UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LEONARDO CORRÁ PIRES TRONE RIBEIRO

RECURSOS EDUCACIONAIS PARA O ENSINO INTRODUTÓRIO À COMPUTAÇÃO USANDO IMPRESSORA 3D E GAMIFICAÇÃO

MONOGRAFIA

CAMPO MOURÃO
2019

LEONARDO CORRÁ PIRES TRONE RIBEIRO

RECURSOS EDUCACIONAIS PARA O ENSINO INTRODUTÓRIO À COMPUTAÇÃO USANDO IMPRESSORA 3D E GAMIFICAÇÃO

Trabalho de Conclusão de Curso de graduação apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Bacharelado em Ciência da Computação do Departamento Acadêmico de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Marco Aurélio Graciotto Silva

CAMPO MOURÃO
2019

ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

Às **15:50** do dia **27 de novembro de 2019** foi realizada na sala **B107** da UTFPR-CM a sessão pública da defesa do Trabalho de Conclusão do Curso de Bacharelado em Ciência da Computação do(a) acadêmico(a) **Leonardo Corrá Pires Trone Ribeiro** com o título **Recursos educacionais para o Ensino Introdutório à Computação usando Impressora 3D e Gamificação**. Estavam presentes, além do(a) acadêmico(a), os membros da banca examinadora composta por: **Prof. Dr. Marco Aurélio Graciotto Silva** (orientador(a)), **Prof. Ms. Narci Nogueira da Silva e Prof. Ms. Paulo Cesar Gonçalves**. Inicialmente, o(a) acadêmico(a) fez a apresentação do seu trabalho, sendo, em seguida, arguido(a) pela banca examinadora. Após as arguições, sem a presença do(a) acadêmico(a), a banca examinadora o(a) considerou _____ na disciplina de Trabalho de Conclusão de Curso 2 e atribuiu, em consenso, a nota _____ (_____). Esse resultado foi comunicado ao (à) acadêmico(a) e aos presentes na sessão pública. A banca examinadora também comunicou ao (à) acadêmico(a) que este resultado fica condicionado à entrega da versão final dentro dos padrões e da documentação exigida pela UTFPR ao professor responsável do TCC no prazo de **onze dias**. Em seguida foi encerrada a sessão e, para constar, foi lavrada a presente Ata que segue assinada pelos membros da banca examinadora, após lida e considerada conforme.

Observações:

Campo Mourão, **27 de novembro de 2019**

Prof. Ms. Narci Nogueira da Silva
Membro 1

Prof. Ms. Paulo Cesar Gonçalves
Membro 2

Prof. Dr. Marco Aurélio Graciotto Silva
Orientador

A ata de defesa assinada encontra-se na coordenação do curso.

Agradecimentos

É chegado ao fim um ciclo da minha vida. Sendo assim, dedico este trabalho a todos que fizeram parte desta etapa. À minha mãe e meu pai que me apoiaram e acompanharam durante a realização da graduação. Aos familiares que sempre se preocuparam comigo. Aos amigos que fiz e que compartilharam dos momentos bons e ruins dessa jornada. Ao meu orientador Marco Aurélio Graciotto Silva por todo apoio e paciência ao longo da elaboração do meu projeto final.

Resumo

Corrá, Leonardo. Recursos educacionais para o Ensino Introdutório à Computação usando Impressora 3D e Gamificação. 2019. 45. f. Monografia (Curso de Bacharelado em Ciência da Computação), Universidade Tecnológica Federal do Paraná. Campo Mourão, 2019.

Contexto: Estudos da literatura apontam uma alta evasão em cursos superiores da área de Computação. Dada a importância deste setor para o desenvolvimento tecnológico do país, é de suma importância o desenvolvimento de métodos de ensino que fomentem o interesse de alunos do ensino médio que possivelmente ingressarão na área e, consequentemente, que reduzam a evasão em cursos de Computação. Sabe-se que novas tecnologias costumam atrair o interesse das pessoas. A impressão 3D é um dos exemplos de tecnologia em maior evidência atualmente. Além disso, conceitos de jogos também auxiliam atrair o interesse cognitivo das pessoas no quesito de ensino. Neste contexto, este trabalho propõe uma abordagem que une conceitos de impressão 3D e gamificação a fim de ampliar a atratividade de cursos da área de computação.
Objetivo: O objetivo deste trabalho foi a geração de um conjunto de recursos educacionais para ensinar o básico de programação, de modo a introduzir a área da computação.
Método: Para tal, vários métodos de ensino de programação foram analisados, junto a ferramentas de modelagem 3D. De forma a tentar unir ambos os conceitos, uma ferramenta de modelagem 3D utilizando programação chamada Madeup foi modificada, adicionando conceitos de gamificação buscando aumentar a atratividade da mesma. Além disso, um curso baseado nessa ferramenta foi desenvolvido, ensinando o básico para utilização da ferramenta, conceitos inicias de programação e conceitos de modelagem 3D, de forma a ser possível a realização de atividades no estilo *(*maker).
Resultados: Com o fim da pesquisa, temos uma ferramenta gamificada para modelagem 3D utilizando programação, assim como um curso de ensino de programação que se aproveita do conceito de modelagem e impressão 3D assim como da gamificação da ferramenta.
Conclusões: O desenvolvimento de recursos educacionais requer muita pesquisa e fundamentação sobre os conceitos que se deseja abordar com esses recursos, sendo que quanto maior a quantidade de conceitos a serem abordados, maior é a dificuldade para relacioná-los e uni-los. Isso também se aplica aos mecanismos motivacionais, que podem ser difíceis de se relacionar, visto que, no caso desse trabalho, são dois mecanismos completamente diferentes. Como trabalho futuro temos a aplicação dos recursos educacionais com alunos do ensino médio para avaliação de eficiência

dos recursos e a avaliação da ferramenta. Além disso, uma pesquisa está sendo realizada para identificar estudos que utilizam modelagem e impressão 3D como mecanismos motivacionais.

Palavras-chaves: Computação, Programação, Impressão em 3D, Gamificação, Educação, Recursos Educacionais

Abstract

Corrá, Leonardo. Educational Resources for Introductory Computer Teaching Using 3D Printer and Gamification. 2019. 45. f. Monograph (Undergradute Program in Computer Science), Federal University of Technology – Paraná. Campo Mourão, PR, Brazil, 2019.

Context: Studies in the literature point to a high dropout in higher education courses. Given the importance of this sector for the technological development of the country, it is extremely important to develop teaching methods that foster the interest of high school students who may enter the area, so that there is a reduction in this dropout in computer courses. New technologies are known to attract people's interest and 3D printing is one of the most prominent examples of technology today. In addition, game concepts also help to attract people's cognitive interest in learning. In this context, this work proposes an approach that combines concepts of 3D printing and gamification in order to increase the attractiveness of courses in the computing field.
Objective: The aim of this work was to generate a set of educational resources to teach the basics of programming in order to introduce the area of computing.
Method: To this end, several programming teaching methods were analyzed, together with 3D modeling tools. In order to try to unite both concepts, a 3D modeling tool using programming called Madeup was modified, adding gamification concepts to increase its attractiveness. In addition, a course based on this tool was developed, teaching the basics of using the tool, early programming and 3D modeling concepts, in order to be able to perform maker-style activities.
Results: With the end of the research, we have a gamified tool for 3D modeling using programming, as well as a programming teaching course that takes advantage of the concept of 3D modeling and printing as well as the gamification of the tool.
Conclusions: The development of educational resources requires research and reasoning about the concepts to be addressed with these resources, and the greater the number of concepts to be considered, the greater the difficulty to relate and unite them. This also applies to motivational mechanisms, which can be difficult to relate to, as in the case of this work they are two completely different mechanisms. As future work, we will use the educational resources with high school students to evaluate their efficiency and to evaluate the usage of the tool. In addition, a research is being conducted to identify studies that use 3D modeling and printing as motivational mechanisms.

Keywords: Computer Science, Programming, 3D printing, Gamification, Education, Educational Resource

Lista de figuras

Lista de acrônimos

CAD *Computer Aided Design* 12

NURBS *Non Uniform Rational Basis Spline* 13

AP **CSP** *Advanced Placement Computer Science Principles* 18

ECS *Exploring Computer Science* 18

AP *Advanced Placement* 18

STEM *Science, Tecnology, Engineer and Math* 19

STEAM *Science, Tecnology, Engineer, Arts and Math* 21

IoT Internet das Coisas 30

ARCS Attention Relevance Confidence Satisfaction 11

Sumário

Capítulo 1

Introdução

O ensino de computação é uma área que vem evoluindo constantemente deste de que surgiu na década de 1950. Um dos principais problemas quanto ao ensino de computação hoje é o fato da evasão dos alunos que ingressam na área. Com cerca de apenas 23% dos ingressantes nos cursos de Computação concluindo os cursos (SBC, 2015), há a necessidade de manter esses alunos não concluintes interessados na área. Além disso, a maior parte dos ingressantes se concentram na região sudeste do país, tendo pouca concentração de alunos nas outras regiões do país (SBC, 2015), então também é necessário aumentar o interesse dos alunos de ensino médio das demais regiões a ingressarem em cursos de Computação. Com isso, muitos estudos (IBÁÑEZ et al., 2014; SNOW et al., 2017; ASTRACHAN; OSBORNE, 2016) foram desenvolvidos no sentido de se introduzir a área da Computação para possíveis ingressantes ao ensino superior, tentando descobrir formas para atrair a atenção desses alunos e trazê-los para o curso de Computação.

Como questão principal deste trabalho, temos a geração de recursos educacionais, com foco em auxiliar o ensino de programação, visando facilitar o ensino dos conceitos chave da programação e aumentar o interesse em relação a área da computação. Os recursos focados foram a modificação de uma ferramenta de modelagem 3D, adicionando mecanismos de gamificação para ensino de programação e um curso abordando a ferramenta, ensinando os conceitos básico de programação, uso da ferramenta e modelagem 3D.

Utilizamos como ferramenta de modelagem 3D o Madeup (JOHNSON; BUI, 2015a, 2015b; JOHNSON et al., 2016a; JOHNSON, 2017), linguagem de programação, tanto escrita como no formato em blocos, para a criação de um curso para ensino de programação com modelagem 3D de 3 módulos, ensinando, respectivamente, o uso da ferramenta, conceitos de programação e conceitos de modelagem 3D, feito tanto para utilização da linguagem escrita do Madeup quanto para a linguagem em blocos. Junto ao curso, foi feita uma modificação no Madeup, para acrescentar gamificação à ferramenta, buscando incentivar ainda mais os possíveis participantes à realizarem as atividades e aprenderem sobre computação. Também foi elaborado um questionário para

avaliação de motivação dos participantes, baseado no modelo Attention Relevance Confidence Satisfaction (ARCS). O questionário não foi aplicado, mas foi criado por serem úteis na aplicação dos recursos educacionais, sendo uma ótima ferramenta para analisar a experiência de se aplicar os recursos.

O texto está dividido em 4 capítulos. No próximo capítulo são tratados os conceitos básicos sobre educação em computação, impressão em 3D e gamificação para introduzir melhor esses elementos que serão utilizados no trabalho. No capítulo 3 são apresentados os resultados da pesquisa, discorrendo sobre o curso e ferramenta gerados. No último capítulo, temos as conclusões sobre a pesquisa.

Capítulo 2

Conceitos

Neste capítulo são apresentados os conceitos de educação em computação, impressão em 3D e gamificação que serão utilizados posteriormente na monografia.

2.1. Impressão 3D

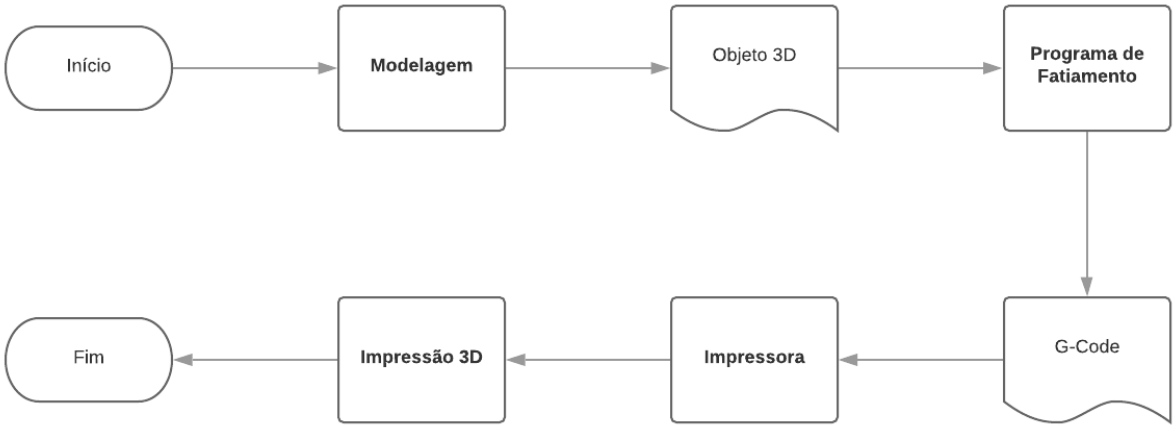
Impressão 3D é uma inovação tecnológica que permite ao usuário fabricar um objeto sólido tridimensional a partir de um modelo tridimensional digital. Este objeto comumente é criado pela impressora através da sobreposição de sucessivas camadas do material de produção, por ser o método mais barato de impressão. Os materiais podem variar desde diversos tipos de polímeros sintéticos, e até tecidos sintéticos. Essa tecnologia se tornou bastante popular pois é uma forma mais acessível de fabricação aditiva que torna viável a fabricação de peças personalizadas e em baixo volume, além do baixo custo.

O fluxo de uma impressão em 3D segue como mostra a Figura 2.1, começando pela modelagem, comumente feita através de software de *Computer Aided Design* (CAD), que geram o objeto 3D. Por sua vez, esse objeto é mandado para um programa de fatiamento, gerando um código legível pela impressora 3D, no formato G-Code. Com esse código, a impressora tem todas as coordenadas e instruções necessárias para realizar a impressão.

