

Automata Defense 2.0: reedição de um jogo educacional para apoio em Linguagens Formais e Autômatos

Rômulo César Silva¹, Ricardo Luis Binsfeld², Izaura Maria Carelli¹, Rodrigo Watanabe¹

¹Centro de Engenharia e Ciências Exatas (CECE) – Universidade Estadual do Oeste do Paraná (Unioeste)

²Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)

{romulocearsilva, ricardobins, imcarelli, rodwatanabe}@gmail.com

Abstract. *This paper describes the educational game Automata Defense 2.0 and an experience as teaching resource in Formal Languages and Automata. It also reports the results of usability test and a preliminary evaluation of the pedagogical effectiveness.*

Resumo. *Este artigo descreve o jogo educacional Automata Defense 2.0 e a experiência de sua utilização como apoio pedagógico em Linguagens Formais e Autômatos. São apresentados os resultados de teste de usabilidade e uma avaliação preliminar da sua eficácia pedagógica.*

1. Introdução

A disciplina de Linguagens Formais e Autômatos (LFA) é um dos pilares na formação de um bacharel em Ciência da Computação. Sua aplicação vai desde aspectos teóricos tais como decidibilidade, computabilidade e complexidade até aplicações práticas variadas como construção de compiladores para linguagens de programação e reconhecimento de linguagens naturais [Menezes 2001] [Ramos, Neto and Vega 2009].

A falta de familiaridade dos alunos com a representação formal [Dognini and Raabe 2003] e o grande número de conceitos complexos, abstratos e aparentemente muito similares [Devedzic, Debenham and Popovic 2000] contribuem potencialmente para monotonia e dificuldade de aprendizado.

Para contornar tais dificuldades, têm sido propostas diversas ferramentas pedagógicas para LFA, incluindo simuladores [Chesñevar, Cobo and Yurcik 2003] e sistemas tutores inteligentes [Devedzic, Debenham and Popovic 2000].

Outra forma de auxiliar na aprendizagem é fazer uso de jogos educacionais que proporcionem um ambiente de aprendizado atraente e eficaz, devido aos aspectos lúdico e interativo que um jogo pode oferecer.

Este artigo apresenta o jogo educacional Automata Defense 2.0 e uma avaliação preliminar de sua utilização para apoio na disciplina LFA. O jogo pode ser obtido em <http://www.foz.unioeste.br/~detae>.

2. Trabalhos Relacionados

Java Formal Languages and Automata Package (JFLAP) [JFLAP 2009] é um software educacional desenvolvido em Java para experimentação de diversos tópicos de autômatos e gramáticas, incluindo autômatos finitos determinísticos (AFD), autômatos finitos não determinísticos (AFND), Máquinas de Mealy, Máquinas de Moore, Autômatos com Pilha (AP), Máquina de Turing (MT), expressões regulares, entre outros. A ferramenta possui um editor gráfico para criação e alteração de autômatos e permite visualizar a execução de diversos algoritmos.

O Language Emulator [Vieira, Vieira and Vieira 2004] é uma ferramenta desenvolvida no curso de pós-graduação da Universidade Federal de Minas Gerais (UFMG) com a finalidade de ajudar os estudantes a compreender os conceitos da teoria da computação. Desenvolvido em Java, permite a manipulação de Expressões Regulares, Gramáticas Regulares, Autômatos Finitos (AF), Máquinas de Moore e de Mealy.

O Software para Criação e Teste de Modelos Formais (SCTMF) [Costa, Meneses and Uber 2008] é um ambiente multiplataforma para modelar os formalismos estudados em LFA e testá-los, permitindo aos usuários, por exemplo, fornecer sentenças como entrada, retornando o resultado de aceitação ou rejeição do reconhecimento pelo autômato modelado.

MTSolution [Martinez et al. 2007] é uma ferramenta educativa, visual e interativa implementada em Microsoft Visual C++ .NET que integra autômatos e gramáticas. Baseia-se em uma arquitetura cliente-servidor, permitindo assim que os alunos pratiquem, tanto em um computador pessoal em sua própria casa ou em ambientes de rede tais como laboratórios. MTSolution permite definir e simular AFD, AFN, AP e MT.

