|  |  |
| --- | --- |
|  | [FURB – Universidade Regional de Blumenau](http://www.furb.br/)  [DSC – Departamento de Sistemas e Computação](http://www.inf.furb.br/dsc/)  DE - Departamento de Educação  Disciplina: Sistemas Multimídia - prof. Dalton Solano dos Reis  Disciplina: Tecnologia Educacional e Aprendizagem - Prof.º Maurício Capobianco Lopes |

**Projeto Interdisciplinar de construção de material educacional**

**\_\_\_\_\_ Etapa 1: definição do assunto**

**1.1 Equipe:**

Computação Alexandre Vicenzi (líder) / Janaína Carraro Mendonça

Pedagogia Jessiel Odilon Junglos (líder) / Jessica Araujo / Diana Meyer

**1.2 Título**: Marionete Wireless: Controle de uma marionete virtual com o leap motion

**1.3 Problema/Dúvida:**

Será desenvolvido um aplicativo que desafie as crianças a explorar a sua imaginação, conciliando criatividade, coordenação motora e tecnologia. Se observa no campo educativo a ausência de proposições que estimulem a coordenação motora, a noção de lateralidade e a capacidade de criação, utilizando-se do faz de conta e do lúdico para crianças pequenas. Assim surgiram alguns questionamentos: como a tecnologia poderia auxiliar nesse campo, sem ser desenvolvido um produto obsoleto? As crianças conseguirão utilizar o Leap Motion?

**1.4 Objetivos:**

**Objetivo Geral:**

Desenvolver um aplicativo que explore o imaginário da criança, em que ela irá controlar uma marionete virtual através de seus movimentos das mãos captados pelo Leap Motion.

**Objetivos Específicos:**

1. ampliar o repertório cultural através da linguagem artística cultural;
2. estimular a coordenação motora;
3. desenvolver a noção espacial e lateralidade;
4. utilizar o Leap Motion como interface de entrada para uma aplicação.

**1.5 Público-alvo:** Crianças 04 a 10 anos e professores(as) interessados(as).

**1.6 Relevância:**

O aplicativo visa ajudar o desenvolvimento de alguns aspectos importantes durante o processo de crescimento de uma criança, dentre eles: noção espacial; lateralidade; coordenação motora fina; ludicidade.

Como o Leap Motion é um dispositivo recente no mercado, ainda existem poucas aplicações disponíveis para que possamos utilizar alguma como base para o desenvolvimento do nosso trabalho. Porém, está disponível uma boa documentação, com exemplos em várias linguagens de programação, além do suporte deles ser eficiente, com respostas rápidas e coerentes às nossas necessidades. É imprescindível que seja estudado o SDK do dispositivo em questão, além da escolha de um mecanismo para trabalhar a parte da interface gráfica, onde podemos optar pelo uso de uma engine de renderização (Unity 3D, Unreal, MonkeyEngine).

**1.7 Equipamento escolhido**: Leap Motion.

**\_\_\_\_\_ Etapa 2: Trabalhos correlatos e o Proposto**

|  |  |
| --- | --- |
|  |  |

2.1 TRABALHOS CORRELATOS

A seguir estão relacionados três trabalhos correlatos ao proposto. O item 2.1.1 descreve o Marionete Zoo, aplicativo do Leap Motion que utiliza o movimento das mãos para movimentar marionetes virtuais. O item 2.1.2 apresenta o Puppeteer, um jogo para o Play Station 3 onde o personagem principal é uma marionete. No item 2.1.3 é descrito o Form and Function 3D, uma aplicação para o Leap Motion que permite visualizar estruturas anatômicas.

