

Desenvolvimento de Jogos para o Aperfeiçoamento na Aprendizagem de Disciplinas de Ciência da Computação

Luciano A. Digiampietri
Universidade de São Paulo (EACH-USP)
digiampietri@usp.br

Diogo D. Kropiwiec
Universidade Estadual de Campinas (IC-UNICAMP)
diogo@las.ic.unicamp.br

Abstract

Atualmente, os professores de ensino superior em cursos ligados à computação apresentam duas grandes preocupações. A primeira é quanto à diminuição de candidatos por vaga nesses cursos. O que, de certa forma, contradiz as necessidades do mercado de trabalho que ainda carece de pessoal qualificado nessa área. A segunda preocupação diz respeito à insatisfação dos graduandos com algumas disciplinas, principalmente por não conseguirem vislumbrar a aplicação prática de diversos conceitos teóricos, o que pode, em casos extremos, aumentar a evasão destes cursos.

O objetivo deste projeto é ajudar na resolução destes problemas por meio do desenvolvimento de um ambiente computacional para a implementação e o gerenciamento de jogos de computador que poderá ser usado por professores de diversas disciplinas como exemplo prático dos diversos conceitos teóricos ensinados na sala de aula. Além disso, grande parte do conteúdo desenvolvido, incluindo código fonte, jogos, tutoriais, simuladores e apresentações, estará disponíveis via Internet, possibilitando que qualquer um interaja com os mesmos.

Keywords:: Inteligência Artificial; Desenvolvimento de Jogos; Jogos na Educação; Ensino de Computação

1 Introdução

Atualmente, existem dois grandes desafios no ensino de cursos de graduação ligados à Ciência da Computação: tornar o curso mais atrativo aos estudantes do ensino médio para que estes ingressem nesses cursos e tornar o curso mais interessante para aqueles que já o estão cursando [Guzdial 2003; Forte and Guzdial 2005].

Após a grande procura por cursos ligados à computação, que ocorreu principalmente na década de 1990, observamos uma constante queda na procura por este tipo de curso. O número de candidatos por vaga nas principais instituições públicas de ensino no país está caindo em todos os níveis de ensino (graduação, mestrado e doutorado). Há diversas hipóteses sobre esse fenômeno, tais como (i) a computação não é mais o curso da “moda”; (ii) a grande maioria dos alunos do ensino médio já sabe “utilizar” o computador e por isso não vêem necessidade de um curso de graduação neste assunto; ou mesmo (iii) a impressão de que o perfil de uma pessoa formada nestes cursos é de alguém que ficará o dia inteiro sentado em frente ao computador digitando coisas (potencialmente desinteressantes) sem se envolver com desafios de outras áreas (consideradas mais interessantes por muitos alunos).

Por outro lado, considerando os graduandos em cursos de computação (Ciência da Computação, Sistemas de Informação, Análise de Sistemas, etc), há duas reclamações constantes: (i) as disciplinas desses cursos são ministradas de maneira disjuntas (sem existir uma grande ligação entre o conteúdo de uma disciplina em relação às disciplinas já ministradas) e (ii) há disciplinas excessivamente teóricas, nas quais os alunos não conseguem e/ou não tem oportunidade de aplicar o aprendizado teórico em problemas reais.

Acreditamos que a crescente desmotivação pela procura por cursos de computação dá-se devido à desinformação quanto ao conteúdo do curso; quanto às possibilidades multidisciplinares permitidas pela área; e, até mesmo, sobre o mercado de trabalho. Este fato foi comprovado muitas vezes no evento Unicamp de Portas Abertas (UPA), no qual a universidade abre suas portas e apresenta alguns de seus projetos a estudantes do ensino médio. Nos estandes dedicados aos cursos de computação, é constante a surpresa desses

alunos ao saber que todos os projetos apresentados (envolvendo os mais variados assuntos) são projetos em computação.