O EduLing [Dognini and Raabe 2003] é um software educacional composto por 3 módulos (Tutorial, Experimentação Livre e Desafio) que possibilita o desenvolvimento de atividades práticas sobre a construção de linguagens regulares, seja através de AFDs, AFNDs, expressões regulares ou tabelas de transição. O módulo Tutorial consiste em capítulos compostos de textos com exemplos ilustrativos e questões de múltipla escolha. O módulo Experimentação Livre permite ao aluno especificar linguagens através de expressões regulares ou autômatos finitos (por diagrama de transição ou formato tabular), e visualizar suas representações equivalentes. O módulo Desafio deve ser alimentado previamente pelo professor com questões de problemas propostos e respectivas soluções, sendo os problemas apresentados aleatoriamente para o aluno.

Formal Languages and aUTomata Environment (FLUTE) é um sistema tutor inteligente (STI) criado com o objetivo de tornar a aprendizagem de linguagens formais mais atrativa para os estudantes e mais concretos os conceitos abstratos da disciplina. Seu projeto é baseado em um *framework* que utiliza ontologias para construção de STIs. Cobre uma extensa gama de conceitos sobre LFA. Mais detalhes podem ser encontrados em [Devedzic, Debenham and Popovic 2000].

As ferramentas JFLAP, Language Emulator, SCTMF, MTSolution são basicamente simuladores que permitem criar formalismos (AFDs, APs, ERs, MTs, etc.) e testar seu funcionamento, no entanto não propõem desafios aos usuários.

A ferramenta EduLing funciona como um simulador, porém também permite o cadastro de exercícios pelo professor. Já FLUTE é baseada em lições compostas de tópicos, conceitos, fatos, teoremas, problemas e exercícios propostos relacionados. Nessas ferramentas o desafio consiste em responder às questões ou solucionar os problemas propostos.

3. Automata Defense 2.0

O projeto do jogo Automata Defense foi inspirado em [Prensky 2008], que assinala os estudantes como sendo os melhores desenvolvedores de jogos educacionais, no caso ex-alunos da disciplina LFA. O objetivo principal do jogo é servir de complemento pedagógico em LFA. A versão 1.0 cobre somente o conceito de autômatos finitos determinísticos (AFD). Para ampliação do conteúdo pedagógico, foi feita uma reedição, resultando na versão apresentada neste artigo, que contempla também os tópicos autômatos finitos não determinísticos (AFND) e autômatos com pilha determinísticos (APD).

Desenvolvido em Adobe Flex [ADOBE 2010], o Automata Defense pode ser classificado como um *mini-game* (curta duração) de estratégia, e categoria *tower defense* [Binsfeld et al., 2009]. Diferentemente das ferramentas desenvolvidas para apoio em LFA apresentadas na seção 2, o Automata Defense propõe ao usuário um desafio subjacente: projetar autômatos para vencer o jogo. Assim, o jogo utiliza os aspectos lúdico e interativo para motivar a aprendizagem do conteúdo.

3.1 Elementos do Jogo

Como outros jogos da categoria *tower defense*, o jogo consiste em um tabuleiro com uma entrada e uma saída. No decorrer do jogo, diversas criaturas surgem na entrada e se dirigem à saída. O objetivo do jogador é impedir as criaturas de alcançarem a saída. Para tal, devem ser posicionadas torres capazes de atacar ou retardar os movimentos das criaturas. A Figura 1 exibe a interface do jogo, em que estão assinaladas a entrada e saída do tabuleiro, e algumas torres.

