas atualmente: o computador como uma máquina de ensinar e o computador como uma ferramenta (VALENTE, 1993).

Segundo Valente (1993, p. 6), o computador, como máquina de ensinar, trata-se apenas de “uma versão computadorizada dos métodos tradicionais de ensino”. Existem diversas modalidades que se enquadram nessa categoria, como *softwares* tutoriais, jogos, simulações e *softwares* instrucionais ou *drill-and-practice* que estimulam o exercício da memorização.

Já os jogos e simulações, ainda que classificados como máquinas de ensinar, ladeiam com abordagens diferentes. Os jogos têm grande receptividade pelos alunos, devido aos seus gráficos, *scores* e diversão aliados aos conteúdos que se pretende ensinar. Porém, os jogos voltados ao ensino dificilmente cumprem seu papel pedagógico satisfatoriamente, havendo sempre mais ênfase para os aspectos competitivo e de distração propiciados (RIBEIRO et al., 2015).

As simulações beiram potenciais mais arrojados. Embora simulações de qualidade sejam mais complicadas de serem desenvolvidas e não devam ser usadas desassociadas de outras ferramentas de ensino, elas trazem à sala de aula

a possibilidade de explorar situações que fogem da realidade, como conceitos mais abstratos, experimentos complicados, caros ou inviáveis de serem realizados em sala.

As potencialidades educacionais se ampliam significativamente quando a simulação permite alteração de condições e interações com o aluno, como o autor destaca:

Nos casos onde o programa permite um maior grau de intervenção do aluno no processo sendo simulado (por exemplo, definindo as leis de movimento dos objetos da simulação) o computador passa a ser usado mais como ferramenta do que como máquina de ensinar (VALENTE, 1993, p. 8).

Mesmo sendo interessantes os *softwares* dessas categorias, todos eles colocam o computador como detentor do processo de aprendizagem: é a máquina que ensina, que transmite o conhecimento.

Já a possibilidade do computador como uma ferramenta educacional parte do princípio de que o estudante não será mais ensinado pela ferramenta, mas esta servirá como meio para o desenvolvimento de alguma atividade pelo aluno. Essas atividades podem ser as mais diversas possíveis, desde a pesquisa em bancos de dados até a criação de objetos através de linguagem de programação, criação de obras musicais e atividades laboratoriais com o controle de processos via computador.

Dentro dessa categoria, estão as ferramentas de manipulação de dados e informações, que podem ser usadas tanto por professores quanto por alunos. A resolução de problemas via computador também se enquadra como uma ferramenta educacional. Esses métodos trazem benefícios que não poderiam ser alcançados por meios tradicionais, como a necessidade da descrição metódica do problema em questão, sem contar que essas resoluções podem ser verificadas e validadas ou não durante sua elaboração (VALENTE, 1993).

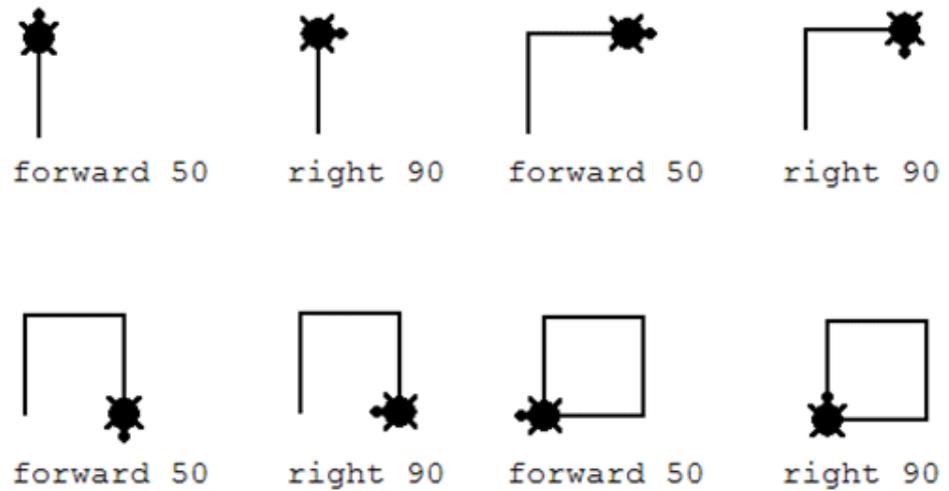
É importante salientar que a linguagem de programação não é o objeto de estudo, ou seja, o foco não é o aluno dominar a linguagem de programação, pois esta é apenas a ferramenta que possibilita a elaboração e a construção das ideias pelo aluno.

2.3 A LINGUAGEM LOGO

Uma linguagem de programação desenvolvida especialmente para a educação – inicialmente voltada ao aprendizado de Geometria – é a linguagem LOGO, criada por Seymour Papert em 1967, no MIT. Trata-se de uma linguagem completamente compreensível aos estudantes e para pessoas leigas em programação. Apesar disso, a LOGO oferece um largo potencial de criação, equiparando-se às demais linguagens utilizadas oficialmente para tais fins, como *Basic* e *Pascal*.

A linguagem em si é baseada em listas de comandos simples e repetições destes comandos elementares. Tais comandos vão se aperfeiçoando de acordo com a evolução no aprendizado da criança. Eles são intuitivos, como por exemplo: “PARAFRENTE PARADIREITA PARAESQUERDA” (Figura 1 – Representação dos comandos da linguagem LOGO).

Figura 1 – Representação dos comandos da linguagem LOGO



Fonte: CORE Education (2014).

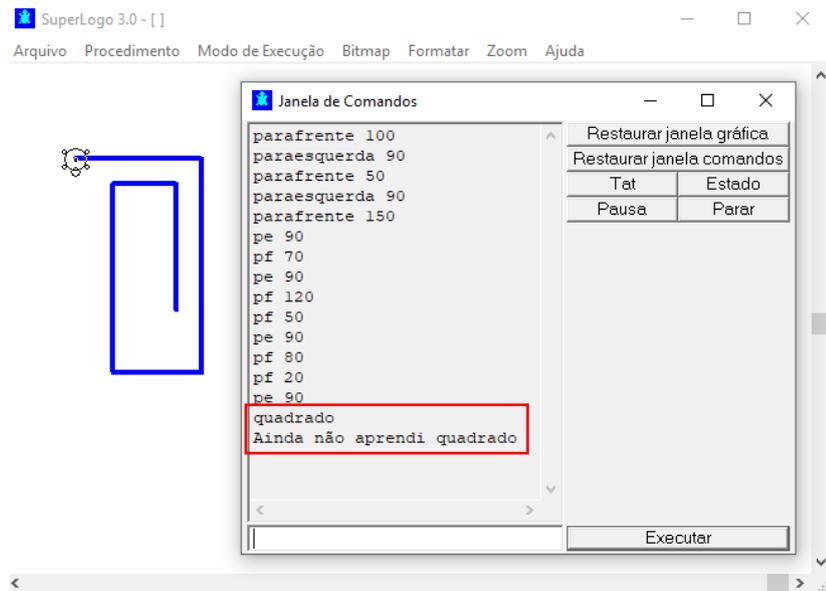
A proposta da linguagem LOGO era permitir que a criança controlasse, através dos comandos em LOGO, um robô (conhecido como Tartaruga) ou a representação desse robô na tela do computador (Figura 1 – Representação dos comandos da linguagem LOGO). O robô, então, aprenderia com a criança o que deveria fazer.

Com o domínio dos comandos mais elementares, o estudante já é capaz de elaborar pequenas ações que são realizadas pela Tartaruga através de comandos dados ao computador. Essas ações são procedimentos programados que, embora inicialmente simples, podem compor, posteriormente, um programa mais elaborado (MOURA; ACUNZO, 1985). Esse método de representar as ideias e soluções de uma determinada situação acabam por representar os pensamentos da criança, dando um significado novo ao aprendizado. A ferramenta está a serviço da construção dos saberes da criança, e esta passa a não ser mais uma mera receptora passiva.

Segundo Papert (1985), é possível delimitar também uma segunda parte da linguagem LOGO, a qual ele chamou de ambiente LOGO. Esse ambiente refere-se às condições em que a criança deve aprender e utilizar a linguagem para que esta seja eficiente dentro da teoria Construcionista.

O autor defende ser ideal que a LOGO seja aprendida de maneira flexível e ativa pela criança. Ela deve realizar suas interações com o computador de maneira não dirigida. Isso significa dar à criança a autonomia de desenvolver suas ideias e construir seus saberes ativamente. Dessa forma, até os próprios erros cometidos durante a programação passam a ter uma interpretação diferente: o erro não tem mais conotação negativa e nem danos psicológicos devido à exigência de respostas corretas. Afinal, o erro nessa linguagem ocorre não porque a criança errou um conceito, mas porque a máquina não soube interpretar o que ela quis representar, como ilustrado ao final da lista de comandos na Figura 2.

Figura 2 – Exemplo da linguagem LOGO⁶ e de uma mensagem de erro⁷



Fonte: Autores.

Essa abordagem é clara nas mensagens de erro da linguagem (Figura 2) e traz uma nova perspectiva aos erros: são parte do processo de solução de um problema, pois em caso de erros assim, o estudante deve ensinar a máquina a como realizar o novo comando.

2.4 MICROMUNDO FÍSICO: A PROPOSTA ORIGINAL DE PAPERT

Ao fazer uma analogia com os obstáculos encontrados na aprendizagem tradicional da Matemática, Papert (1985) discute as dificuldades enfrentadas pelos alunos na compreensão de conceitos da Física newtoniana.

Segundo ele, os currículos tanto de Matemática quanto de Física apresentam falhas semelhantes, que colaboram para a rápida perda do interesse pela disciplina, por parte dos alunos. Essas falhas concentram-se, em parte, no excesso de pré-requisitos para a disciplina, que acabam tomando quase todo o tempo disponível e minando a motivação inicial dos alunos pela compreensão dos conceitos físicos.

A aprendizagem das Leis de Newton por si só apresenta uma grande contradição para os estudantes: a lei da Inércia, do movimento perpétuo na ausência de forças resultantes, por exemplo. Este caso em especial gera uma enorme controvérsia entre as experiências cotidianas do aluno e o conhecimento científico. Em geral, os estudantes sabem (decoram) as leis fundamentais da mecânica, porém, suas explicações para fenômenos de seus cotidianos geralmente apelam para a Física aristotélica.

Segundo Palangana (2015), Piaget propunha que a assimilação do conhecimento ocorre quando o aluno absorve o novo saber a partir do anterior, construindo, assim, o conhecimento quando age sobre o objeto. Mas, no caso dos princípios da mecânica, o aluno provavelmente não tem experiências anteriores que remetam à Física como ela é descrita nos livros e apostilas.

