

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/267942894>

# Alunos como designers: relato de experiência para aprendizagem de Linguagens Formais e Autômatos

Article

CITATIONS

0

READS

45

4 authors, including:



**Ricardo Luis Binsfeld**

Universidade Federal do Rio Grande do Sul

2 PUBLICATIONS 4 CITATIONS

[SEE PROFILE](#)



**Rômulo C Silva**

Universidade Estadual do Oeste do Paraná, Foz do Iguaçu, Brazil

6 PUBLICATIONS 7 CITATIONS

[SEE PROFILE](#)



**Izaura Carelli**

Universidade Estadual do Oeste do Paraná

6 PUBLICATIONS 22 CITATIONS

[SEE PROFILE](#)

## Alunos como *designers*: relato de experiência para aprendizagem de Linguagens Formais e Autômatos

Ricardo L. Binsfeld<sup>1</sup>   Rodrigo Watanabe<sup>1</sup>   Rômulo C. Silva<sup>1</sup>   Izaura M. Carelli<sup>1</sup>

<sup>1</sup>Universidade Estadual do Oeste do Paraná, DETAE, Brasil

**Resumo:** Prensky (2008) defende que os melhores designers de jogos educacionais são os próprios estudantes. O objetivo deste artigo é relatar a experiência de design de um jogo educacional pelos acadêmicos que já cursaram a disciplina de Linguagens Formais e Autômatos (LFA), na sua segunda versão, incorporando as funcionalidades levantadas no teste de usabilidade da sua primeira versão.

**Palavras-chaves:** jogos educacionais; aprendizagem de LFA; aluno como game designer

### Autores:

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

## 1. Introdução

Atualmente, a disciplina de Linguagens Formais e Autômatos (LFA), um dos pilares na formação de um bacharel em ciência da computação, cumpre dupla função: (1) apoiar outros aspectos teóricos da Ciência da Computação, como por exemplo, decidibilidade, computabilidade e complexidade computacional e (2) fundamentar diversas aplicações computacionais, tais como processamento de linguagens, reconhecimento de padrões e modelagem de sistemas (Menezes, 2001).

LFA descreve a parte da Teoria da Computação correspondente ao uso de máquinas abstratas para o estudo dos problemas que podem ser resolvidos por sistemas computacionais automáticos. Tem como objetivo compreender linguagens geradas por gramáticas, escrever gramáticas que representem linguagens e compreender o potencial e o limite de diferentes tipos de máquinas teóricas para manipulação de linguagens.

Porém não é tarefa fácil levar alunos a entenderem e se apropriarem dos fundamentos dessa área de conhecimento. Para auxiliá-los na assimilação dos conceitos relacionados a LFA é recomendável o uso de atividades que possibilitem a manipulação de premissas que compõem esses conceitos.

O uso de recursos computacionais para a criação de objetos de aprendizagem (OA) é amplamente adotada pelos professores em qualquer nível educacional. Um OA é considerado uma unidade de instrução/ensino que é reutilizável (Wiley, 2000), isto inclui todo o tipo de artefato digital, incluindo jogos educacionais.

Se integrados a prática pedagógica, os jogos propiciam um ambiente de aprendizagem cativante e eficaz, devido ao seu caráter lúdico.

Prensky (2008) defende que os melhores desenvolvedores de jogos educacionais são os próprios estudantes, que aprenderam determinado conteúdo e conhecem as dificuldades na aprendizagem, podendo, desse modo, identificar as melhores soluções.

O objetivo deste artigo é relatar a experiência no desenvolvimento de um jogo educacional, AutomataDefense, demonstrando que o design pode ser elaborado pelos próprios acadêmicos que já tenham se apropriado da temática a ser abordada pelo jogo criando ferramentas de aprendizagem para outros acadêmicos.

Na seção 2, discute-se como jogos educacionais podem ser utilizados para aprendizagem de LFA. Na seção 3, relata-se todo o processo do design do AutomataDefense 1.0, o teste de usabilidade e a implementação da sua continuidade.

## 2. Jogos Educacionais para a aprendizagem de LFA

Como LFA demanda a aprendizagem de conceitos complexos, não é simples desenvolver qualquer tipo de OA. Um levantamento de OAs identificou somente simuladores de autômatos e visualização da execução de algoritmos relacionados a autômatos através de computação gráfica, por exemplo, a ferramenta JFLAP (JFLAP, 2009). Não se referencia jogos educacionais relacionados à disciplina de LFA.

