

JSUBSTANCE: UMA FERRAMENTA COMPUTACIONAL LÚDICA PARA AMBIENTES DE ENSINO-APRENDIZAGEM

Jackson Amaral da SILVA (1); Jefferson Amaral da SILVA (2); Gentil SERRA JR (3)

(1) (2) (3) Centro Federal de Educação Tecnológica do Maranhão, Av. Getúlio Vargas, Nº 4 – Monte Castelo
São Luís – MA, (98) 32179000, (98) 32179067

(1) e-mail: jackson.amarals@gmail.com

(2) e-mail: jefferson.amarals@gmail.com

(3) e-mail: gentil@cefet-ma.br

RESUMO

O ambiente do jogo propicia motivação, concentração, desenvolvimento do raciocínio, dentre outros fatores utilizados em um ambiente de ensino-aprendizagem. Os jogos em rede podem ser utilizados para potencializar o ensino através da negociação, colaboração e cooperação entre os aprendizes. Nesse contexto, mostra-se a implementação e vantagens de utilizar-se recursos lúdicos computacionais na aprendizagem de química por alunos do ensino médio. Nessa pesquisa experimental, utiliza-se os conceitos de aprendizagem cooperativa computadorizada para conduzir os aprendizes à construção do conhecimento. Além disso, utiliza-se conceitos de Inteligência Artificial para configurar a dificuldade do jogo de acordo com o perfil do jogador e modelar o conhecimento. A implementação dessa ferramenta computacional é feita com o uso de tecnologias livres, tais como Java 2D API e tuProlog. Com o uso dessa ferramenta, os aprendizes têm a possibilidade de construir seu próprio conhecimento no decorrer do jogo em rede, além disso, o professor poderá identificar as unidades de conhecimentos adquiridas pelos aprendizes e ainda construir um sociograma que mostra os relacionamentos entre os alunos.

Palavras-chave: Aprendizagem Cooperativa Computadorizada, Inteligência Artificial na Educação, Jogos em Rede.

1. INTRODUÇÃO

De uma forma geral, os jogos fazem parte da nossa vida desde os tempos mais remotos, estando presentes não só na infância, mas como em outros momentos. Os jogos podem ser ferramentas instrucionais eficientes, pois eles divertem enquanto motivam, facilitam o aprendizado e aumentam a capacidade de retenção do que foi ensinado, exercitando as funções mentais e intelectuais do jogador (TAROUCO et al, 200). Os jogos, sob a ótica de crianças e adolescentes, se constituem a maneira mais divertida de aprender. Além disso, eles proporcionam a melhora da flexibilidade cognitiva, pois funcionam como uma ginástica mental, aumentando a rede de conexões neurais e alterando o fluxo sanguíneo no cérebro quando em estado de concentração.

O processo de ensino-aprendizagem é complexo e muitas vezes se depara com problemas das mais diversas naturezas como falta de motivação por parte de alunos e professores, dificuldade de socialização dos alunos entre si, conteúdos expostos de formas maçantes e pouco atrativas, falta de recursos físicos e/ou didáticos etc. Portanto, a utilização de jogos computadorizados na educação proporciona ao aluno motivação, desenvolvendo também hábitos de persistência no desenvolvimento de desafios e tarefas.

É conhecida a capacidade que os jogos têm de motivar as pessoas e desenvolver sua capacidade de concentração e raciocínio bem como, em se tratando de jogos em rede, a possibilidade de promover a interação com outros indivíduos, característica dos jogos eletrônicos, que pode ser explorada como potência para aprendizagens e desenvolvimento sócio-cognitivo em contextos cooperativos e de compartilhamento afetivo (BASTOS FILHO et al., 2004). Daí, a importância desse modelo de software dentro do âmbito educacional e sua subsequente escolha para desenvolvimento deste projeto.

Neste sentido, os jogos educacionais podem ser um elemento catalisador, capaz de contribuir para o processo de resgate do interesse do aprendiz, na tentativa de melhorar sua vinculação afetiva com as situações de aprendizagem (BARBOSA, 1998), além de incentivar a socialização com outros participantes do processo.