Já as críticas dos graduandos em cursos de computação estão intimamente ligadas a falta de um aspecto prático das disciplinas. Aspecto este que é muito procurado pelo mercado de trabalho e que, muitas vezes, não é satisfeito pelos cursos das universidades públicas. De um modo geral, os cursos de computação nas universidades públicas são mais abrangentes e mais teóricos do que os cursos equivalentes de universidades particulares; o que é muito bom, considerando os diferentes objetivos destes dois tipos de instituição. Porém, julgamos que um projeto que possibilite a aplicação prática dos diversos conceitos teóricos pode ajudar o aluno a assimilar o conteúdo teórico, além de servir como um facilitador na entrada deste aluno no mercado de trabalho.

Este artigo apresenta um ambiente de desenvolvimento de jogos para ajudar a enfrentar estes dois desafios. Enquanto o principal enfoque será no segundo desafio, por meio do desenvolvimento de atividades práticas de Algoritmos e Estruturas de Dados, Inteligência Artificial, Interface Humano-Computador e Engenharia de Software, o primeiro desafio será enfrentado como consequência do segundo, por meio da disponibilização dos softwares desenvolvidos, de materiais de apoio e de palestras a serem apresentadas em feiras de profissões e, possivelmente, nas escolas de ensino médio.

O restante deste artigo está organizado da seguinte maneira. Seção 2 detalha as metas e objetivos. A Seção 3 sumariza alguns trabalhos correlatos. Seção 4 apresenta a infra-estrutura atual. A Seção 5 apresenta o trabalho que está sendo atualmente desenvolvido e os próximos passos. Por fim, a Seção 6 apresenta as conclusões e um resumo das contribuições.

2 Objetivos

Este artigo apresenta o protótipo de um ambiente computacional a ser utilizado por professores e alunos para o aperfeiçoamento no aprendizado de disciplinas ligadas à computação. O domínio “desenvolvimento jogos” foi escolhido pelos seguintes motivos: (i) por envolver os mais diversos assuntos da computação; (ii) por ser de interesse dos alunos e das empresas; (iii) por apresentar desafios científicos; e (iv) pela sua grande visibilidade para alunos do ensino médio. A seguir, cada um desses motivos é descrito.

(i) Envolvimento dos diversos assuntos da computação.

O desenvolvimento de jogos envolve aspectos de Engenharia de Software: projeto, implementação e testes; Interface Humano-Computador: interface gráfica, aspectos de acessibilidade e jogabilidade; Inteligência Artificial: representação de conhecimento; algoritmos de busca para implementação de *bots*; Sistemas Distribuídos; Redes Sociais; Algoritmos e Estruturas de Dados; entre outros. Cada uma destas disciplinas poderia tirar proveito deste ambiente para o desenvolvimento de projetos práticos.

(ii) Interessante aos alunos e às empresas.

Diversos alunos têm interesse no desenvolvimento de jogos não apenas pelo fato de jogos serem divertidos, mas também porque o resultado do desenvolvimento é “palpável” no sentido que, assim que o protótipo de um jogo estiver desenvolvido, o desenvolvedor chama usuários para testá-lo e imediatamente receberá um retorno se o jogo está agradando ou não. Consideramos que esta experiência com usuários reais é muito importante, porém inexistente para muitos graduandos. Por outro lado, o mercado de jogos de computadores tem crescido bastante nos últimos anos e há uma demanda não atendida por profissionais com alguma experiência na

área. Desta forma, alunos que participassem ativamente no desenvolvimento do ambiente proposto provavelmente teriam vantagens ao tentar ingressar nas empresas desta área.

(iii) Desafios científicos. Todas as disciplinas de computação apresentadas no item (i) apresentam desafios científicos (por exemplo, questões de acessibilidade, algoritmos de inteligência artificial, sistemas distribuídos, etc). A principal meta científica deste projeto é ter alunos de iniciação científica trabalhando nesses desafios enquanto ajudam a desenvolver e ampliar o ambiente computacional proposto.