O aluno pode jamais ter presenciado uma pena e uma barra de ferro caírem à mesma aceleração numa câmara de vácuo ou uma cadeira, ao ser empurrada, continuar em movimento eternamente, a menos que alguma força resultante a faça parar. O aluno pode não conseguir relacionar que essa força que faz a cadeira parar é a força de atrito, baseando-se apenas em suas vivências cotidianas. Para ele, esses detalhes são tão intrínsecos ao dia-a-dia que acabam passando despercebidos. Para o aluno, pode ser impossível imaginar um ambiente sem a existência de atrito ou matéria, simplesmente porque ele não presencia nada semelhante a isso em seu cotidiano. Portanto, insistir nessa idealização sem fundamentos palpáveis torna a Física ainda mais mirabolante e contraintuitiva. Nesse aspecto, o autor destaca que:

Acredito que a aprendizagem da Física consiste em colocar o conhecimento de Física em contato com o conhecimento pessoal bastante diversificado. E, para que isto aconteça, devemos permitir que o aprendiz construa e trabalhe com sistemas transitórios que os físicos talvez se recusem a reconhecer como Física (PAPERT, 1985, p. 150-151).

Com isso, apesar de toda a evolução científica que alcançamos, a interação dos alunos com o ambiente cotidiano gera experiências que não convergem diretamente para um pensamento newtoniano. Então, quando esses alunos estudam as leis do movimento, elas simplesmente não fazem sentido algum: não houve experimentações cotidianas com objetos newtonianos como descritos nos livros e com as condições que permitem a verificação clara dessas leis. Diante dessas falhas, é comum os professores recorrem a modelos matematizados para representar objetos newtonianos e conjecturas de situações ideais. Primeiro problema: o aluno jamais presenciou essas condições ideais, sendo que tais são completamente contraintuitivas para ele; segundo problema: o aluno provavelmente não terá o aporte matemático necessário para trabalhar com esses modelos matematizados.

Essa contradição entre a idealização da teoria e as experiências cotidianas, somadas à dificuldade de aplicar princípios de Matemática básica, resultam em uma das grandes dificuldades encontradas no aprendizado da disciplina de Física.

Diante disso, Papert propõe o desenvolvimento de uma nova sequência de aprendizagem para as leis de Newton da mecânica. Nessa sequência, inclui-se uma abordagem mais lúdica, o que não significa que seja mais fácil. Afinal, as leis de Newton continuarão não fazendo sentido, a menos que elas sejam aplicadas de modo que possam ser vistas mais como algo familiar do que apenas um capítulo nos livros acadêmicos. Para fazer isso, Papert propõe:

[...] um ambiente de aprendizagem interativa baseado no computador onde os pré-requisitos estão embutidos no sistema e onde os aprendizes podem tornar-se ativos, arquitetos construtores de sua própria aprendizagem (PAPERT, 1985, p. 151).

Esse ambiente interativo é o micromundo computadorizado. Um mundo simulado onde o aluno terá acesso à Física newtoniana pura. Ao contrário do aprendizado convencional, no micromundo físico não há a necessidade de o aluno ter o domínio inicial de equações matemáticas. Longe disso, o micromundo primeiro irá proporcionar ao aluno as intuições sobre as leis do movimento, dentro de um contexto simulado e interessante.

Dessa forma, o micromundo de Papert parte do princípio de conhecer intuitivamente o mundo simulado da mecânica primeiro, para, após a assimilação desses conhecimentos experenciados, fazer a interligação com os formalismos intrínsecos à Física newtoniana. Assim, os pré-requisitos não minarão o interesse inicial do aluno pela Física e o conhecimento se construirá de forma completa e dinâmica, proporcionado pela prática no micromundo.

Porém, como foi salientado por Valente (1993), o fato de uma ferramenta ser interativa, não significa necessariamente que ela proporcionará uma aprendizagem ativa, como é o caso de simulações, por exemplo. Nesse sentido, Papert explica que o micromundo físico supre essa falha através da liberdade que esse ambiente proporciona ao aluno, permitindo que ele modifique as condições e explore suas próprias hipóteses acerca das leis:

Aprendizes num micromundo físico são capazes de inventar seus próprios conjuntos de suposições sobre esse micromundo e suas leis, e são capazes de torná-los reais. Eles podem moldar a realidade em que irão trabalhar hoje, podem modificar e construir alternativas. Esta é uma maneira eficiente de aprender [...] (PAPERT, 1985, p. 155).

Para apresentar essa proposta de micromundo físico, Papert utilizou as bases do que chamou de Matelândia, um micromundo criado anteriormente com a finalidade de auxiliar a aprendizagem da Geometria. Esse micromundo para a Matemática era protagonizado pela mesma Tartaruga, orientada pelos comandos da linguagem LOGO.

Um aspecto interessante dessa linguagem no micromundo é que quando o aluno erra algum comando, isso não é interpretado pelo *software* como um erro que deve ser consertado rapidamente, como ocorreria numa aula convencional de Matemática. Na verdade, o termo erro não é empregado na linguagem LOGO, mas sim *bug*. Dessa forma, o aluno é estimulado a estudar o problema e eliminá-lo, através do processo chamado *debugging*, no qual ele analisa por que ocorreu o *bug* e encontra uma solução para o mesmo. O erro deixa de soar como algo pejorativo e passa a fazer parte do processo de aprendizagem.

Analogamente à Matelândia, o micromundo proposto por Papert para o estudo das leis do movimento na Física utiliza uma Tartaruga Dinâmica, a *DinaTart*. Esta é mais dinâmica, pois, além dos componentes orientação e posição, possui também a componente de velocidade e aceleração.

Assim como na Matelândia, onde as ideias matemáticas podiam nascer e desenvolver-se facilmente, como numa incubadora de conhecimento, o micromundo proposto por Papert para a Física newtoniana proporciona o desenvolvimento de pensamentos mais concretos e favorece o desenvolvimento das estruturas intelectuais do indivíduo. Para tanto, não basta que o micromundo traga apenas a idealização do mundo das Leis de Newton como elas são de maneira correta. A vantagem do micromundo neste caso é a possibilidade de exemplificar visões e abordagens diferenciadas sobre o movimento, ilustrando desde o mundo Aristotélico do movimento, fazendo assim uma abordagem referente à história da Física e até mesmo relacionando-o com o senso comum, por exemplo. Sem contar que é possível ir além de Newton, explorando conceitos mais complexos até mesmo a partir de Einstein. Fora as teorias físicas em si, o micromundo simulado oferece ao indivíduo a liberdade de desenvolver suas próprias leis do movimento,

como situações intermediárias, de modo a explorá-las e compará-las a modelos anteriores já comprovados. Quanto a isso, Papert destaca:

Desse modo, os aprendizes podem progredir de Aristóteles até Newton e mesmo até Einstein através de tantos mundos intermediários quanto eles desejarem. [...] os obstáculos matéticos⁸ à compreensão de Newton são superados: os pré-requisitos estão baseados no conhecimento pessoal e o aprendiz é envolvido numa exploração criativa das ideias e da variedade de leis do movimento (PAPERT, 1985, p. 78).

De acordo com o autor, para que o micromundo seja capaz de auxiliar na compreensão das leis de Newton do movimento, ele precisa ser projetado de forma a superar os três grandes bloqueios que os estudantes enfrentam ao ter contato com as leis de Newton:

1. A discrepância existente entre a realidade que o aluno conhece, que experiencia e a descrição das leis. Papert defende que antes de colocar as definições padrão e contraintuitivas das Leis de Newton, é interessante que os alunos desenvolvam o conceito de lei de movimento através de exemplos mais simplificados
2. As leis, como são enunciadas, têm como fim somente a resolução de exercícios abstratos. Dessa maneira, não dão espaço às aplicações manipuláveis dessas leis por parte do aluno. Nesse âmbito, o micromundo oferece a possibilidade de manipulações dessas leis, tornando as atividades nele um trabalho relevante e menos preso às definições teóricas.
3. Os conceitos pré-requisitados para a compreensão das leis de Newton, como os de estado, de inércia ou mesmo de ausência de atrito, vácuo. Grande parte desses conceitos não são, a rigor, experienciados no cotidiano de ninguém e são assimilados pelos estudantes de forma meramente abstrata, sem significado claro. O micromundo proposto fornece a experimentação desses conceitos de forma ativa dentro das simulações.

A proposta do micromundo de Papert em 1985 fundamentava-se, inicialmente, em remodelar as regras de movimento do personagem interativo do ambiente virtual (*DinaTart*) às leis de Newton do movimento, desenvolvendo as chamadas leis de movimento da Tartaruga. A *DinaTart* mover-se-ia obedecendo aos operadores definidos para ela, os quais modificavam direção, velocidade e aceleração. Por exemplo, de acordo com o próprio Papert (1985, p. 79), o operador PARAFRENTE muda a posição, enquanto o operador PARADIREITA muda a orientação da *DinaTart*. Dessa forma, é possível relacionar essa mudança de *estado* à atuação de uma força. O efeito dessa força é mudar a velocidade da *DinaTart*, ou mesmo o seu momento, surgindo então uma aceleração. Dessa forma, o micromundo seria uma porta para a compreensão da essência das leis de Newton.

Para Santos et al. (2010), através desse processo de analogia, o estudante conheceria, de início, outras leis de movimento mais simples e não tão contra intuitivas como o são as de Newton. Tais simplificações abririam caminho para os conceitos mais complexos envolvidos nas leis de Newton propriamente ditas.

Assim, o autor sugerira que, para diminuir ainda mais a diferença entre o mundo da Tartaruga e de Newton, poderiam ser projetados micromundos cada vez

mais semelhantes às condições newtonianas, à medida que o estudante fosse evoluindo na compreensão dos conceitos envolvidos.

Entretanto, após mais de 30 anos da proposta original de Seymour Papert, poucos trabalhos propuseram-se a desenvolver micromundos ou a utilização de aplicativos existentes com esse fim para aprendizagem.

PROPOSTAS EXISTENTES

Desde o surgimento dos primeiros computadores, o potencial das simulações vem sendo explorado para os mais diversos fins, tais como o cálculo de trajetórias complexas de satélites e foguetes ou fluxo de fluidos, por exemplo (HAVOK⁹, 2008, p. 371). Atualmente, as simulações, mundos virtuais e tecnologias imersivas estão cada vez mais presentes não só no campo técnico, mas também no cotidiano das pessoas, em especial dos estudantes, como entretenimento e diversão, sendo em maioria fiéis à realidade.

Na educação, tais realidades simuladas podem, segundo Walker (1990), colaborar para a aprendizagem através da interação com os instrumentos virtuais, os avatares e as próprias regras e condições definidas pelo mundo virtual.