2.1.1. Modelos CAD

CAD é o nome genérico para softwares que facilitam o projeto e desenho técnicos, utilizados comumente na engenharia, arquitetura e design. Estes sistemas fornecem uma série de ferramentas para construção de entidades geométricas planas (como linhas, curvas, polígonos) ou mesmo objetos tridimensionais (cubos, esferas, etc.). Também disponibilizam ferramentas para relacionar essas entidades ou esses objetos, por exemplo: criar um arredondamento entre duas linhas ou subtrair as formas de dois objetos tridimensionais para obter um terceiro. Geralmente é feita uma

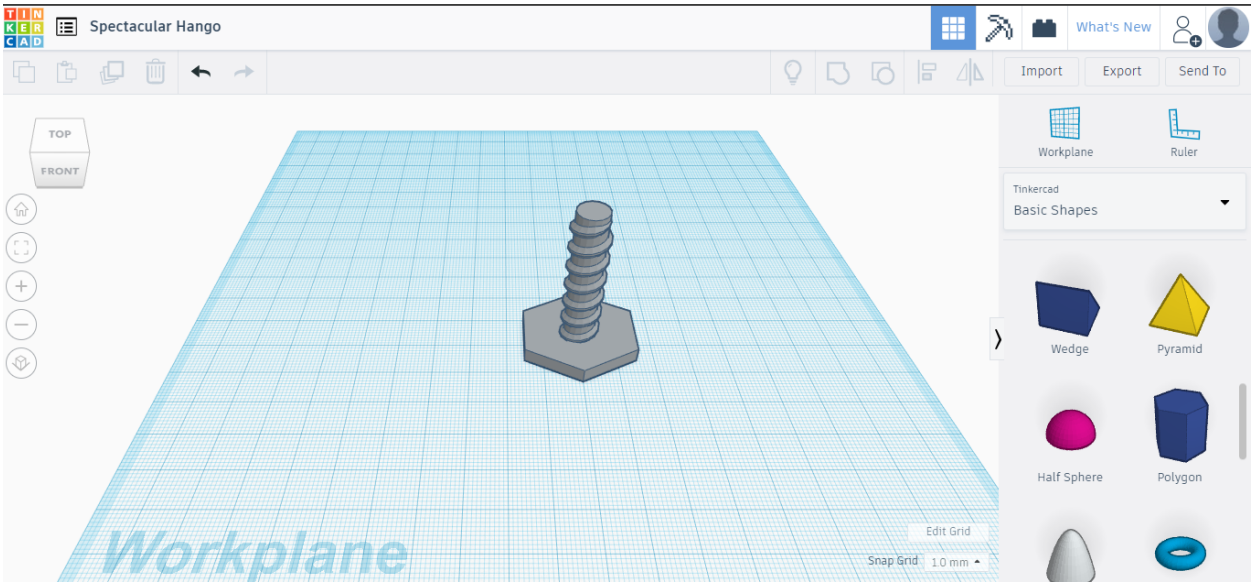
Figura 2.1. Fluxograma de impressão 3D



Fonte: Autoria própria.

divisão básica entre os softwares CAD com base na capacidade do programa em desenhar apenas em 2 dimensões ou criar modelos tridimensionais também, sendo estes últimos subdivididos ainda em relação a que tecnologia usam como modelador 3D, sendo esses gerados por polígonos ou por *Non Uniform Rational Basis Spline* (NURBS). Na Figura 2.2 temos um parafuso feito no *TINKERCAD*[1], ferramenta de CAD online e gratuita.

Figura 2.2. Parafuso feito no TINKERCAD



Fonte: Autoria própria.

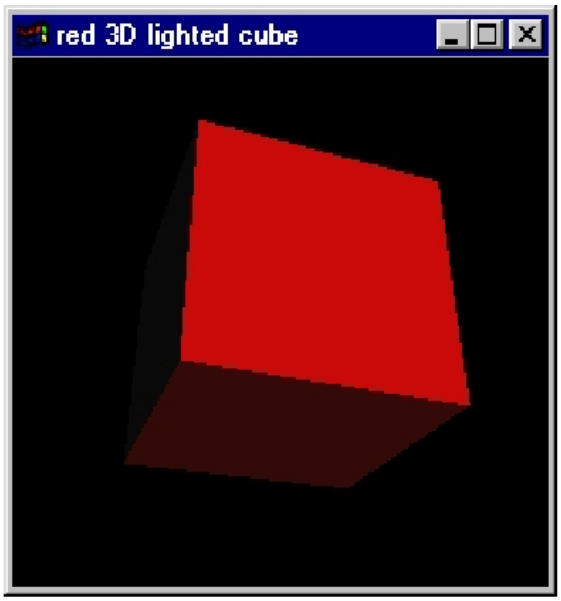
[1] https://www.tinkercad.com/

2.1.2. Modelos gerados por programação

Modelagem em 3D se utilizando de linguagens de programação são diferentes, já que, em suma, seu uso é focado através de bibliotecas ou, alguns casos, em linguagens especificas apenas para a programação de modelos 3D, porém, o meio mais comum sendo as plataformas voltadas apenas para a modelagem, como os CADs. Dito isso, alguns exemplos da utilização de bibliotecas podem ser vistas em C (Usando a biblioteca *OpenGL*, mostrado na Figura 2.3) ou em Java (Usando, por exemplo, a biblioteca LWJGL), além das plataformas como o *OpenSCAD* (KINTEL; WOLF, 2011), mostrado na Figura 2.4, e *FormWriter* (GROSS, 2001) que são linguagens de programação em texto especificas para modelagem 3D, que permitem a composição de modelos complexos com primitivas mais simples, porém são em grande parte declarativas e agrupadas com uma biblioteca padrão contendo muitas primitivas pré-fabricadas.

Figura 2.3. Cubo feito em OpenGL

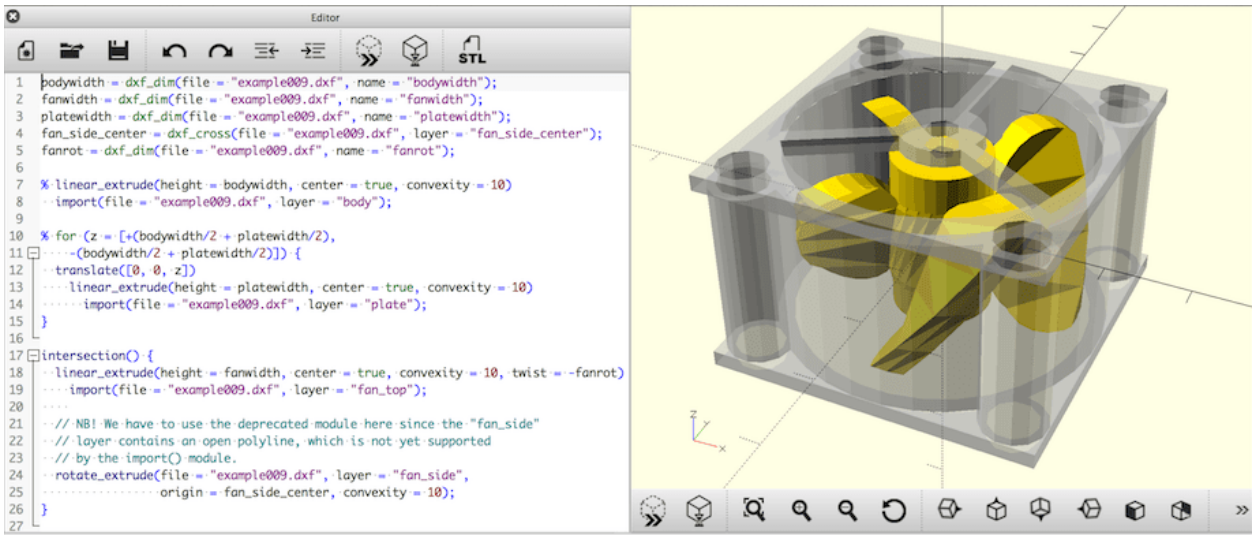
```
void
drawBox(void)
{
  int i;

  for (i = 0; i < 6; i++) {
    glBegin(GL_QUADS);
    glNormal3fv(&n[i][0]);
    glVertex3fv(&v[faces[i][0]][0]);
    glVertex3fv(&v[faces[i][1]][0]);
    glVertex3fv(&v[faces[i][2]][0]);
    glVertex3fv(&v[faces[i][3]][0]);
    glEnd();
  }
}

void
display(void)
{
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  drawBox();
  glutSwapBuffers();
}

void
init(void)
{
  /* Setup cube vertex data. */
  v[0][0] = v[1][0] = v[2][0] = v[3][0] = -1;
  v[4][0] = v[5][0] = v[6][0] = v[7][0] = 1;
  v[0][1] = v[1][1] = v[4][1] = v[5][1] = -1;
  v[2][1] = v[3][1] = v[6][1] = v[7][1] = 1;
  v[0][2] = v[3][2] = v[4][2] = v[7][2] = 1;
  v[1][2] = v[2][2] = v[5][2] = v[6][2] = -1;

  /* Enable a single OpenGL light. */
  glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
  glLightfv(GL_LIGHT0, GL_POSITION, light_position);
  glEnable(GL_LIGHT0);
  glEnable(GL_LIGHTING);
```



Fonte: https://www.opengl.org/archives/resources/code/samples/glut_examples/examples/examples.html.

Com isso, a necessidade de um ambiente mais simplificado de programação voltado para a modelagem 3D surge, visto que a utilização de bibliotecas e linguagens como o *OpenSCAD* acabam sendo inviáveis para iniciantes em computação, que tem conhecimento mínimo ou nulo em programação.

Figura 2.4. *Cooler* feito em OpenSCAD



Fonte: https://www.openscad.org/index.html.

Assim sendo, uma linguagem com o conceitos simples de modelagem, como o Madeup, é atraente, já que envolve comandos simples de movimentação e solidificação dos objetos para realizar a renderização dos objetos, com *feedback* visual do resultado do código em tempo real.

Continuando nesse caminho, a utilização de programação em blocos facilita o processo, sendo um método mais simples e visual de programação. A programação em blocos funciona basicamente como um quebra-cabeça. Os blocos são coloridos dependendo do seu tipo e a execução flui de cima para baixo através da sequência dos blocos. Esse formato é mais simples para uma aprendizagem focada em pessoas inexperientes em programação, já que não se tem a preocupação com pontuação ou sintaxe.

Utilizando programação em blocos, duas ferramentas se destacam, o Beetle Blocks (KOSCHITZ; ROSENBAUM, 2012) e o Madeup (JOHNSON; BUI, 2015a, 2015b; JOHNSON et al., 2016a; JOHNSON, 2017), ambas como linguagens de programação voltada para modelagem em 3D utilizando o formato de programação em blocos. Neste trabalho optamos pela utilização do Madeup.

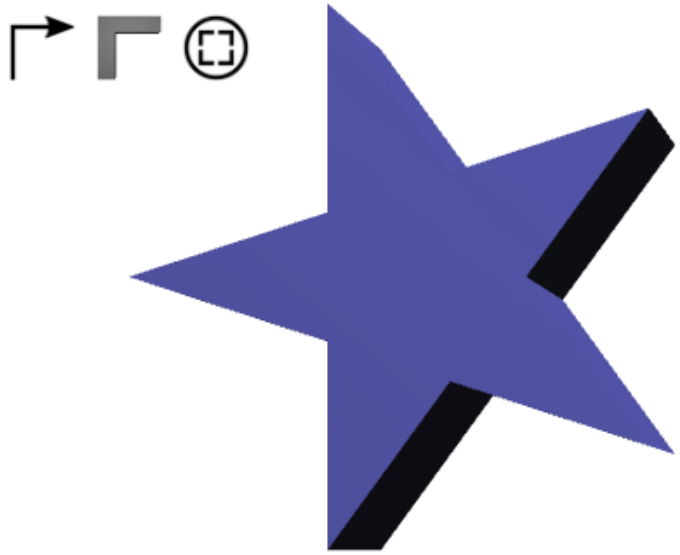
Na Figura 2.5 é mostrada a implementação de uma estrela em 3D criada pelo Madeup em seu formato textual e na Figura 2.6 temos a implementação da mesma estrela feita com o formato em blocos.

2.1.3. Fatiamento e G-Code

Para a impressora realizar a impressão do objeto 3D, o mesmo tem que passar por um software de fatiamento, que por sua vez gera o G-Code que a impressora executará para realizar a impressão. G-Code nada mais é do que uma linguagem de programação baseada em letras do alfabeto, em que cada letra tem uma função. Pelo fato da letra G abrigar as funções preparatórias, a linguagem

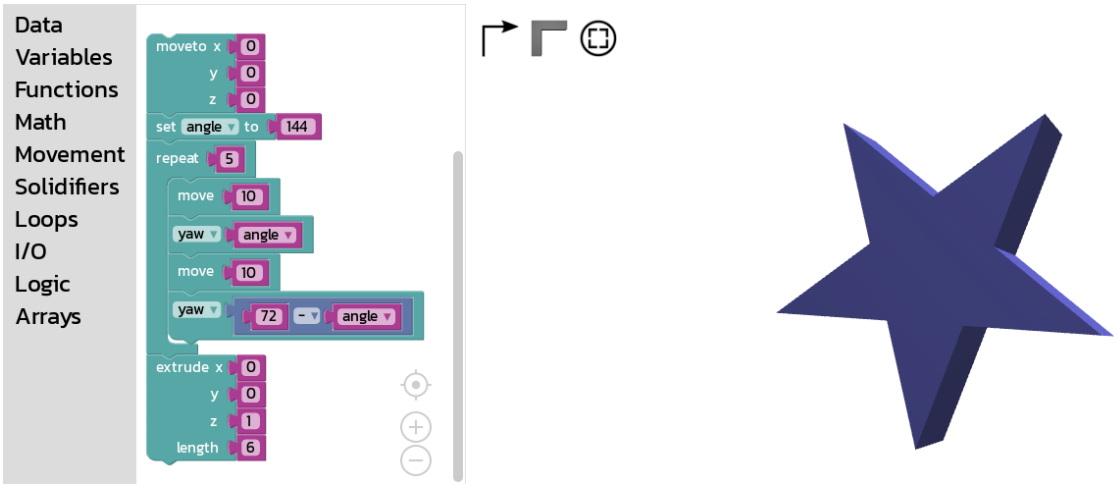
Figura 2.5. Estrela em 3D feita no Madeup com programação em texto

```
moveto 0,0,0

angle = 144
repeat 5
  move 10
  yaw angle
  move 10
  yaw 72 - angle
end

extrude 0, 0, 1, 6
```



Fonte: Autoria própria.

Figura 2.6. Estrela em 3D feita no Madeup com programação em blocos



Fonte: Autoria própria.

é chamada de G-Code. Outra letra importante da linguagem é o M, que refere as diversas funções de máquina da linguagem. Como funções mais utilizadas no G-Code temos:

- G00 (Posicionamento Rápido): Move cada eixo em sua velocidade máxima até atingir sua quantidade vetorial. O vetor mais curto geralmente termina primeiro (com velocidades de eixo semelhantes). Isso importa porque pode render um movimento de perna de cachorro ou taco de hóquei, o que o programador precisa considerar dependendo de quais obstáculos estão próximos, para evitar um acidente. Algumas máquinas oferecem corredeiras interpoladas como recurso para facilitar a programação.
- G01 (Interpolação Linear): O programa especifica os pontos inicial e final, e o controle calcula (interpola) automaticamente os pontos intermediários para passar através de uma

linha reta (portanto, “linear”). O controle então calcula as velocidades angulares para girar os parafusos do eixo através de seus servo-motores ou motores de passo.

- G28 (Retornar a posição inicial ou zero da máquina, também conhecido como ponto de referência da máquina): Toma endereços em X, Y, Z que definem o ponto intermediário que a ponta da ferramenta passará no seu caminho de casa para a origem da máquina. Eles são em termos da posição zero do programa, não da máquina.
- G53 (Sistema de coordenadas de máquina): Toma as coordenadas absolutas (X, Y, Z) como referência à posição zero da máquina em vez de zero do programa. Pode ser útil para mudanças de ferramentas. Não modal e absoluto.
- G20 e G21: Programam o sistema para utilização de coordenadas em polegadas e em milímetros, respectivamente.
- G97 (Velocidade constante do fuso): Toma um inteiro após um S, que é interpretado como revoluções por minuto (rpm), modo de velocidade padrão por parâmetro do sistema se nenhum modo estiver programado.

Também há funções para controle de temperatura, quantidade de filamento a ser extrudado e outras mais específicas, porém, como o foco deste trabalho é o ensino simplificado para alunos de ensino médio, serão explicadas apenas as funções de movimentação, as quais possuem relação direta com os conceitos de programação das impressões.

Na Figura 2.7 temos um trecho do G-Code da estrela 3D mostrada no exemplo de modelagem do Madeup.