Como os jogos educacionais podem auxiliar os acadêmicos de Ciência da Computação a vivenciar a complexidade da disciplina LFA numa linguagem que lhe é conhecida, a linguagem de jogos, associado à proposta de Prensky (2008), surgiu a ideia de desenvolver um jogo com esta finalidade, pois os jogos podem ser uma nova metodologia de aprendizagem, que permite mudar o paradigma educacional centrado no professor para centrado no aluno, possibilitando que eles internalizem conceitos fundamentais desta disciplina.

### 3. Alunos como *Designers*

Seguindo a teoria de Prensky, o acadêmico Watanabe (2008) desenvolveu o jogo AutomataDefense, o qual auxilia no aprendizado de conceitos relacionados à disciplina de Linguagens Formais e Autômatos, mais especificamente o conceito de autômatos finitos determinísticos, permitindo a assimilação de regras, a compreensão do contexto envolvendo este conceito, e a prática do mesmo de modo interativo e atrativo. Este jogo foi implementado em Adobe Flex, que fornece as estruturas necessárias para a criação de uma interface dinâmica e atrativa.

O jogo AutomataDefense é classificado como do tipo *Tower Defense*. De modo geral, os jogos deste tipo são considerados como jogos de estratégia, porque os resultados obtidos no jogo dependem diretamente do planejamento estratégico aplicado pelo jogador. De um ponto de vista mais amplo, pode-se classificá-lo também como um mini-game, isto é, jogo de curta duração, que para fins educacionais, possui uma vantagem, pois permite abordar conceitos mais específicos de uma determinada área.

Tendo como objetivo auxiliar os alunos na aprendizagem de um tópico de LFA, mais especificamente, autômatos finitos determinísticos, implementou-se um editor de autômatos em que o jogador cria autômatos e posteriormente associa-os às torres que passam a atacar somente as criaturas que são reconhecidas pelo referido autômato.

#### 3.1 AutomataDefense 1.0

Os jogos desse estilo são, geralmente, compostos basicamente por um tabuleiro, que possui uma entrada e uma saída (ver Figura 1) diversas criaturas, as quais entram no tabuleiro no decorrer do jogo e buscam, de algum modo, alcançar a saída do mesmo; e torres têm a capacidade de atacar as criaturas e podem ser colocadas aleatoriamente no tabuleiro.



Figura 1 – Layout do tabuleiro do AutomataDefense

O objetivo do jogo é impedir que algumas destas criaturas alcancem a saída do tabuleiro, para isso, o jogador deve distribuir as torres que atacam estas

criaturas pelo tabuleiro, geralmente formando labirintos, para dificultar a movimentação das mesmas e destruí-las a tempo. Uma regra fundamental é que não se pode bloquear completamente o acesso das criaturas à saída do tabuleiro, para não perder a idéia do jogo.

Há diferentes tipos de torres disponíveis no jogo, como mostrado na Figura 2 que possuem características distintas, isto é, vantagens e desvantagens, em que, por exemplo, algumas podem ter uma melhor média de tiros por segundo, outras maior área de danos, outras que diminuem a velocidade das criaturas dentre outras.

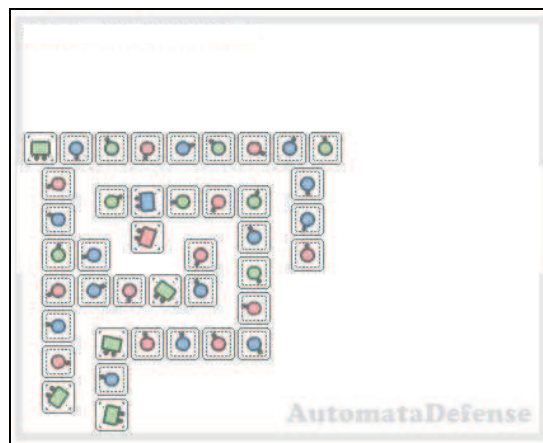


Figura 2 – Exemplos dos tipos de torres

O jogo é bastante dinâmico envolvendo alto grau de estratégia e administração dos recursos disponíveis, tanto que não há uma única solução e nem mesmo uma melhor solução para completar o objetivo do jogo.

O jogo AutomataDefense 1.0, além de possuir funcionalidades particulares para atender o objetivo de relacioná-lo com o conceito de autômatos finitos determinísticos, herda todas as características pertencentes à categoria Tower Defense.

Cada uma das criaturas que tenta atravessar o tabuleiro possui uma palavra relacionada. Há duas categorias de criaturas: monstros (Figura 3) e civis (Figura 4).