(iv) Visibilidade aos alunos do ensino médio. Alunos do ensino médio (assim como muitos outros adolescentes e jovens) têm muito interesse em jogos de computadores. Geralmente, jogos são vistos apenas como uma atividade lúdica, porém eles têm grande utilidade como ferramentas educativas. Pretende-se que o ambiente proposto forneça ferramentas (jogos e outros materiais didáticos) para ajudar no ensino de disciplinas tais como lógica, algoritmos, física, etc. O ambiente computacional proposto também fornecerá interfaces simples para a criação de novos jogos ou fases sem a necessidade de conhecimento de programação. Por exemplo, a criação de uma nova fase do jogo Newmings (ver Seção 4.e), ou a especificação de um jogo de xadrez onde todas as peças, exceto o rei, sejam cavalos. Este tipo de atividade permite que usuários sem conhecimentos em computação interajam na construção do ambiente. Julgamos que esta combinação entre este viés colaborativo e o viés competitivo comumente encontrado nos jogos pode ser muito construtiva. Além disso, pretendemos executar tarefas (palestras, tarde de jogos, etc) juntamente com a organização Atlética de Alunos da EACH-USP a fim de divulgar os mais variados aspectos dos cursos de computação e tentando aumentar a visibilidade destes para os alunos do ensino médio.

3 Trabalhos Correlatos

Diversos experimentos foram realizados tanto na área de Educação em Informática quanto em Jogos de Computadores para demonstrar que o uso de jogos em disciplinas introdutórias de cursos ligados à computação é benéfico para motivar os alunos e aumentar o aprendizado dos mesmos [Forte and Guzdial 2005; Clua 2008].

Jogos têm sido usados em diversas iniciativas como incentivo ao aprendizado para crianças, adolescentes, jovens e adultos [Klawe 1999; Distasio and Way 2007]. Inclusive, novas metodologias de ensino surgiram, como o Aprendizado Baseado em Jogos que está sendo utilizada em diversos sistemas e, atualmente, existem eventos científicos inteiramente dedicados a esta metodologia [Tan et al. 2007; Burgos et al. 2008].

Enquanto a maioria dos trabalhos existentes está focada no uso de jogos para o aprendizado de disciplinas introdutórias em computação, nosso trabalho se destaca em três características: (i) o ambiente proposto visa ajudar no aprendizado de disciplinas desde as introdutórias como Algoritmos e Estruturas de Dados, até disciplinas mais específicas, tais como, Inteligência Artificial, Engenharia de Software, Interface Humano-Computador e Tecnologias para Web; (ii) participação ativa dos alunos na especificação e desenvolvimento do ambiente: além dos módulos básicos especificados e desenvolvidos pelos proponentes, o restante do sistema está sendo implementado de acordo com as sugestões dos alunos; (iii) ambiente de pesquisa em computação: um dos objetivos do ambiente é apoiar atividades científicas nas mais diversas áreas, como Inteligência Artificial, Segurança de Redes e Interfaces.

4 Infra-estrutura atual

O ambiente proposto está em desenvolvimento e já possui os seguintes sistemas (em fase de teses): (a) servidor de jogos; (b) aplicação deflexion¹; (c) aplicação multi-jogadora; (d) bots jogadores de jogo da velha; (e) Newmings; (f) gerador de Sudoku;

¹Deflexion é um jogo de estratégia que utiliza espelhos e laser: www.deflexion.biz

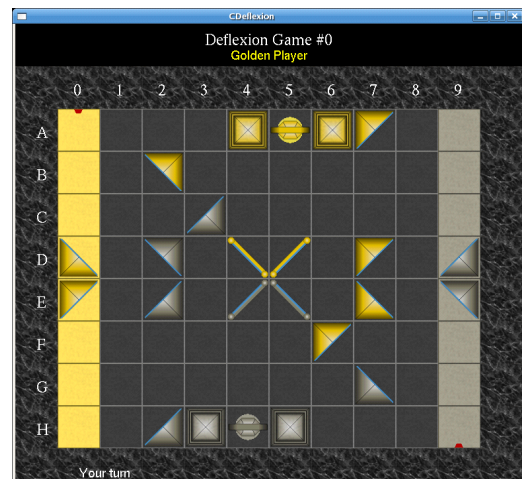


Figure 1: Cópia de tela da Aplicação Deflexion

e (g) conjunto de jogos em Prolog. A idéia do ambiente é possuir uma variedade de jogos desenvolvidos utilizando tecnologias distintas para, assim, poderem ser utilizados nas atividades práticas de diversas disciplinas. A seguir, cada um destes sistemas/programas é descrito:

a) Servidor de Jogos: trata-se de um servidor de jogos de tabuleiro on-line. Já possui implementada a lógica dos seguintes jogos: jogo da velha 2D (convencional) e 3D; damas; xadrez; e deflexion. O servidor está implementado em Java (sendo independente de plataforma) e as aplicações se comunicam com ele via mensagens SOAP, o que permite que aplicações desenvolvidas nas mais diversas linguagens de programação e sistemas operacionais possam interagir com o servidor. Este sistema é composto por três módulos principais: Jogador, Gerenciador de Jogos e Interface de Comunicação. O Jogador armazena as informações de cada jogador (cadastro) além de um histórico de seus resultados. O módulo Gerenciador de Jogos contém as classes básicas para a criação de um jogo: Tabuleiro, Peça e Jogo. Para se implementar um novo jogo basta estender cada uma dessas classes. Dentro do Gerenciador de Jogos também estão as extensões dessas classes para os jogos já implementados (jogo da velha, damas, xadrez, etc). O módulo Comunicação é responsável por manter portas de comunicação para o recebimento das mensagens SOAP, tradução dessas mensagens para o módulo Gerenciador de Jogos e envio das respostas às mensagens recebidas. Uma outra característica interessante deste servidor é que há duas maneiras de se adicionar bots: (i) estendendo as classes já implementadas para o processamento automático de jogadas ou (ii) através de um serviço Web que receba o estado do jogo e sugira a próxima jogada.

b) Aplicação Deflexion: é um “programa cliente” que interage com o Servidor de Jogos para se jogar o jogo deflexion. Enquanto toda a lógica do jogo encontra-se no servidor, esta aplicação é responsável por fornecer uma interface gráfica para que um usuário possa facilmente interagir com o servidor. Esta aplicação foi desenvolvida em C++ e utiliza algumas bibliotecas C++ exclusivas do Linux. Por se tratar de uma aplicação local, o usuário deve executá-la de sua máquina e passar como parâmetro o endereço do servidor de jogos (endereço IP e número da porta de comunicação). A Figura 1 apresenta uma cópia de tela da Aplicação Deflexion.

c) Aplicação Multi-Jogadora: esta aplicação permite que um usuário jogue, via Internet, qualquer um dos jogos do servidor. Enquanto a Aplicação Deflexion foi desenvolvida exclusivamente para interagir com o jogo deflexion (e, por isso, possui uma interface gráfica muito mais detalhada e específica), a Aplicação Multi-Jogadora possui uma interface genérica sem grandes detalhes gráficos, mas que possibilita a interação com qualquer um dos jogos. Desenvolvida inicialmente para testar cada um dos novos jogos incluídos no Servidor de Jogos, ela também pode ser utilizada



Figure 2: Cópia de tela do programa Newmings

por aqueles que queiram interagir com o servidor de jogos sem a necessidade de baixar nenhuma aplicação. Esta aplicação foi desenvolvida em Perl e utiliza o formato CGI para possibilitar a interação com o usuário via navegador de Internet.

d) Bots Jogadores de Jogo da Velha: este sistema não está diretamente relacionado com os anteriores, pois sua intenção é apresentar uma aplicação simples de aspectos teóricos; neste caso, sobre Inteligência Artificial. Além de permitir ao usuário jogar Jogo da Velha, este sistema também fornece três *bots* (do inglês, *robots*, ou seja, programas de computador capazes de jogar um dado jogo) que implementam diferentes métodos da Inteligência Artificial: (i) busca MiniMax; (ii) sistema especialista (baseado em regras); e (iii) uso de dicionário de jogadas. O Jogo da Velha foi escolhido para este sistema por se tratar de um jogo extremamente simples, no qual é fácil explicar e exemplificar o funcionamento de cada um dos métodos de Inteligência Artificial implementados.