2.1.1 Marionette Zoo

Aplicativo do Leap Motion que utiliza o movimento das mãos e dedos para movimentar marionetes virtuais (LEAP MOTION, 2014). É possível escolher até 02 personagens (animais) e diferentes cenários. Com os movimentos é possível promover a interação entre os bonecos. No momento em que as mãos são retiradas da área de interação, retorna para a tela inicial onde é possível escolher novos personagens e cenários. A figura 1 apresenta um exemplo de uso da aplicação. Demonstração disponível no link: <https://www.youtube.com/watch?v=-Vq-SPJXHU0>



Figura 1 - Exemplo de uso do Marionette Zoo

2.1.2 Puppeteer

O Puppeteer é um jogo desenvolvido pelo SCE JAPAN Studio exclusivamente para o PlayStation 3 (SONY, 2014). O ambiente é em um cenário mágico de marionetes, onde os jogadores percorrem um conto de fadas exuberante e sombrio, onde encontram diversas surpresas. O jogo apresenta cenários interativos e dinâmicos que mudam constantemente e levam o jogador a uma jornada de descobertas. Com narrativa teatral, apresenta plateia, luzes, ambientação, música e narrativa.

Pode ser jogado por um ou dois jogadores. O analógico da direita pode controlar o personagem que acompanha Kutaro (o personagem principal), que muda conforme a história progride. Também pode ser controlado através do PlayStation Move. Usando o analógico e o botão R2, é possível interagir com o fundo do cenário (PS3 BRASIL, 2013). A Figura 2 apresenta uma cena do jogo.



Figura 2 - cena do jogo Puppeteer.

Fonte: Sony (2014).

2.1.3 Form and Function 3D

O Form and Function 3D (POLLEY, 2014) é uma aplicação educacional para Leap Motion com o objetivo de desenvolver o senso espacial e o aprendizado da anatomia animal. Hoje é possível escolher entre o coração de três animais: do tubarão, da salamandra ou do gato. Através dos movimentos da mão, o usuário pode manipular o coração no espaço 3D, ver as informações de cada parte do órgão, além de poder escolher visualizar o fluxo sanguíneo. A Figura 3 apresenta um exemplo de uso do aplicativo.

Figura 3 - Movimento do coração e visualização do fluxo sanguíneo.



Fonte: Leap Motion (2014).

2.2 CENÁRIO DO TRABALHO PROPOSTO

O aplicativo consistirá em um jogo onde o participante poderá escolher um entre os cenários disponíveis e um personagem (marionete) para que, com os movimentos das mãos e dedos, captados pelo Leap Motion, possa interagir com o personagem e cenário para criar ou reproduzir uma história.

No Quadro 1 podem ser observadas as principais características dos trabalhos correlatos e sua comparação com o trabalho proposto.

Quadro 1 – proposto versos correlatos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Características | Proposto | Correlato 1 | Correlato 2 | | Correlato 3 |
|  |  | Marionette Zoo | Puppeteer | Form and Function 3D | |
| Marionete | sim | sim | sim | não | |
| Mais de um personagem | sim | sim | sim | não | |
| Leap Motion | sim | sim | não | sim | |
| Permite criar histórias | sim | sim | não | não | |

**Revisão bibliográfica**

3.1 CONCEITOS RELACIONADOS A EDUCAÇÃO E SISTEMAS MULTIMÍDIA

O Leap Motion é um dispositivo USB projetado para o rastreamento de ambas as mãos e todos os 10 dedos em tempo real. Para o reconhecimento das mãos, ele faz uso de duas câmeras IR monocromáticas e três LEDs infravermelho. O dispositivo consegue observar uma área hemisférica de aproximadamente 1 metro. As câmeras geram quase 300 frames por segundo de dados refletidos, já os LEDs geram um padrão de luz IR.

Todos os dados obtidos a partir do dispositivo são enviados ao computador para que sejam analisados e processados pelo Leap Motion Controller. Este software faz uso de matemática complexa para sintetizar em forma 3D as posições das mãos obtidas pelos frames 2D.

O Kinect oferece uma tecnologia semelhante, porém ele é capaz de rastrear o corpo todo em um espaço do tamanho de uma sala. Já o Leap Motion consegue rastrear todos os 10 dedos com uma precisão de 1/100 milímetros.

