

The Enlightenment Challenge

Protótipo de sistema especialista baseado em técnicas de RPG

para o ensino de matemática e física

Alexandre Zeferino Lima
Universidade Federal do ABC
Santo André, Brasil
alexandre.lima@aluno.ufabc.edu.br

Tiago Henrique Simionato Machado
Universidade Federal do ABC
Santo André, Brasil
tiago.simionato@aluno.ufabc.edu.br

Resumo: O presente trabalho apresenta o desenvolvimento de um protótipo de sistema especialista que visa ensinar matemática e física por meio de um jogo de RPG - Role Playing Game ou Jogo de Interpretação de Personagem, em português. Para alcançar tal objetivo, perguntas são feitas ao jogador, inseridas em um contexto de aventura que o faça aplicar conceitos vistos em sala de aula de modo a superar os desafios. O protótipo foi implementado na linguagem de programação PROLOG, uma linguagem declarativa capaz de trabalhar com regras e fatos.

Palavras-chave: RPG; matemática; física; educação; jogos; sistema especialista; prolog.

I. INTRODUÇÃO

Computação. Esta é uma ferramenta que vem crescendo nas últimas décadas e é uma das principais no mundo contemporâneo. É inegável que a computação está presente na maioria das áreas do conhecimento, seja direta ou indiretamente. Este trabalho busca algo similar: juntar a computação à educação. De forma mais específica, ensinar matemática e física por meio de um jogo de RPG (*Role Playing Game*, jogo de interpretação de personagem), em que são impostas situações que farão com que o jogador tenha que pensar para prosseguir na história, sendo isto feito por meio da interação humano-computador.

A origem da ideia remonta às conversas comuns aos autores: os videogames. Imaginou-se que juntar jogos com o ensino de alguma disciplina poderia incentivar os alunos a participarem, resolvendo problemas em situações de sobrevivência no jogo, aplicando os conceitos até então vistos em sala de aula.

O artigo de título “O Desenvolvimento de um Protótipo de Sistema Especialista Baseado em Técnicas de RPG para o Ensino de Matemática”^[1] é a principal fonte da qual o projeto se baseia. Os autores buscaram a criação de um RPG de forma que os alunos se interessassem no aprendizado da disciplina de matemática, interpretando e resolvendo problemas impostos pela aventura.

Para o desenvolvimento do sistema, a linguagem de programação PROLOG (programação lógica) foi utilizada. Por ser uma linguagem declarativa, funciona sobre uma base de dados e regras que são fornecidas pelo programador e que podem ser consultadas pelo programa e/ou usuário. Para testar o código, foram utilizados os programas SWI-PROLOG e SWISH SWI-PROLOG (este uma plataforma online).

Este projeto deixa em aberto a possibilidade de adição de outras disciplinas às perguntas da aventura, contribuindo para uma maior interdisciplinaridade.

II. LÓGICA PROPOSICIONAL: CONCEITOS FUNDAMENTAIS

Diversos conceitos da lógica proposicional foram utilizados neste projeto. As tabelas abaixo contêm os principais^[2].

TABELA 1: EQUIVALÊNCIAS DA LÓGICA PROPOSICIONAL.

Nome	Equivalência
Leis da Comutatividade	$p \wedge q \equiv q \wedge p$ $p \vee q \equiv q \vee p$

Leis da Idempotência	$p \wedge p \equiv p$ $p \vee p \equiv p$
Leis da Identidade	$p \wedge V \equiv p$ $p \vee F \equiv p$
Leis do Limite Superior	$p \wedge F \equiv F$ $p \vee V \equiv V$
Lei da Contrapositiva	$p \rightarrow q \equiv \sim q \rightarrow \sim p$

TABELA 2: REGRAS DE INFERÊNCIA DA LÓGICA PROPOSICIONAL.

Nome	Premissas	Conclusão
Modus Ponens	$p \rightarrow q, p$	$\vdash q$
Modus Tollens	$p \rightarrow q, \sim q$	$\vdash \sim p$
Introdução da Conjunção	p, q	$\vdash p \wedge q$
Silogismo Disjuntivo	$p \vee q, \sim p$ $p \vee q, \sim q$	$\vdash q$ $\vdash p$

III. LÓGICA DE PREDICADOS: CONCEITOS FUNDAMENTAIS

Diversos conceitos da lógica de predicados foram utilizados neste projeto. A tabela abaixo contém os principais^[3].

TABELA 3: QUANTIFICADORES.

Nome	Equivalência
Enunciado Universal Afirmativo	$\forall X(P(X) \rightarrow Q(X))$
Enunciado Universal Negativo	$\forall X(P(X) \rightarrow \sim Q(X))$
Enunciado Particular Afirmativo	$\exists X(P(X) \wedge Q(X))$
Enunciado Particular Negativo	$\exists X(P(X) \wedge \sim Q(X))$

IV. TRABALHOS RELACIONADOS

A principal fonte de consulta e pesquisa está na dissertação de mestrado “O Desenvolvimento de um Protótipo de Sistema Especialista Baseado em Técnicas de RPG para o Ensino de Matemática”, como já citado. A dissertação mostra como incentivar os alunos a aplicarem o conhecimento obtido em sala de aula de modo divertido, por meio de um jogo de RPG.