Os jogos educacionais se baseiam numa abordagem auto-dirigida, isto é, aquela em que o sujeito aprende por si só, através da descoberta de relações e da interação com o software. Neste cenário, o professor tem o papel de moderador, mediador do processo, dando orientações e selecionando softwares adequados e condizentes com sua prática pedagógica. Ele vai além do simples coletor de informações, percebendo a necessidade de pesquisar, selecionar, elaborar e confrontar visões, metodologias e os resultados esperados (TAROUCO et al, 2004).

Existem jogos eletrônicos de vários tipos, tais como jogos de ação, aventura, cassino, lógicos, estratégicos, esportivos e *role-playing games* (RPGs). Independente do tipo dos jogos, eles podem ser utilizados de diferentes formas, conforme destaca Botelho (2004):

[...] para treinamento de habilidades operacionais, conscientização e reforço motivacional, desenvolvimento de insight e percepção, treinamento em comunicação e cooperação, integração e aplicação prática de conceitos aprendidos e até mesmo *assessment* (avaliação de aprendizagem).

Nesse contexto, propõe-se o desenvolvimento de uma ferramenta computacional lúdica que venha a se tornar um aparato utilizado pelo professor para sanar ou minimizar os problemas citados. Tal ferramenta está sendo implementada em forma de um jogo educacional a ser executado em rede por grupos de alunos. O jogo proposto neste trabalho, o JSubstance, se enquadra no tipo RPG.

A Seção 2 deste trabalho apresenta o jogo proposto e seu contexto educacional de aplicação. Em seguida, na Seção 3, são apresentadas as tecnologias utilizadas para desenvolvê-lo. Posteriormente, na Seção 4, descreve-se como ocorrem as interações entre os aprendizes/jogadores dessa ferramenta. Finalmente, a Seção 5 apresenta as considerações finais.

2. JSUBSTANCE: UMA FERRAMENTA COMPUTACIONAL LÚDICA PARA O ENSINO DA QUÍMICA

O JSubstance é um jogo eletrônico que está sendo desenvolvido para auxiliar o ensino da Química, mais especificamente no que diz respeito à separação de misturas, funções e reações químicas, enquadrados dentro da Química Geral. Esse conteúdo é abordado no primeiro ano do ensino médio e foi escolhido em razão da grande dificuldade dos alunos em relação a ele. Essa dificuldade se dá em razão da falta de motivação dos alunos, devido aos conteúdos serem apresentados de forma pouco visual.

O jogo consiste basicamente de um conjunto de cenários onde o personagem, controlado pelo aluno, recebe, ao longo do percurso, lições sobre Química. O jogador encontra pelo seu caminho substâncias, como água, por

exemplo, e adquire ferramentas e habilidades necessárias para separar os átomos dessas substâncias. Por exemplo, para separar a água ele utiliza eletrólise, após fazer isso, o jogador guarda seus átomos em um Banco de Átomos. Esse Banco é um “espaço” onde guarda-se os itens adquiridos. Em seguida, o jogador é capaz de reagrupar os átomos adquiridos para conseguir novamente a substância inicial ou uma substância diferente, de acordo com suas tarefas.

A motivação do aluno para realizar as tarefas dentro do jogo é que, em determinados momentos, o jogador encontra pela sua frente desafios que exigem conhecimentos de Química para serem superados. Por exemplo, o jogador chega a um vilarejo onde, conversando com os moradores, ele percebe que o lago do local está com a água muito ácida o que causa a morte dos peixes, esse cenário pode ser observado na Figura 1. Conversando um pouco mais com outros alunos (à distância ou não), o jogador percebe que para neutralizar o teor ácido da água ele pode adicionar a esta uma substância base. Então, uma vez possuindo os átomos necessários em seu Banco de Átomos, o jogador agrupa esses átomos para formar essa substância e a aplica no lago com água ácida. Mas caso o jogador não possua os átomos necessários, ele poderá percorrer o cenário para procurá-los ou interagir com outros alunos, pela Rede, para solicitá-los. Após realizar essa tarefa o jogador recebe a permissão para avançar no jogo.