O *Marionete Wireless* é um aplicativo que se propõe a estimular a ludicidade, a coordenação motora fina e lateralidade de crianças de seis a dez anos. O aplicativo proporcionará ao operador ou operadora um maior conhecimento e estímulo do seu lado dominante (lateralidade). Exigirá uma coordenação motora desenvolvida e estimulará a ludicidade e imaginação na criação de histórias. Traremos a seguir a definição desses conceitos – ludicidade, coordenação motora fina e lateralidade – que embasaram a criação do aplicativo.

O termo ludicidade vem sendo bastante utilizado, principalmente no âmbito pedagógico. Remete a expressões como infância, brincadeira, criança, jogos, alegria e liberdade, porém, o lúdico já existe desde os tempos primitivos, acompanhando o homem nas mais variadas culturas. Inicialmente assimilado aos adultos em suas festas religiosas, nas sociedades medievais, onde não havia grande distinção entre adultos e crianças, a ludicidade se encontra em todas as etapas da vida, e segue em permanente transformação. Bruner afirma:

A atividade lúdica se caracteriza por uma articulação muito frouxa entre o fim e os meios. Isso não quer dizer que as crianças não tendam a um objetivo quando jogam e que não executem certos meios para atingí-lo, mas é freqüente que modifiquem seus objetivos durante o percurso para se adaptar a novos meios ou vice-versa [...], portanto, o jogo não é somente um meio de exploração, mas também de invenção (Bruner, apud Brougère, 1998, p.193).

Assim, entende-se por lúdico o divertimento que tem por objetivo o ensinamento, despertando o interesse e motivação da criança no processo ensino-aprendizagem favorecendo as interações, estimulando então reações sociais, cognitivas, afetivas, linguísticas e culturais.

Conceituar coordenação motora pode ser difícil, pois há muitos autores que a conceituam olhando-a no aspecto de sua área de estudo. Por esse motivo trazemos aqui um dos conceitos encontrados e que mais se aproxima do que queremos desenvolver com o aplicativo

KIPHARD (1976) *apud* Silva et al. (1995, p.20) entende por coordenação de movimentos a "interação harmoniosa e econômica de músculos, nervos e sentidos com o fim de produzir ações cinéticas precisas e equilibradas (motricidade voluntária) e reações rápidas e adaptadas a situações (motricidade reflexa).

Dessa forma podemos compreender que coordenação motora a ação desenvolvida através dos músculos, nervos e seus sentidos. É um movimento coordenado. Dessa forma o aplicativo desenvolvido tem por objetivo estimular o amadurecimento da coordenação motora da criança.

Já a lateralidade é um conceito bastante amplo, pois abrange a dominância que um lado do nosso corpo tem em comparação ao outro. Essa dominância se manifesta nos membros inferiores e superiores, nos olhos e ouvidos. Pode ser *Homogênea*, quando a criança é destra ou canhota do olho, pé, mãos e ouvido, *Cruzada*, quando a criança é destra de algum membro e canhota de outro e *Ambidestra* quando os dois lados são desenvolvidos. Faria (2001, p. 69) define a lateralidade como

a consciência interna que a criança tem dos lados direito e esquerdo de seu corpo, não se tratando de um conceito adquirido, mas de programas de atividades cuja finalidades é o desenvolvimento de acuidades sensoriais e habilidades motoras, já que a criança é exposta a uma grande quantidade de estímulos que aumentam esta consciência.

3.2 AMBIENTES UTILIZADOS NO DESENVOLVIMENTO

O desenvolvimento é separado em duas áreas, uma para obtenção e manipulação dos dados do dispositivo e outra para a renderização dos dados obtidos para o usuário. Toda esta manipulação será feita pelo Web Browser, sendo assim, é necessário o uso de bibliotecas JavaScript para o processamento dos dados.