Ainda quanto às realidades simuladas, Dede (1992)¹⁰ salienta que essa abordagem para o ensino auxilia o estudante na aplicação de conceitos mais abstratos, justamente por situá-los em contextos virtuais semelhantes à realidade, onde de fato aqueles conceitos serão utilizados posteriormente.

Entretanto, falando do ensino de Física e do movimento newtoniano, a semelhança com a realidade desses mundos virtuais acaba por apenas aumentar os problemas. Pois os estudantes praticamente não têm experiências com a Física puramente newtoniana no seu cotidiano. As simulações realísticas vêm apenas trazer condições simuladas ainda mais limitadas, seja pelas interfaces fechadas, não permitindo qualquer modificação na estrutura ou pela imposição de parâmetros do mundo virtual que devem ser seguidos, como a física padrão imposta aos objetos, por exemplo.

Dessa forma, os estudantes não encontram a possibilidade de experienciar de fato as condições das diferentes leis de movimento, muito menos têm a liberdade de inventar suas próprias leis. Mas antes, são compelidos a assimilarem as teorias corretas sobre o movimento, mesmo que não tenham os pré-requisitos mínimos para compreendê-las. Portanto, mesmo com o uso dos mundos virtuais fiéis à realidade, ao final do processo, o movimento newtoniano é aprendido novamente através da manipulação de equações, em vez da manipulação dos próprios objetos.

3.1 PROPOSTA DE MICROMUNDO FÍSICO NO *SECOND LIFE*

Desde a proposta original de Papert (1985), vêm sendo construídos micromundos com fins pedagógicos – ou não, em diversas plataformas, desde as primitivas Tartarugas em linguagem LOGO até ambientes com tecnologia de realidade simulada (ZORZAL, 2008).

Dentro dessa última categoria, enquadra-se o *Second Life* (SL). Trata-se de um mundo virtual 3D que simula ambientes e situações cotidianas da vida real dos

seres humanos. Foi fundado por Philip Rosedale, em 2003, e permanece mantido pela empresa *Linden Lab*. Embora os ambientes, avatares e objetos do SL possam ser modificados, eles respeitam as leis físicas estabelecidas, o que mantém a semelhança com o mundo real (BESTEBREURTJE, 2007).

No quesito de educação, o *Second Life* já despertou o interesse de instituições de ensino e até empresas privadas, sendo utilizado para o ensino de línguas estrangeiras e conversação, salas de aula virtuais e exploração de ambientes simulados em quesito de biomas, Química e Física. Porém, estas tentativas enfrentaram dificuldades e não obtiveram resultados consideráveis no sentido de ir além dos parâmetros do ensino tradicional, pois constatou-se que “[...] a primeira tendência de exploração/utilização dessa nova tecnologia é levada a cabo numa lógica de reposição das práticas educativas habituais” (BETTENCOURT; ABADE, 2008, p. 14).

Quanto à Física, especificamente, as experiências dos usuários incluem a visualização de situações e fenômenos físicos dentro do mundo virtual, além da própria física simulada no mundo. Tal física e suas leis majoritárias definem que avatares não devem ultrapassar paredes e que os objetos em geral comportem-se como é de se esperar num mundo semelhante ao nosso, ou seja, obedecendo quesitos de massa, atrito, densidade, gravidade, etc. Para que tais leis sejam consideradas no mundo virtual, é utilizada uma *engine* de física, um *software* destinado a simular modelos teóricos e realizar cálculos físicos dentro das situações do jogo ou simulação.

Essa Física do SL beira duas definições, sem enquadrar-se plenamente em nenhuma: nem totalmente realística, por permitir movimentos anormais como em quedas de objetos, por exemplo; nem tão surreal, por não viabilizar uma física mirabolante ou fruto dos ensejos de seus usuários.

Embora o SL, devido a todas as suas características, desperte o interesse dos educadores para o seu potencial em promover simulações que auxiliem no ensino da Física, não se conhecem relatos concretos de sua utilização para tal fim. Conhecem-se apenas objetos deixados à livre manipulação pelos usuários, mas sem finalidades objetivas. Portanto, segundo o próprio autor da pesquisa relacionada à questão da física no *Second Life*:

A Física do *Second Life* não é nem a virtualização da Física do ‘mundo real’ nem a da ‘ideal’ Galileana/Newtoniana, para concluir que ela é ‘surreal’ e, ainda assim, um ambiente rico para discussão da realidade/irrealidade das leis físicas ensinadas na escola (SANTOS, 2008, p. 8).

Diante da considerável dificuldade na utilização do *Second Life*, na criação de objetos e na própria compreensão da linguagem de programação do *software*, a *Linden Scripting Language* (LSL), e considerando o potencial do SL, o próprio autor anterior propõe, após quatro anos das primeiras pesquisas, o desenvolvimento de micromundos físicos dentro do *Second Life*, utilizando a chamada *TATIlogo*, uma linguagem derivada da primitiva LOGO, que embora limitada, evitaria a necessidade de alto grau de conhecimento sobre a nativa linguagem de programação do *software* SL. Nas figuras a seguir, está ilustrado um exemplo de objetos criados com a interface TATI (Figura 3 – Objetos criados com a interface TATI), desenvolvidos pelo autor.

Figura 3 – Objetos criados com a interface TATI



Fonte: Santos (2012).

A TATILogo proposta por Santos (2012) prevê operadores com funções idênticas às das *DinaTart* da proposta original de Papert (1985), como as tartarugas de velocidade e aceleração, por exemplo. Outras sequências de comandos apresentadas no trabalho definem o desenho de uma circunferência e a projeção de uma trajetória em forma de circunferência. Ao concluir, o autor salienta que:

[...] a sequência de tipos de objetos, tal como no exemplo acima das trajetórias em circunferência, realiza a proposta de Papert de uma sequência de aprendizagem piagetiana para a Física Newtoniana (PAPERT, 1985, p. 152), [...] contribuindo para melhorar a generalizada dificuldade em distinguir entre os conceitos de velocidade e aceleração (SANTOS, 2012, p. 7).

O autor defende ainda que a linguagem TATI facilita a criação de simulações no SL, sendo que os estudantes poderiam relacionar os novos conhecimentos aos já existentes e, além disso, apropriar-se deles brincando, criando coisas novas através deles, condizendo com a idealização de Papert.

Mesmo assim, a linguagem TATI apresenta-se bastante limitada, buscando simplificar os problemas com a programação original do SL. Além disso, há demais obstáculos na proposta, como as próprias limitações da física do *Second Life* e das restrições quanto à criação de leis de movimento diferentes das corretas.

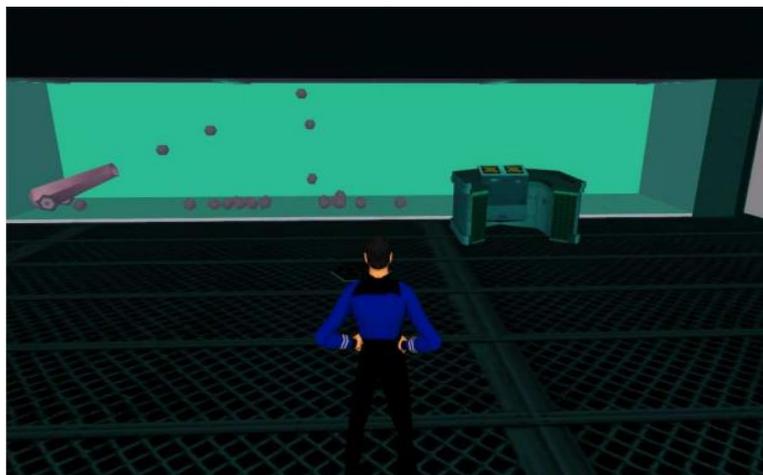
Para simular situações que não sigam estritamente o movimento newtoniano, Santos (2011) implementou uma simulação que compara dois parâmetros de trajetórias de lançamento já conhecidas: uma primeira trajetória parabólica como prevista pela teoria de Galileu/Newton, Figura 4 – Projéteis seguindo a usual trajetória parabólica da Mecânica Newtoniana, e outra como prevista pela teoria do Ímpetus de Buridan¹¹, Fonte: Santos (2011). Figura 5 – Projéteis reproduzindo a trajetória prevista pela Teoria do Impetus, muito difundida no período Medieval. A alternância entre as duas formas de trajetória é controlada por dois botões em um painel de controle.

Figura 4 – Projéteis seguindo a usual trajetória parabólica da Mecânica Newtoniana



Fonte: Santos (2011).

Figura 5 – Projéteis reproduzindo a trajetória prevista pela Teoria do *Impetus*



Fonte: Santos (2011).

Segundo o próprio autor (Santos, 2011, p. 9), o simulador do canhão “embora muito simples é o primeiro dispositivo que realiza a proposta de Papert (1985).” Entretanto, considerando-se as premissas do micromundo físico estabelecidas por Papert, a simulação do canhão atende somente ao requisito de proporcionar a experiência de outra lei do movimento, além da de Newton. Mesmo assim, esse requisito é atendido parcialmente, uma vez que se limita à apenas uma teoria alternativa (a do Ímpetus de Buridan). Além disso, outros fundamentos essenciais do micromundo físico não são considerados. O estudante é representado pelo avatar que observa as simulações dentro do micromundo. Logo, ele não tem um papel ativo no mundo virtual, sendo apenas espectador. Por isso, o estudante não pode agir sobre os objetos, modificar o micromundo e suas condições iniciais, muito menos criar suas próprias leis de movimento, como seria adequado. Portanto, a proposta de Santos (2012) se assemelha às simulações destinadas ao ensino de Física já encontradas no mercado atualmente, como é o caso das simulações PhET¹².

Nesse sentido, os mundos de realidade simulada conhecidos até então, voltados ao ensino de Física, de modo geral, não atendem aos requisitos de um

micromundo físico capaz de quebrar o bloqueio do aprendizado dos conceitos físicos, tal como concebido por Papert (1985), pois, 1: não proporcionam a compreensão anterior do conceito de leis de movimento com exemplos mais simplificados; 2: permitem a manipulação limitada dos objetos, mas não das próprias leis do movimento, sendo estas definidas como padrões no mundo virtual e 3: as realidades simuladas, em geral, não oferecem maneiras de compreender o movimento sem antes exigir do estudante os pré-requisitos que ele sequer tem a oportunidade de conhecer no mundo real.

Portanto, percebeu-se que as propostas até então apresentadas permanecem com lacunas, sem satisfazer a rigor a proposta construcionista de Papert (1985), tanto pelas limitações da ferramenta utilizada quanto pelas falhas dos próprios micromundos implementados.