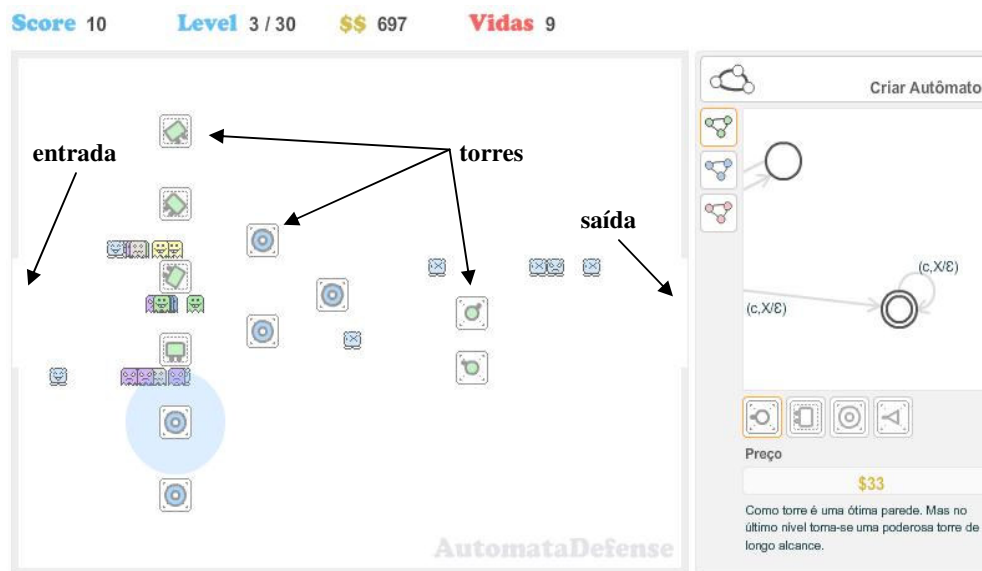


Figura 1. Interface do jogo Automata Defense

No início do jogo, o jogador deve escolher o tópico educacional a ser usado (AFD, AFND ou APD) e em seguida o nível de dificuldade (fácil, médio ou difícil), que determina o intervalo de tempo que as criaturas surgem na entrada do tabuleiro entre uma e outra invasão.

No Automata Defense há 2 (dois) tipos de criaturas: civis e monstros, mostrados nas Figuras 2 e 3, respectivamente. O jogador deve impedir apenas a passagem de monstros, pois a morte de civis implica na perda de *score*, enquanto o abate de monstros implica em ganho. Além disso, inicialmente o jogador possui 20 vidas, que diminuem à medida que monstros conseguem atingir a saída do tabuleiro. Se o número de vidas atingir o valor zero, o jogador perde o jogo.



Figura 2. Exemplos de civis

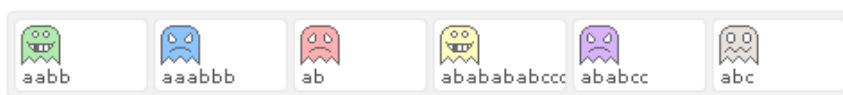


Figura 3. Exemplos de monstros

O Automata Defense associa palavras escolhidas aleatoriamente dentro de um conjunto de linguagens formais fixas (6 regulares e 7 livres de contexto) às criaturas (civis e monstros), sendo que as linguagens de monstros são distintas das linguagens de civis, mas não necessariamente disjuntas.

3.2 Estratégias do Jogo

Cada torre posicionada no tabuleiro é configurada com um autômato. Se uma criatura que tenha palavra reconhecida pelo autômato passa na área de abrangência da torre, a mesma é atacada. A edição do autômato é feita através do editor de autômatos disponível na interface do jogo, mostrado na Figura 4.

O editor de autômatos opera conforme a escolha de conteúdo (AFD, AFND ou APD) feita no início do jogo. No caso de ter sido escolhido AFD ou APD, o editor não permite a inclusão de transições não determinísticas. Além disso, para APD deve ser definido adicionalmente o alfabeto de pilha. O autômato pode ser salvo em memória secundária para posterior carga e utilização.