Para o reconhecimento do Leap Motion e captura dos dados de entrada deste dispositivo será utilizado o LeapJS. O LeapJS é o SDK oficial para a linguagem JavaScript, e com ele é possível criar aplicações que interagem com os dados e funções para tratamento de dedos e mãos de forma fácil.

Para o desenvolvimento da interface gráfica será utilizado o Three.js (THREE JS, 2015), uma biblioteca em Javascript que utiliza o WebGL para renderização em 3D. Isto torna mais fácil o uso do WebGL, por exemplo, para desenhar um cubo em WebGL puro seriam necessárias muitas linhas de código, com o Three.js é possível renderizar um objeto equivalente com bem menos linhas.

Além das bibliotecas para desenvolvimento, é necessário a instalação do Leap Motion Control Center. Isto permite acessar o dispositivo e alterar algumas configurações básicas de funcionamento. Este aplicativo é necessário não só no desenvolvimento, mas também no uso da aplicação.

Como JavaScript é uma linguagem interpretada, não será feito o uso de nenhuma IDE para o desenvolvimento, apenas faz-se necessário o uso de um editor de texto, que no nosso caso será o Sublime Text. Também será feito o uso das ferramentas JSHint e JSLint, para análise do código gerado, encontrando possíveis problemas de programação, como sintaxe errada. Ambas ferramentas dependem do NodeJS para funcionamento.

3.3 REQUISITOS PRINCIPAIS DO PROBLEMA TRABALHADO

O aplicativo desenvolvido deverá:

1. permitir o uso das duas mãos para manipular duas marionetes ao mesmo tempo (Requisito Funcional - RF);
2. exibir um cenário de fundo (RF);
3. apresentar um menu com opções de configuração do jogo (RF);
4. permitir ao usuário escolher o cenário de fundo (RF);
5. permitir ao usuário escolher um personagem (RF);
6. utilizar o dispositivo Leap Motion como interface de entrada dos dados (Requisito Não Funcional - RNF);
7. ser desenvolvido para a plataforma Web (RNF);
8. ser compatível com os navegadores Chrome 40 e Firefox 36 (RNF);

3.4 ESPECIFICAÇÃO

A especificação do aplicativo foi feita utilizando a Unified Modeling Language (UML) através da ferramenta Dia, onde foi realizada a modelagem e diagrama de casos de uso apresentado na seção 3.4.2 e de classes na seção 3.4.3. A seção 3.4.1 apresenta um mapa mental construído com a ferramenta Mindomo, onde estão descritas as etapas do desenvolvimento deste projeto.

3.4.1 Etapas do desenvolvimento do projeto

A figura 4 apresenta um mapa mental onde estão representadas de forma sucinta as etapas do desenvolvimento do projeto.

