Desenvolvimento de um algoritmo para interpretação e visualização de equações na plataforma Unity

Development of an algorithm for interpretation and visualization of equations in the Unity platform

Adriano Aragão dos Santos
adrianosantos.snr@gmail.com
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

Eduardo Pezutti Beletato dos Santos
epsantos@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

Edson Tavares de Camargo
edson@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

Heloisa Cristina Silva
heloisasilva@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

Roberto Milton Scheffel
roberto.scheffel@gmail.com
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

RESUMO

Este trabalho foi desenvolvido no contexto do projeto de extensão Desenvolvimento de Jogos Digitais para Ensino de Matemática Básica. Basicamente, jogos elaborados pelos alunos de licenciatura em matemática disponíveis no LEM (Laboratório de Ensino da Matemática) são projetados e programados em uma versão digital usando a plataforma Unity. Uma característica comum aos jogos digitais é a necessidedade de exibir expressões matemáticas em formato amigável. No entanto, a plataforma Unity não possui meio nativo para este fim. Deste modo foi desenvolvido um algoritmo capaz de receber como entrada uma expressão matemática em formato LaTeX e produzir uma imagem correspondente à equação, que pode ser exibida nos jogos. A solução desenvolvida faz parte de um componente essencial para a criação de uma variedade de jogos voltados para o ensino de matemática.

PALAVRAS-CHAVE: Matemática. Unity. Jogos.

ABSTRACT

This work was developed in the context of the extension project “Desenvolvimento de Jogos Digitais para Ensino de Matemática Básica”. Essentially, games prepared by the students of mathematics majors available in the LEM (Laboratório de Ensino da Matemática) are designed and implemented as a digital version utilizing the Unity platform. One common feature of such digital games is the necessity of displaying mathematical expressions. However, the Unity platform does not have native means for such a goal. Thus was developed an algorithm capable of receiving, as input, a mathematical expression in LaTeX form and producing an image according with the expression that may be displayed in the games. The solution developed is part of an essential component for the creation of a variety of games aimed at the teaching of mathematics.

KEYWORDS: Math. Unity. Games.

INTRODUÇÃO

O projeto de extensão intitulado “Desenvolvimento de Jogos Digitais para Ensino da Matemática Básica” tem o objetivo de desenvolver jogos para auxiliar no ensino de matemática básica, uma proposta que busca estimular maior engajamento e interesse dos alunos ao posicionar os conteúdos relevantes em um ambiente já buscado por eles. Os jogos elaborados por alunos da licenciatura em matemática durante seus estágios e outras disciplinas ao longo do curso são programados em formato digital usando a plataforma Unity.

A plataforma Unity foi escolhida devido a um conjunto de fatores, entre os quais, destacam-se: **a)** Permite a publicação dos jogos em múltiplas plataformas, como Windows, Linux e Android; **b)** O alto nível de desenvolvimento e popularidade da plataforma, tendo vários jogos já desenvolvidos em Unity (UNITY TECHNOLOGIES, 2021); **c)** A grande disponibilidade de informações, documentação, extensões e outros elementos que facilitam o desenvolvimento.

Um dos componentes essenciais para jogos de matemática é a exibição de expressões em um formato adequado para o público-alvo. Foi identificado que a plataforma não possui esta funcionalidade e foram consideradas duas possíveis soluções. A primeira, visando minimizar o tempo de programação do jogo, produziria imagens em tempo de desenvolvimento em uma ferramenta externa para integrá-las ao jogo. A segunda opção foi desenvolver um algoritmo próprio dentro da plataforma Unity. A segunda opção foi escolhida ao se considerar os desenvolvimentos futuros do projeto, em que se busca desenvolver uma plataforma que possibilite professores desenvolverem suas próprias atividades com expressões personalizadas, cobrindo conteúdos específicos que possam ser automaticamente integradas aos jogos. Para alcançar este objetivo, seria mais prático utilizar um algoritmo que produzisse as imagens em tempo de execução de acordo com as entradas dos professores.