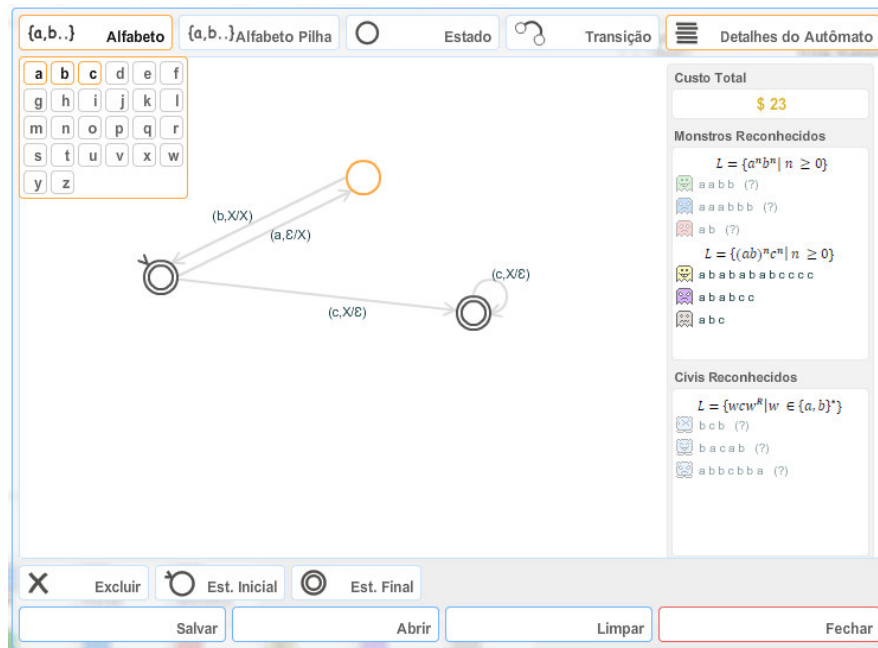


Figura 4. Editor de Autômatos

No projeto da interface foi dado preferência a escrever por extenso os termos alfabeto, estado e transição ao invés de usar símbolos comumente usados em livros da área para que alunos ainda não familiarizados com a notação possam fazer melhor uso do jogo.

A cada jogo, duas linguagens de monstros e uma de civis são escolhidas aleatoriamente dentre as linguagens disponíveis e mostradas junto com exemplos de palavras na interface do editor de autômatos. À medida que o autômato é editado, simultaneamente a interface mostra as palavras que serão reconhecidas.

A criação de autômatos também exige do jogador um raciocínio estratégico, pois cada estado ou transição adicionada implica em um custo a ser pago pelo jogador com dinheiro virtual, levando-o assim a procurar desenvolver um autômato menor. No início do jogo, o jogador tem disponível \$1000 (dinheiro virtual).

Após a criação do autômato, o jogador deve escolher o tipo de torre a ser usada, cujo dano, alcance, e custo variam conforme seu tipo. Durante o jogo, ao selecionar-se uma torre posicionada no tabuleiro com o botão esquerdo do *mouse* é mostrada sua área de abrangência e exibido um painel com informações específicas tais como valor da torre, dano, alcance e preço para *upgrade*, conforme pode ser visto na Figura 5. Nesse painel é possível vender ou realizar o *upgrade* da torre. No caso de venda, o valor atual da torre é creditado na conta do jogador e a mesma é retirada do tabuleiro. Já para o *upgrade*, deve ser pago o custo indicado na interface, e seu efeito é a melhoria dos atributos conforme o tipo da torre (aumento do dano causado, aumento do alcance, etc.).