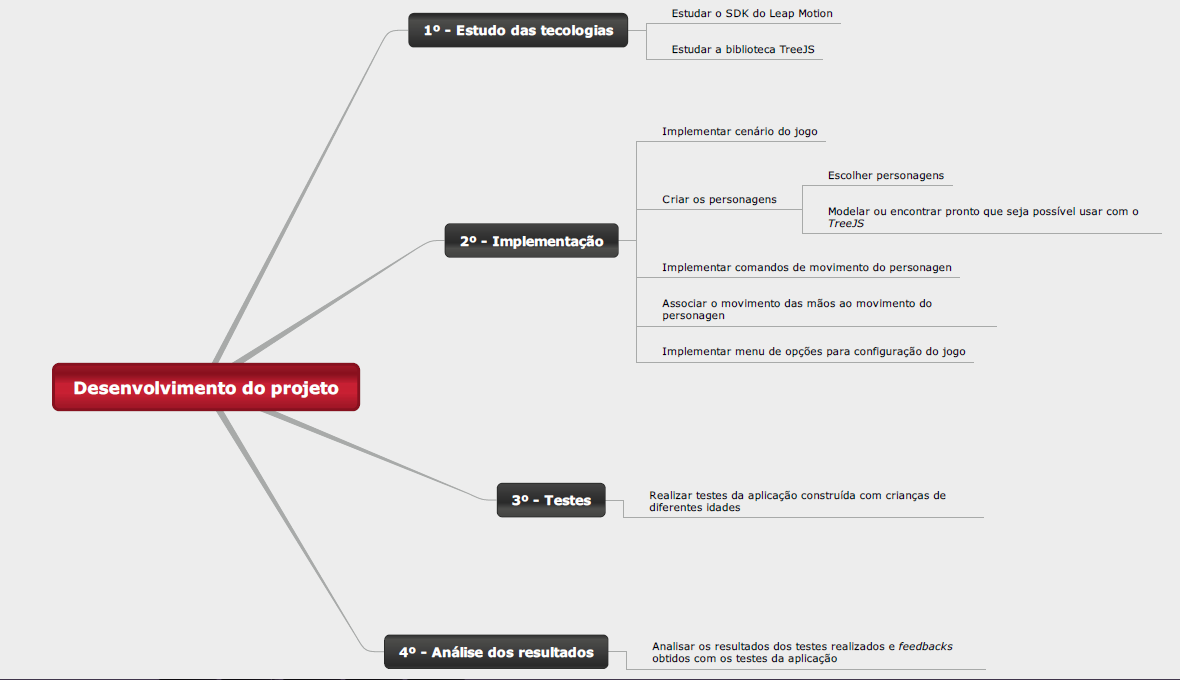
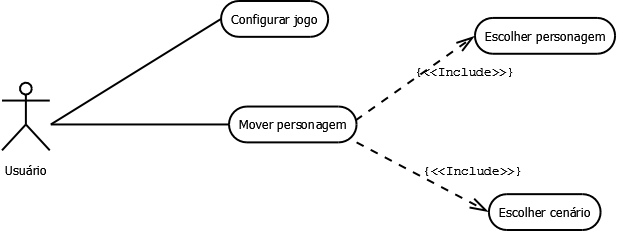


Figura 4 - Etapas do desenvolvimento do projeto.

3.4.2 Diagrama de Casos de Uso

A figura 5 apresenta o diagrama de casos de uso do aplicativo. Foi identificado apenas um usuário, que assume o controle sobre o aplicativo realizando ações como escolher o personagem e o cenário, configurar opções do jogo e mover o personagem da forma como desejar.

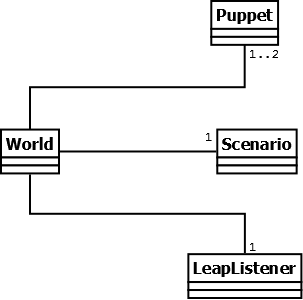
Figura 5 - Diagrama de Casos de uso



3.4.3 Diagrama de Classes

A figura 6 apresenta o diagrama de classes do aplicativo. Foram identificadas apenas 4 classes para o desenvolvimento do mesmo. A classe *Puppet*, que representa a marionete (personagem) do jogo. A classe *Scenario* que representa o cenário onde se passa o jogo. A classe *LeapListener*, responsável por “escutar” os eventos de movimentos das mãos e transmití-los aos objetos da classe *Puppet*. E por fim, a classe *World*, que gerencia os objetos das classes citadas anteriormente, sendo responsável por manter o ciclo de vida do jogo.

Figura 6 - Diagrama de classes.



**4. Desenvolvimento do Projeto**

Neste capítulo serão abordadas as etapas do desenvolvimento e resultados do projeto. Inicialmente, a seção 4.1 descreve os detalhes da implementação do aplicativo. Na seção 4.2 é apresentada a descrição do objeto de aprendizagem. Em seguida, a seção 4.3 apresenta os resultados e discussão do que foi desenvolvido. Na seção 4.4 demonstra as conclusões e por fim, a seção 4.5 cita possíveis extensões para o aplicativo desenvolvido.