e) Newmings: inspirado no jogo Lemmings de 1991, este programa apresenta um jogo de lógica cujo objetivo é salvar um determinado número de pingüins. Ao usuário, cabe dar algumas capacidades específicas aos pingüins de forma a atingirem seu objetivo. Este jogo foi desenvolvido em C++ e uma das características interessantes é que, para se adicionar novas fases/níveis basta a especificação de dois arquivos. Ou seja, um usuário que não conhece nada de computação pode “criar um novo jogo” sem precisar programar uma linha de código. A Figura 2 apresenta uma cópia de tela da primeira fase do programa Newmings.

f) Gerador de Sudoku: gera automaticamente “tabuleiros” de Sudoku. Desenvolvido em C# (.net), o usuário pode configurar o grau de dificuldade solicitando ao programa que varie a quantidade de números que precisarão ser preenchidos para se resolver o Sudoku. A Figura 3 apresenta uma cópia de tela do Gerador de Sudoku.

g) Jogos em Prolog: este sistema é composto por quatro jogos desenvolvidos em Prolog: (i) sistema especialista jogador de jogo da velha; (ii) sistema especialista jogador de truco; (iii) jogador de damas, baseado em busca MiniMax; e (iv) o Mundo de Wumpus (nome original: *Hunt the Wumpus*).

Por enquanto, estes sistemas estão disponíveis para testes apenas para os alunos de algumas disciplinas específicas dentro da Universidade de São Paulo, por exemplo, Introdução a Ciência da Computação, Algoritmos e Estruturas de Dados e Inteligência Artificial. Esta interação com os alunos está possibilitando o aperfeiçoamento do ambiente proposto para que, nos próximos meses, este seja disponibilizado via Internet.

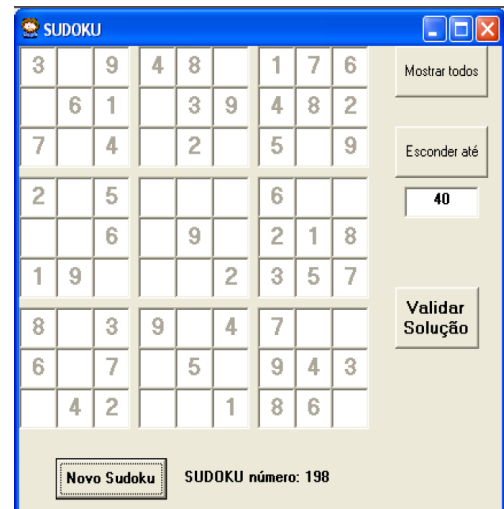


Figure 3: Cópia de tela do programa Gerador de Sudoku

5 Próximos passos

Conforme apresentado na seção anterior, diversos sistemas já foram desenvolvidos (ou prototipados). A tarefa atual consiste em melhorar a documentação do que já foi implementado, desenvolver novos programas, estender e integrar esses sistemas e disponibilizar tudo via Internet criando um ambiente interativo onde qualquer usuário possa jogar, criar novos jogos e baixar jogos e materiais didáticos relacionados.

Neste semestre, parte do sistema está sendo testado junto aos alunos das disciplinas Inteligência Artificial e Introdução a Ciência da Computação II. Para o semestre que vem, serão desenvolvidos simuladores para apoiar o ensino de algoritmos clássicos de estruturas de dados, tais como, gerenciamento de filas, pilhas, listas e árvores, Torres de Hanói, algoritmos para a resolução do Cubo Mágico, etc.

Também estão sendo desenvolvidas ferramentas para permitir um torneio de *bots* para todos os jogos que compõem o Servidor de Jogos citado anteriormente.

Futuramente, pretendemos desenvolver um projeto em Aprendizado Baseado em Jogos que será acrescentado ao ambiente proposto. No Aprendizado Baseado em Jogos, jogos são criados com a única finalidade de serem ferramentas para o aprendizado. Esta área tem se desenvolvido bastante atualmente, tanto em aspectos teóricos quanto em aspectos práticos. Acreditamos que este tipo de projeto combinado ao ambiente apresentado neste artigo serão uma grande contribuição para o aprendizado em computação.

O conteúdo desenvolvido até o momento está sendo organizado e documentado e será disponibilizado via Internet em dois locais distintos: (i) no site do projeto (<http://www.uspleste.usp.br/digiampietri/jogos>) e no Portal *Microsoft Faculty Connexion* (<http://www.microsoft.com/education/facultyconnection/br>).