Figura 1 - Tela de um desafio presente no JSubstance

No decorrer do jogo, o jogador, encontrará outros desafios nos mesmos moldes do citado anteriormente, mas com pequenas particularidades. Podendo encontrar inimigos móveis que devem ser derrotados com o uso de ferramentas e substâncias adequadas.

Os jogos convencionais têm, em geral, a característica de tentar derrotar o jogador, impedir que este avance no jogo, em um ambiente educacional isso poderia ser desmotivador para o aprendiz. Sendo assim, é relevante que um jogo educacional utilize do modelo de usuário construído com base em informações como tempo que o jogador leva para superar os desafios, número de erros, número de acertos, número de derrotas etc., para reconfigurar seu nível de dificuldade de acordo com o perfil do jogador. Outra característica importante no JSubstance é essa capacidade de adaptação. Portanto, este ambiente é capaz de ajudar o aprendiz a avançar caso tenha dificuldades. Isso é possível fornecendo-lhe dicas ou facilitando os desafios, no entanto, sem retirar o fator desafio, ou por outro lado tornar seu ambiente mais complexo caso o jogador avance rapidamente sem dificuldades. Um exemplo de jogo adaptativo já desenvolvido é o fIOW (CHEN, 2007). Este é um jogo desenvolvido utilizando a tecnologia Flash (ADOBE, 2007) que ajusta seu nível de dificuldade dependendo de como o usuário joga. Ele utiliza uma técnica chamada de "active DDA" (dynamic difficulty adjustment).

3. TECNOLOGIAS UTILIZADAS

Para desenvolvimento dessa ferramenta foram escolhidas tecnologias robustas, confiáveis e livres e/ou *opensource*. Tais tecnologias provêm os recursos necessários para implementação das funcionalidades do jogo, tais como as linguagens de programação Java (SUN, 2007) e Prolog (SURI, 2007) e os *frameworks* ProActive (OBJECT WEB, 2007) e tuProlog (TUPROLOG, 2007).

3.1. A Linguagem Java

A linguagem de programação, base do desenvolvimento do sistema, selecionada para implementar o JSubstance foi a linguagem Java, por ser uma linguagem robusta, utilizada dentro da comunidade de desenvolvedores e com grande quantidade de recursos (HORSTMANN e CORNEL 2003). Dentre alguns desses recursos pode-se citar a portabilidade entre diferentes plataformas. Dessa maneira, o JSubstance poderá ser executado em qualquer sistema operacional que disponibilize uma máquina virtual Java (*Java Runtime Environment*). Outra grande vantagem do Java é que, por ser uma linguagem orientada a objetos, permite que a aplicação possa ser dividida em camadas especializadas, através de um padrão de projeto chamado MVC (*Model View Control*), permitindo que uma camada seja modificada sem “danificar” a outra. Sendo assim, o sistema fica extensível de forma que cada camada possa evoluir sem causar danos ao sistema.

A tecnologia Java 2D, parte integrante da API (*Application Program Interface*) Java, foi utilizada para fazer os desenhos e realizar os movimentos dos jogadores. Essa tecnologia possibilita o desenvolvimento de aplicações gráficas incorporando gráficos e animações 2D de alta qualidade, texto e imagens. Ela permite tais criações sem necessidade de interação mais direta com o *hardware*. A sua escolhida foi dado por uma série de vantagens que possui, tais como (BRACKEEN, 2004):