Figura 2.7. Trecho final do G-Code da estrela 3D

```
G1 X107.396 Y105.301
G0 F7200 X107.892 Y105.37
G1 F1800 X108.541 Y104.721 E348.40341
G1 X108.611 Y104.651
G0 F7200 X108.549 Y105.279
G1 F1800 X108.881 Y105.512 E348.41199
G1 X109.214 Y105.745 E348.41615
G0 F7200 X109.173 Y105.787
G0 X102.698 Y104.257
G0 X99.989 Y107.051
G1 F1800 X98.241 Y108.799 E348.45726
G1 X98.171 Y108.869
G0 F7200 X98.483 Y109.123
G1 F1800 X98.241 Y109.365 E348.46295
;TIME_ELAPSED:631.012068
G1 F1500 E341.96295
M107
M104 S0 ;extruder heater off
M140 S0 ;heated bed heater off (if you have it)
G91 ;relative positioning
G1 E-1 F300  ;retract the filament a bit before lifting the nozzle, to release some of the pressure
G1 Z+0.5 E-5 X-20 Y-20 F9000 ;move Z up a bit and retract filament even more
G28 X0 Y0 ;move X/Y to min endstops, so the head is out of the way
M84 ;steppers off
G90 ;absolute positioning
M82 ; absolute extrusion mode
M104 S0
;End of Gcode
```

Fonte: Autoria própria.

2.2. Educação em Computação

Educação em computação é um tópico bem abrangente, visto que a própria área da computação em si é muito vasta, com várias subáreas incluídas nesse contexto. Assim, foram considerados currículos e abordagens de ensino de computação para agregar mais informação sobre o ensino de programação abordado nesses trabalhos.

2.2.1. Currículos

Dito isso, existem algumas abordagens de currículo para ensino em computação, como por exemplo o *Advanced Placement Computer Science Principles* (AP CSP)[2] (ASTRACHAN; OSBORNE, 2016) e o *Exploring Computer Science* (ECS)[3] (SNOW et al., 2017).

ECS é um curso que consiste em 6 unidades. O curso foi desenvolvido em torno de uma estrutura de conteúdo de ciência da computação e prática computacional. As atribuições e instruções são contextualizadas para serem socialmente relevantes e significativas para estudantes diversos. As áreas da computação tratadas pelo currículo do ECS são Interação Humano-Computador, Resoluções de Problemas, Design Web, Programação, Computação e Análise de Dados e Robótica. As unidades utilizam uma variedade de ferramentas e plataformas que culminam com projetos finais.

O AP CSP é um curso de informática *Advanced Placement* (AP) e exame oferecido pelo College Board para estudantes do ensino médio como uma oportunidade de ganhar crédito universitário para um curso de ciência da computação de nível universitário. Concebido para ampliar a participação da informática no ensino médio e além, sendo acessível aos professores que possam ser novos em ciência da computação. O AP CSP engloba certas práticas de pensamento computacional, como:

- Computação de conexão: Os alunos aprendem a estabelecer conexões entre diferentes conceitos de computação.
- Criação artefatos computacionais: Os alunos se envolvem nos aspectos criativos da computação, criando e desenvolvendo artefatos computacionais, bem como aplicando técnicas de computação para resolver de forma criativa problemas.
- Abstração: Os alunos usam abstração para desenvolver modelos e simulações de fenômenos naturais e artificiais, usando-os para fazer previsões sobre o mundo e analisar sua eficácia e validade.
- Análise de problemas e artefatos: Os alunos projetam e produzem soluções, modelos e artefatos, avaliam e analisam o próprio trabalho computacional, bem como o trabalho computacional que os outros alunos produziram.

[2] https://apstudent.collegeboard.org /apcourse/ap-computer-science-principles
[3] http:// www.exploringcs.org/curriculum

- Comunicação: Os alunos descrevem a computação e o impacto da tecnologia e computação, explicando e justificando o design e a adequação de suas escolhas computacionais. Também analisam e descrevem os artefatos computacionais gerados e os resultados ou comportamentos desses artefatos.
- Colaboração: Os alunos colaboram em uma série de atividades, incluindo a investigação de questões utilizando conjuntos de dados e a produção de artefatos computacionais.

O currículo AP CSP foi fundamentado em uma abordagem inclusiva e baseada na comunidade, de forma a construir um novo curso que é quase sem precedentes. Neste trabalho, a utilização dos elementos de Computação de conexão, Criação artefatos computacionais, Análise de problemas e artefatos e Comunicação são essenciais, visto que em parte as atividades buscam a criação de objetos de acordo com a criatividade dos participantes, porém sempre com uma questão chave envolvida.

2.2.2. Abordagens

A grande dificuldade do ensino, não só em computação, mas em qualquer área, é ter o engajamento dos alunos no conteúdo ensinado. Além disso, muitas das matérias da computação envolvem conceitos de programação, que não são apresentados no ensino médio comum. Um bom exemplo de como simplificar a aprendizagem neste contexto é a programação em blocos que funciona como um quebra-cabeça. Os blocos são coloridos dependendo do seu tipo e a execução flui de cima para baixo através da sequência dos blocos. Alguns exemplos de programação em blocos podem ser vistos nos trabalhos *Scratch: Programming for All* (RESNICK et al., 2009), que incentiva o uso criativo da programação em blocos em diversas aplicações como, por exemplo, animações e jogos, e *Blocks in, blocks out- A language for 3D models* (JOHNSON; BUI, 2015a), que se utiliza de impressão e modelagem em 3D juntamente a uma linguagem de programação em blocos para atrair a atenção dos alunos e fornecer os conceitos mínimos de programação. Na Figura 2.8, temos um exemplo de código feito no *Scratch*, que realiza a animação do desenho de um gato andando.

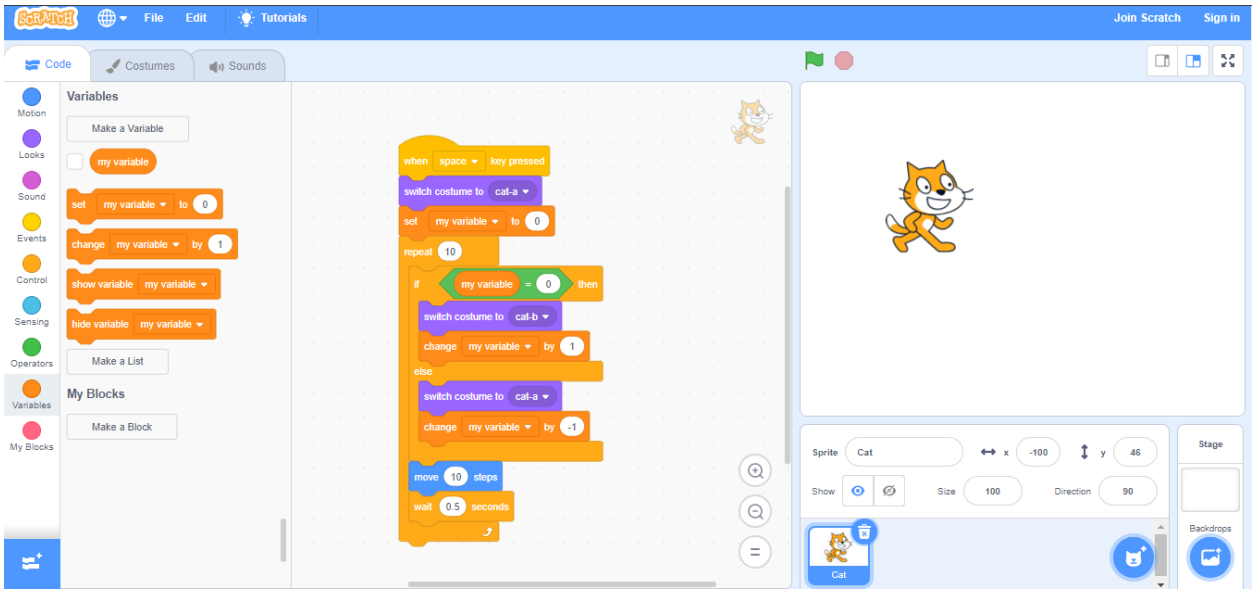
2.3. Educação em Computação com impressão 3D

Muitas pesquisas se utilizam de impressão em 3D para ensino de computação. Abaixo são citados alguns exemplos de trabalhos desenvolvidos no contexto de aprendizagem em computação com impressões em 3D.

2.3.1. Impressão em 3D no ensino médio

No artigo (STANSELL; TYLER-WOOD, 2016) foi analisado o impacto da utilização de projetos *Science, Tecnology, Engineer and Math* (STEM) com impressora 3D para o ensino fundamental. Noventa e nove alunos do ensino fundamental foram selecionados para experimentar um ambiente

Figura 2.8. Exemplo de código feito no Scratch



Fonte: Autoria própria.

de aprendizado baseado em STEM e parte desses alunos utilizaram uma impressora 3D para completar um desafio de impressão. Esses alunos utilizaram cálculos e medidas matemáticas exatas, raciocínio científico em seus projetos e tecnologia para transformar seus pensamentos em uma solução de engenharia real. O período de intervenção se baseou num pré-teste, realizado antes da intervenção, e num pós-teste, aplicado após a intervenção, em que todos os alunos, tanto os que utilizaram a impressora quantos os que não utilizaram, fizeram os testes. Os testes possuíam vários tipos de dados, como dados acadêmicos, de interesse da atividade, percepção da escola e de interesse STEM. A parte acadêmica possuía questões de matemática e ciências desde o 5º ao 9º ano, limitadas a questões de múltiplas escolhas (STANSELL; TYLER-WOOD, 2016). Como resultado, os dados obtidos não mostraram diferença entre o aprendizado dos dois grupos na área de ciências, mas na área de matemática houve uma queda no rendimento dos alunos que não utilizaram impressora 3D, enquanto os que utilizaram mantiveram resultados semelhantes (STANSELL; TYLER-WOOD, 2016).

2.3.2. Smart Cup, Wisdom Creation: A Project-Based Learning Initiative for Maker Education

Esse artigo (WANG et al., 2016) ilustra como as habilidades de modelagem, programação e modelagem 3D dos estudantes são adquiridas e aprimoradas através do processo de fazer um Copo Inteligente e também propõe um modelo de aprendizagem baseado em projetos. O processo de criação de um Copo Inteligente envolve duas fases: primeiramente, projetar as características do copo, para desenvolver a capacidade de projetar e escrever programas (feitos em Arduíno) e, em

seguida, imprimir o corpo do Copo, para desenvolver a capacidade de modelagem em 3D (WANG et al., 2016).

O copo foi projetado a ter as seguintes características: exibir a temperatura da água no Copo, lembrar o usuário do Copo de beber uma quantidade de água dependendo da temperatura da água e permitir que usuários estabeleçam alertas de temperatura para diferentes tipos de bebidas. Para a impressão do corpo do Copo, os alunos foram incentivados a atualizar o modelo do Copo, o que ajuda a gerar um conhecimento e habilidade na modelagem em 3D.

O modelo proposto no artigo é composto por quatro estágios lineares, porém inter-relacionados, que são a entrada do projeto, o processamento do projeto, a saída do projeto e o monitoramento do projeto, sendo que em cada estágio os alunos devem estar sempre no centro, com sua experiência de participação como prioridade o tempo todo. O estágio de entrada do projeto é onde ocorre a criação de um cenário para resolução de problemas, cenário o qual deve ser suficiente para motivar a participação dos alunos no projeto. O estágio do processamento do projeto visa aprender e experimentar o conteúdo do curso, que geralmente compreende três seções, e em cada seção incentiva-se os alunos a participar e explorar. A primeira seção é criar, a segunda é integrar *Science, Tecnology, Engineer, Arts and Math* (STEAM) e a terceira é auto-extensão. O estágio da saída do projeto se baseia no compartilhamento de resultados de aprendizagem, onde os alunos analisam e compartilham o que aprendem, demonstrando o que criam. Enquanto isso, o nível de aprendizagem dos alunos deve demonstrar a capacidade dos mesmos de resolver problemas práticos e compreender o conhecimento científico adquirido. O estágio de monitoramento do projeto serve para avaliar as criações dos alunos, com um sistema racional e eficaz. A avaliação (que pode incluir avaliações diagnóstica, formativa e somativa) é aconselhável para executar todo o processo de aprendizagem (WANG et al., 2016).

Como conclusão, eles viram que era necessário retestar o modelo de estudos proposto e melhorá-lo de acordo com a necessidade (WANG et al., 2016).

2.3.3. Madeup

O Madeup é uma ferramenta para modelagem em 3D baseada em uma linguagem de programação, que visa ensinar alunos de ensino fundamental e médio conceitos básicos de programação e computação através da criação e impressão dos objetos 3D modelados (JOHNSON, 2017). Além de seu formato padrão de programação textual, o Madeup também possui uma interface de programação em blocos. Esse tipo de interface é mais fácil para iniciantes navegarem, além de eliminar a preocupação sobre detalhes superficiais como espaços em branco e pontuações desnecessárias. O programador trabalha mais diretamente com a árvore de sintaxe abstrata do programa, tornando a interação com o programa mais direta e simples (JOHNSON; BUI, 2015a). A interface de programação em blocos foi inserida na ferramenta para aumentar a presença do mundo real no processo de aprendizagem, visto que esse é um dos focos do Madeup, que também se utiliza da impressão dos modelos 3D para trabalhar em cima da cognição e experiência física

dos alunos (JOHNSON; BUI, 2015a). Esse ambiente mais físico facilita o aprendizado e, somado à saída que é um modelo 3D impresso, os desenvolvedores tiveram um grande progresso em relação a barreira do pensamento computacional.

Dentro da ferramenta a modelagem é feita da seguinte maneira: um modelador atravessa um caminho 1D representativo ou uma polilinha através do espaço 3D usando comandos de movimento e curvatura. O modelador pode representar dados simbolicamente, usar operadores matemáticos e compor funções. Os comandos podem ser sequenciados usando estruturas de controle de fluxo padrão (JOHNSON, 2017). À medida que o usuário escreve o programa, o caminho que está sendo percorrido é rastreado em tempo real e um cursor 3D mostra a posição e a direção atuais. O caminho percorrido é então interpretado como uma seção transversal ou esqueleto de um objeto sólido, onde o modelador emite um comando de solidificador apropriado para produzir um modelo sólido 3D.

2.4. Gamificação

O conceito de gamificação trata do uso de elementos e dinâmicas presentes em jogos em áreas fora deste contexto, no intuito de trazer uma experiência satisfatória e divertida para envolver e incentivar certas atitudes dos possíveis usuários (IBÁÑEZ et al., 2014). A Gamificação é uma área de constante crescimento hoje em dia, tendo grande uso na área comercial, sendo utilizada por diversas marcas para apoiar o engajamento dos usuários e promover a atividade do usuário e a interação social. Como alguns mecanismos de gamificação, temos, por exemplo:

- ***Placares*:** São tabelas que demonstram um ranking entre usuários de acordo com os objetivos alcançados pelo mesmo. Incentiva a competitividade entre os usuários, aumentando o rendimento individual.
- **Objetivos:** São metas que podem ser definidas pelo sistema ou até mesmo pelo próprio usuário. Tem o intuito de definir o foco do usuário em certa atividade, recompensando-o de alguma forma.
- ***Badges*:** São recompensas por atingir certos objetivos. Geralmente estão presentes no perfil do usuário e podem ser compartilhados com outros usuários. O sentimento de recompensa em fazer certas atividades atrai o interesse dos usuários.
- **Indicadores de Progresso:** Mostram ao usuário sua evolução durante o processo de uso. A sensação de progresso em algo incentiva a continuar as atividades para continuar evoluindo.
- **Estimular Competitividade:** Utilizar da competitividade focando nos elementos recreativos da gamificação gera um fator motivacional para engajar os colaboradores no esforço adequado.
- **Divulgação de Méritos Alcançados:** O reconhecimento ajuda a engajar e motivar as pessoas e, se utilizando de gamificação, pode-se sistematizar esse processo tornando-o gerenciável e desafiador.