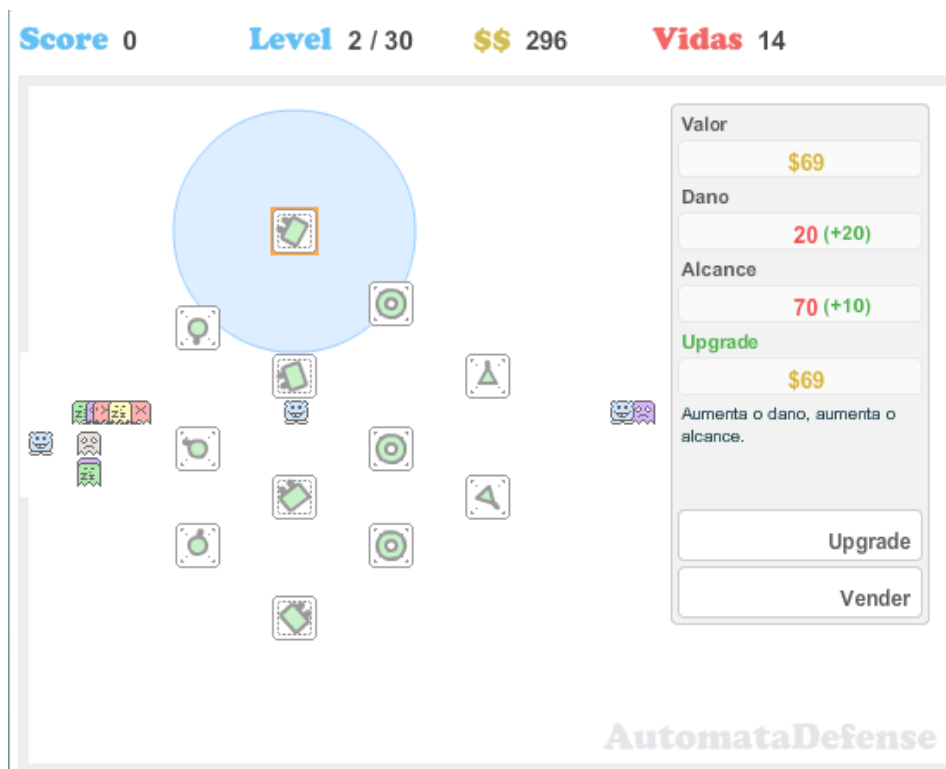


Figura 5. Upgrade/Venda de Torre

4. Uso Pedagógico e Teste de Usabilidade

Foi realizada uma pesquisa sobre o jogo Automata Defense 2.0 usando duas abordagens distintas, porém conjugadas: um teste de usabilidade e uma avaliação preliminar de sua eficácia pedagógica. Para tal, foi passado a cada aluno um termo de ciência e de consentimento livre e esclarecido individual, no qual o mesmo concordava que a participação na pesquisa não implicaria no aproveitamento das notas obtidas nos testes para a disciplina, que o mesmo era livre para abandonar a participação na pesquisa quando o desejasse e não haveria divulgação de informações pessoais.

Para a realização da pesquisa foram utilizados 3 (três) dias de aula seguidos da disciplina, sendo 1(um) dia por semana com 2 horas-aulas de 50 minutos cada, distribuídos da seguinte maneira:

- 1º dia: aplicação de teste pré-sessão (1h 40 min)
- 2º dia: aplicação de questionário sobre perfil do jogador, uso do jogo em laboratório com um computador para cada participante, e aplicação de questionário de avaliação do jogo, nesta ordem (1h e 40 min)
- 3º dia: aplicação de teste pós-sessão (1h 40 min)

Eram 36 alunos matriculados na disciplina LFA (ano-base: 2010) do curso Ciência da Computação da Universidade A, dos quais 31 assinaram o termo de ciência e de consentimento livre e esclarecido. Desses, 26 participaram efetivamente de todas as etapas da pesquisa, sendo os questionários e testes dos demais descartados.

Os testes pré e pós-sessão versavam sobre os tópicos AFD, AFND e questões teóricas gerais tais como critérios de aceitação/rejeição de palavras e relação entre alfabetos e linguagens. Não foram incluídas questões sobre APD porque no momento da aplicação do teste, o referido conteúdo ainda não havia sido ministrado.

Cada teste consistia de 9 questões, sendo 8 objetivas e 1 aberta. As questões 5 (objetiva) e 9 (aberta) pedem ao aluno exatamente o mesmo tipo de raciocínio exigido no jogo: projetar um autômato finito determinístico que reconheça uma dada linguagem.

Foram selecionadas questões comparativamente mais difíceis para o teste pós-sessão em relação ao teste pré-sessão, com exceção da questão 6, mantida idêntica nos dois testes.

Na Tabela 1 são mostradas as linguagens formais usadas nas questões 5 e 9 dos testes pré e pós-sessão.

Tabela 1. Linguagens Formais usadas nas questões 05 e 09 dos testes pré e pós-sessão

	Teste pré-sessão	Teste pós-sessão
Questão 5	$L = \{ w \in \{x,y\}^* \mid w \text{ é ímpar} \}$	$L = \{ w \in \{x,y\}^* \mid w \text{ é ímpar e } \text{qtd}(x) \text{ é par} \}$
Questão 9	$L = \{ w \in \{a,b,c\}^* \mid w \text{ não tem } bca \text{ como subpalavra} \}$	$L = \{ w \in \{a,b,c\}^* \mid w \text{ tem } ac \text{ como subpalavra e } w \text{ é par} \}$

Para a correção da questão 09 (aberta) foram adotados os seguintes critérios:

- AFD projetado corretamente: 1,0 ponto
- Se o AFD projetado tivesse o número correto de estados, mas faltando ou sobrando apenas algumas transições: 0,5 ponto
- Se o AFD projetado tivesse mais ou menos estados: 0,25 ponto
- Múltiplos erros: zero.