Nas seções seguintes este artigo apresentará, brevemente, detalhes de como esta implementação foi realizada.

MATERIAIS E MÉTODOS

A plataforma Unity é um ambiente de desenvolvimento de jogos bidimensionais, tridimensionais, de realidade virtual e realidade aumentada que possibilita o rápido desenvolvimento através de cenas que agrupam objetos. Estes utilizam componentes ou *scripts* de código para exibirem algum comportamento desejado.

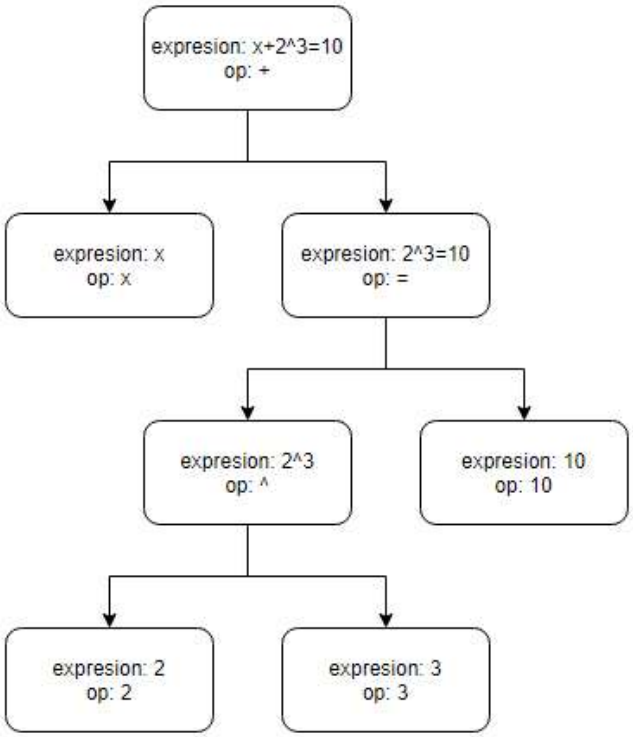
Para programar a exibição de equações, primeiramente foi necessário definir o formato de entrada das expressões. Foi escolhida a utilização do formato de texto em sintaxe LaTeX (THE LATEX PROJECT,

2021) em razão desta ser amplamente utilizada para escrita de artigos científicos e textos com expressões matemáticas. A escolha também visa facilitar a utilização da plataforma pelos professores, pois o LaTex já é amplamente difundido no meio acadêmico.

A seguir, identificou-se a necessidade de dois subcomponentes para desempenhar a tarefa: um subcomponente para interpretar o texto das expressões e produzir uma estrutura de dados para descrevê-la de uma maneira conveniente para o segundo subcomponente. Por sua vez, o segundo componente seria responsável por receber a estrutura da equação matemática, analisá-la e retornar uma imagem para ser utilizada nos jogos. Para o projeto, estes componentes foram denominados Interpretador e Renderizador, respectivamente.

O interpretador foi definido com a seguinte premissa: para produzir uma imagem adequada, é necessário apenas identificar os membros, as funções a que se referem, seus argumentos e os operadores. Ou seja, não seria necessário observar a prioridade de operadores pois apenas precisamos observar o posicionamento e a dimensão de cada elemento na imagem final. Com esta premissa definida, foi projetada uma estrutura de grafo para ser produzida a partir de uma linha de texto contendo uma expressão. De maneira similar à uma árvore (BIANCHI, 2014), seriam produzidos nós para representar os operadores enquanto as folhas representam os parâmetros. Como exemplo, se for recebida a expressão “x+2^3=10”, deve ser produzida a estrutura da Figura 1. Deve-se notar que o algoritmo espera entradas em formato LaTeX, ou seja, argumentos de funções devem ser especificados entre chaves e com marcação adequada, como por exemplo: “\frac{(1+2)}{2}” em vez de “(1+2)/2” para exibir uma fração com parêntesis no numerador.