4.1 IMPLEMENTAÇÃO

A implementação do projeto foi dividida em duas partes, uma responsável pela interface gráfica do aplicativo e outra encarregada por tratar os dados recebidos pelo Leap Motion. A seção 4.1.1 descreve os detalhes da construção da interface gráfica.

4.1.1 Interface gráfica

Foi utilizada a biblioteca Three.js para a construção da cena do aplicativo, o objeto MarionetteDrawer.js é o responsável pela renderização. O Quadro 2 apresenta a função *setup*, responsável por criar e configurar a cena.

Quadro 2 - Código que cria e configura a cena do aplicativo.

|  |
| --- |
| setup.png |

Nas primeiras linhas, é criada a cena e o renderizador WebGL. A linha 65 adiciona um pondto de luz à cena. A linha 67 cria a câmera e define suas propriedades básicas.

A seguir é apresentado o Quadro 3, que demonstra a função que cria e adiciona um cavalo à cena. O modelo do cavalo está em um arquivo com formato JSON, portanto é carregado utilizando a classe JSONLoader do Three.js. Na linha 200 é adicionado um som de cavalo para ser executado quando o cavalo aparecer na tela.

Quadro 3 - Criação do cavalo

|  |
| --- |
| addHorse.PNG |

O Quadro 4 apresenta as funções encarregadas por mover e rotacionar os personagens.

Quadro 4 - Código que move e rotaciona um personagem

|  |
| --- |
| move_rotate.PNG |

A seguir, é demonstrado no Quadro 5 o código responsável por adicionar o cenário de fundo da tela do aplicativo.

Quadro 5 - Função que cria o cenário de fundo

|  |
| --- |
| setBackground.PNG |

Por fim, o Quadro 6 apresenta a função *render*, que é invocada a partir do *loop* de renderização da interface.

Quadro 6 - Código que redesenha a tela

|  |
| --- |
| render.PNG |

4.2 DESCRIÇÃO DO OBJETO DE APRENDIZAGEM

|  |  |  |
| --- | --- | --- |
| **Geral** | **Título** | Marionete Wireless |
| **Idioma** | Português (BR) |
| **Descrição** | Aplicativo voltado a crianças e educadores com a proposta de estimular a imaginação e coordenação motora, se utilizando da tecnologia no movimento de marionetes através dos movimentos captados pelo Leap Motion. |
| **Palavras chave** | Marionete. Leap Motion. Tecnologia Educacional |
| **Cobertura** | Educação Básica (Infantil e Fundamental I) Podendo ser utilizada em diversas áreas, dependendo do contexto dado pelo professor. |

|  |  |  |
| --- | --- | --- |
| **Dados Técnicos** | **Formato** | Página html |
| **Tamanho** | - |
| **Localização** | https://github.com/alexandrevicenzi/marionette |
| **Requisitos** | Aparelho Leap Motion |
| **Observações para instalação** | Primeiramente é necessário conectar o dispositivo Leap Motion ao computador e em seguida instalar o Leap Motion Controller, se ainda não estiver instalado.  Em seguida basta acessar o link <http://alexandrevicenzi.github.io/marionette/dist/index.html> e usar o aplicativo. |
| **Outros requisitos para execução** | Conexão com internet |
| **Duração** | - |

|  |  |  |
| --- | --- | --- |
| **Aspectos Educacionais** | **Tipo de Interatividade** | O usuário dá movimento e significado aos personagens do aplicativo. Interagindo de forma intensa com o aplicativo. |
| **Tipo de Recurso de Aprendizagem** | Simulação |
| **Nível de Interatividade** | Alta |
| **Contexto de Aprendizagem** | O aplicativo serve como um estimulador de criatividade e imaginação para o sujeito que o manipula, além de contribuir para o desenvolvimento e aprimoramento da coordenação motora fina e lateralidade. |
| **Tempo aproximado de uso para aprendizagem** | Variável para cada criança. |