A Tabela 2 mostra o número de participantes do total de 26 alunos, que acertaram cada uma das questões dos testes pré e pós-sessão.

Analisando as questões ligadas diretamente ao jogo, vê-se que o índice de acerto da questão 5 (objetiva) aumentou 43,75% no teste pós-sessão. Na questão 9 (aberta) somente 19,23% dos alunos obtiveram alguma pontuação nos 2 (dois) testes.

Tabela 2. Números de acertos por questão

Questões	1	2	3	4	5	6	7	8	9
Pré-sessão	17	25	15	21	16	7	22	18	4(0,5) 1(0,25)
Pós-sessão	17	11	11	18	23	7	19	22	3 (0,5) 2(0,25)

O aumento apenas do índice de acerto da questão 5 (objetiva) e a manutenção praticamente igual da questão 9 (aberta) pode ser justificado pelo uso do jogo por relativo curto espaço de tempo (1h 40min, incluindo o preenchimento dos questionários do teste de usabilidade). O jogo foi usado sem manual de instruções, sendo feita apenas uma rápida explanação sobre seu objetivo pelo professor.

Na Tabela 3 são apresentadas as informações obtidas sobre o perfil dos jogadores: predominantemente indivíduos do sexo masculino, a maioria joga videogame desde os 8 anos de idade, e gastam em média 3,6 horas semanais jogando.

Tabela 3. Perfil dos participantes do teste de usabilidade

Sexo	Masc.: 23 Fem.: 3	
Idade (média)	20 anos	
Joga videogame? Desde que idade?	Sim: 19 Não: 7	Média: 8,6
Horas semanais de jogo (média)	3,6 horas	

Foi solicitando ao participante que fizesse uma auto-avaliação de seu conhecimento sobre LFA, especificamente sobre AFD, AFND e APD, antes e depois de jogar. Também foi pedido que avaliasse se o jogo contribuiu para ampliar o conhecimento sobre o assunto. As Tabelas 4 e 5 sumarizam as respostas.

Tabela 4. Auto-avaliação quanto ao conteúdo de LFA

	Antes de Jogar			Depois de jogar		
	AFD	AFND	APD	AFD	AFND	APD
Ótimo	06	03	zero	07	02	zero
Médio	10	10	03	14	11	4
Básico	10	13	08	03	02	2
Insuficiente ou nenhum	zero	zero	15	zero	zero	2
Não informou	zero	zero	zero	02	11	18

Todos os 26 alunos computados na pesquisa jogaram AFD. 15 jogaram AFND e apenas 5 APD. A grande maioria avaliou que o Automata Defense contribuiu para a compreensão dos conceitos de LFA.

Tabela 5. Auto-avaliação quanto à contribuição pedagógica do jogo

Ampliação do conhecimento	AFD	AFND	APD
Contribuiu muito	2	1	zero
Contribuiu	19	11	4
Contribuiu pouco	2	2	3
Não contribuiu	3	1	2
Não informado	zero	11	17

Na Tabela 6 são apresentados os resultados da avaliação do jogo. A maioria achou-o divertido ou muito divertido. 26,92% dos participantes encontraram algum *bug*.

Além das perguntas objetivas, também foi pedido ao participante que apontasse aspectos positivos/negativos do jogo e sugestões para melhoria da interface e ou conteúdo. Dentre os pontos positivos, os mais apontados foram: a melhor compreensão do conteúdo da disciplina a partir do jogo e o raciocínio estratégico necessário para jogar. Dentre os negativos, o mais apontado foi a ausência das regras do jogo e dificuldade de entender a diferença de funcionalidade entre os botões Limpar e Excluir do editor de autômatos.

Tabela 6. Avaliação do jogo Automata Defense

Gostou de jogar?		Acho divertido?		Interface		Encontrou erro durante a execução?	
Gostei muito	06	Muito divertido	06	Ótima	03	Sim	07
Gostei	14	Divertido	12	Boa	20		
Gostei pouco	05	Pouco divertido	08	Regular	03	Não	19
Não gostei	01	Nada divertido	zero	Insuficiente	zero		
				Péssima	zero		

6. Conclusão e Trabalhos Futuros

A partir da experiência preliminar com o uso do jogo Automata Defense é possível concluir que ele pode ser bastante útil como complementação pedagógica para melhor compreensão dos conceitos de autômatos finitos vistos na disciplina LFA.