- **Enredo:** Utilizar-se de um enredo envolvendo as atividades desejadas visam torna-las mais divertidas e interessantes para as pessoas envolvidas.

Esses mecanismos têm o intuito de atrair o interesse cognitivo do usuário, para incentivá-lo a interagir mais com outros usuários.

Como um bom exemplo da utilização de gamificação para educação em computação temos a *Khan Academy* (MORRISON; DISALVO, 2014), organização sem fins lucrativos que proporciona aulas online de diversas áreas diferentes, sendo a computação uma delas. Na *Khan Academy*, as aulas são apresentadas em vídeos online, com um mecanismo que gera exercícios baseados no nível de aprendizado de cada aluno. Para as aulas de computação, há uma área para programação junto a uma janela de execução a cada vídeo apresentado. A aplicação da gamificação no site é feita pelo uso de *badges* e pontos de energia como recompensa pela realização de atividades, indicadores de progresso para acompanhamento da evolução do estudante nas atividades realizadas e um *streak* de estudos que aumenta a quantidade de pontos de energia ganhos para cada dia seguido que o aluno entra na plataforma, para incentivar o mesmo a entrar todo dia na página *web*.

2.5. Considerações finais

Como a essência do trabalho é o ensino de programação e simplicidade para esse ensino, visto que o foco do ensino é para alunos de ensino médio, unimos os conceitos mais simples de Impressão 3D, Ensino de Computação e Gamificação apresentados, no capítulo seguinte, para implementação da ferramenta proposta.

Capítulo 3

Resultados

Aqui, será descrito sobre as características do curso criado, mostrando os passo para a escolha dos mecanismos de gamificação, sobre o conteúdo e a mecânica do curso. Também será falado o processo de gamificação do Madeup e sobre as avaliações que serão aplicadas com os participantes quando os recursos forem aplicados.

3.1. Curso introdutório à Computação com gamificação e impressão 3D

Como mostrado na seção 3 do capítulo anterior, trabalhos utilizando-se de impressões em 3D para atrair o interesse de alunos para a área da computação já foram desenvolvidos, mas estão num estado inicial e a questão de motivação geralmente não é explicitamente abordada nos trabalhos, sendo brevemente comentada ou nem ao menos citada. Já neste trabalho, o foco está voltado para essa questão, utilizando a gamificação para tentar aumentar a motivação dos participantes.

A união desses dois conceitos foi realizada através de missões, cada uma relativa a algum conhecimento de programação ou modelagem 3D, que, quando completas, recompensavam os participantes com pontos. Além disso, também se pode votar em objetos, recompensando o criador do mesmo com pontos. A ferramenta gamificada pode ser aplicada junto a um curso, ensinando o básico para o uso do Madeup na modelagem de objetos 3D, com atividades entre os módulos para os alunos criarem objetos de acordo com a sua criatividade.

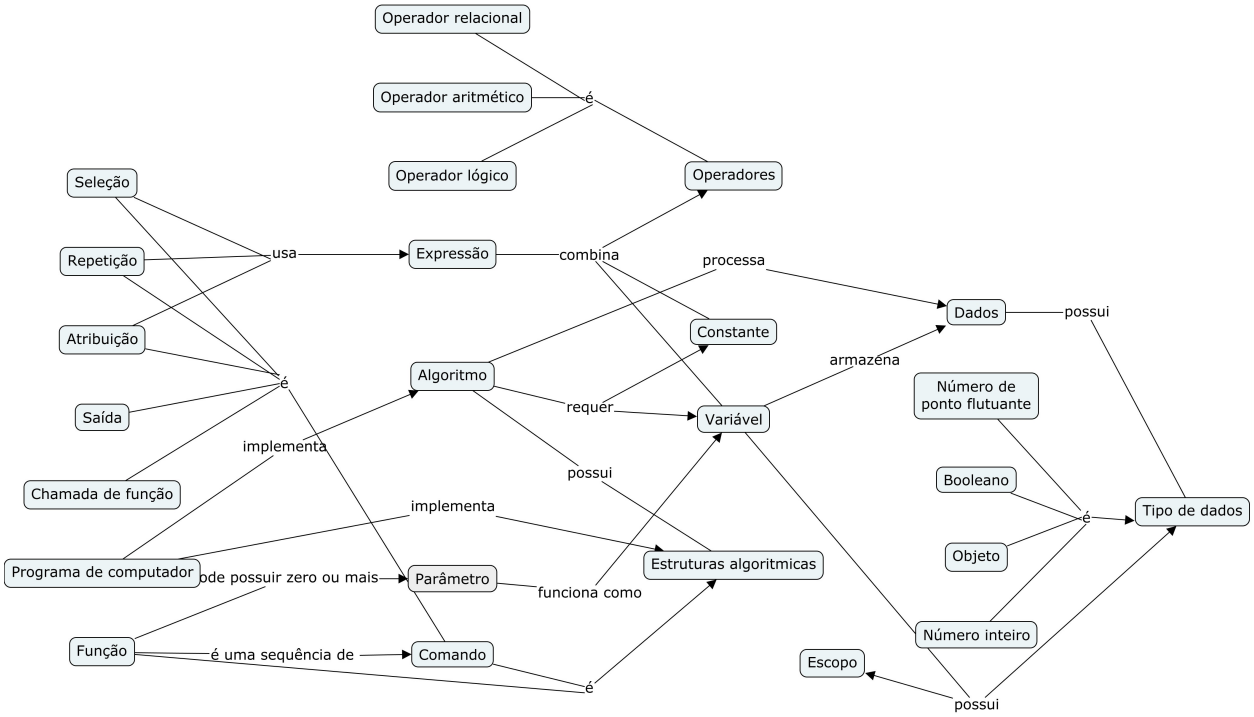
3.1.1. Projeto instrucional

Para a aplicação do curso, temos dois elementos a serem trabalhados com os alunos: programação, tendo como objetivo o ensino do básico de programação para pessoas com pouco ou nenhum conhecimento na área; e modelagem 3D, também por permitir uma representação direta e, com a

impressão 3D, real, dos resultados dos programas criados, facilitando o entendimento do domínio da Computação para os iniciantes na área.

Como o foco do trabalho é o ensino de programação, os principais conceitos abordados são aqueles básicos de programação. Conceitos como laços de repetição e condicionais são essenciais na programação, sendo praticamente inviável a construção de certos programas sem a utilização dos mesmos. Já conceitos como funções são questões organizacionais, para manter o código mais compreensível e limpo. Esses três são conceitos chave na área da programação, independente de linguagens de programação, o que os torna muito interessantes de se ensinar para iniciantes da área, para formar uma base de conhecimento em programação. Na Figura 3.1 se encontra um mapa conceitual com os principais conceitos de programação que abordamos neste trabalho ao utilizar o Madeup.

Figura 3.1. Mapa Conceitual de Programação

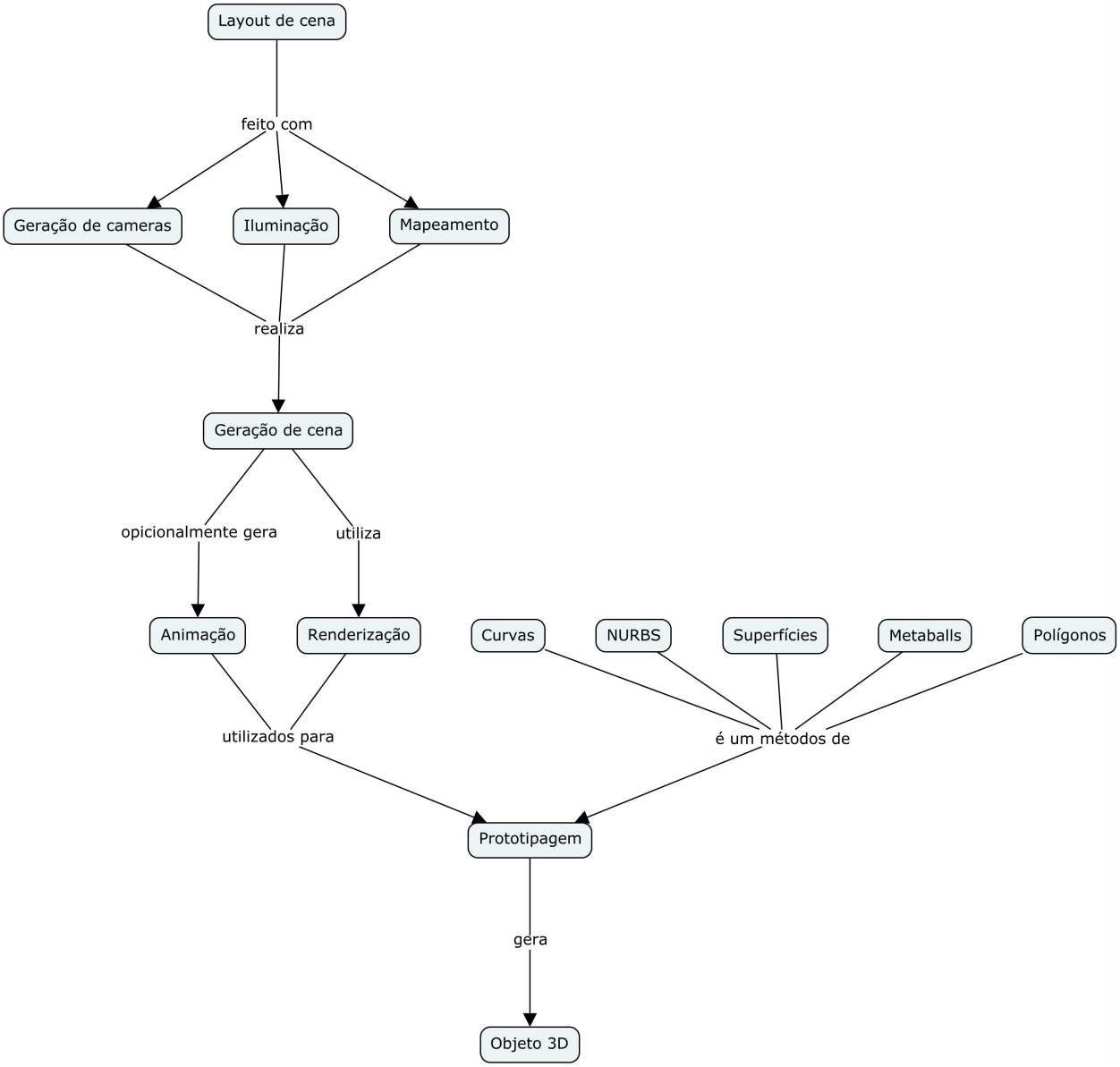


Fonte: Autoria própria.

Porém, ao se usar impressões em 3D para se engajar os alunos, também é necessário ensinar conceitos de modelagem de objetos. Normalmente, a modelagem de objetos 3D é realizada em *softwares* do tipo CAD, onde o fluxo de criação é separado em três fases (Layout de cena, Geração de cena e Prototipagem), cada fase com etapas mais especificas. Na Figura 3.2, descrevemos os conceitos chave para modelagem 3D.

Porém, na ferramenta Madeup, a modelagem 3D ocorre de maneira diferente. A modelagem é iniciada pela criação de uma linha 2D, utilizando comandos de movimentação para traçar a linha, o que se equivaleria ao mapeamento no layout de cena, e em seguida essa

Figura 3.2. Mapa Conceitual de Modelagem 3D



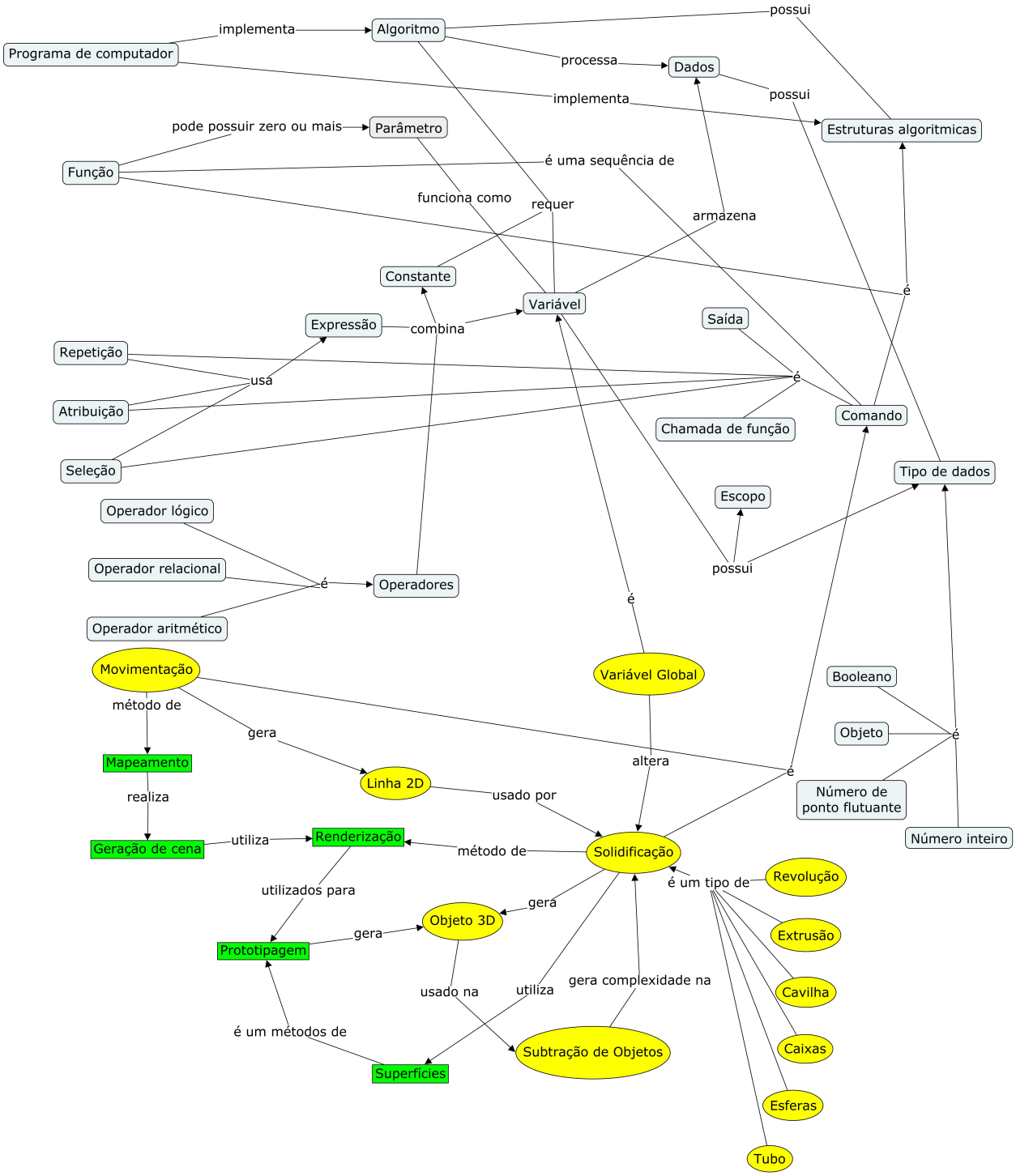
Fonte: Autoria própria.

linha é solidificada para um objeto, utilizando comandos de solidificação da ferramenta, sendo a renderização na geração de cena para gerar o protótipo. Além disso, algumas características dos objetos podem ser alteradas utilizando variáveis globais, como o raio ou cor do objeto, por exemplo. Logo, para se utilizar o Madeup, é preciso entender esses conceitos de movimentação, solidificação e de variáveis globais, visto que são através deles que são criados os objetos no Madeup. Ainda nesse quesito, também temos a subtração de objetos, para se gerar objetos mais complexos, muito importante para permitir que os participantes tenham uma liberdade maior para criar objetos focando em incentivar a criatividade deles.