Figura 1 – Exemplo de estrutura



Fonte: Autor (2021)

A árvore foi construída através de um algoritmo recursivo que, através de um conjunto de expressões regulares, identifica o primeiro membro da expressão, operadores e seus parâmetros e operadores. A seguir o algoritmo separa este membro da expressão total, produzindo os nós e folhas adequadas para o membro indicado. Por fim um último nó é criado e o algoritmo invoca a si mesmo utilizando aquele último nó como raiz e o restante da expressão como parâmetro.

Deste modo, podem ser vistas na Figura 1 as folhas com os argumentos que são números ou variáveis e os nós com os operadores que são soma, subtração, multiplicação, potenciação ou marcações LaTeX de expressões como frações e raízes.

É Importante destacar que o interpretador foi implementado sem base de código, assim a solução atual tem suporte apenas para os tipos de funções que foram necessárias ao longo do projeto, ou seja: soma, subtração, multiplicação, divisão, fracionamento, exponenciação e radiciação. De acordo com estas necessidades, foram identificadas categorias de funções que se referem a quantidade de parâmetros que cada função pode receber e a posição de cada um destes na imagem final. Por exemplo, uma função contendo uma raiz pode receber um ou dois argumentos, sendo estes o índice e o radicando. A estrutura deve diferenciar tal, visto que cada um destes deve ter dimensionamento e posicionamento específico no resultado. Tendo o primeiro subcomponente e suas estruturas produzidas, pode-se implementar o segundo subcomponente, o Renderizador.

Os requisitos Renderizador são o tamanho das fontes de cada argumento e seus posicionamentos. Devem, também, disponibilizar a funcionalidade de escrever símbolos matemáticos não existentes em fontes comuns como o símbolo de raiz.

Para exemplificar o processo, com a expressão “2x+1-\frac{x}{2}=\sqrt{25}+x^2”, é buscada produzir uma textura similar a Figura 2.

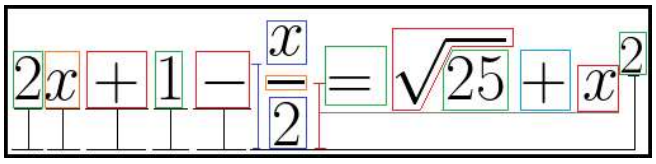
Figura 2 – Imagem a ser produzida

$$2x + 1 - \frac{x}{2} = \sqrt{25} + x^2$$

Fonte: Autor (2021)

Na Figura 3 pode ser observada a mesma expressão com destaque as posições e dimensões que devem ser levadas em consideração para produzir a imagem correta.

Figura 3 – Dimensões e posições a serem consideradas



Fonte: Autor (2021)

Este processo é realizado através de um algoritmo recursivo que busca as folhas da estrutura produzida pelo interpretador. Como mostrado anteriormente, os elementos nestas são as variáveis e os valores literais. O algoritmo então utiliza objetos da plataforma para escrever estes elementos em um câmera oculta que renderiza para uma textura, também oculta, e realiza uma captura de pixels daquele elemento.

Estes dados são salvos na folha correspondente como uma textura 2D e o algoritmo percorre a estrutura a partir das folhas até a raiz, unindo as texturas de acordo com o operador do nó.