- **Renderização transparente:** Diferentemente de outras API's, a API Java 2D possibilita renderização dos desenhos e movimentos que compõem o jogo sem se preocupar diretamente como esse processo se dar a nível de *hardware*, assim sendo, pôde-se focar a atenção no desenvolvimento propriamente dito do jogo sem se ater a como ele será “desenhado na tela do computador”.
- **Suporte a imagens geradas externamente:** É possível fazer uso de imagens criadas anteriormente com o uso de outras ferramentas ou mesmo fotografias dentro de *softwares* desenvolvidos com a API Java 2D, com isso se pode realizar a etapa de desenho dos personagens, cenários e objetos do jogo separadamente do desenvolvimento do jogo. É possível fazer uso de quaisquer imagens nos formatos GIF, PNG e JPEG.
- **Compatibilidade com outros recursos da linguagem Java:** Por ser parte integrante da linguagem de programação Java, a API Java 2D pode se comunicar e usufruir de todo o arcabouço de recursos dessa linguagem, como acesso a banco de dados, suporte a execução em rede, suporte a leitura e gravação de arquivos, utilização de uma rica interface gráfica com o usuário, utilização de *multithreading*, etc.

Além disso, outra característica que foi decisiva para a escolha da linguagem Java no desenvolvimento do jogo educacional foi a possibilidade de integração dessa com outras tecnologias necessárias ao seu desenvolvimento. Dessa forma, percebe-se que com o uso da tecnologia Java no desenvolvimento da ferramenta proposta a torna bem estruturada, robusta e extensível.

3.2. O ProActive

Os jogos, de uma forma geral, quando executados em rede cativam mais o jogador, pois ele interage não só com o computador mas também com outros jogadores podendo estes estarem em uma mesma sala ou separados por uma grande distância geográfica.

Para propiciar que o jogo seja executado em rede foi escolhido o *framework* ProActive (OBJECT WEB, 2007). Esse *framework* provê recursos necessários para “montar” o jogo em uma arquitetura Cliente-Servidor, assim sendo, o jogo é dividido em duas partes: um Servidor que é responsável por receber chamadas dos Clientes e tornar a comunicação entre eles possível e um Cliente que faz chamadas ao Servidor e utiliza as informações disponibilizadas pelo mesmo. Este modelo também permitiu que fosse construído um *chat* embutido dentro do jogo. Através desse *chat* os jogadores podem conversar uns com os outros. Essa comunicação se dá no momento em que um cliente envia uma mensagem para o Servidor implementado com o ProActive. Este Servidor se encarrega de repassar a mensagem para os demais Clientes conectados a ele.

O ProActive é um *framework* escrito completamente na linguagem Java o que faz dele naturalmente compatível seus recursos, e, assim como a linguagem em que foi escrito, este *framework* também é distribuído de forma livre (código aberto), não necessitando que seja paga qualquer taxa para sua utilização.

3.3. O Prolog

Jogos em geral possuem recursos de inteligência artificial embutidos na programação para definir estratégias e ações do sistema perante as interações com o usuário. Sendo assim, além de aumentar o nível de desafio imposto pelo jogo é um atrativo a mais para os jogadores. Podendo criar uma base de conhecimento a ser

utilizada em interações personalizadas. Para o desenvolvimento do JSubstance, foi selecionada a linguagem Prolog (*Programmation en Logique*) como linguagem de programação lógica, a qual é integrável à linguagem Java.

Prolog é uma linguagem de uso geral que é especialmente associada com a inteligência artificial e a lingüística computacional (SURI, 2007). É uma linguagem declarativa, onde o usuário especifica o conhecimento sem se preocupar com a sua aplicação. Um programa Prolog é uma especificação executável é um conjunto de propriedades e relações, ou melhor, é um conjunto de afirmações. O código não é uma descrição da solução do problema, em Prolog, um problema é um conjunto de perguntas sobre os objetos do universo da aplicação. A execução de um programa Prolog implica a dedução de novas relações a partir das afirmações do código.

Prolog se difundiu e passou a ser utilizada em aplicações de diversas áreas, tais como: Solução de Equações Simbólicas, Banco de Dados, Sistemas Especialistas, Planejamento Automático de Atividades, Jogos Eletrônicos, Ambientes Educacionais, Compiladores, Análise Bioquímica, etc. Certamente, essa popularidade se deve à facilidade com que se pode descrever os problemas nessa linguagem, dado que os conhecimentos relacionados a esses domínios são facilmente descritos por meio de fatos (dados ou relações conhecidas) e regras (definições de relacionamentos), ou seja, relações entre objetos da base de dados. A execução do programa Prolog carregado em memória consiste em realizar uma pergunta de forma interativa: o intérprete gerará por inferência os resultados que se deduzem a partir do conteúdo da base de dados.