Juntando todos esses conceitos, temos todo o conhecimento necessário a se passar para os participantes do curso. Esses conceitos foram formuladas em missões(que serão tratadas na

próxima subseção), para ensiná-los aos participantes de forma mais atrativa. O mapa conceitual final, juntando todos os conceitos é mostrado na Figura 3.3, onde os conceitos em cinza são os relacionados a programação, os em verde à modelagem 3D e os em amarelo os específicos para o Madeup.

Figura 3.3. Mapa Conceitual de Modelagem 3D e Programação



Fonte: Autoria própria.

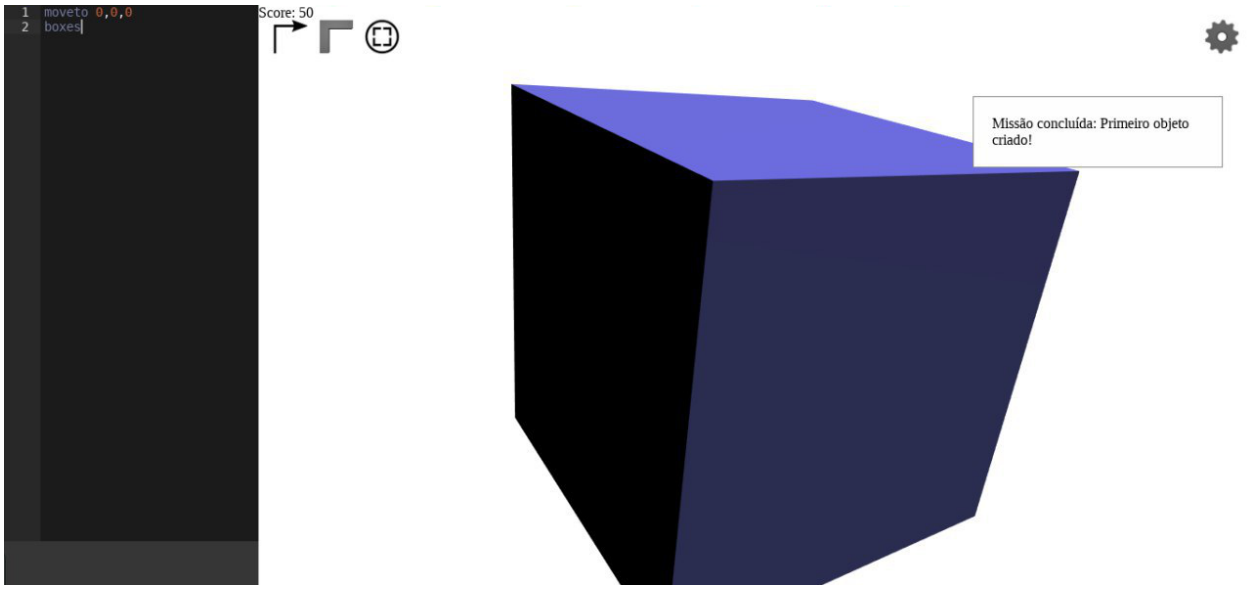
3.1.2. Mecanismos de gamificação

Para os mecanismos de gamificação, começamos pensando em relação aos mecanismos mostrados na Seção 2.4. Dentre eles, escolhemos os mecanismos mais simples e optamos pelos pontos, para estimular o sentimento de progressão, e objetivos, para estimular o sentimento de recompensa, juntamente para mostrar os conceitos mais importantes de forma intuitiva. Também adicionamos uma votação para os objetos criados, onde os participantes votam em objetos que gostaram, também para que vejam os objetos de outros para terem ideias e incentivá-los a serem mais criativos. Com isso, definimos os seguintes mecanismos de gamificação:

- **Missões:** Objetivos a serem completados pelos participantes para receber pontos; utilizado para demonstrar os conceitos de programação e modelagem 3D.
- **Votação dos Objetos criados:** Onde os participantes podem votar em um objeto para recompensar o criador do mesmo; Também servem para dar inspiração para os participantes, mostrando os objetos dos outros participantes para terem mais ideias.
- **Pontos:** Recompensa por completar missões ou receber votos, usado para poder imprimir um objeto criado.

Tanto as missões quanto os votos recompensam os alunos com pontos, que são mostrados no placar. As missões recompensam os participantes por completarem seis tarefas simples, relacionadas a criação de objetos, que no caso são: primeira, criar um objeto; segunda, criar um objeto utilizando variáveis globais; terceira, criar um objeto utilizando condicionais; quarta, criar um objeto utilizando laços de repetição; quinta, criar um objeto utilizando subtração de objetos; sexta, criar um objeto utilizando todos os conceitos anteriores. Cada missão tem como objetivo principal apresentar de maneira mais atrativa os conceitos necessários no curso, sendo a primeira e a segunda missão responsáveis pela criação básica de um objeto, usando os conceitos de movimentação e solidificação; e mostrar as variáveis globais para alterar atributos dos objetos. A terceira e quarta missões são responsáveis pela parte de programação, mostrando o uso de condicionais e laços de repetição para facilitar a criação de objetos. A quinta missão serve para mostrar a subtração de objetos, para a criação de objetos mais complexos. Vale lembrar que até a quinta missão, todas as missões serão tratadas em módulos do curso, sendo ensinados esses conceitos aos participantes. A última missão tem o foco em exercitar a imaginação dos participantes, envolvendo todos os conceitos aprendidos anteriormente para a criação de um objeto. Os pontos de recompensa por completar cada objetivo são maiores de acordo com a missão completada, começando com 50 pontos para a primeira missão e indo até 250 pontos para a última. Quando uma missão é completa, uma notificação é mostrada na tela no canto superior direito da área de renderização da ferramenta, avisando qual missão foi concluída e a quantidade de pontos do participante aumenta no canto superior esquerdo da área de renderização. Isso é mostrado na Figura 3.4.

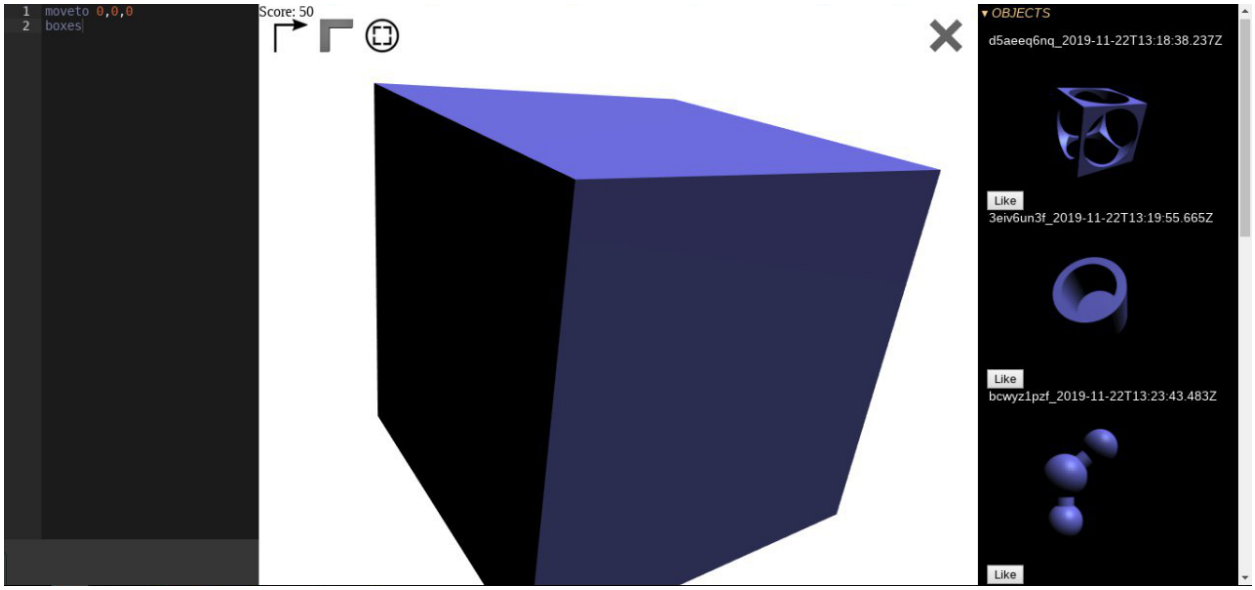
Figura 3.4. Madeup gamificado: missão sendo completa



Fonte: Autoria própria.

Para a votação, na ferramenta Madeup foi criada uma aba de objetos, onde são listados todos os objetos criados pelos participantes. Um participante pode votar num objeto, para recompensar o criador com pontos, caso goste do objeto. Além disso, ao se mostrar os objetos criados, os participantes podem se motivar a criar objetos melhores vendo objetos complexos que eles não imaginariam normalmente, aumentando as possibilidades de criação para os mesmos. Cada voto recompensa o criador do objeto com 100 pontos. A aba de votação é como mostra a Figura 3.5.

Figura 3.5. Madeup gamificado: missão sendo completa



Fonte: Autoria própria.

Os pontos são responsáveis por unir os dois mecanismos anteriores, recompensando os participantes por completarem as missões ou receberem votos. Além disso, os alunos com a maior pontuação poderão imprimir um de seus objetos na impressora 3D, como recompensa por sua criatividade.

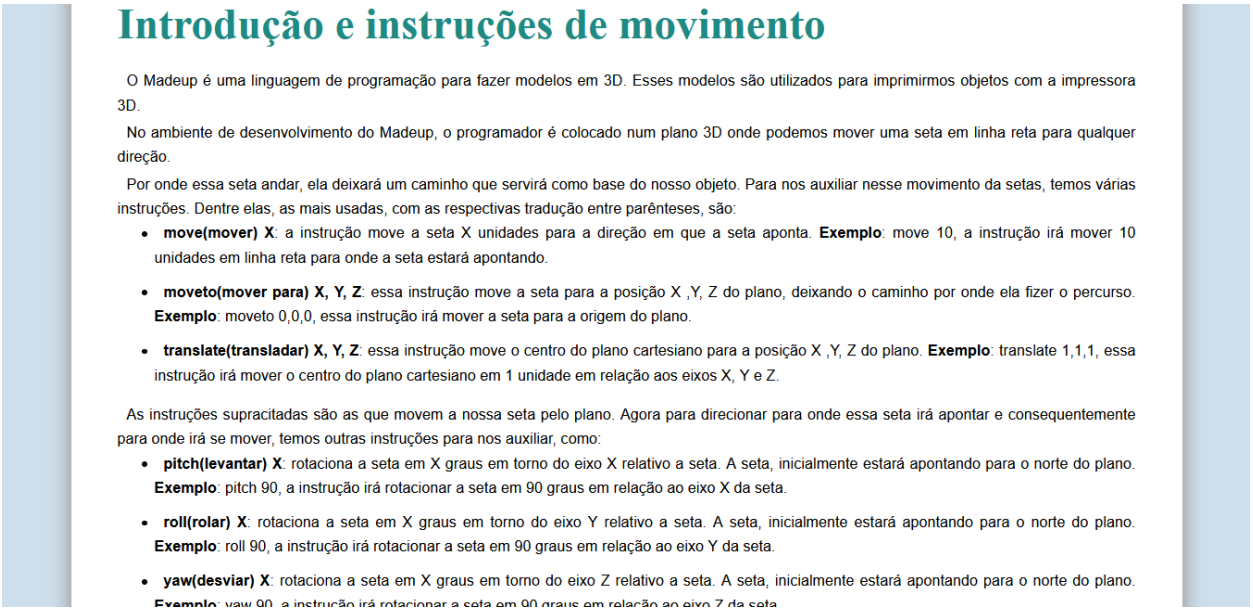
3.2. Curso: conteúdo e mecânica

Para o desenvolvimento do curso, utilizamos como base um curso que já vinha sendo desenvolvido no projeto de extensão sobre impressão em 3D e Internet das Coisas (IoT). Esse curso já utilizava o Madeup como base, porém ele explorava apenas os conceitos do próprio Madeup e alguns conceitos iniciais de programação. A partir do curso inicial, os módulos foram alterados para conter o Madeup gamificado e também para melhorar a descrição dos conceitos, os conceitos de variáveis e condicionais, que eram mostrados num mesmo módulo, foram separados em módulos individuais e acrescentamos um módulo para subtração de objetos. Além disso, o curso foi feito também para a linguagem em blocos do Madeup.

Então, nesse novo curso aprimorado, temos três módulos, para ensinar conceitos relativos ao Madeup, programação e modelagem 3D, respectivamente.

O primeiro módulo do curso dá enfoque à modelagem 3D da ferramenta, ensinando os conceitos de movimentação, solidificação e variáveis globais. A implementação do módulo 1 do curso é mostrada entre as Figuras 3.6 e 3.8, organizando-se em três sub-módulos. No sub-módulo 1.1, é gerado um objeto abstrato, cujo o único intuito é demonstrar os comando mais básicos de movimentação. Ao fim desse módulo será aplicado um exercício para os alunos criarem uma

Figura 3.6. Módulo 1.1 do curso



Introdução e instruções de movimento

O Madeup é uma linguagem de programação para fazer modelos em 3D. Esses modelos são utilizados para imprimirmos objetos com a impressora 3D.

No ambiente de desenvolvimento do Madeup, o programador é colocado num plano 3D onde podemos mover uma seta em linha reta para qualquer direção.

Por onde essa seta andar, ela deixará um caminho que servirá como base do nosso objeto. Para nos auxiliar nesse movimento da setas, temos várias instruções. Dentre elas, as mais usadas, com as respectivas tradução entre parênteses, são:

- **move(mover) X**: a instrução move a seta X unidades para a direção em que a seta aponta. **Exemplo**: move 10, a instrução irá mover 10 unidades em linha reta para onde a seta estará apontando.
- **moveto(mover para) X, Y, Z**: essa instrução move a seta para a posição X ,Y, Z do plano, deixando o caminho por onde ela fizer o percurso. **Exemplo**: moveto 0,0,0, essa instrução irá mover a seta para a origem do plano.
- **translate(transladar) X, Y, Z**: essa instrução move o centro do plano cartesiano para a posição X ,Y, Z do plano. **Exemplo**: translate 1,1,1, essa instrução irá mover o centro do plano cartesiano em 1 unidade em relação aos eixos X, Y e Z.