A partir do teste de usabilidade, foram identificadas que a disponibilização das regras dentro do próprio jogo ao modo de um *help*, e uma maior diferenciação gráfica entre as imagens de civis e monstros trariam uma melhora considerável na interface. Embora alguns participantes tenham apontado como aspecto negativo a ausência das regras, a maioria conseguiu entender o funcionamento e objetivo do jogo. Além disso, também foi identificada a necessidade de maior depuração do código-fonte, pois 26,92% encontrou algum tipo de erro durante a execução.

A partir dos resultados da pesquisa, foram percebidas as seguintes alterações na própria metodologia da pesquisa que podem torná-la mais precisa quanto instrumento de aferição da eficácia pedagógica do jogo: aumentar para 50% o número de questões dos testes pré-sessão e pós-sessão relacionadas diretamente ao jogo (cerca de 5 questões por teste), manter o nível de dificuldade dos testes pré-sessão e pós-sessão no mesmo patamar, e permitir aos alunos usarem o jogo por 2 (dois) ou mais dias de aula seguidos. Pretende-se continuar a pesquisa de eficácia pedagógica seguindo esta metodologia modificada, agora para APD.

Como trabalhos futuros foram identificadas as seguintes alterações no jogo:

- Criar um banco de linguagens (regulares e livres de contexto determinísticas) e uma interface em que mediante uma senha o professor possa registrar novas linguagens.
- Incluir o tratamento para autômatos com pilha não determinísticos (APN) e Máquina de Turing (MT).

Referências

- ADOBE. “Adobe Flex 3” (2010). Disponível em: <http://www.adobe.com/products/flex/>. Acesso em 13 Julho 2010
- Binsfeld, R. L., Watanabe, R., Silva, R. C. and Carelli, I. M.(2009) “Alunos como designers: relato de experiência para aprendizagem de Linguagens Formais e Autômatos”, In: VIII Brazilian Symposium on Games and Digital Entertainment, Rio de Janeiro.
- Chesñevar, C. I., Cobo, M. L. and Yurcik, W. (2003) “Using Theoretical Computer Simulators for Formal Languages and Automata Theory”, In: ACM SIGCSE, Volume 35, Issue 2, 2003, ISSN 0097-8418
- Costa, Y. M.G., Meneses, R. C. and Uber, F. R. (2008) “Uma ferramenta para auxílio didático no ensino de teoria da computação”, In: XXVIII Congresso da SBC, Belém-PA.
- Dognini, M. J. and Raabe, A. L. A. (2003), EduLing – Software Educaional para Linguagens Regulares, In: XIV Simpósio Brasileiro de Informática na Educação, Rio de Janeiro.
- Devedzic, V. , Debenham, J. and Popovic, D. (2000) “Teaching Formal Languages by an Intelligent Tutoring System”, In: Educational Technology & Society 3(2), ISSN 1436-4522.
- JFLAP (2009). Disponível em: <<http://www.jflap.org>> . Acesso em 13 Julho 2010.
- Martinez, M., Barbuzza, R., Mauco, M. V. and Favre, L. (2009) “MTSolution: A Visual and Interactive Tool for a Formal Languages and Automata Course”, In: Information Systems Education Journal, 7(10), <http://isedj.org/7/10/> , ISSN 1545-679X.
- Menezes, P. B. (2001), “Linguagens Formais e Autômatos”, Ed. Sagra Luzzatto, Porto Alegre, 4a edição.
- Prensky, M. (2008) “Students as Designers and Creators of Educational Computer Games: Who else?”, In: British Journal of Educational Technology, England.
- Ramos, M. V., Neto, J. J. and Vega, I. S. (2009), Linguagens Formais: Teoria, Modelagem e Implementação, Ed. Bookman, 1ª edição, Porto Alegre.
- Vieira, L. F. M., Vieira, M. A. M. and Vieira, N. J. (2004) “Language Emulator, uma ferramenta de auxílio no ensino de teoria da computação” In: ACM SIGCSE.