3.4. tuProlog: Integração entre Java e Prolog

A integração entre Java e Prolog visa basicamente a utilização do mecanismo de busca e inferência de Prolog em programas Java e/ou a utilização das vantagens de Java em programas Prolog. Tal integração pode ser implementada basicamente em três arquiteturas distintas:

- **Arquitetura em uma camada:** Um programa Prolog é transformado em um programa Java. A partir de um tradutor Prolog/Java todo o código Prolog é convertido para código Java, o que faz com que a cada mudança na base Prolog haja a necessidade de uma nova conversão (Exemplo de *framework*: PrologCafe);
- **Arquitetura em duas camadas:** A Máquina de inferência Prolog é implementada em Java com métodos para carregar uma base Prolog, para realizar uma consulta ou busca etc. (Exemplo de *framework*: tuProlog);
- **Arquitetura de três ou mais camadas:** dois processos sendo executados, sendo uma máquina virtual Prolog e uma máquina virtual Java. Esses processos se comunicam via uma terceira linguagem (por exemplo, linguagem C) através de API ou via soquetes com conversão bi-direcional de objetos Java para termos Prolog (Exemplo de *framework*: InterProlog).

A arquitetura escolhida para desenvolver a integração no jogo foi a de duas camadas, onde ocorre a emulação do Prolog, ou seja, o motor de inferência Prolog é implementado em Java. A base Prolog fica em uma camada e a programação orientada a objetos (Java) em outra camada. Sendo assim, o código Java fica separado do código Prolog. Tal arquitetura garante independência entre as camadas de forma que alterações na base de dados não reflete, necessariamente, em mudanças no código Java, e vice-versa. Assim mostrando-se indicada para ser utilizada no software proposto.

O *framework* utilizado para a integração na arquitetura desejada foi o tuProlog (TUPROLOG, 2007) uma ferramenta estável e gratuita. O arquivo que contém as regras Prolog é carregado para dentro de um objeto Java em tempo de execução e a partir de então são utilizados métodos que chamam o motor Prolog emulado, executam consultas, retornam os resultados, etc. Tudo a partir de objetos e métodos o que torna natural o tratamento dos resultados dentro de um programa Java.

O tuProlog é um projeto *Open Source* licenciado sob LGPL e por se tratar de Java, pode ser utilizado em qualquer plataforma operacional. Isso foi decisivo na escolha desse *framework*, pois existem outras tecnologias que realizam essa integração na arquitetura desejada de maneira bem similar, contudo, são tecnologias proprietárias como é o caso do JIProlog (*Java Internet Prolog*) (UGOS' WEB, 2007) que foi testado em uma versão para avaliação (*shareware*), mas quando se fazia uma consulta à base de dados uma mensagem aparecia informando sobre a compra da sua versão comercial.

4. INTERAÇÃO ENTRE JOGADORES E GRUPOS

A cooperação é uma das atividades humanas mais importantes. Porém, nas salas de aula tradicionais a cooperação não é um fator principal, ela é até vista como uma forma de "pesca" por alguns educadores. A colaboração entre aprendizes existe, mas não é estimulada pelos docentes e acontece de forma muito tímida. Enquanto isso, a competição pela recompensa é o comportamento dominante nas aulas, isto implica que o sucesso de um estudante reduz as chances de recompensa dos outros. Entende-se que essas três modalidades de interação são importantes, desde que aplicadas em momentos oportunos.