A. RELAÇÕES ENTRE OS TRABALHOS

O trabalho acima citado tem foco no ensino de matemática. Este projeto tenta ampliar para mais uma área de conhecimento: a física. Ambos, porém, utilizam o conceito do RPG para alcançar o propósito final, o ensino.

V. BASE DE CONHECIMENTO

B. Descrição Geral

A base de conhecimento deste projeto contém cerca de cem regras, muitas delas interligadas; e os fatos, estes que comportam as respostas das perguntas, a base do trabalho.

C. Fatos

Os fatos presentes no projeto correspondem às respostas das perguntas.

Tais fatos são acessados quando o jogador responde alguma pergunta; uma resposta é dada e há uma regra que verifica se o fato com a resposta dada está no banco de dados. Caso esteja, o jogador acertou a resposta; caso contrário, errou.

TABELA 4: RESUMO DOS FATOS CONTIDOS NA BASE DE CONHECIMENTO.

Fato	Descrição
resposta(X, Y).	Indica que a resposta da pergunta X é Y.

D. Regras

O projeto possui dezenas de regras. Os próximos parágrafos trarão detalhes sobre o que cada regra faz.

- “iniciar”: é a regra responsável por iniciar o sistema. Ela chama as principais regras do programa em sequência;
- “regras”: imprime as regras do jogo para que o jogador possa aproveitar o jogo;
- “introdução”: imprime o início da história, situa o contexto no qual o jogador será inserido;
- “acontecimentos”: esta regra organiza a sequências de eventos que ocorrerão no decorrer da história;
- “encerramento”: imprime os créditos;
- “dados_jogador(Nome)”: coleta o nome do jogador;

- “classes(Classe)”: coleta a informação de qual classe o jogador escolheu;
- “enunciado(X)”: imprime os textos que contém as perguntas que serão feitas ao jogador. Possuem a seguinte estrutura: “enunciado(número da pergunta)”;
- “evento(X)”: imprime textos que ocorrem após dada uma resposta certa ou entre as salas do jogo. Possuem a seguinte estrutura: “evento(nº da questão)”;
- “caminho(X, Y)”: imprime textos que dão ao jogador uma opção de caminho, esta que será captada pela função/regra “acontecimentos”. Possui a seguinte estrutura: “caminho(nº do caminho, resposta do jogador)”;
- “escolha(X, Y, Z)”: imprime textos que dão ao jogador uma opção de escolha, esta que será captada pela função/regra “acontecimentos”. Contribui para decidir qual será o fim de jogo do jogador. Possui a seguinte estrutura: “escolha(nº da escolha, resposta do jogador)”;
- “consequencia(X, Y)”: imprime textos que dão as consequências baseadas nas decisões do jogador. Trabalha em conjunto com a função/regra “escolha”. Possui a seguinte estrutura: “consequencia(nº da escolha, resposta do jogador)”;
- “final(X)”: imprime textos que dizem ao jogador o final do jogo. Possui a seguinte estrutura: “final(nº do final de jogo)”. O final do jogo varia conforme as decisões do jogador. Um dos finais é possível de ser desbloqueado exclusivamente pelos jogadores que optarem por jogar com a classe cientista;
- “dica(X)”: imprime uma dica relacionada à questão X.
- “vidas(X)”: regra que atua como sinalizador de vidas. Indica se o jogador tem todas as vidas, se perdeu alguma vida ou se está sem vidas;
- “perde(X)”: imprime textos encadeados que fazem com que o jogador perca vidas. Dificilmente perdas de vidas serão iguais entre salas. Seguem a seguinte estrutura: “perde(nº do tipo de perda de vida)”;
- “pergunta(Y, Classe)”: chama a regra(Y). Utiliza também a classe escolhida pelo jogador para ativar o bônus de classe conforme a situação;
- “vidas(Y, V, K, Classe)”: sistema de vidas, recolhe a resposta do jogador e verifica se está correta. Caso sim, a história prossegue; caso não, o jogador perde uma vida. Y corresponde ao número da pergunta, V ao número de vidas, K indica se o jogador quer ou não uma dica;
- “checa_resposta(N, X, D, V, NV, Classe, K)”: verifica se a resposta dada pelo jogador está certa ou errada. N corresponde ao número da pergunta, X à resposta da pergunta, D indica se o jogador quer ou não uma dica, V indica o número de vidas atual, NV corresponde ao sinalizador de vida e K ao tipo de perda de vida;

- “checa_dica(Y, D, Classe)”: exibe uma dica para o jogador. Caso a classe escolhida tenha sido ciborgue, uma dica extra é exibida. Y corresponde ao número da pergunta e D indica se o jogador quer ou não uma dica;
- “perde_vidas(Y, K)”: sistema que escolhe de forma aleatória o modo como o jogador perderá uma vida. Y corresponde ao número da questão e K o tipo de perda de vida que será posteriormente utilizado na regra perde(X);
- “decide_final(E1, E2, E4, F)”: regra que decide o final do jogo baseado nas decisões E1 e E4 do jogador.