MICROMUNDO FÍSICO WEBGL

A década de 1980 oportunizou mudanças substanciais no setor da computação, as quais abriram caminho para o acelerado desenvolvimento tecnológico da sociedade. Dentre tais mudanças, duas merecem especial destaque: a popularização do computador pessoal, com sua disseminação a preços acessíveis para grande parcela da população. A segunda mudança foi a conexão em rede desses computadores, ou seja, a *Internet*, que assim como os computadores, também passou a ser acessível a usuários domésticos e a empresas menores. Esses dois fatores aliados favoreceram a evolução meteórica da tecnologia desde então, remodelando o funcionamento e as relações na sociedade humana (WAZLAWICK, 2017).

Diante de tais mudanças, os teóricos da educação buscaram compreender como essa tecnologia poderia agregar ao setor educacional, propondo um ensino que seguisse os novos parâmetros que a sociedade passara a adotar. Mesmo assim, o avanço nesse quesito é lento e ainda hoje poucas tecnologias foram utilizadas com sucesso na educação.

Diversas propostas e ferramentas surgiram para fazer uso da tecnologia no ensino de forma significativa e envolvente pelos estudantes, começando pela linguagem LOGO desenvolvida por Seymour Papert e sua equipe do MIT¹³.

Com a expansão e a diversificação das ferramentas tecnológicas à disposição, a linguagem LOGO original passou a ser menos utilizada. Porém, em 2007, foi desenvolvido o projeto *Scratch*, pelo grupo do *Lifelong Kindergarten Group* do MIT Media Lab. O *Scratch* pode ser considerado um descendente da linguagem LOGO original e permite a programação de histórias, jogos e animações interativas através da combinação de blocos lógicos¹⁴.

4.1 TECNOLOGIA WEBGL

Outras tecnologias não foram desenvolvidas especialmente voltadas à educação, porém foram adotadas para tal fim através de diversas abordagens, inclusive tentativas de implementar os micromundos físicos propostos por Papert ainda na década de 1980, sendo uma destas o *Second Life* (apresentado na subseção 3.1).

Além dos obstáculos já discutidos, o SL trata-se de um *software* que deve ser baixado e instalado no computador do usuário, o que, dependendo do poder de processamento da máquina, pode comprometer o seu desempenho.

Em contrapartida, cada vez mais percebe-se o aumento da preferência por plataformas e/ou ferramentas que funcionem diretamente no navegador *Web*, sem necessitar de *plug-ins* ou instalações de *softwares* extras. Uma das tecnologias que oferecem essa versatilidade é chamada de WebGL¹⁵, do inglês *Web Graphics Library*, que é uma API *JavaScript*¹⁶ projetada para renderizar gráficos 3D interativos e gráficos 2D apenas através de um navegador *Web* compatível.

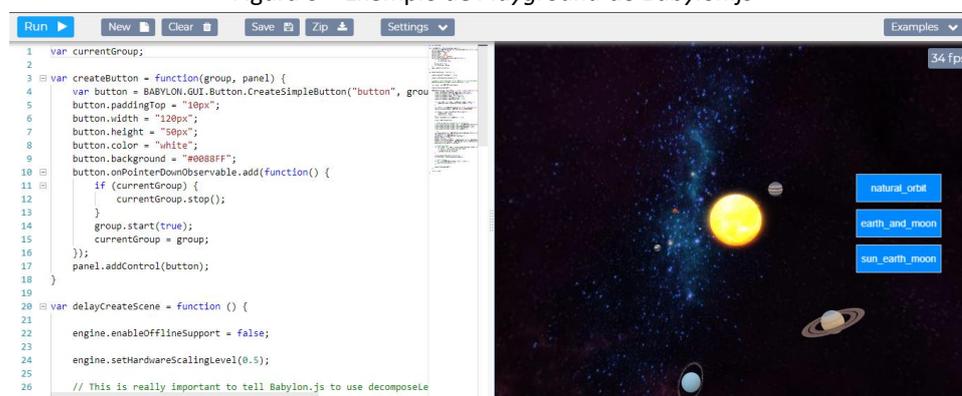
4.2 BABYLON.JS

As possibilidades abertas pelas novas tecnologias Web3D¹⁷ e sua versatilidade permitiram o desenvolvimento de um poderoso mecanismo de criação de jogos, simulações, objetos 3D e ambientes virtuais em *JavaScript*: o *Babylon.js*, lançado inicialmente em 2013.

Seus criadores projetaram o *Babylon.js* de forma que fosse uma *engine* 3D fácil de usar e acessível, escrita em *JavaScript*. O projeto cresceu e atualmente o *Babylon.js* encontra-se na versão 3.2 (maio de 2018), sob a licença do *Apache 2.2*. Assim, o *Babylon.js* se define como uma estrutura completa de *JavaScript* para a criação de jogos 3D com HTML 5 e WebGL. Seu *site* oficial contém muitos tutoriais para iniciantes e para usuários mais avançados, com demonstrações para cada recurso e cenas como exemplos (MOREAU-MATHIS, 2016)

O *Babylon.js* foi projetado para desenvolver jogos, animações e experiências em 3D com WebGL, em tempo real. Para isso, a interface da plataforma, chamada de *Playground*¹⁸ (Figura 6), é dividida entre o editor de programação, à esquerda, e a tela de renderização, à direita, onde ficam os objetos e ambientes simulados. Isso proporciona a visualização instantânea de qualquer modificação feita no código. Na tela de renderização, é o *framework*¹⁹ WebGL que renderiza a cena que é escrita em *JavaScript* no editor.

Figura 6 – Exemplo de *Playground* do *Babylon.js*



Fonte: BABYLON.JS: Solar System.

Além disso, *Babylon.js* é um projeto cujo código é aberto, para que qualquer pessoa possa estudá-lo, modificá-lo e aprimorá-lo. Esse fato também permite que o *software*/plataforma seja colaborativo entre usuários. Também por se tratar de

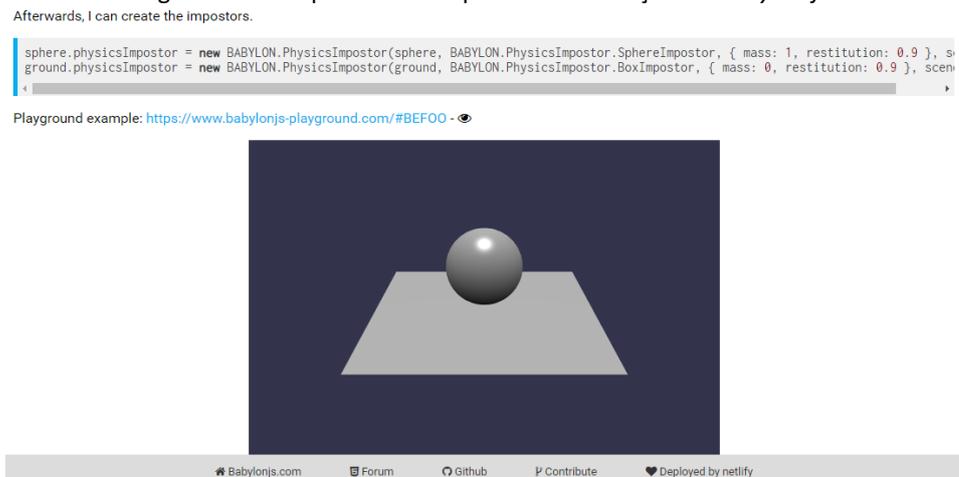
uma plataforma de código aberto, o *Babylon.js* está em contínuo desenvolvimento, com novos recursos sendo acrescentados, pois conta com fóruns e grupos voltados ao esclarecimento de dúvidas e ao desenvolvimento de novos projetos.

Quanto à física, o *Babylon.js* a suporta através do motor (ou *engine*) de física *Cannon.js*, que é uma versão escrita à mão do *Ammo.js*. Com essa *engine*, pode-se criar uma cena completa com luzes, câmeras, materiais e malhas. Também suporta um mecanismo de animação, um sistema de partículas, colisões, mapas de altura, malha e efeitos de pós-processamento da câmera, renderização de camadas, texturas de vídeo, etc. (STAMOULIAS, 2014).

O *Babylon.js* também oferece em seu *site* uma documentação completa sobre seus recursos e possibilidades. Apesar de ainda não haver uma versão na Língua Portuguesa, a documentação é de fácil compreensão, bem estruturada e com uma linguagem simples. Em cada tópico, são fornecidos exemplos não só da programação, mas também na tela de renderização, diretamente no *Playground*, como ilustra a Figura 7.

Tais exemplos e o recurso de visualização rápida favorecem a compreensão e a experimentação simplificada de todos os recursos da plataforma, mesmo por usuários leigos.

Figura 7 – Exemplo fornecido pela documentação do *Babylon.js*



Fonte: Documentação do *Babylon.js*.

4.3 POSSIBILIDADE DE MICROMUNDOS FÍSICOS NO *BABYLON.JS*

Foram realizadas, no decorrer deste artigo, pesquisas em busca de trabalhos voltados ao ensino que propusessem ou desenvolvessem de fato micromundos físicos ou apenas implementassem mundos virtuais voltados ao ensino. Porém, até o momento da elaboração deste trabalho, não foram encontradas propostas de implementação de micromundos físicos utilizando o *framework Babylon.js*, embora existam propostas diversas utilizando *softwares* como o *Modellus*, mesmo que não empreguem a denominação de micromundo ou não atendam todos os requisitos para serem definidos como tal.

A proposta de Santos (2011) para o uso do *SL* como aporte para os micromundos físicos de Papert, voltada especificamente para o ensino de Física, embora, aparentemente, pioneira nesse quesito, não atende, adequadamente, aos pré-requisitos para um micromundo físico construcionista como idealizado por Papert, como já discutido em tópico anterior (3.1 PROPOSTA DE MICROMUNDO FÍSICO NO *SECOND LIFE*).

Diferentemente da tecnologia do *Second Life*, o *framework Babylon.js* é uma estrutura de código aberto, programável em *JavaScript* e dispensa quaisquer instalações extras de ferramentas ou *softwares*, sendo capaz de renderizar simulações e jogos através da tecnologia *WebGL* diretamente no navegador. Além disso, o *Playground* do *Babylon.js* é estruturado de forma a dividir a tela entre o editor do código e a tela de renderização. No editor interno do *Playground*, o código é programado em linguagem *JavaScript* e os resultados da programação podem ser visualizados instantaneamente na tela de renderização ao lado. Isso permite uma interação muito mais efetiva com a animação que está sendo produzida. Além disso, mesmo que o usuário acesse uma simulação já previamente elaborada, ele tem a liberdade de modificá-la da forma como lhe convier, não ficando preso somente aos controles oferecidos na tela de renderização.

4.4 PROGRAMAÇÃO EM JAVASCRIPT

A linguagem *JavaScript* (JS) é amplamente utilizada no desenvolvimento *Web* e o código é executado no próprio navegador *Web*, não no servidor. Embora tenha sido influenciada pela linguagem *Java*, ela possui uma sintaxe muito diferente desta, assemelhando-se à linguagem *C*.