4.1 Modalidades de Interações

Com a cooperação sendo utilizada no processo de ensino-aprendizagem pressupõe-se a existência de um grupo de alunos voltados para o mesmo objetivo e que colocam, acima dos interesses pessoais, os interesses do grupo. Todos procuram ajudar e pensam juntos, discutindo a elaboração de um plano que lhes facilite a realização de tarefas, dividindo atribuições, mas que permita a posterior consecução do objetivo dentro do mesmo espírito de equipe. Há uma interdependência positiva entre as realizações de objetivo dos estudantes. Eles percebem que podem alcançar seus objetivos de aprendizagem se, e somente se, os outros aprendizes do grupo alcançam também os seus. Em situações cooperativas de aprendizagem, o sucesso de um membro da equipe ao realizar uma determinada tarefa depende do esforço individual deste aluno e do esforço de todos os outros componentes do grupo, cada um contribuindo com conhecimentos, habilidades e todos os outros recursos necessários à conclusão da tarefa. Logo, o objetivo e sucesso de um grupo são também de todos os alunos pertencentes a ele. Os alunos interagem cooperativamente entre si, buscando uma maior eficiência na aprendizagem, pois se um componente falhar todos falham (SERRA JR, 1999).

A colaboração se assemelha muito com a cooperação, mas diferencia-se em razão dos objetivos dos participantes do processo serem distintos. A interação ocorre como uma forma de companheirismo apenas, ou seja, um ajuda ao outro sem se envolver com seus objetivos. Esse tipo de interação é muito comum nas salas de aula tradicionais, onde os alunos não são desafiados pelo professor a compartilhar um desafio comum, então eles são conduzidos a trocar idéias sem comprometer-se realmente com a tarefa do outro.

No contexto da competição (negociação), o aluno/grupo tem metas a alcançar sem, no entanto, necessitar de outros. Dessa forma, os alunos podem interagir entre grupos, através da negociação na busca de interesses individuais. Os aprendizes tem a impressão de que vão obter seus objetivos se, e somente se, os outros grupos não os obtêm. Isso pode ser útil em determinados momentos em que é necessário criar desafios para motivar a aprendizagem.

4.2 Interações no JSubstance

Um jogo, de maneira geral, sempre aborda o "espírito de competição" entre os participantes. Mas um jogo com fins educacionais, além da competição, deve promover a colaboração e cooperação para que a aprendizagem se dê de maneira coletiva e possa se construir os conhecimentos através da troca de idéias. O JSubstance utiliza essas três modalidades de forma que o jogador possa cooperar com sua equipe e colaborar ou negociar com membros de outros grupos. Isso acontece da seguinte forma:

- **Colaboração:** Jogadores de grupos diferentes trocam seus itens do Banco de Átomos para cumprir metas distintas. Por exemplo, um grupo de jogadores precisa de um determinado item para prosseguir no jogo, se ele não possuir poderá pedir que alguém de um outro grupo possa ceder este item. A característica marcante desta modalidade de interação é o fato de que os jogadores envolvidos não dependerem mutuamente uns dos outros para prosseguir. A colaboração não é obrigatório, é uma ajuda voluntária de um grupo/aluno para com outro;
- **Cooperação:** Caracteriza-se quando diferentes jogadores se deparam com uma situação na qual necessitam agir em grupo para poder prosseguir. Por exemplo, um grupo de jogadores tem que coletar uma determinada quantidade de uma substância e utilizá-la para avançar. Se um dos jogadores não utilizar a parte que coletou não será possível para nenhum deles prosseguir no jogo, logo para avançar é necessário que eles cooperem. Essa modalidade de interação é realizada realizada por integrantes de um mesmo grupo que tenham um objetivo comum. Diferentemente da colaboração, na cooperação os jogadores desfrutam dos benefícios ou malefícios resultantes dessa interação;
- **Negociação:** É caracterizada quando grupos de jogadores se comunicam a fim de negociar itens ou recursos do jogo. Por exemplo, um grupo necessita de N itens para prosseguir no jogo mas não possui

todos, esse grupo pode contatar um outro grupo a fim de saber se alguém possui os itens necessários. O grupo contatado poderá negociar a troca de itens, podendo querer algo em troca para ceder o que foi solicitado. Observa-se que essa modalidade de interação só é realizada por membros de grupos diferentes e que cada parte visa única e exclusivamente benefício próprio. Ao negociar, o participante tentará abrir mão do mínimo possível para que possa “levar vantagem” em relação aos demais grupos de aprendizes. Uma negociação poderá ter sucesso caso os negociantes cheguem a um acordo, caso contrário, eles podem simplesmente procurar outros grupos para negociar.