4.3 RESULTADOS E DISCUSSÃO

4.3.1 Avaliação do aplicativo

|  |  |  |  |
| --- | --- | --- | --- |
| **AVALIAÇÃO DO APLICATIVO MARIONETE WIRELESS** | | | |
| CRITÉRIOS  ITENS | Não Contempla | Contempla Parcialmente | Contempla Totalmente |
| Marionete |  | X |  |
| Variedade de personagens |  |  | X |
| Interação com o usuário |  |  | X |
| Atratividade para o usuário |  |  | X |
| Diferentes cenários | X |  |  |
| Sons |  | X |  |
| Página inicial interativa que possibilite a escolha de cenário e personagem | X |  |  |
| Interação de duas mãos simultaneamente |  |  | X |
| Compatibilidade com a realidade da escola (financeira) |  |  | X |
| Aplicabilidade em vários contextos |  |  | X |
| Possibilidade de adicionar cenários e personagens pelos usuários | X |  |  |

4.3.2 Utilização do projeto

Um vídeo foi desenvolvido para demonstrar a utilização do aplicativo, encontra-se disponível no youtube: [Leap Motion - Marionete Wireless](https://www.youtube.com/watch?v=s4VwsRK7D2c)

4.4 CONCLUSÕES

A proposta de provocar a imaginação, brincando com o movimento das mãos capturados pelo aparelho demonstrados em tela na manipulação de marionetes, foi atingida conforme o esperado. Experimentado pela primeira vez por uma criança em sala, no dia da apresentação da primeira versão do aplicativo, o Marionete Wireless surpreendeu a equipe desenvolvedora, considerando imprevistos e dificuldades em seu desenvolvimento. A criança voluntária no experimento conseguiu movimentar os personagens abrindo um sorriso ao ver que estes acompanhavam os movimentos das suas mãos, além disso, a criança demonstrou interesse ao perceber que suas mãos passaram “para o mundo virtual” vendo seus comandos na tela do computador.

4.5 EXTENSÕES

Como extensão par o aplicativo, propõe-se:

a) Adicionar objetos gráficos onde seja possível controlar o movimento das articulações;

b) Permitir ao usuário escolher o cenário e o personagem;

c) Permitir configurar sons;

d) Permitir *upload* do cenário e personagem.

**REFERÊNCIAS BIBLIOGRÁFICAS**

FARIA, Alcídia Magalhães. **Lateralidade: implicações no desenvolvimento infantil.** Rio de Janeiro: Sprint, 2001. 117 p.

LEAP MOTION. **Form and Function 3D**. [S.l.], 2015. Disponível em: <<https://apps.leapmotion.com/apps/form-and-function-3d/windows>>. Acesso em: 08 abr. 2015.

\_\_\_\_\_ **Marionette Zoo**.2014. Disponível em: <https://apps.leapmotion.com/apps/marionette-zoo/windows h>. Acesso em: 08 abr. 2015.

POLLEY, Brendan. **Research**. 2014. Disponível em: <http://brendanpolley.com/>. Acesso em: 08 abr. 2015.

PS3 BRASIL. **Análise Puppeteer**. 2013. Disponível em: <<http://www.ps3brasil.com/analise/322/>>. Acesso em: 12 abr. 2015.

SONY. **Puppeteer**. 2014. Disponível em: <<https://www.playstation.com/pt-br/games/puppeteer-ps3/>>. Acesso em: 12 abr. 2015.

SANTOS, Élia Amaral do Carmo. O lúdico no processo ensino-aprendizagem. Disponível em: <<http://need.unemat.br/4_forum/artigos/elia.pdf>>. Acesso em: 29/04/2015

SILVA, Renato de Oliveira e GIANNICHI, Ronaldo Sergio. **Coordenação motora: uma revisão de literatura.** Revista Mineira de Educação Física. UFV - Viçosa, 3(2): 17-4 1, 1995

THREE JS. **What is Three.js?**. [S.l.], 2015. Disponível em: <<http://threejs.org/docs/index.html#Manual/Introduction/Creating_a_scene>>. Acesso em: 22/04/2015.