Quanto à programação em *JavaScript*, a mesma não é tão complexa e é uma das linguagens mais indicadas para quem começam a programar. Isso se aplica também às crianças, pois o JS permite a produção de animações e jogos, o que, segundo Strom (2018), faz com que a programação se torne um caminho prazeroso para alcançar um objetivo cativante para a criança.

4.5 A ENGINE DE FÍSICA DO BABYLON.JS

A *engine* de física diferencia-se de um sistema de colisões interna porque ela calcula a dinâmica dos corpos e objetos em uma cena, emulando interações que ocorreriam no mundo real. O *Babylon.js* suporta três *engines*: *Cannon.js*, *Oimo.js* e *Energy.js*.

Por questão de versatilidade e por ser mais completa, neste trabalho optou-se por descrever a *engine Cannon.js*. Assim, para ativar esse mecanismo de física no *Playground*, basta chamar a função `enablePhysics` na cena, como na Figura 8:

Figura 8 – Ativando a *engine* de física

```
var scene = new BABYLON.Scene(engine);  
var gravityVector = new BABYLON.Vector3(0,-9.81, 0);  
var physicsPlugin = new BABYLON.CannonJSPlugin();  
scene.enablePhysics(gravityVector, physicsPlugin);
```

Fonte: Documentação do *Babylon.js*.

Para que seja possível a interação física entre objetos, as *engines* utilizam um *impostor*, que é em geral um corpo rígido, uma representação simplificada de objetos mais complexos. Cada *engine* de física fornece *impostors* diferentes. Para a *Cannon.js*, tem-se os *impostors* do Quadro 1:

Quadro 1 – Impostors da engine Cannon.js

ENGINE Cannon.js	Tipo de impostor (INGLÊS)
Caixa	Box
Esfera	Sphere
Partícula	Particle
Plano	Plane
Cilindro	Cylinder
Malha	Mesh
Mapa de altura	Heightmap

Fonte: Adaptado da Documentação do *Babylon.js*.

Para habilitar a física em um determinado objeto, é preciso atribuir a ele um desses *impostors* físicos citados no Quadro 1, como mostrado na Figura 9:

Figura 9 – Novo objeto *impostor*

```
new BABYLON.PhysicsImpostor(object: IPhysicsEnabledObject, type: number, options: PhysicsImpostorParameters, scene: BABYLON.Scene);
```

Fonte: Documentação do *Babylon.js*.

Os *impostors* ficam da seguinte maneira:

Figura 10 – Tipos de *impostors*

```
BABYLON.PhysicsImpostor.SphereImpostor;  
BABYLON.PhysicsImpostor.BoxImpostor;  
BABYLON.PhysicsImpostor.PlaneImpostor;  
BABYLON.PhysicsImpostor.MeshImpostor;  
BABYLON.PhysicsImpostor.CylinderImpostor;  
BABYLON.PhysicsImpostor.ParticleImpostor;  
BABYLON.PhysicsImpostor.HeightmapImpostor;
```

Fonte: Documentação do *Babylon.js*.

Pode-se atribuir física à cena básica do *Playground*, a qual consiste em um solo (*ground*) e uma esfera (*sphere*), permitindo que haja interações físicas entre esses dois objetos. Para tal, basta ativar a *engine* de física (`scene.enablePhysics();`) e em seguida criar os *impostors* para o solo e a esfera:

Figura 11 – Criando *impostors* para o solo e a esfera

```
sphere.physicsImpostor = new BABYLON.PhysicsImpostor(sphere, BABYLON.PhysicsImpostor.SphereImpostor, { mass: 1, restitution: 0.9 }, scene);  
ground.physicsImpostor = new BABYLON.PhysicsImpostor(ground, BABYLON.PhysicsImpostor.BoxImpostor, { mass: 0, restitution: 0.9 }, scene);
```

Fonte: Documentação do *Babylon.js*.

No exemplo acima, a variável *mass* é a massa do objeto, definida em quilogramas, e *restitution* é a quantidade de força que o corpo devolve ao colidir. Um valor baixo não resultará em nenhum salto, já o valor de 1 resultará em uma interação mais intensa.

Outra função da *engine* de física é a velocidade linear, que é basicamente o movimento de um objeto na direção de um determinado eixo de coordenadas. Para definir a velocidade linear do objeto, usa-se `impostor.setLinearVelocity(new BABYLON.Vector3(x,y,z))`; definindo os valores de velocidade (em metros por segundo) nas coordenadas X, Y e Z.

Para obter a velocidade angular de um objeto, utiliza-se: `impostor.getAngularVelocity()`; cujo valor pode ser definido nesta função: `impostor.setAngularVelocity(new BABYLON.Quaternion(0,A,0,0))`; Neste caso, o valor da velocidade angular é o segundo a ser definido (A).

Para variar valores de velocidade (angular e linear), é necessária a aplicação de uma força ou impulso (segunda lei de Newton do movimento). Para isso, utiliza-se a função para aplicar impulso no *impostor*: `impostor.applyImpulse(new BABYLON.Vector3(0, 0, 0), sphere.getAbsolutePosition())`; A primeira variável é a direção e a quantidade de impulso a ser aplicada. A segunda se refere ao local do corpo em que o impulso será aplicado. No exemplo, `sphere.getAbsolutePosition()`, indica que o impulso será aplicado no centro de massa do corpo.

Existe também opções para a criação de impulsos ou explosões radiais e forças de campo, como a gravitacional. Tais forças apenas são aplicadas em *impostors* de massa diferente de 0. Também é possível adicionar funções de retorno que são acionadas quando um *impostor* colide com outro.

Todas as funções oferecidas pelas *engines* de física suportadas pelo *Babylon.js* podem ser consultadas em sua documentação²⁰, a qual conta com códigos de exemplo para cada função, além de diversos exemplos executáveis diretamente no navegador pelo *Playground*.

METODOLOGIA

Essa pesquisa possui uma metodologia de caráter exploratório, por tratar de uma proposta diferenciada das publicações anteriores, considerando os aspectos fundamentais do micromundo físico de Seymour Papert e da tecnologia adotada. Assim, pretende-se iniciar uma nova linha de investigação acerca desse tema e, conforme resultados, buscar a validação, ou não, da proposta.

Seymour Papert (1985) defendia um micromundo para a aprendizagem das leis de Newton do movimento, utilizando a *DinaTart* e modificações na linguagem LOGO. Neste trabalho, foi implementado um micromundo físico dentro da proposta construcionista de Papert, porém as possibilidades de aprendizagem foram ampliadas, não se restringindo somente às leis do movimento. O micromundo implementado, como exemplo, se constitui em duas câmaras de queda que têm a mesma estrutura fundamental, porém uma delas aborda a queda livre no vácuo e a outra a queda com resistência.

5.1 QUEDA LIVRE E QUEDA COM RESISTÊNCIA

Em Mecânica, a queda livre é o termo adotado para descrever o movimento de um corpo sujeito somente à força gravitacional da Terra, ou seja, no vácuo ou quando se é desprezada a resistência do fluido. Esse movimento é uniformemente variado²¹, de trajetória vertical retilínea, sendo que a aceleração é uma constante para todos os corpos nas proximidades da superfície terrestre, independentemente de sua massa, forma ou densidade e vale aproximadamente $9,81 \text{ m/s}^2$.

Como conceito físico, a aceleração significa a variação da velocidade (Δv) pelo respectivo intervalo de tempo (Δt). No caso da queda livre, a aceleração é a própria aceleração da gravidade (g), expressa por $g = 9,81 \text{ m/s}^2$. Dessa maneira pode-se deduzir:

$$\Delta v = g \cdot \Delta t$$

$$v - v_0 = g \cdot \Delta t$$

$$v = v_0 + g \cdot \Delta t$$

Considerando $t_0 = 0$:

$$v = v_0 + g \cdot t \quad \mathbf{(1)}$$

Esta é a função horária da velocidade do movimento uniformemente variado e estabelece como varia a velocidade escalar de um móvel no decorrer do tempo.

Integrando a função (1), tem-se:

$$\Delta s = \int_{t_0}^t v \, dt \quad \mathbf{(2)}$$

Δs = variação da posição em função do tempo (m)

A velocidade média ($v_{méd}$) da função (1) para um intervalo de tempo t_0 até t é:

$$v_{méd} = v_0 + \frac{1}{2}at \quad \mathbf{(2.1)}$$

Agrupando os termos com o valor de v dado pela função (1), obtém-se:

$$v_{méd} = \frac{1}{2}(v_0 + v)$$

Da definição de velocidade média, onde s é:

$$s = s_0 + v_{méd}t \quad \mathbf{(2.2)}$$

Substituindo (2.1) em (2.2), obtém-se:

$$s - s_0 = v_0t + \frac{1}{2}gt^2$$
$$s = s_0 + v_0t + \frac{1}{2}gt^2 \quad \mathbf{(3)}$$

s = espaço final (m)

s_0 = espaço inicial (m)

t = instante de tempo (s)

As funções (1) e (3) são utilizadas para analisar o movimento em condições ideais, sem considerar coeficiente de resistência do fluido onde ocorre o

movimento, como se a queda se desse no vácuo. Portanto, as características do objeto, como forma ou massa, não são consideradas para efeito de cálculo.

Já para analisar o movimento como ocorre na realidade, deve-se considerar que um objeto em queda através de um fluido sofre uma resistência a esse movimento. A força de resistência do ar (\vec{F}_{ar}) pode ser representada pela equação:

$$\vec{F}_{ar} = -f(v)\hat{v}$$

Onde \hat{v} é o vetor unitário da velocidade e a função $f(v)$ é definida como:

$$f(v) = av + bv^2$$

As constantes a e b dependem do formato do objeto. Assim, a função possui um termo linear e um quadrático. O segundo termo predomina com o aumento da velocidade, podendo em certos casos desconsiderar o termo linear.

Cabe destacar que nos casos em que a velocidade é muito baixa, como na situação de partículas de poeira no ar, por exemplo, pode-se considerar que a força de resistência do ar é diretamente proporcional à velocidade. Em muitos casos – e neste trabalho – a velocidade atinge valores mais altos e a equação da força de arrasto a ser utilizada deve ser a quadrática e não a linear. Assim, a resistência do ar (F_d) com o quadrado da velocidade pode ser definida por:

$$F_d = \frac{1}{2} \rho Av^2 C_d$$

ρ = densidade da substância (kg/m^3)

A = área da seção transversal do objeto (m^2)

C_d = coeficiente de arrasto

Definindo a o coeficiente de resistência do ar, k :

$$k = \frac{1}{2} \rho AC_d$$

k em $\left[\frac{\text{kg}}{\text{m}}\right]$

Que implica:

$$C_d = \frac{2}{\rho A} k$$

Pode-se reescrever a força de arrasto como:

$$F_d = kv^2$$

Da Segunda Lei de Newton do movimento, pode-se escrever as forças que agem sobre o objeto em queda sob ação da gravidade (F_g) e da força devido à resistência do fluido (F_d):

$$F = F_g - F_d$$

$$ma = -mg + kv^2$$

Cabe destacar que o tratamento é escalar e não vetorial, tendo em vista que foi considerada a velocidade na mesma direção da força gravitacional.