Dadas as modalidades de interação, o JSubstance promove o uso de todas essas de forma natural. Por exemplo, um jogador pertence a um grupo que necessita de uma lista de itens e não possui todos. Esse grupo, cooperando entre si, poderá solicitar a outros grupos os itens necessários. Alguns participantes de outros grupos poderão fornecer gentilmente os itens necessários (colaboração) ou poderão querer negociar esses itens, pedindo em troca um outro item desejado (negociação).

Toda a comunicação é feita através do *chat* embutido no jogo. Dentro do Servidor são armazenadas todas as informações sobre quem interage com quem e de que forma. Inclusive, os diálogos poderão ser armazenados também. Esse armazenamento fica a critério do professor que faz uso do jogo como ferramenta educacional. Afinal de contas, armazenar os diálogos poderia inibir os alunos quanto às interações. O ideal é não manter registros dos diálogos e somente o quantitativo das interações, assim os alunos podem interagir livremente sem a impressão de estarem sendo vigiados.

4.3 Formação dos Grupos

Cada grupo de jogadores poderá ser formado por um ou mais aprendizes, ficando a critério de definições estabelecidas pelo professor. Uma vez que o JSubstance é aplicado em uma turma de alunos, é possível verificar que alunos têm mais afinidade para interagir uns com os outros. Dessa forma, pode-se identificar sub-grupos existentes dentro da turma. Com isso, tem-se dados que podem ajudar na promoção da integração de alunos que não interagem com outros. Obviamente que esses dados são disponibilizados apenas para os educadores desse processo.

Um dos principais fatores que determinam a eficiência na aprendizagem é a forma como os alunos são distribuídos em grupos. Os grupos são formados de maneira tal que alguns critérios pedagógicos pré-estabelecidos sejam respeitados. Mas, em que consiste uma boa formação de grupo? Tendo-se definido o que é um bom grupo, como dividir uma grande turma de alunos em bons grupos? Em (SERRA JR, 2001) são descritos fatores pedagógicos que têm sido utilizados para formar grupos em ambiente de aprendizagem, através dos Perfis de Aluno e Sociograma.

5. DISCUSSÃO E CONSIDERAÇÕES FINAIS

Atualmente, o JSubstance é composto por um módulo Cliente e um módulo Servidor. O módulo Cliente é manipulado pelo jogador e é nele que se encontra o jogo com seus recursos (arquivos de som e imagens). O módulo Servidor tem a responsabilidade de permitir que vários Clientes se comuniquem, recebendo informações, fazendo seu devido tratamento e as repassando aos jogadores. Além disso, o Servidor poderá armazenar informações pertinentes às interações entre os participantes.

Com relação às tecnologias selecionadas e utilizadas, mediante pesquisa e análise das mesmas, conseguiu-se integrá-las entre si e usar o que melhor elas têm a oferecer. Como as tecnologias usadas giram em torno da linguagem de programação Java, para que o jogo seja executado basta se ter um computador com a máquina virtual Java (*JRE - Java Runtime Environment*) para exercer a função de Servidor, então outros participantes à distância, também com a JRE em seus computadores, estariam interligados ao Servidor por meio de uma rede de computadores. Todas as tecnologias utilizadas no desenvolvimento da ferramentas proposta são gratuitas. O JSubstance pode ser também utilizado gratuitamente e sem limitações de uso. Inclusive, o jogo é independente de sistema operacional, ou seja, os participantes podem ter sistemas diferentes (Windows, Linux, etc). A codificação foi feita seguindo os conceitos da orientação a objetos e usando os padrões de projeto, com isso obteve-se um código de fácil extensão e manutenção, tornando o jogo adaptável à novas situações, mudanças e extensões.