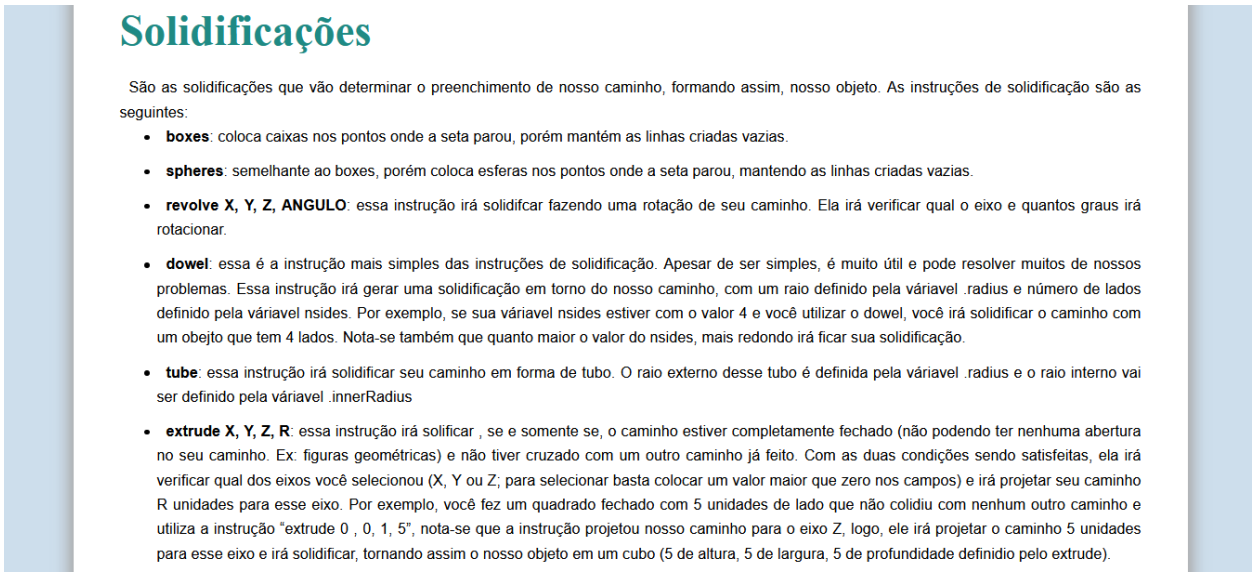
As instruções supracitadas são as que movem a nossa seta pelo plano. Agora para direcionar para onde essa seta irá apontar e consequentemente para onde irá se mover, temos outras instruções para nos auxiliar, como:

- **pitch(levantar) X**: rotaciona a seta em X graus em torno do eixo X relativo a seta. A seta, inicialmente estará apontando para o norte do plano. **Exemplo**: pitch 90, a instrução irá rotacionar a seta em 90 graus em relação ao eixo X da seta.
- **roll(rolar) X**: rotaciona a seta em X graus em torno do eixo Y relativo a seta. A seta, inicialmente estará apontando para o norte do plano. **Exemplo**: roll 90, a instrução irá rotacionar a seta em 90 graus em relação ao eixo Y da seta.
- **yaw(desviar) X**: rotaciona a seta em X graus em torno do eixo Z relativo a seta. A seta, inicialmente estará apontando para o norte do plano. **Exemplo**: yaw 90, a instrução irá rotacionar a seta em 90 graus em relação ao eixo Z da seta.

Fonte: Autoria própria.

pirâmide usando os comandos de movimentação, sem solidificar o objetos. No sub-módulo 1.2, são gerados alguns objetos, para mostrar os diferentes tipos de solidificadores existentes na ferramenta. Após esse módulo, novamente o exercício de montagem da pirâmide será aplicado, porém com os solidificadores inclusos, para mostrar o auxílio que eles oferecem. Aqui será completada a missão de criação do primeiro objeto. No sub-módulo 1.3 são gerados quatro objetos, cada um tendo uma variável global alterada, mostrando as alterações que os objetos sofrem ao se alterar esses atributos. Nesse módulo a missão de criação de objeto usando variáveis globais será concluída. O módulo 1 faz uso dos conceitos de modelagem 3D, sendo no sub-módulo 1.1 apresentado o conceito de layout da cena, com os comandos de movimentação, no sub-módulo 1.2 o conceito de

Figura 3.7. Módulo 1.2 do curso



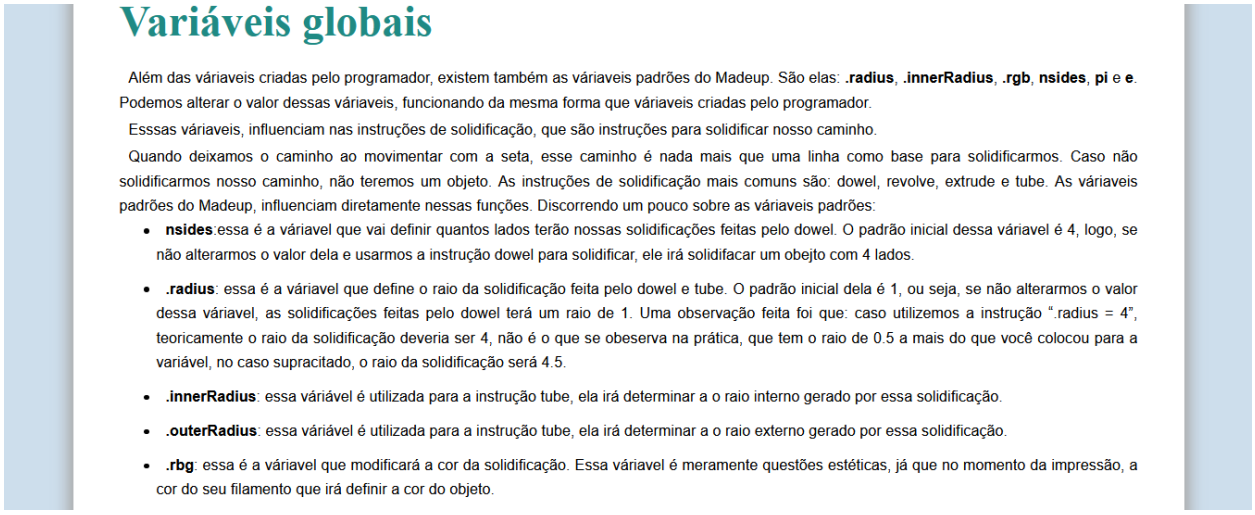
Solidificações

São as solidificações que vão determinar o preenchimento de nosso caminho, formando assim, nosso objeto. As instruções de solidificação são as seguintes:

- **boxes**: coloca caixas nos pontos onde a seta parou, porém mantém as linhas criadas vazias.
- **spheres**: semelhante ao boxes, porém coloca esferas nos pontos onde a seta parou, mantendo as linhas criadas vazias.
- **revolve X, Y, Z, ANGULO**: essa instrução irá solidifcar fazendo uma rotação de seu caminho. Ela irá verificar qual o eixo e quantos graus irá rotacionar.
- **dowel**: essa é a instrução mais simples das instruções de solidificação. Apesar de ser simples, é muito útil e pode resolver muitos de nossos problemas. Essa instrução irá gerar uma solidificação em torno do nosso caminho, com um raio definido pela váriavel .radius e número de lados definido pela váriavel nsides. Por exemplo, se sua váriavel nsides estiver com o valor 4 e você utilizar o dowel, você irá solidificar o caminho com um obejto que tem 4 lados. Nota-se também que quanto maior o valor do nsides, mais redondo irá ficar sua solidificação.
- **tube**: essa instrução irá solidificar seu caminho em forma de tubo. O raio externo desse tubo é definida pela váriavel .radius e o raio interno vai ser definido pela váriavel .innerRadius
- **extrude X, Y, Z, R**: essa instrução irá solificar , se e somente se, o caminho estiver completamente fechado (não podendo ter nenhuma abertura no seu caminho. Ex: figuras geométricas) e não tiver cruzado com um outro caminho já feito. Com as duas condições sendo satisfeitas, ela irá verificar qual dos eixos você selecionou (X, Y ou Z; para selecionar basta colocar um valor maior que zero nos campos) e irá projetar seu caminho R unidades para esse eixo. Por exemplo, você fez um quadrado fechado com 5 unidades de lado que não colidiu com nenhum outro caminho e utiliza a instrução "extrude 0 , 0, 1, 5", nota-se que a instrução projetou nosso caminho para o eixo Z, logo, ele irá projetar o caminho 5 unidades para esse eixo e irá solidificar, tornando assim o nosso objeto em um cubo (5 de altura, 5 de largura, 5 de profundidade definidio pelo extrude).

Fonte: Autoria própria.

Figura 3.8. Módulo 1.3 do curso



Variáveis globais

Além das váriaveis criadas pelo programador, existem também as váriaveis padrões do Madeup. São elas: **.radius**, **.innerRadius**, **.rgb**, **nsides**, **pi** e **e**. Podemos alterar o valor dessas váriaveis, funcionando da mesma forma que váriaveis criadas pelo programador.

Esssas váriaveis, influenciam nas instruções de solidificação, que são instruções para solidificar nosso caminho.

Quando deixamos o caminho ao movimentar com a seta, esse caminho é nada mais que uma linha como base para solidificarmos. Caso não solidificarmos nosso caminho, não teremos um objeto. As instruções de solidificação mais comuns são: dowel, revolve, extrude e tube. As váriaveis padrões do Madeup, influenciam diretamente nessas funções. Discorrendo um pouco sobre as váriaveis padrões:

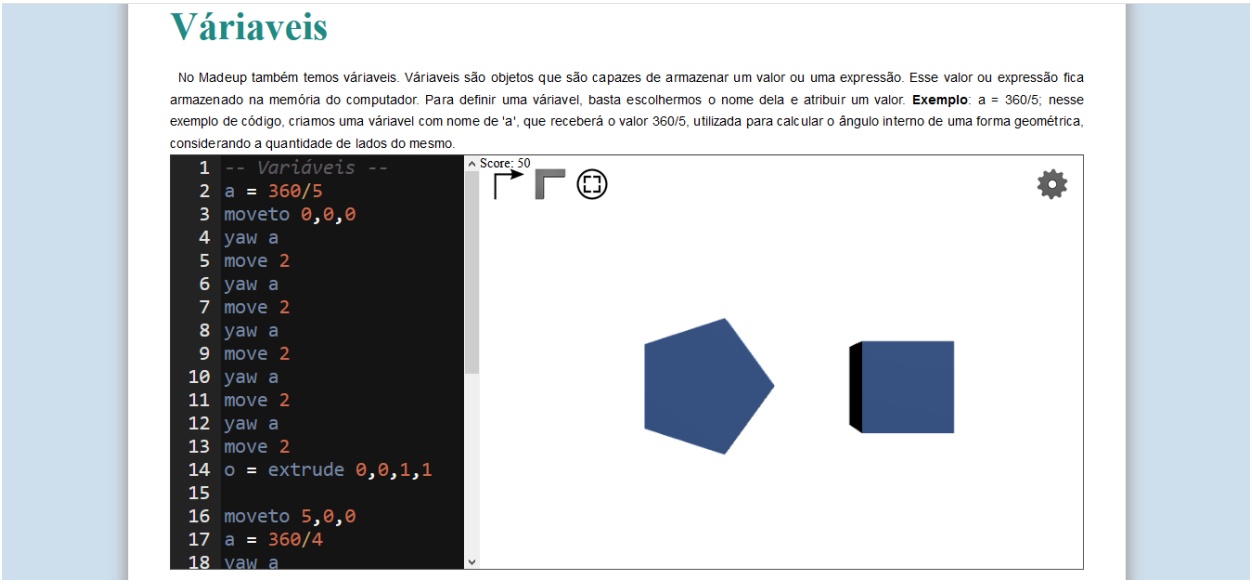
- **nsides**:essa é a váriavel que vai definir quantos lados terão nossas solidificações feitas pelo dowel. O padrão inicial dessa váriavel é 4, logo, se não alterarmos o valor dela e usarmos a instrução dowel para solidificar, ele irá solidifacar um obejto com 4 lados.
- **.radius**: essa é a váriavel que define o raio da solidificação feita pelo dowel e tube. O padrão inicial dela é 1, ou seja, se não alterarmos o valor dessa váriavel, as solidificações feitas pelo dowel terá um raio de 1. Uma observação feita foi que: caso utilizemos a instrução ".radius = 4", teoricamente o raio da solidificação deveria ser 4, não é o que se obeserva na prática, que tem o raio de 0.5 a mais do que você colocou para a variável, no caso supracitado, o raio da solidificação será 4.5.
- **.innerRadius**: essa váriável é utilizada para a instrução tube, ela irá determinar a o raio interno gerado por essa solidificação.
- **.outerRadius**: essa váriável é utilizada para a instrução tube, ela irá determinar a o raio externo gerado por essa solidificação.
- **.rbg**: essa é a váriavel que modificará a cor da solidificação. Essa váriavel é meramente questões estéticas, já que no momento da impressão, a cor do seu filamento que irá definir a cor do objeto.

Fonte: Autoria própria.

geração de cena com os comandos de solidificação e o sub-módulo 1.3 apresenta o conceito de variáveis globais. Os conceitos mostrados no módulo 1 tem como objetivo ensinar o básico de modelagem 3D realizada no Madeup.

O segundo módulo do curso dá enfoque em conceitos de programação, visando o ensino de conceitos básicos, tais como condicionais e variáveis, laços e funções. No sub-módulo 2.1 temos como exemplo uma caixa redonda aberta, que ao se alterar uma variável de controle fecha a caixa, para mostrar um possível uso de condicionais. Nesse módulo será realizada a missão de utilização de condicionais. No sub-módulo 2.2 geramos uma esfera utilizando um laço de repetição, para mostrar como laços podem diminuir um código. A missão de utilização de laços de repetição será completa nessa parte. No sub-módulo 2.3 temos a mesma esfera, porém com listras coloridas, feitas através de uma função, para mostrar uma possível utilidade para as funções. O exercício sugerido ao fim desses módulos é a criação de uma pirâmide com caixas empilhadas, onde a base da pirâmide tem um número de caixas descrito como parâmetro de uma função, para que os alunos utilizem todos os conceitos aprendidos nesse módulo. Os conceitos mostrados no módulo 2 são focados em programação, sendo o sub-módulo 2.1 voltado para o conceito de variáveis, o sub-módulo 2.2 mostrando o conceito de condicionais(comandos de seleção), no sub-módulo 2.3 apresentando o conceito de laços e no sub-módulo 2.4 o conceito de funções. Os conceitos do módulo 2 servem para introduzir os participantes à programação, para melhorar seus códigos, deixando-os mais sucintos e dinâmicos. Esse módulo é mostrado entre as Figuras 3.9 e 3.12.

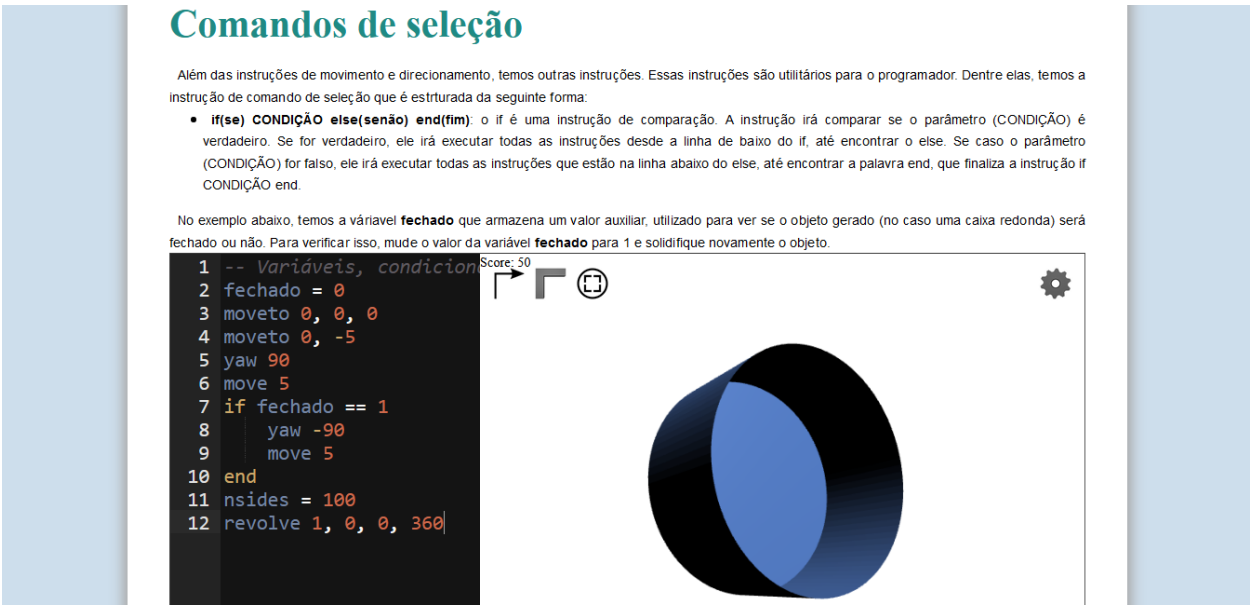
Figura 3.9. Módulo 2.2 do curso



Fonte: Autoria própria.