Sendo que a aceleração é a variação da velocidade pelo decorrer de um intervalo de tempo e após dividir a equação acima por m , tem-se:

$$\frac{dv}{dt} = -g + \frac{k}{m}v^2 \quad \text{(I)}$$

Considerando $v(0) = 0$, a integração de (I),

$$\int_0^v \frac{1}{-g + \frac{k}{m}v^2} dv = \int_0^t dt$$

Fornece a função horária da velocidade em função do tempo para um objeto em queda sob ação da gravidade e da resistência do fluido:

$$v(t) = -\sqrt{\frac{mg}{k}} \tanh\left(\sqrt{\frac{kg}{m}} t\right) \quad \text{(4)}$$

m = massa do objeto (kg)

t = tempo (s)

g = aceleração da gravidade (m/s^2)

Sabendo que $v = \frac{dy}{dt}$, tem-se:

$$\frac{dy}{dt} = -\sqrt{\frac{mg}{k}} \tanh\left(\sqrt{\frac{kg}{m}} t\right) \quad \text{(II)}$$

A partir da integração de (II), chega-se, finalmente, à função horária das posições de um objeto em queda com resistência, considerando $y(0) = y_0$ como constante:

$$y(t) = -\frac{m}{k} \ln\left(\cosh\left(\sqrt{\frac{kg}{m}} t\right)\right) + y_0 \quad \text{(5)}$$

y_0 = posição inicial do objeto

5.2 A CÂMARA DE VÁCUO – QUEDA LIVRE

A primeira etapa do micromundo projetado no *Babylon.js* parte do princípio de uma câmara de vácuo, que simula o movimento de duas esferas em queda livre, movimento este descrito inicialmente pelas funções horárias de velocidade (1) e do deslocamento (3) do movimento retilíneo uniformemente variado (MRUV), respectivamente.

Inicialmente, foram definidas as variáveis fundamentais utilizadas no micromundo, como ilustrado na Figura 12:

Figura 12 – Variáveis fundamentais do micromundo

```
//Inserção das variáveis e da biblioteca numeric js
var gravidade = 9.81;
var massa = 1;
var massa2 = 1;
var velocidade0 = 0.0;
var url = "https://fisicagames.com.br/numeric-1.2.6.js";
```

Fonte: Autores.

A variável gravidade refere-se à aceleração da gravidade na Terra, cujo valor é $9,81 m/s^2$. As variáveis massa e massa2 referem-se aos valores iniciais de massa,

em kg, das esferas 1 e 2 na câmara, respectivamente. A variável `velocidade0` corresponde à velocidade inicial das esferas, que vale 0 m/s para ambas. E a variável `url` acrescenta a biblioteca matemática *Numeric JavaScript*²² em JS para permitir que funções mais complexas possam ser programadas posteriormente dentro do micromundo.

Em seguida, foram criadas as duas esferas e a câmara de vácuo, além de detalhes gráficos como a textura do solo, a semitransparência da câmara e o fundo de céu.

Foi criada também uma interface gráfica para o usuário (GUI: *Graphic User Interface*), que embora seja simples, possui um marcador do tempo de queda e de velocidade para cada esfera, além de uma barra de rolagem que permite a variação das massas das esferas (de 1 kg a 10 kg). A GUI também traz o botão executar, responsável por iniciar o movimento de queda das esferas e o botão Reiniciar, que reinicia a simulação (Figura 13 – GUI 1 e 2). Essa GUI serve como uma porta de entrada para as interações com o micromundo, pois, posteriormente, o usuário, já mais familiarizado, poderá modificar o ambiente totalmente através do console em JS.

Figura 13 – GUI 1 e 2



Fonte: Autores.

Para o cálculo dos valores de velocidade, foi utilizada a função horária da velocidade do movimento uniformemente variado (1), para cada uma das esferas. E para o cálculo das posições, foi utilizada a função horária do deslocamento (3), novamente para cada esfera.

Figura 14 – Funções horárias de velocidade e deslocamento

```
//Cálculo de velocidade e posição: esfera 1
scene.registerBeforeRender(function () {
    tempo += 1/60;
    if(executarGatilho == true) {
        tempoGUI.text = "Tempo de queda: " + tempo.toFixed(1)+ " s";
        sphere.position.y = 37-(velocidade0*tempo+(1/2)*gravidade*(tempo*tempo));
        sphere.velocidade1 = velocidade0+gravidade*tempo;
        velocidadeGUI.text = "Velocidade da esfera 1: " + sphere.velocidade1.toFixed(1) + " m/s";
    }
//Cálculo de velocidade e posição: esfera 1
    tempoGUI2.text = "Tempo de queda 2: " + tempo.toFixed(1) + " s";
    sphere2.position.y = 37-(velocidade0*tempo+(1/2)*gravidade*(tempo*tempo));
    sphere2.velocidade2 = velocidade0+gravidade*tempo;
    velocidadeGUI2.text = "Velocidade da esfera 2: " + sphere2.velocidade2.toFixed(1) + " m/s";
}
```

Fonte: Autores.

A primeira câmara de vácuo do micromundo, ainda no *Playground*, ficou como ilustra a Figura 15 – Micromundo da câmara de vácuo no *Playground*²³.

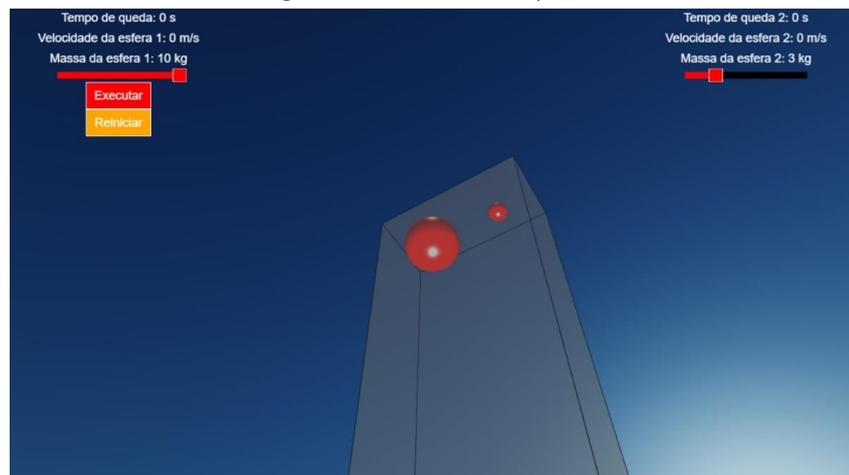
Figura 15 – Micromundo da câmara de vácuo no *Playground*²³



Fonte: Autores.

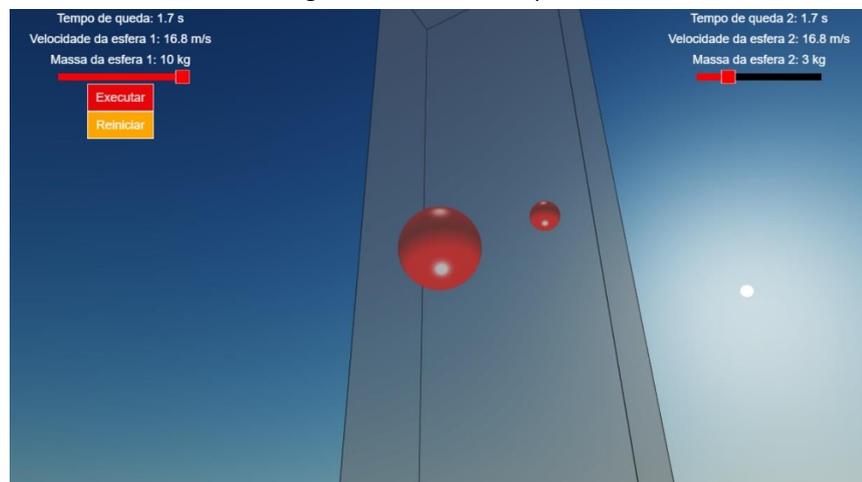
As figuras 16, 17 e 18 ilustram três momentos da queda das esferas na simulação dentro do micromundo, em modo *fullscreen*.

Figura 16 – Esferas em repouso



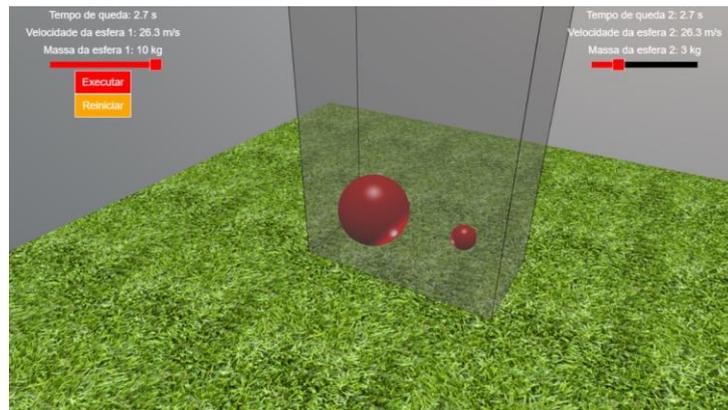
Fonte: Autores.

Figura 17 – Durante a queda



Fonte: Autores.

Figura 18 – Ao final do movimento



Fonte: Autores.

5.3 A CÂMARA DE QUEDA COM RESISTÊNCIA

A segunda câmara projetada diferencia-se da apresentada anteriormente por trazer em seu código as condições fundamentais que permitem a simulação de queda das esferas quando sob ação da resistência do ar.

Como visto anteriormente, a queda de um objeto com resistência depende, entre outros fatores, da forma desse objeto. Para esta segunda câmara, optou-se por fixar o valor do raio das esferas em 0,2 m. Dessa maneira, pode-se calcular a área da seção transversal das esferas:

$$A = 4\pi r^2$$

$$A = 4\pi(0,2)^2$$

Logo a área da seção transversal de cada esfera é:

$$A = 0,5 \text{ m}^2$$

O coeficiente de arrasto (C_d) no caso das esferas de raio igual a 0,2 m em queda no ar a 20 °C (cujo valor de k é 0,15 kg/m) corresponde a um valor adimensional de 0,5. O menor valor de k possível, depois do vácuo, é de $\cong 0,01$ kg/m, que corresponde ao gás Hidrogênio.