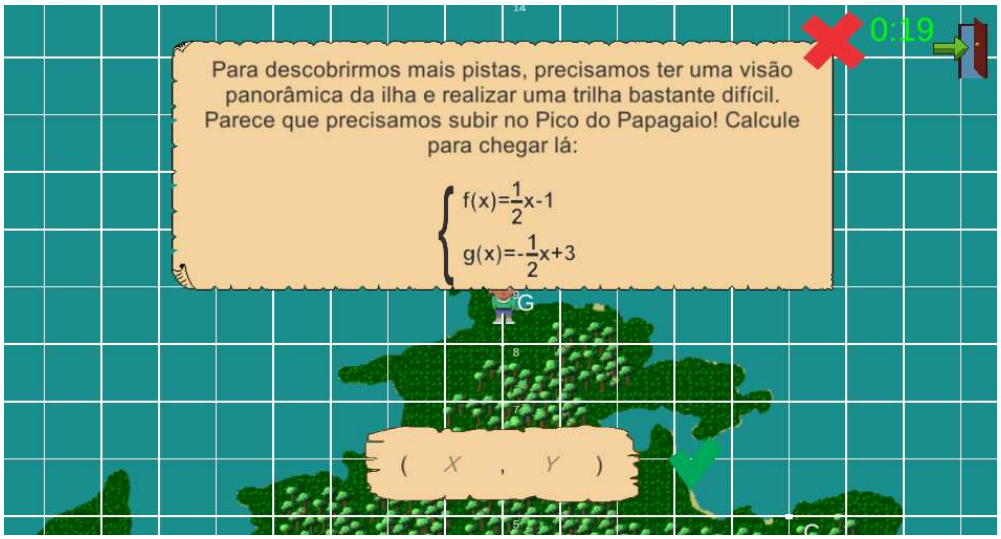
Pode-se detalhar que para operadores simples como soma, subtração, multiplicação ou divisão é realizada uma união simples criando uma textura maior e colocando as menores lado a lado. No entanto para operadores como fração deve-se atentar para posicionar o denominador abaixo e o numerador acima, além disto deve ser desenhada proceduralmente uma barra de comprimento adequada para os argumentos da fração. Questões similares se aplicam para outros operadores como potenciação e radiciação.

RESULTADOS E DISCUSSÕES

Foi produzido um algoritmo que recebe entrada de texto contendo expressões em sintaxe LaTeX e que retorna uma imagem que pode ser utilizada em jogos produzidos na plataforma Unity.

O algoritmo foi consequentemente utilizado, de maneira satisfatória, na implementação do jogo Arquipélago da Matemática, como visto na Figura 4 onde temos uma tela do jogo em que foram utilizadas as expressões “f(x)=/frac{1}{2}2-1” e “g(x)=-/frac{1}{2}x+3” para exibir um sistema linear.

Figura 4 – Tela de jogo com imagens de expressões produzidas pelo componente



Fonte: Autor (2021)

Pode ser destacado este passo do projeto como um dos pilares para o desenvolvimento de uma plataforma educacional e engajante que poderá ser utilizada por professores para criar atividades personalizadas para suas turmas.

O componente desenvolvido neste estágio deste projeto poderá ser reutilizado para todos os jogos desenvolvidos posteriormente, de modo que o processo destes poderá ocorrer de forma mais ágil.

CONCLUSÃO

Este trabalho proporcionou o desenvolvimento de habilidades que foram além do que foi adquirido no curso de Engenharia de Computação. Além disso, contribuiu e realizou os passos iniciais para eventual expansão das ofertas curriculares da matriz. Com iniciativa para disponibilização de cursos de extensão e

ambiente de prática com ferramentas e plataformas gratuitas, pode-se antecipar uma possível evolução para disciplina optativa e de enriquecimento curricular.

Ademais, o projeto segue no processo de produção de uma plataforma de ensino que poderá ser utilizada pela comunidade externa e apresenta oportunidades de estímulo de interesse dos alunos pelo aprendizado.

Um terceiro potencial seria despertar o interesse dos alunos pela universidade ao se destacar que nesta são disponibilizadas atividades para a aprendizagem sobre desenvolvimento e publicação de jogos.

AGRADECIMENTOS

Ficam os agradecimentos aos professores e supervisores que atuaram na concepção e direcionamento do projeto, atuando também na manutenção e promoção das atividades. Agradecemos, também, a Universidade Tecnológica Federal do Paraná por disponibilizar bolsas de estímulo que se tornaram essenciais para o desenvolvimento das atividades.

REFERÊNCIAS

Bianchi, F. Estrutura de Dados e Técnicas de Programação. Grupo GEN, 2014. 9788595152588. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788595152588/. Acesso em: 9 setembro 2021.

The LaTeX Project. LaTeX – A document preparation system. Disponível em https://www.latex-project.org/. Acesso em 9 setembro 2021.

Unity Technologies. +70 case studies from studios & firms using Unity. Disponível em https://unity.com/case-study. Acesso em 9 setembro 2021.