Uma das principais estratégias do jogo é a criação de autômatos finitos que reconheçam apenas palavras associadas aos monstros, pois civis não devem ser atacados. Os autômatos então criados são associados a torres que serão posicionadas no tabuleiro.



Figura 3 – Exemplos de monstros



Figura 4 - Exemplos de civis

Assim, outra funcionalidade do jogo é o editor de autômatos finitos determinísticos, mostrado na Figura 5 a partir do qual o jogador pode criar e editar autômatos a qualquer momento do jogo. Cada torre adicionada ao tabuleiro é associada a um único autômato, a qual passa a atacar somente as criaturas cuja palavra seja reconhecida pelo autômato associado. Cada civil atacado ou monstro que alcance a saída do tabuleiro são descontados pontos de vida.

O editor de autômatos não permite a inclusão de não-determinismo nos autômatos, impedindo que o jogador adicione uma transição de estado lendo um símbolo já utilizado em outra transição a partir do mesmo estado.

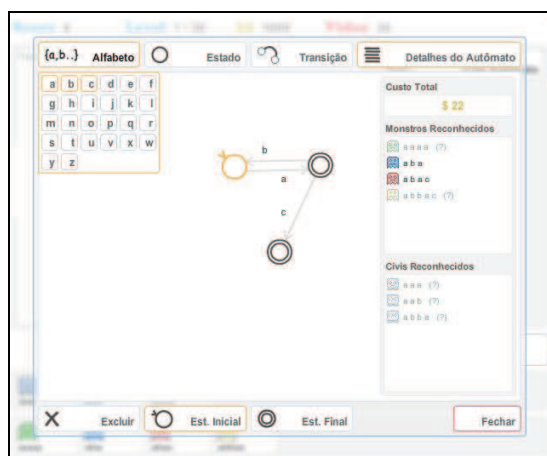


Figura 5 – Editor de autômatos finitos determinísticos

No AutomataDefense, o jogador é desafiado a criar autômatos finitos determinísticos que reconheçam apenas as palavras associadas a monstros. Essa habilidade está diretamente relacionada à capacidade de projetar autômatos para reconhecer linguagens específicas, frequentemente exigida de alunos da disciplina LFA.

O AutomataDefense, como os demais jogos, possui algumas regras fundamentais para controlar as ações do jogador, com a finalidade de torná-lo competitivo e desafiador.

Após o início do jogo, as invasões dos monstros no tabuleiro ocorrem em intervalos constantes de tempo, sendo que, a cada invasão, a dificuldade do jogo aumenta, portanto os monstros ficam mais difíceis de serem destruídos. O *level* do jogo começa em zero, aumentando em uma unidade a cada invasão dos monstros.

Inicialmente, o jogador possui uma determinada quantia de *gold*, sendo que esse valor aumenta a cada monstro destruído, variando conforme o *level* do jogo. Há ações no jogo que exigem o uso desse dinheiro virtual por possuírem um custo associado a elas, como, por exemplo, a construção de torres ou, até mesmo, inserção de novos elementos em um autômato. Esse valor deve ser administrado da melhor maneira possível, pois ele faz parte da estratégia do jogo e afetando diretamente o seu desempenho do mesmo.

Além do *gold*, o jogador também possui uma pontuação, chamada *score*, que varia conforme suas ações no jogo. Essa pontuação não afeta diretamente o jogo, ela tem apenas a finalidade de permitir que o jogador avalie suas estratégias.

Conforme a idéia do jogo, deve-se evitar que os monstros cheguem à saída do tabuleiro e, do mesmo modo, evitar que os civis sejam atacados pelas torres. Para que se mantenha este objetivo, descontam-se pontos de vida do jogador para cada monstro que escapa ou civil que seja abatido. Inicialmente o jogador possui um valor fixo de vidas, definido como vinte, porém pode ser alterado. Caso esse valor chegue a zero, o jogador perde o jogo, isto é, implica em *Game Over*, sendo essa a única maneira de ocorrer derrota.

O fim do jogo, com vitória, ocorre quando o jogador atingir o *level* máximo, definido atualmente como trinta, ou, com derrota, quando ocorre *Game Over*, como descrito anteriormente.

### 3.2 Teste de usabilidade no AutomataDefense 1.0

Foi realizado teste de usabilidade com acadêmicos que estão cursando a disciplina de LFA, cujos resultados foram publicados em (Silva et al, 2009).