Assim, as variáveis fundamentais desta segunda câmara de queda são as mesmas da câmara de vácuo, acrescentando-se apenas o k , que é o coeficiente de resistência do ar, em kg/m, e depende da área da seção transversal da esfera (0,5 m²) e a variável $posicao0$ que define a posição inicial das esferas: 37 m acima do solo (Figura 19 – Variáveis fundamentais).

Figura 19 – Variáveis fundamentais

```
//Inserção da biblioteca numeric js e constantes
var gravidade = 9.81;
var massa = 1;
var massa2 = 1;
var velocidade1 = 0.0;
var velocidade2 = 0.0;
var posicao0 = 37;
var k = 0.0001;
var url = "https://fisicagames.com.br/numeric-1.2.6.js";
```

Fonte: Autores.

O valor de k definido refere-se a um valor aproximado de 0 para simular inicialmente o vácuo, sendo que valores maiores de resistência podem vir a ser inseridos diretamente no código ou através da barra de rolagem acrescentada com este fim à GUI, conforme a Figura 20. A barra permite valores desde $\cong 0$ até 1,43 kg/m, o qual se trata de um valor grande, cuja substância deve ter alta densidade. Valores ainda maiores podem ser testados diretamente no código.

Figura 20 – GUI 1 e 2



Fonte: Autores.

Para o cálculo da velocidade das esferas, foi utilizada, neste caso, a função horária da velocidade (4), considerando a velocidade inicial como 0, pois ambas as esferas partem do repouso. Para o cálculo das posições, foi utilizada a função horária das posições (5), considerando a posição inicial como 37 m. As funções foram aplicadas a cada esfera, como ilustra a Figura 21. Para a inserção destas funções, foram utilizados recursos específicos da biblioteca *Numeric JS*.

Figura 21 – Funções horárias de velocidade e deslocamento com resistência do ar

```
//Cálculo de velocidade e posição - esfera 1:
scene.registerBeforeRender(function () {
    tempo += 1/60;
    if(executarGatilho == true) {
        tempoGUI.text = "Tempo de queda: " + tempo.toFixed(1)+ " s";
        sphere.position.y = (-massa/k)*Math.log(Math.cosh(tempo*Math.sqrt(k*gravidade/massa)))+posicao0;
        sphere.velocidade1 = Math.sqrt(massa*gravidade/k)*Math.tanh(Math.sqrt(k*gravidade/massa)*tempo);
        velocidadeGUI.text = "Velocidade da esfera 1: " + sphere.velocidade1.toFixed(1) + " m/s";
//Cálculo de velocidade e posição - esfera 2:
        tempoGUI2.text = "Tempo de queda 2: " + tempo.toFixed(1) + " s";
        sphere2.position.y = (-massa2/k)*Math.log(Math.cosh(tempo*Math.sqrt(k*gravidade/massa2)))+posicao0;
        sphere2.velocidade2 = Math.sqrt(massa2*gravidade/k)*Math.tanh(Math.sqrt(k*gravidade/massa2)*tempo);
        velocidadeGUI2.text = "Velocidade da esfera 2: " + sphere2.velocidade2.toFixed(1) + " m/s";
        camera.position.z -= (sphere.position.y-sphere2.position.y)*0.01
        camera.setTarget(sphere.position);
    }
}
```

Fonte: Autores.

O *Playground* da câmara de queda com resistência do ar ficou como ilustra a figura a seguir.

Figura 22 – Micromundo da câmara de queda com resistência: Playground24



Fonte: Autores.

Como dito anteriormente, as dimensões de área da seção transversal das esferas foram fixadas em $0,5 \text{ m}^2$, para ambas. Portanto, embora o ato de deslizar a barra de massa das esferas altere visualmente suas dimensões, essa variação não é aplicada ao cálculo, servindo apenas para efeito visual durante o aumento da massa. Tal configuração teve por intuito evitar o emprego de cálculos ainda mais complexos nesse micromundo, aspecto que não limita a criação de outro *playground* para explorar especificamente o efeito da alteração das dimensões do objeto no movimento de queda com resistência.

Quanto à ocorrência de velocidade terminal, a mesma pode ser visualizada no caso de baixos valores de densidade de fluido. Logo, é possível notar o momento em que as esferas atingem a velocidade terminal na simulação através da observação dos marcadores de velocidade de cada uma delas na GUI. Também é possível, em propostas futuras, acrescentar meios para uma observação mais detalhada dos casos em que a força peso é anulada pela resistência do fluido ao movimento.

RESULTADOS E DISCUSSÃO

Diante do que foi analisado, notou-se que o *Babylon.js* oferece todo o aparato fundamental ao desenvolvimento de um ambiente virtual para aprendizagem construcionista, pois:

- Os objetos e funções modelam as propriedades físicas do mundo real, podendo também agir de outras maneiras, seguindo uma física idealizada pelo usuário (basta não ativar a *engine* de física), ou simulando situações que normalmente não se experencia no cotidiano, permitindo assim várias representações das propriedades subjacentes à realidade;
- Permite a compreensão a e aplicação direta de conceitos da Física;
- O *Playground* faz um papel análogo ou superior à linguagem LOGO ao possibilitar que comandos em *JavaScript* do editor sejam imediatamente executados e visualizados na tela de renderização;
- A primitiva tartaruga LOGO é substituída pelos objetos variados e dinâmicos, os quais podem inclusive ser inventados pelo aluno e tornam-se o caminho para ele lidar com conceitos mais complexos;

- Permite a aprendizagem de forma exploratória, permitindo ao aluno testar suas hipóteses enquanto constrói seu conhecimento;
- Favorece o conceito de *debugging*, pois os erros (no código ou conceito) tornam-se uma rica fonte de aprendizado, afinal um erro no editor se transforma na busca por uma nova solução, por novas estratégias.

As funções do MRUV utilizadas na primeira parte do micromundo, embora muito recorrentes no ensino de Cinemática, retratam uma situação que exige o pré-requisito do conceito de vácuo. Esse conceito é possível de ser observado através de aparatos experimentais, embora nem sempre esteja acessível a todos os alunos devido à limitação de recursos nas escolas.

Nesse sentido, o micromundo desenvolvido fornece tanto a simulação de uma câmara de vácuo com a queda livre como também uma segunda câmara que simula a queda das esferas com resistência do ar, permitindo não só a simples visualização de um movimento de queda livre em condições ideais e com resistência, como possibilita também a interação direta com o micromundo. Como a *engine* não foi habilitada neste micromundo, propositalmente, ele não simulará restritamente as leis físicas conhecidas, abrindo caminho para a exploração de alternativas pelos estudantes, independentemente de qualquer *engine* padrão.

Nos primeiros contatos com o micromundo, o estudante provavelmente limitar-se-ia a alterar as variáveis de massa das esferas na GUI, e através disso perceberia que de fato no MRUV os valores de massa não influenciam no movimento de queda. Já na segunda câmara, há a possibilidade de variar, além da massa, valores de k diferentes através da GUI.

Uma observação rápida do código permite perceber que a *engine* de física não foi previamente ativada, portanto, constantes como a aceleração da gravidade podem ser facilmente alteradas. Assim, o aluno poderia analisar o movimento de queda livre e com resistência em outros planetas, por exemplo.

As funções que descrevem o movimento das esferas também podem ser alteradas, não limitando-se somente às funções do MRUV e de queda com resistência, pois pode-se sugerir inclusive que o próprio estudante desenvolva funções alternativas para o movimento antes de compreender as que estão incluídas no micromundo.

Quanto às funções (4) e (5), devido à sua maior complexidade, pode-se utilizá-las normalmente, porém apenas com o intuito de compreender quais os fatores que influenciam no movimento de queda com resistência e como influenciam, não se fazendo necessária, portanto, a dedução a rigor dessas funções.

No caso das supracitadas funções, torna-se indispensável o acompanhamento de um professor para mediar o processo de ensino aprendizagem, auxiliando o aluno na compreensão de cada variável presente, qual é o seu significado e as suas influências no micromundo.

Também é importante destacar que o micromundo implementado ainda apresenta partes incompletas, como a descontinuação da queda de uma das esferas quando outra atinge o solo antes, além do assincronismo da câmera em relação ao movimento das esferas. Porém, a proposta do trabalho e do próprio referencial teórico adotado não é entregar um micromundo que dispense

modificações e intervenções, antes o contrário. Trata-se de implementar uma estrutura inicial para que, a partir dela, o estudante possa construir, testar e analisar suas hipóteses e construir novos conhecimentos.

Quanto à programação em *JavaScript* do micromundo, como já discutido, trata-se de uma linguagem de baixa dificuldade para compreensão. Além disso, dentro da proposta construcionista para micromundos, a programação é um aspecto de suma importância, pois em essência a programação viabiliza o desenvolvimento de um produto de interesse do usuário pela da construção detalhada do mesmo através do código. Ainda sobre a importância da programação no micromundo, Hoyles et al. (2002) argumenta:

Programação é a ferramenta prototípica para a visão construcionista, e um micromundo sem programação corre o risco de evitar justamente aquilo que dá ao micromundo seu poder (HOYLES et al., 2002, p. 29).

Ademais, a atividade de programar ou modificar a programação do micromundo coloca em prática a premissa construcionista de que o aluno se coloca por inteiro na atividade, dizendo ao computador o que ele deve fazer e experienciando todas as possibilidades possíveis dentro do problema abordado. Para isso, a utilização do micromundo não deve sobrepujar teorias equivocadas do senso comum trazidas pelos estudantes, mas sim permitir que elas sejam exploradas também (ALMEIDA, 1999).

CONCLUSÃO

Considerando os fundamentos construcionistas, o micromundo implementado através da plataforma do *Babylon.js* neste trabalho não se trata, de forma alguma, de um produto final. Antes traz uma estrutura inicial para que, a partir dela o aluno ou usuário teste suas próprias conjecturas, modifique-as, entenda como as leis do movimento funcionam e os fatores que as influenciam e dessa maneira, construa seu próprio conhecimento, agindo sobre o micromundo inicial e até criando novos micromundos.

Este trabalho também não buscou analisar a eficácia do micromundo para o aprendizado de Física, aplicando-o a estudantes, embora este possa ser o objetivo de trabalhos futuros.

A partir da implementação do micromundo WebGL das câmaras de queda livre e com resistência no *Babylon.js*, pôde-se concluir que a proposta de Papert de um micromundo físico que permitisse a interação e manipulação das condições físicas contraintuitivas foi revolucionária, embora a tecnologia da época ainda não permitisse o desenvolvimento de ferramentas apropriadas para esse fim. Tentativas posteriores de implementação dos micromundos físicos, como a de Santos (2011), mostraram-se falhas em aspectos fundamentais, como o fato de que o *Second Life* não permite a modificação de variáveis e das próprias leis do movimento, fazendo com que a proposta não atendesse às características de um micromundo físico a rigor.