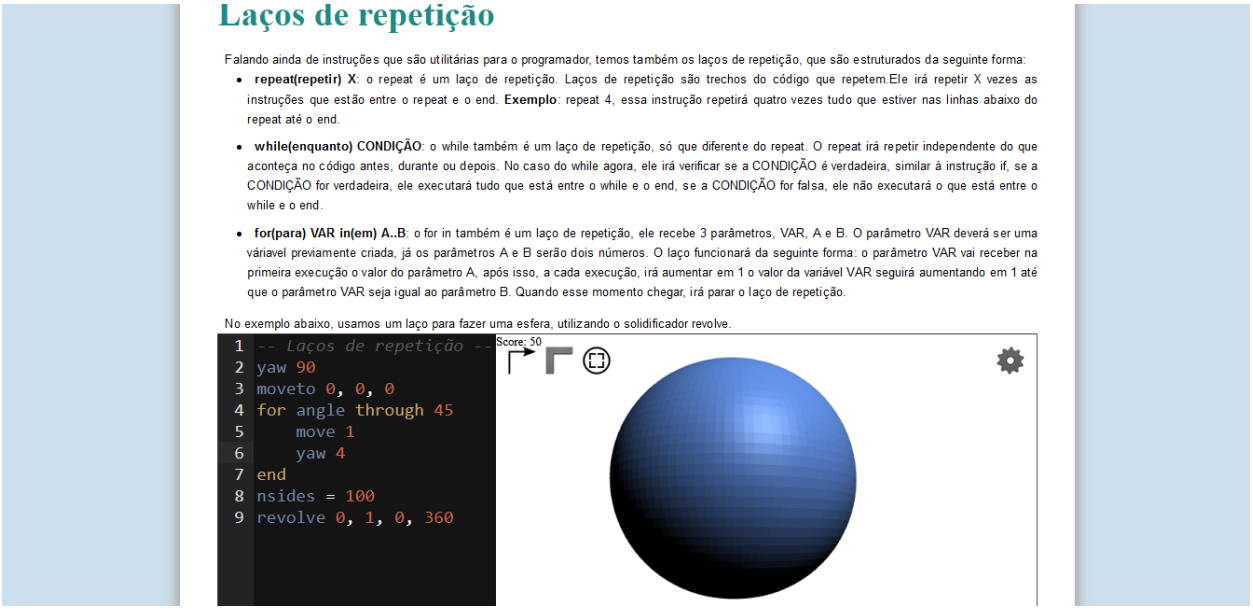
O terceiro e último módulo ensina a subtração de objetos 3D, conforme mostrado na Figura 3.13. Esse módulo possui apenas um sub-módulo, que possui uma parede com um arco no meio, feita através da subtração da parede com o arco. Aqui será completada a penúltima

Figura 3.10. Módulo 2.2 do curso



Fonte: Autoria própria.

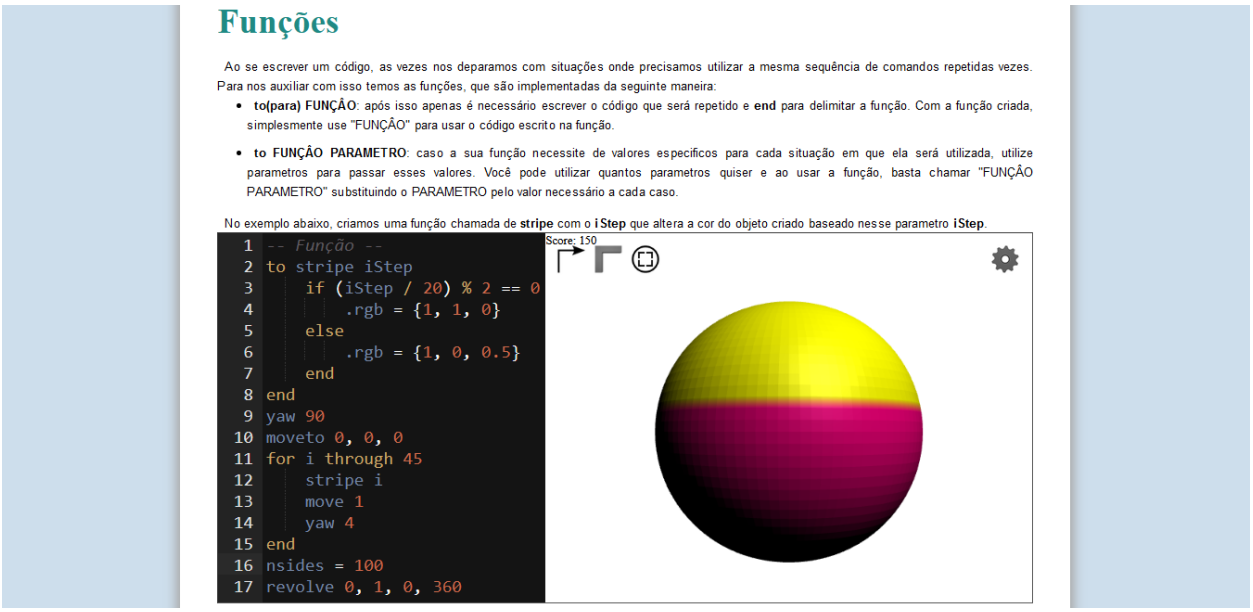
Figura 3.11. Módulo 2.3 do curso



Fonte: Autoria própria.

missão(criação de objeto utilizando subtração de objetos). Com esse material, espera-se que os alunos consigam desenvolver objetos 3D mais complexos, assim como aprender o básico de programação. Assim, o exercício sugerido ao fim desse módulo é livre, para os alunos criarem objetos que lhes interessem, utilizando todos os conceitos apresentados anteriormente, e estimule-os a pensar. No módulo 3, apenas um conceito é apresentado e é o conceito de subtração 3D,

Figura 3.12. Módulo 2.4 do curso



Fonte: Autoria própria.

Figura 3.13. Módulo 3 do curso

Fonte: Autoria própria.

usado para ensinar um método para gerar objetos mais complexos facilitando para os participantes criarem objetos que eles imaginarem.

O curso completo com os três módulos[1] e a ferramenta do Madeup gamificada[2] [3] estão disponíveis abertamente no GitHub.

[1] https://github.com/hackerspace-utfpr-cm/madeup-oer
[2] https://github.com/hackerspace-utfpr-cm/madeup
[3] http://hackerspace.net.br/iot/madeup/

3.3. Implementação de elementos e dinâmicas de jogos no Madeup

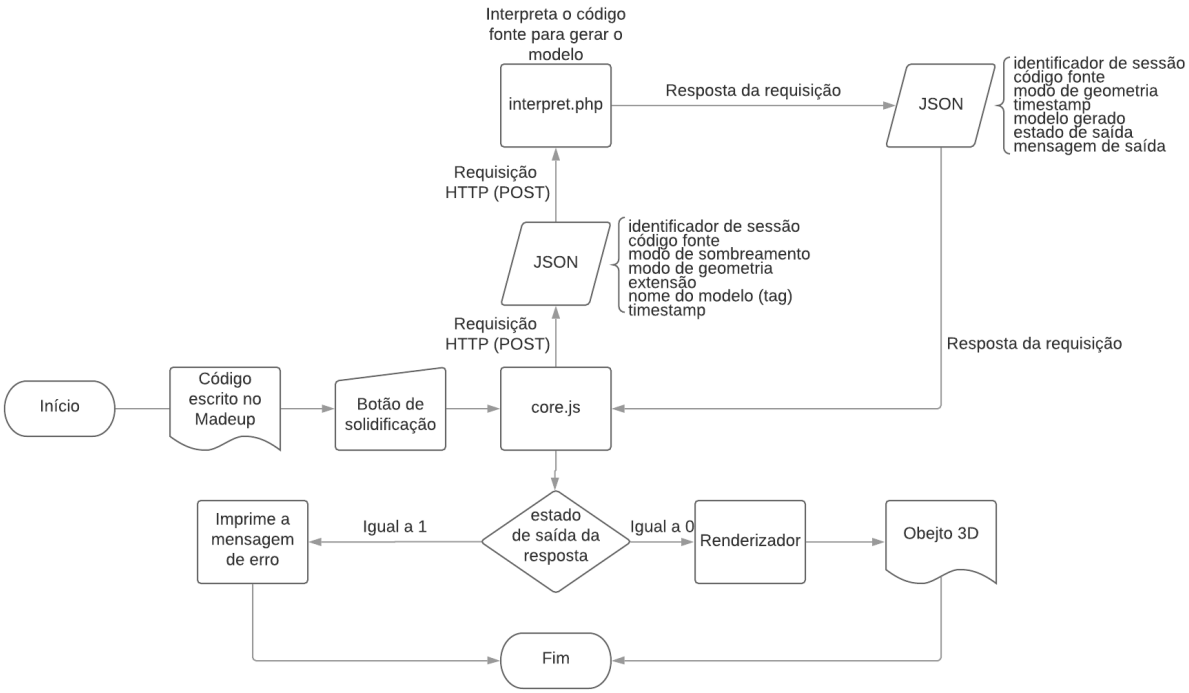
Nesta seção, será discutida a implementação do trabalho, explicando tanto o funcionamento da ferramenta Madeup quanto as modificações realizadas para acrescentar a gamificação na ferramenta.

3.3.1. Madeup

O Madeup é uma aplicação Web que agrega um ambiente de programação para o usuário, mantido no cliente Web e programado em Javascript. Junto a isso, se utiliza de componentes para renderização de modelos 3D, utilizando a biblioteca *THREE*; e para coordenação das chamadas entre o ambiente de programação e o servidor, usando o *AJAX*.

Na parte do servidor, o Madeup trabalha em suma com PHP, utilizando os arquivos PHP para processamento dos dados enviados pelo cliente. No servidor são realizadas operações de interpretação do código fonte para o formato que o renderizador aceita. Para o nosso trabalho, esse processo de interpretação do código da linguagem Madeup para o formato a ser utilizado para criar os objetos 3D é o foco, onde trabalhamos em cima desse fluxo para a realização da gamificação. Esse fluxo, modelado no fluxograma da Figura 3.14, segue os passos mostrados a

Figura 3.14. Fluxograma de interpretação de objetos no Madeup



Fonte: Autoria própria.

seguir. Um aluno escreve um código no Madeup e clica no botão de solidificação. Com isso, no arquivo core.js, um objeto JSON é criado, contendo os seguintes dados (com nome dos atributos entre parênteses): identificador de sessão (*id*); código fonte (*source*); modo de sombreamento (*shading_mode*) e modo de geometria (*geometry_mode*); extensão (*extension*, deve ser json, stl ou obj); nome do modelo usado para exportar (*tag*) e horário em que a requisição foi submetida (*timestamp*). Esse objeto é enviado via requisição HTTP do tipo POST para o arquivo interpret.php, onde são realizadas as devidas interpretações para passar o código fonte para o formato que o renderizador THREE utiliza para gerar os objetos. O interpretador armazena o processamento dos dados em um arquivo temporário no formato JSON para ao fim, se não ocorrer nenhum erro, serem mandados de volta para o core.js. As informações que são retornadas para o core.js são identificador de sessão (*id*), código fonte (*source*), modo de geometria (*geometry_mode*) e *timestamp*, sem nenhuma modificação, e o modelo gerado (*model*), estado de saída (*exit_status*) e uma mensagem de saída (*stdout*). Ao receber esses dados no core.js, caso o *exit_status* for zero, ou seja, nenhum erro ocorreu na execução, os dados do model são passados para o renderizador para renderizar o objeto. Caso o *exit_status* seja um, a renderização não é realizada e o *stdout* é mostrado no console do Madeup, para indicar o erro ocorrido.

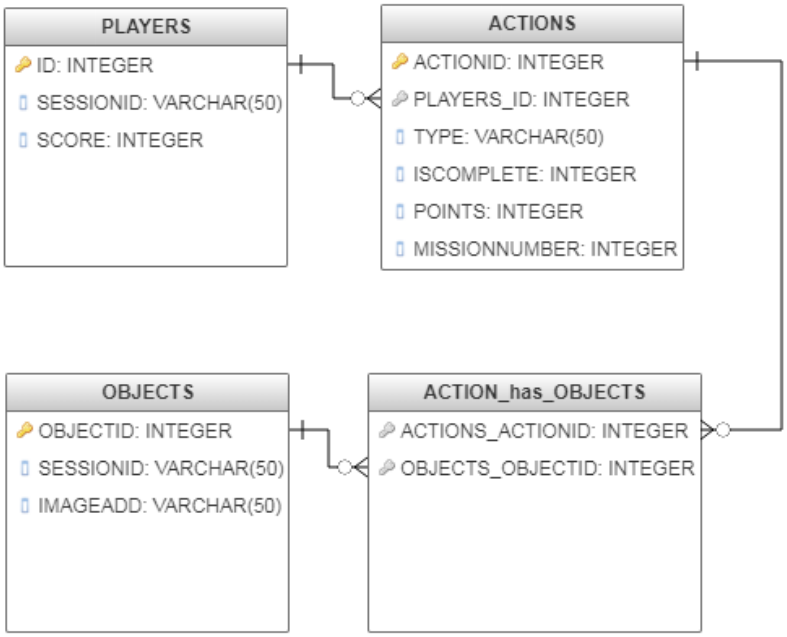
3.3.2. Inclusão de elementos e dinâmicas de jogos no Madeup: Elementos comuns

A gamificação da ferramenta Madeup foi feita através dos arquivos PHP que realizam o *back-end* da mesma. Utilizamos os arquivos de interpretação dos códigos do Madeup para obter as informações necessárias junto a arquivos criados para processamento dessas informações e para criação e acesso a banco de dados. utilizando o SQLite embutido do PHP. O banco criado é mostrado na Figura 3.15. No banco temos quatro tabelas, PLAYERS, ACTIONS, OBJECTS e ACTIONS_has_OBJETCS, sendo a última uma tabela pra armazenar o relacionamento entre as tabelas ACTIONS e OBJECTS. Cada PLAYER tem múltiplas ACTIONS, que podem ser do tipo missão ou voto e ACTIONS do tipo voto são relacionadas a OBJECTS, sendo que um OBJECT pode ter vários votos.

Sempre ao se solidificar uma imagem, é verificado no banco se existe um usuário com o *sessionid* da sessão atual, se não existir, um novo item é criado no banco na tabela **PLAYERS**, com um ID de chave primária, seu *sessionid* e um *score* igual a zero. Em seguida, seis missões são criadas na tabela **ACTIONS**, contendo um *ACTIONID* de chave primária, o ID do usuário, como chave estrangeira da tabela **PLAYERS**; o tipo da ação, que no caso é *mission*; se ela foi completa ou não, no campo *ISCOMPLETE* e o número da missão. Isso é feito pela classe MyDB, descrita no arquivo *database.php*.