O JSubstance ainda não está completamente concluído em seu desenvolvimento, muito embora já se tenha as funcionalidades básicas, tais como a comunicação em rede, o *chat* embutido e o controle das interações. Além disso, já tem-se parte da base de regras modeladas que inclui parte do conhecimento de Química abordado no jogo. Pretende-se, dentro em breve, disponibilizar para testes com alguns alunos do CEFET-MA, então fazer

os ajustes finais necessários para disponibilizar seu uso em aulas de Química Geral, alcançando assim seu objetivo que é de tornar o ensino mais lúdico e, conseqüentemente, mais produtivo para alunos e educadores.

REFERÊNCIAS

ADOBE. Disponível em: <<http://www.adobe.com/>> Acesso em: 07-09-2007.

CHEN, J. **Flow in Games**. Disponível em:
<[http://www.jenovachen.com/flowinggames/flowing.htm](http://www.jenovachen.com/flowingames/flowing.htm)> Acesso em: 07-09-2007.

BARBOSA, Laura Monte Serrat. **Projeto de trabalho: uma forma de atuação psicopedagógica**. 2.ed. Curitiba: L. M. S, 1998.

BASTOS FILHO, O.; AXT, M.; FONSECA, L.; LABIDI, S.; GUIMARÃES, L.; THOMAZ, A.; NASCIMENTO, E. **Jogos inteligentes como sistemas abertos: singularizando as possibilidades de interação com desafios lógicos**. Novas Tecnologias na Educação . V. 2, N. 2. CINTED-UFRGS , 2004.
<http://www.cinted.ufrgs.br/renote/nov2004/artigos/othon_jogos_inteligentes.pdf>

BOTELHO, Luiz. **Jogos educacionais aplicados ao e-learning**. Disponível em:
<http://www.elearningbrasil.com.br/news/artigos/artigo_48.asp> Acesso em: 07-09-2007.

BRACKEEN, David e BARKER, Bret e VANHELSUWÉ, Laurence.
Developing Games in Java. New Riders Publishing, 2004.

HORSTMANN, S. e CORNEL, G.. **Core Java volume I (Conceitos Básicos) e volume II (Recursos Avançados)**. Editora Pearson Makron Books. 2003.

OBJECT WEB. Disponível em <<http://www.objectweb.org/phorum/read.php?f=29&i=10258&t=10258>>
Acesso em 15-07-2007.

SERRA JR, G. (1999) “**Proposta de um Modelo de Avaliação para o Sistema SHIECC de Ensino Cooperativo Computadorizado**”. Monografia de Graduação. Bacharelado em Ciência da Computação. Universidade Federal do Maranhão].

SERRA JR, G. **Agente de Modelagem do Aprendiz para o Sistema MATHNET de Ensino Inteligente Cooperativo Computadorizado**. Dissertação de Mestrado. Pós-graduação em Engenharia de Eletricidade. Universidade Federal do Maranhão. 2001.

SUN. Disponível em: <<http://www.sun.com/>> Acesso em 10-09-2007.

SURI, R. **Introduction to Prolog**. Alpha Science Intl Ltd, Jun/2007.

TAROUCO, L.; ROLAND, L.; FABRE, M.; KONRATH, M. **Jogos educacionais**. Novas Tecnologias na Educação. CINTED-UFRGS, 2004. <<http://www.cinted.ufrgs.br/renote/mar2004/artigos/30-jogoseducacioanis.pdf>>.

TUPROLOG. Disponível em <<http://tuprolog.sourceforge.net/>> Acesso em: 15-07-2007.

Ugos' Web site. Disponível em:
<<http://www.ugosweb.com/jiprolog/index.aspx>> Acesso em: 07-09-2007.

AGRADECIMENTOS

Agradecemos ao Departamento Acadêmico de Informática do CEFET-MA pela disponibilização do Laboratório de Projetos para o desenvolvimento desta pesquisa.