Nesse sentido, a plataforma do *Babylon.js* supre tais insuficiências e ainda traz possibilidades ampliadas no quesito da Física, pois além de permitir o uso de uma *engine* de física padrão, possibilita também o desenvolvimento de leis físicas subjacentes. O *Playground do Babylon.js* também permite que os usuários possam

criar versões modificadas das simulações em uma abordagem construcionista, além de possibilitar o processo de *debugging*. Assim, a plataforma de simulação aqui apresentada possui os requisitos necessários para a implementação de micromundos físicos, como idealizados por Papert (1985). Por permitir a programação com uma abordagem construcionista, pela qualidade na renderização de gráficos, a programação de fácil aprendizagem, a acessibilidade da tecnologia *WebGL* e a documentação explicativa do *Babylon.js*, a plataforma mostrou-se promissora para o desenvolvimento de micromundos físicos que contemplem diferentes conteúdos da Física, ficando esse desafio para trabalhos futuros, inclusive para a criação de uma galeria de micromundos físicos de acesso público e de código aberto.

IMPLEMENTATION OF THE PHYSICS MICROWORLDS PROPOSED BY SEYMOUR PAPERT FOR TEACHING PHYSICS

ABSTRACT

The objective of this work is to implement a physics microworld developed with WebGL technology for teaching Physics. Supported by the constructionist theory of Seymour Papert, this work proposes the implementation of a microworld as proposed by Papert, that is, so that the student defines the conditions to be analyzed in the physics microworld, changing the variables, observing and understanding the results. With this proposal, the developed microworld is expected to be consistent with Papert's initial proposal, which must allow the understanding of physics contents that require a higher level of abstraction by means of interactions with the simulated environment and contents that are often too abstract to be spontaneously related to everyday situations.

KEYWORDS: Physics microworld. Teaching physics. Constructionism.

NOTAS

1 Linguagem de programação desenvolvida por Seymour Papert em 1967, no *Massachusetts Institute of Technology* (MIT), em Boston, EUA, especialmente voltada à educação.

2 Scratch é um *software* baseado em blocos lógicos idealizado por Mitchel Resnick e desenvolvido em 2007 no Instituto de Tecnologia de Massachusetts (MIT).

3 <https://www.babylonjs.com/>

4 MIT: Seymour A. Papert Biography. Disponível em: <https://www.media.mit.edu/people/papert/overview/>. Acesso em: 17 out. 2018.

5 Disponível em: <http://blog.core-ed.org/blog/2014/08/the-logo-turtle-and-seymour-paperts-mindstorms-thirty-four-years-on.html>. Acesso em: 12 jun. 2018.

6 Exemplo de funcionamento das mensagens mostradas quando o comando dado ainda não foi aprendido pela máquina na linguagem LOGO.

7 SuperLogo. Version 3.0. Universidade Estadual de Campinas, 2004.

8 A matemática é definida como a arte do discente de conhecer e aprender algo. Opõe-se à didática.

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

10 DEDE, Chris. The future of multimedia: Bridging to virtual worlds. **Educational Technology**, v. 32, n. 5, p. 54-60, 1992.

11 “Um motor, ao colocar um corpo em movimento, infunde nele certo *impetus*, um poder capaz de mover o corpo na direção em que o motor o colocou em movimento, seja para cima, para baixo, para o lado ou em círculo. [...] é por este *impetus* que a pedra é movida após o lançador ter deixado de movê-la” (Buridan, 1360), citado por Crombie (1957, p. 251).

BURIDAN, Jean, **Quæstiones super Octo Libri Physicorum**, Book 8, Question 12, 1360.

12 PhET. Physics Education Technology. Disponível em: https://phet.colorado.edu/pt_BR/. Acesso em: 03 out. 2018.

13 MIT. Logo History. Disponível em: http://el.media.mit.edu/logo-foundation/what_is_logo/history.html. Acesso em: 5 out. 2018.

14 MIT. Scratch: About. Disponível em: <https://scratch.mit.edu/about>. Acesso em: 5 out. 2018.

15 THE KHRONOS GROUP INC. WebGL Overview. Disponível em: <https://www.khronos.org/webgl/>. Acesso em: 5 out. 2018.

16 *JavaScript* é uma linguagem de programação completa que, quando aplicada a um documento HTML, pode fornecer interatividade dinâmica em sites (MDN, 2005).

17 Tecnologia gráfica tridimensional (3D) suportada pela World Wide Web (WWW).

18 <https://playground.babylonjs.com/>

19 Framework: conjunto de classes orientadas a objetos que são integradas para executar conjuntos definidos de comportamentos virtuais. Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/deffw.htm>. Acesso em: 8 out. 2018.

20 https://doc.babylonjs.com/how_to/using_the_physics_engine

21 Ou MRUV: Movimento Retilíneo Uniformemente Variado.

22 <http://www.numericjs.com/>

23 <https://www.babylonjs-playground.com/#WAK0C0#6>

24 <https://www.babylonjs-playground.com/#LWBKWT>

REFERÊNCIAS

ALMEIDA, M. E. B. DE. Informática e formação de professores. **Coleção Informática Aplicada na Educação**. São Paulo: MEC/SEED/PROInfo, 1999.

BESTEBREURTJE, T. **Second Life: A model for applications - Generic Web support for serious games in Second Life and beyond**. Amsterdam, 2007.

BETTENCOURT, Teresa; ABADE, Augusto. Mundos Virtuais de Aprendizagem e de Ensino-uma caracterização inicial. **IE Comunicaciones: Revista Iberoamericana de Informática Educativa**, n. 7, p. 3-16, 2008.

BIELSCHOWSKY, Carlos Eduardo. Tecnologia da informação e comunicação das escolas públicas brasileiras: o programa Proinfo Integrado. **Revista e-curriculum**, v. 5, n. 1, 2009.

CROMBIE, Alistair C., **Augustine to Galileo: The History of Science a.d. 400-1650**. William Heinemann, Ltd., London, 1957.

HOYLES, C.; NOSS, R.; ADAMSON, R. Rethinking the Microworld Idea. **Journal of Educational Computing Research**, v. 27, n. 1-2, p. 29-53, 2002.

MALTEMPI, M. V. Novas tecnologias e construção de conhecimento: reflexões e perspectivas. **Congresso Ibero-Americano de Educação Matemática**, 2005.

MDN WEB DOCS. **JavaScript basics**. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript>. Acesso em: 6 out. 2018.

MOREAU-MATHIS, Julien. **Babylon.js Essentials**. Packt Publishing Ltd, 2016.

MOURA, M. L. S. DE; ACUNZO, I. M. M. Raízes do Logo: uma análise de seus fundamentos psicológicos. **Arquivos Brasileiros de Psicologia**, v. 37, n. 4, p. 27–33, 1985.

PALANGANA, I. C. **Desenvolvimento e Aprendizagem em Piaget e Vygotsky: A relevância do social**. Summus Editorial, 2015.

PAPERT, S. **A família em rede: ultrapassando a barreira digital entre gerações**. Lisboa: Relógio d'água, 1997.

PAPERT, S. **Constructionism: A new opportunity for elementary science education**. Massachusetts Institute of Technology: Epistemology and Learning Group, 1986.

PAPERT, S. **Logo: computadores e educação**. São Paulo: Brasiliense, 1985.

PAPERT, S. **A máquina das crianças: repensando a escola na era da informática**. 1994.

PEREIRA, P. H. C.; PEREIRA, F. F. M. O Uso Da Automação E Robótica Como Forma Lúdica De Consolidação Do Conhecimento No Curso De Engenharia. **Centro Federal de Educação Tecnológica de Minas Gerais – CEFET/MG**, 2006.

PIAGET, Jean. **The Grasp of Consciousness: Action and Concept in the Young Child**. Cambridge, Mass: Harvard University, 1976.

RIBEIRO, R. J. et al. Teorias de aprendizagem em jogos digitais educacionais: um panorama brasileiro. **RENOTE**, v. 13, n. 1, 2015.

SANTOS, R. DOS. O *Second Life* como plataforma para micromundos físicos para o ensino de Física. **RENOTE**, v. 9, n. 1, 2011.

SANTOS, R. P. Manipulando objetos newtonianos: aprendendo física no *Second Life* com TATI. XIV Encontro de Pesquisa em Ensino de Física. **Anais...**, Maresias, 2012.

SANTOS, R. P. Virtual, Real ou Surreal? A Física do *Second Life*. **RENOTE**, v. 6, n. 2, 2008.

SANTOS, R. P.; ZORZAL, E. R.; CARDOSO, A. Novas Realidades e Tecnologias no Ensino de Física. **Ciências Humanas e Sociais em Revista**, v. 32, n. 1, p. 101–120, 2010.

STAMOULIAS, ANDREAS. **Physics Support in Web3d Environments**. 2014. Tese De Doutorado. Technological Educational Institute of Crete.

STROM, C. **3D Game Programming for Kids: Create Interactive Worlds with JavaScript**. Pragmatic Bookshelf, 2018.

SWAINE, M. R.; HEMMENDINGER, D. History of Computing. **Encyclopædia Britannica**, 2018.

VALENTE, J. A. Diferentes usos do computador na educação. **Computadores e conhecimento: repensando a educação**. Campinas: NIED-Unicamp, 1993.

VALENTE, J. A. et al. **O computador na sociedade do conhecimento**. 6. ed. Campinas, 1999.

WALKER, J. Through the Looking Glass. In: B. Laurel. **The art of computer-human interface design**. Menlo Park CA: Addison-Wesley, 1990.

WAZLAWICK, Raul Sidnei. **História da Computação**. Elsevier Brasil, 2017.

ZORZAL, Ezequiel Roberto et al. Ambientes Educacionais Colaborativos com Realidade Aumentada. **RENOTE**, v. 6, n. 1, 2008.

Recebido: 24 jan. 2019.

Aprovado: 11 fev. 2020.

DOI: 10.3895/rbect.v13n2.9420

Como citar: FLOR, F. A.; RIBEIRO, R. J. Implementação dos micromundos físicos propostos por Seymour Papert para o ensino de física. **Revista Brasileira de Ensino de Ciência e Tecnologia**, Ponta Grossa, v.13, n. 2, p. 160-197, mai./ago. 2020. Disponível em: <<https://periodicos.utfpr.edu.br/rbect/article/view/9420>>. Acesso em: XXX.

Correspondência: Fabiana Aparecida Flor - fabiana.flor1@outlook.com

Direito autoral: Este artigo está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.