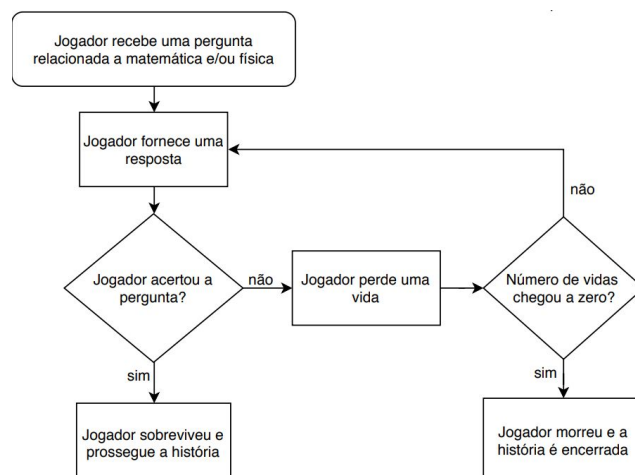
VI. OPERAÇÃO DO SISTEMA

O programa é iniciado ao chamar a regra “iniciar”. Em seguida, pede-se o nome do jogador. A regra “introdução” é chamada, imprimindo o início da aventura. Logo após o breve começo, o jogador deve tomar sua primeira decisão: escolher uma de três classes disponíveis, sendo elas o Cientista, o Ciborgue ou o Androide. Cada classe tem um bônus diferente. O cientista tem a possibilidade de seguir diferentes caminhos no rumo da história; o ciborgue pode receber uma dica extra por pergunta; e o androide, por sua vez, tem uma vida a mais que as demais classes.

Escolhida a classe, o jogador inicia sua jornada na aventura. O sistema de progressão é simples: ao longo da história o jogador toma decisões, e estas levam às perguntas relacionadas à matemática e física. Ao receber uma pergunta, o jogador pode digitar no campo de respostas a palavra “dica”, que o fornecerá uma dica relacionada à questão ou pode simplesmente responder à pergunta.

Neste momento tem-se o sistema de vidas. O jogador inicia cada questão com três vidas (exceto o jogador que tenha escolhido a classe androide, que tem quatro vidas). O sistema funciona de acordo com o diagrama abaixo:

IMAGEM 1: DIAGRAMA DO SISTEMA DE VIDAS.



Caso o jogador erre um número suficiente de vezes para que sua vida se esgote, um final de jogo será iniciado e, logo em seguida, a aventura chegará ao fim; caso contrário, a

história prosseguirá e as vidas do jogador serão restauradas para próxima questão.

Ao fim da última pergunta, o sistema avaliará as decisões feitas e, baseado em critérios internos, desbloqueará um dos três possíveis finais do jogo.

VII. ESTRUTURA TÉCNICA DO SISTEMA BASEADO EM CONHECIMENTO

O sistema foi criado na linguagem de programação PROLOG. É uma linguagem lógica, portanto, declarativa.

O SWI-PROLOG foi utilizado para os testes do código, além da plataforma online SWISH SWI-PROLOG.

VIII. CONCLUSÕES

O maior propósito deste trabalho foi atendido: ensinar tópicos de matemática e física por meio de um jogo de RPG, colocando o jogador em situações que o fizesse pensar e rever conceitos estudados em sala de aula. Além de uma revisão de sua própria base de conhecimento, o jogador pôde obter dicas durante as perguntas, fazendo com que mais conceitos fosse revisitados.

Para uma maior eficácia de ensino, é preciso a adição de mais elementos das áreas sugeridas de forma que mais conceitos possam ser revistos durante a aventura, sendo esta estendida.

E. Dificuldades Encontradas

A criação da história foi um processo complicado, mas interessante. Escolher um ambiente no qual as perguntas fizessem sentido e deixá-las de uma forma que o jogador quisesse prosseguir na história. Além disso, a barreira da nova linguagem de programação foi importante: aprendemos como trabalhar com os elementos básicos do PROLOG que, por ser diferente das linguagens de programação convencionais, tornou-se um desafio recompensador.

F. Trabalhos Futuros

O projeto deixa em aberto uma extensão para sua história, isto é, pode-se adicionar mais elementos para prolongar a aventura e incluir nesta perguntas relacionadas a outras áreas do conhecimento além de matemática e física.

REFERÊNCIAS

- [1] I. Zuchi, “O Desenvolvimento de um Protótipo de Sistema Especialista Baseado em Técnicas de RPG para o Ensino de Matemática”, <<https://repositorio.ufsc.br/bitstream/handle/123456789/78717/170113.pdf>>, acesso em 30 de novembro de 2019.
- [2] M. G. Bruno Marietto, “L.P. - Equivalências e Regras de Inferência”, <<http://professor.ufabc.edu.br/~graca.marietto/HomePage/LPro-EquivalenciasInferencias.pdf>>, acesso em 13 de dezembro de 2019.
- [3] M. G. Bruno Marietto, “Quantificadores e Regras de Inferência”, <<http://professor.ufabc.edu.br/~graca.marietto/HomePage/QuantificadoresInferencias.pdf>>, acesso em 13 de dezembro de 2019.