6 Conclusões

Este artigo apresentou um ambiente computacional para o desenvolvimento e disponibilização de jogos de computador que visa obter contribuições em três diferentes âmbitos: ensino, pesquisa e extensão.

Quanto ao ensino, as contribuições esperadas são a disponibilização de um ambiente computacional para a elaboração de atividades práticas ligadas às áreas de (i) Inteligência Artificial por exemplo, agentes de software, sistemas especialistas, algoritmos de busca e representação de conhecimento; (ii) Engenharia de Software: projeto de software, testes, programação extrema (XP) e padrões de projeto; (iii) Algoritmos e Estruturas de Dados; (iv) Sistemas Distribuídos: sistemas interoperáveis e padrões para a Web;

e (v) Interface Humano-Computado: interfaces gráficas interativas. Além deste ambiente computacional para auxílio no aprendizado de disciplinas em computação que poderá auxiliar centenas de alunos por semestre em nossa unidade de ensino (além dos usuários esporádicos ou mesmo possíveis colaborações com outras unidades de ensino), acreditamos que a interação entre os alunos de computação e este ambiente poderá representar um diferencial no currículo destes alunos, pois muitas empresas de software procuram profissionais que já tenham experiência em uma ou mais das seguintes áreas: programação ágil, desenvolvimento de jogos, desenvolvimento de sistemas interoperáveis e uso de padrões de projeto.

No âmbito de pesquisa, este projeto apresenta contribuições potenciais em dois níveis diferentes. O primeiro está relacionado ao uso do ambiente: pelo fato do ambiente computacional proposto ser interoperável, ele poderá ser usado para testar comparativamente diferentes soluções para, por exemplo, algoritmos jogadores de um dado jogo (*bots*); ou mesmo teste de interfaces, etc. Ou seja, o ambiente proposto pode servir de apoio a diversas atividades de pesquisa. A segunda contribuição é mais direta e está relacionada ao desenvolvimento do ambiente propriamente dito. Existem diversos desafios científicos quanto à interação humano-computador, incluindo questões de acessibilidade e usabilidade; algoritmos de inteligência artificial; e processo de especificação, desenvolvimento e teste de software. Pretendemos comparar e estender as soluções propostas para cada tipo de problema e, com isso, gerar resultados inovadores em algumas destas áreas.

As contribuições ligadas à extensão universitária se referem ao impacto positivo que a disponibilização do ambiente proposto, incluindo material didático, jogos de lógica e código fonte, pode trazer aos alunos do ensino médio. Além de difundir os cursos ligados à computação apresentando aspectos de como estes cursos podem envolver atividades multidisciplinares, o material de apoio também poderá ser útil para aperfeiçoar o entendimento de lógica além de servir como uma introdução destes alunos à área de programação de computadores.

References

- BURGOS, D., MORENO-GER, P., SIERRA, J. L., FERNÁNDEZ-MANJÓN, B., SPECHT, M., AND KOPER, R. 2008. Building adaptive game-based learning resources: The integration of ims learning design and. *Simul. Gaming* 39, 3, 414–431.
- CLUA, E. W. G. 2008. A game oriented approach for teaching computer science. In *Anais do XXVIII Congresso da SBC, Workshop sobre Educação em Computação (WEI)*, 10–19.
- DISTASIO, J., AND WAY, T. 2007. Inclusive computer science education using a ready-made computer game framework. *SIGCSE Bull.* 39, 3, 116–120.
- FORTE, AND GUZDIAL. 2005. Motivation and nonmajors in computer science: Identifying discrete audiences for introductory courses. *IEEE TE: IEEE Transactions on Education* 48.
- GUZDIAL, M. 2003. A media computation course for non-majors. *SIGCSE Bull.* 35, 3, 104–108.
- KLAWE, M. M. 1999. Computer games, education and interfaces: the e-gems project. In *Proceedings of the 1999 conference on Graphics interface '99*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 36–39.
- TAN, P.-H., LING, S.-W., AND TING, C.-Y. 2007. Adaptive digital game-based learning framework. In *DIMEA '07: Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*, ACM, New York, NY, USA, 142–146.