O teste foi realizado com cinco participantes, sendo dividido em três etapas:

- Aplicação de questionário para obtenção do perfil do jogador
- Entrega de documento com as regras do jogo e tempo de 1 hora para testar o jogo
- Aplicação de questionário de avaliação do jogo

O teste revelou que a maioria dos participantes gostou do jogo e o avaliou como divertido (Silva et al, 2009).

Embora todos participantes tenham afirmado que as regras do jogo estavam claras, quatro dos cinco participantes tiveram dúvidas de como jogar, apenas no início, devido à falta de familiaridade com o editor de autômatos. A interface foi avaliada como boa pela maioria. Os acadêmicos sugeriram as seguintes melhorias na interface:

- alteração da cor das transições dos estados (atualmente cinza) para uma cor mais visível;

adicionar opção *limpar autômato*, apagando todo o autômato já desenhado; e

- apresentar opção para salvar os autômatos desenhados em memória secundária para não perdê-los ao fim do jogo, podendo recuperá-los em um jogo futuro.

Quanto ao conteúdo do jogo, os acadêmicos sugeriram:

- A criação de fases no jogo em que seja necessário o projeto de autômatos mais complexos;
- A criação de níveis diferentes de monstros;
- A possibilidade de aumentar o nível de ataque das torres.

### 3.3 AutomataDefense 2.0

Em consonância com a premissa de Prensky de que o aluno é o melhor *designer* de jogo educacional, outro acadêmico aprimorará e ampliará o AutomataDefense 1.0, incorporando as funcionalidades identificadas no teste de usabilidade anteriormente descritas, incluindo duas novas fases implementadas, seguindo a mesma metodologia (descrita na seção 3.1).

As novas fases abordarão dois outros conceitos de linguagens formais: autômatos finitos não-determinísticos e autômatos com pilha. Essa alteração amplia a abrangência de conceitos relacionados a LFA, buscando facilitar a aprendizagem dos alunos.

O grau de complexidade das novas fases demandará a modificação do editor de autômatos, permitindo não somente a criação de autômatos finitos determinísticos, mas também de autômatos finitos não-determinísticos e autômatos com pilha. Além disso, serão feitas adequações nos métodos de geração das palavras às criaturas para que essas sejam representativas quanto ao reconhecimento pelos novos tipos de autômatos.

## 3. Conclusão

Os resultados obtidos no teste de usabilidade da primeira versão do jogo AutomataDefense comprovam a proposta de Prensky (2008), em que ele defende os acadêmicos como designers de jogos educacionais. A partir desses resultados, optou-se por dar continuidade, incluindo na nova versão as funcionalidades sugeridas e a inclusão de novas fases, abordando outros conceitos de LFA.

## Agradecimentos

Os autores agradecem ao ITAI e PTI pelo apoio ao desenvolvimento deste projeto.

## Referências

- JFLAP, 2009. JFLAP 6.4. DISPONÍVEL EM: [HTTP://WWW.JFLAP.ORG](http://www.jflap.org). ACESSO EM: 27 DE JULHO DE 2009.
- MENEZES, P. B., 2001. LINGUAGENS FORMAIS E AUTÔMATOS. SAGRA LUZZATTO, PORTO ALEGRE, 4 EDITION.
- PRENSKY, M., 2008. STUDENTS AS DESIGNERS AND CREATORS OF EDUCATIONAL COMPUTER GAMES: WHO ELSE? BRITISH JOURNAL OF EDUCATIONAL TECHNOLOGY.
- SILVA, R. C.; WATANABE, R.; CARELLI, I. M., 2009. AUTOMATADEFENSE: JOGO EDUCACIONAL PARA APOIO EM LINGUAGENS FORMAIS E AUTÔMATOS. IN: SCGAMES 2009 FLORIANÓPOLIS.
- VIEIRA, N. J., 2006. INTRODUÇÃO AOS FUNDAMENTOS DA COMPUTAÇÃO. PIONEIRA THOMSON LEARNING, SÃO PAULO, 1 EDITION.
- WATANABE, R., 2008. AUTOMATADEFENSE: UM JOGO EDUCATIVO PARA APOIO EM LINGUAGENS FORMAIS E AUTÔMATOS. MONOGRAFIA, UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ, FOZ DO IGUAÇU.
- WILEY, D. A., 2000. CONNECTING LEARNING OBJECTS TO INSTRUCTIONAL DESIGN THEORY: A DEFINITION, A METAPHOR, AND A TAXONOMY. DISPONÍVEL EM: [HTTP://REUSABILITY.ORG/READ/](http://reusability.org/read/). ACESSO EM: 27 DE JULHO DE 2009.