Como explicado na subseção anterior, sempre que um objeto é solidificado, seu código é armazenado em um arquivo json, com todas as informações necessárias para sua interpretação pela ferramenta. Para a realização das missões, o fluxo de interpretação mostrado na Figura 3.14 foi

Figura 3.15. Diagrama Relacional do Banco de Dados

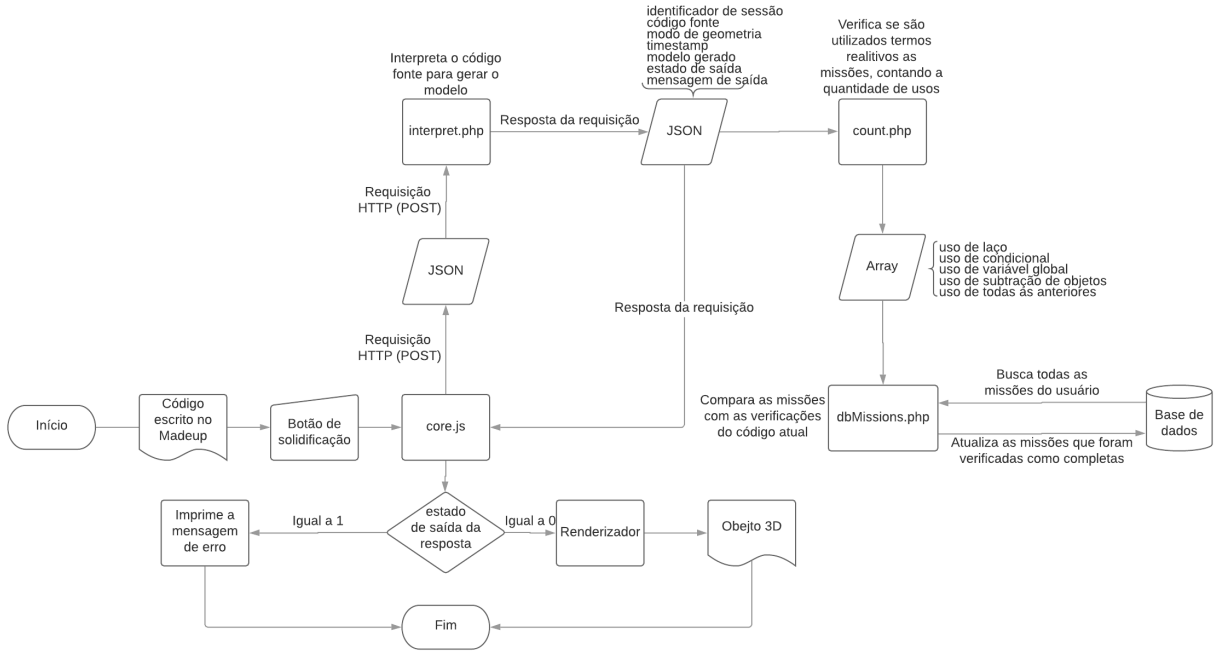


Fonte: Autoria própria.

alterado, de forma à passarmos essas informações para o arquivo *count.php*, para fazer a verificação dos requisitos de cada uma das missões, verificando se é o primeiro objeto que o usuário criou, se são utilizados laços de repetição, condicionais, variáveis globais ou subtração de imagens. Essas verificações são realizadas analisando o código fonte do objeto, contando as utilizações de cada comando no código, verificando se são comandos relativo aos conceitos de laços, aos conceitos de condicionais, se são variáveis globais e/ou se há uma subtração de objetos. Feitas essas verificações, armazena-se no banco se as missões foram completas ou não, através da classe *MissionsDB*, descrita no arquivo *dbMission.php*. Na classe, é verificado se as missões do usuário em questão foram completadas anteriormente, se não, marca a missão como completa, armazena a quantidade de pontos que a missão recompensa e atualiza o *score* do usuário. O fluxo com a adição da verificação de missões ficou como mostra a Figura 3.16

Junto a isso, os objetos são armazenados para serem apresentados numa lista, onde os usuários podem votar em cada objeto, para recompensar o criador do mesmo com pontos. Esses pontos serão utilizados para recompensar o usuário, de forma que com com alguma quantidade de pontos, o usuário possa imprimir um objeto criado. Para salvar o objeto é feita uma requisição jQuery Ajax com o método *POST*, passando o *sessionid* o usuário e a imagem do objeto, obtida através da função *blob* do renderizador *THREE*. Isso é passado para o arquivo *screenshot.php*, que salva a imagem no servidor usando o *sessionid* como nome da imagem. Após isso, outra requisição jQuery é executada, usando o método *GET*, para requisitar uma lista com todos os usuários, utilizada para listar os objetos na ferramenta.

Figura 3.16. Fluxograma de interpretação de objetos no Madeup



Fonte: Autoria própria.

3.4. Avaliação

Para poder avaliar o curso foi desenvolvido um questionário baseado no modelo ARCS, para medir a motivação dos participantes após o término do curso. O questionário conta com trinta e seis questões, com respostas que variam numa escala linear de um a cinco, sendo o um equivalente a concordo e o cinco equivalente a discordo. As questões do modelo ARCS é divido em quatro componentes: Atenção, Relevância, Confiança e Satisfação, que são usados para medir a motivação das pessoas. O questionário criado é mostrado no Apêndice A.

3.5. Considerações finais

Com isso, os recursos educacionais gerados ao fim da pesquisa foram uma ferramenta de modelagem 3D usando programação gamificada, um curso de três módulos, ensinando sobre o uso da ferramenta, conceitos básicos para programação e o conceito de subtração de objetos, para o desenvolvimento de objetos 3D complexos e um questionário de avaliação de motivação seguindo o modelo ARCS que pode ser aplicado após o término do curso.

Capítulo 4

Conclusões

A criação de recursos educacionais envolvem algumas questões, como quais conceitos ensinar, como ensinar, como atrair a atenção dos participante, etc. Definido isso, gerar os recursos considerando todas essas coisas se mostra desafiador, visto que neste trabalho temos alguns conceitos que se diferenciam bastante. Escolhemos os conceitos que seriam priorizados analisando a possibilidades do Madeup e quais conceitos de programação poderiam ser usados. Em seguida separamos desses conceitos os que julgamos essenciais para iniciantes para finalmente uni-los com os conceitos específicos do Madeup, conforme mostrado nos mapas conceituais da Seção 3.1.1.

Em relação a questão de atrair e manter a atenção dos estudantes, a união da gamificação à impressão 3D já existente no Madeup foi uma escolha feita visto os resultados da utilização de ambos em projetos educacionais, individualmente (STANSELL; TYLER-WOOD, 2016; WANG et al., 2016; MORRISON; DISALVO, 2014; IBÁÑEZ et al., 2014; JOHNSON, 2017; JOHNSON et al., 2016b). Porém unir esses conceitos não é algo trivial de se fazer, sendo que cada um tem suas peculiaridades e nem sempre elas convergem para um mesmo caminho. Um exemplo disso foi a tentativa de integração de enredo nos módulos do curso, um dos mecanismos de gamificação que seria incluído. A principal vantagem do enredo seria gerar uma história englobando o curso para atrair os participantes, porém, como o nosso foco em utilizar a impressão 3D é ter uma resposta visual do código e liberar a criatividade dos alunos para a geração de objetos, prendê-los em uma narrativa fechada poderia limitar a liberdade de criação dos mesmos, o que não seria vantajoso para o que visamos conseguir. O uso de missões é uma solução que concilia o suporte e guia permitido em enredos, mas com uma dimensão menor, com a liberdade que desejávamos para a liberdade de criação.

Com isso, a criação dos recursos foi concluída, atendendo os objetivos estabelecidos, tendo em vista que foram gerados os dois recursos educacionais, sendo eles a modificação da ferramenta de modelagem 3D usando programação para ter um ambiente gamificado e a criação de um curso de ensino de programação utilizando a ferramenta modificada. Além disso, um

questionário foi criado para ser utilizado com avaliação da aplicação dos recursos educacionais futuramente.

Como objetivo inicial deste trabalho tínhamos também a realização de uma pesquisa com alunos de ensino médio e técnico para avaliar a utilização de impressão em 3D em conjunto a elementos de gamificação para o ensino introdutório de computação. Entretanto, não foi possível a realização das atividades com os alunos. Como trabalho futuro, buscaremos a realização de ações de pesquisa e extensão, visando a utilização, avaliação e aprimoramento dos recursos educacionais desenvolvidos neste trabalho.

Além disso, uma pesquisa está sendo realizada para identificar estudos que utilizam modelagem e impressão 3D no ensino de programação, avaliando se modelagem e impressão 3D causam um efeito positivo na motivação para o ensino de computação e no alcance das metas de aprendizado para a educação em computação. Desta forma, somando essas evidências com aquelas previstas com a realização dos estudos experimentais e de extensão com os resultados deste trabalho, espera-se proporcionar evidências e meios para o ensino de programação de forma mais atrativa e eficiente.

Referências

ASTRACHAN, Owen; OSBORNE, R Brook. Advanced placement computer science principles (APCSP): A report from teachers. In: ACM, Memphis, TN, EUA. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. New York, NY, EUA, 2016. p. 681–682.

GROSS, Mark D. A little programming language for generating three-dimensional form algorithmically. *CAAD Futures. Eindhoven University of Technology, Eindhoven*, p. 8–10, 2001.

IBÁÑEZ, María-Blanca; DI-SERIO, Angela; DELGADO-KLOOS, Carlos. Gamification for engaging computer science students in learning activities: A case study. *IEEE Transactions on Learning Technologies*, IEEE, v. 7, n. 3, p. 291–301, 2014.

JOHNSON, Chris. Toward computational making with Madeup. In: ACM, Seattle, Washington, EUA. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. New York, NY, EUA, 2017. p. 297–302.

JOHNSON, Chris; AMTHAUER, Heather; HARDT, Ryan; BUI, Peter. Mixing code and 3D printers with Madeup. In: ACM, Memphis, TN, EUA. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. New York, NY, EUA, 2016. p. 721–721.

JOHNSON, Chris; AMTHAUER, Heather; HARDT, Ryan; BUI, Peter. Mixing code and 3D printers with Madeup. In: *47th ACM Technical Symposium on Computing Science Education*. New York, NY, EUA: ACM, 2016. p. 721–721. ISBN 978-1-4503-3685-7.

JOHNSON, Chris; BUI, Peter. Blocks in, blocks out: A language for 3D models. In: IEEE, Atlanta, GA, EUA. *2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond)*. New York, NY, EUA, 2015. p. 77–82.

JOHNSON, Chris; BUI, Peter. Madeup: A language for making things up. In: ACM, Kansas City, Missouri, EUA. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. New York, NY, EUA, 2015. p. 684–684.

KINTEL, Marius; WOLF, Clifford. *OpenSCAD, The Programmers Solid 3D CAD Modeller*. 2011.

KOSCHITZ, Duks; ROSENBAUM, Eric. Exploring algorithmic geometry with 'Beetle Blocks:'a graphical programming language for generating 3D forms. In: *Proceedings of the 15 th International Conference on Geometry and Graphics*. Red Hook, NY, EUA: Curran, 2012. p. 380–389.

MORRISON, Briana B; DISALVO, Betsy. Khan academy gamifies computer science. In: *Proceedings of the 45th ACM technical symposium on Computer science education*. New York, NY, EUA: ACM, 2014. p. 39–44.

RESNICK, Mitchel; MALONEY, John; MONROY-HERNáNDEZ, Andrés; RUSK, Natalie; EASTMOND, Evelyn; BRENNAN, Karen; MILLNER, Amon; ROSENBAUM, Eric; SILVER, Jay; SILVERMAN, Brian; KAFAI, Yasmin. Scratch: Programming for all. *Communications of the ACM*, ACM, New York, NY,EUA, v. 52, n. 11, p. 60–67, nov. 2009. ISSN 0001-0782.

SBC. *Educação Superior em Computação Estatísticas*. 2015.

SNOW, Eric; RUTSTEIN, Daisy; BIENKOWSKI, Marie; XU, Yuning. Principled assessment of student learning in high school computer science. In: *Proceedings of the 2017 ACM Conference on International Computing Education Research*. New York, NY, EUA: ACM, 2017. p. 209–216.

STANSELL, Alicia; TYLER-WOOD, Tandra. Digital fabrication for STEM projects: A middle school example. In: *2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT)*. New York, NY, EUA: IEEE, 2016. p. 483–485.

WANG, Haonan; ZHOU, Chun; WU, Yonghe. Smart cup, wisdom creation: A project-based learning initiative for maker education. In: *2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT)*. New York, NY, EUA: IEEE, 2016. p. 486–488.

Apêndices

APÊNDICE

A

Questionário ARCS

Nesse apêndice estão em anexos o questionário elaborado para aplicação após o uso dos recursos educacionais. O questionário baseado no modelo ARCS tem as seguintes questões:

1. Quando olhei pela primeira vez para o curso, tive a impressão de que seria fácil para mim.
2. Havia algo interessante no começo deste curso que chamou minha atenção.
3. Esse material era mais difícil de entender do que eu gostaria que fosse.
4. Depois de ler as informações introdutórias, me senti confiante de que sabia o que deveria aprender com esse curso.
5. Completar os exercícios deste curso me deu um sentimento satisfatório de realização.
6. É claro para mim como o conteúdo deste material está relacionado a coisas que eu já conheço.
7. Muitas das páginas tinham tantas informações que era difícil escolher e lembrar os pontos importantes.
8. Estes materiais são atraentes.
9. Havia histórias, fotos ou exemplos que me mostraram como esse material pode ser importante para algumas pessoas.
10. Completar este curso com sucesso foi importante para mim.
11. A qualidade da escrita ajudou a prender minha atenção.
12. Essa lição é tão abstrata que era difícil manter minha atenção nela.
13. Enquanto trabalhava neste curso, estava confiante de que poderia aprender o conteúdo.
14. Gostei tanto deste curso que gostaria de saber mais sobre esse assunto.
15. As páginas deste curso parecem secas e desagradáveis.
16. O conteúdo deste material é relevante para os meus interesses.
17. A maneira como as informações são organizadas nas páginas ajudou a manter minha atenção.
18. Existem explicações ou exemplos de como as pessoas usam o conhecimento neste curso.
19. Os exercícios deste curso foram muito difíceis.

20. Este curso tem coisas que estimularam minha curiosidade.
21. Gostei muito de estudar este curso.
22. A quantidade de repetições neste curso me fez ficar entediado às vezes.
23. O conteúdo e o estilo de escrever neste curso transmitem a impressão de que vale a pena conhecer seu conteúdo.
24. Eu aprendi algumas coisas que foram surpreendentes ou inesperadas.
25. Depois de trabalhar neste curso por um tempo, eu estava confiante de que eu seria capaz de passar por um teste nela.
26. Este curso não foi relevante para as minhas necessidades porque eu já sabia a maior parte dela.
27. O feedback após os exercícios, ou de outros comentários nesta lição, me ajudou a sentir recompensado pelo meu esforço.
28. A variedade de passagens de leitura, exercícios, ilustrações, etc., ajudou a manter a atenção na aula.
29. O estilo de escrever é entediante.
30. Consegui relacionar o conteúdo deste curso com coisas que vi, fiz ou pensei em minha própria vida.
31. Há tantas palavras em cada página que é irritante.
32. Foi bom concluir com êxito este curso.
33. O conteúdo deste curso será útil para mim.
34. Eu não conseguia entender muito do material deste curso.
35. A boa organização do conteúdo me ajudou a ter certeza de que aprenderia esse material.
36. Foi um prazer trabalhar em uma aula tão bem projetada.