

Criação de Sistemas de Aprendizagem Educacional Utilizando Jogos Sérios na Unidade do CEFET de Recife

**Bruno MACHADO TRAJANO (1); Anderson LUIZ SOUZA MOREIRA (2); Katyusco DE
FARIAS SANTOS (3)**

(1) Centro Federal de Educação Tecnológica de Pernambuco, Pernambuco, Jaboatão dos Guararapes, 1ª Trav. Rua
Sucupira do Norte, 13 - 54410-361, e-mail: bmtest@gmail.com

(2) Escola Agrotécnica Federal de Vitória de Santo Antão, e-mail: anderson.moreira@cefetpe.br

(3) Centro Federal de Educação Tecnológica de Pernambuco, e-mail: katyusco@cefetpe.br

RESUMO

Para interessados em programação a disciplina de lógica de programação é muito importante. Este artigo fala da aplicação e avaliação de um Jogo Sério, voltado para o curso de Sistemas de Informação do CEFET-PE que foca esta disciplina. Jogos sérios são ferramentas de aprendizado que combinam consigo o conteúdo educacional de uma área do conhecimento facilitando seu entendimento. Para conseguir resultados satisfatórios e cumprir as metas definidas é aplicado o conceito de usabilidade, uma característica que facilita a relação entre usuário e produto. Inicialmente foi realizada uma revisão bibliográfica para compreender os conceitos de jogos sérios e usabilidade de jogos, a jogabilidade. Processo de desenvolvimento de jogos eletrônicos e linguagens de programação mais utilizadas também foram analisados. Avaliou-se a opinião do público-alvo através de questionários. Utiliza-se C++ em conjunto com a biblioteca de desenvolvimento de jogos Allegro. Foram analisadas metodologias existentes para desenvolvimento de jogos eletrônicos computacionais, utilizando o processo que melhor se encaixa ao projeto. Ao final da pesquisa, pretende-se ter um estudo sobre desenvolvimento de jogos sérios com um protótipo jogável e de código-aberto, sobre usabilidade de jogos e sobre os impactos ocasionados nos usuários. Também se deseja disseminar em instituições de ensino o interesse por jogos sérios.

Palavras-chave: jogos sérios, usabilidade, aprendizado, desenvolvimento de jogos.

1. INTRODUÇÃO

Para aqueles que pretendem seguir a carreira de desenvolvedor, entender a disciplina de lógica de programação é algo imprescindível, visto que essa acompanha o trabalho de programação, seja qual for a linguagem de programação utilizada. Por isso mesmo, o projeto tratado aqui foi idealizado visando o desenvolvimento de uma ferramenta de aprendizagem voltado a esta disciplina. Dentre os tipos existentes foi escolhido o desenvolvimento de um Jogo Sério – que é uma modalidade de entretenimento (em *software*) desenvolvido de forma a mesclar entretenimento com o conteúdo da área de conhecimento que se deseja ensinar. Com isso se pretende atribuir ao processo de aprendizagem um aspecto mais leve e descontraído. Através de uma pesquisa entre o público-alvo decidiu-se que o jogo seria um RPG (do acrônimo em inglês: *Role Playing Game*), o estilo de jogo produzido. Isto ajuda o projeto, tendo em vista que este é um tipo de jogo que incentiva o raciocínio do jogador.

Para que o projeto alcance seu objetivo foram aplicados os conceitos de Usabilidade. Assim o usuário conseguirá tirar o máximo de conhecimento durante a interação com o sistema da maneira mais eficiente possível. O projeto também contou com a análise e adequação de processos de desenvolvimento para jogos eletrônicos computacionais, que inclui tanto a modelagem de suas interações e funcionalidades, quanto os testes do aplicativo.

No desenvolvimento deste projeto é aplicada a linguagem de programação C/C++. Focando especificamente a biblioteca de desenvolvimento de jogos Allegro. Isso se deve ao fato de que esta é uma das mais usadas por iniciantes em desenvolvimento de jogos. O que dá boas fontes de referências para a mesma e também por ter seu código aberto, possibilitando a adaptação de suas funções caso seja necessário. Outro bom aspecto é que assim, caso alguém deseje, poderá modificá-lo, para fins de melhorá-lo ou ampliá-lo, por exemplo.

2. BASES TEÓRICAS DO PROJETO

2.1. Fundamentos dos Sistemas de Aprendizagem

Os sistemas de aprendizagem são utilizados para facilitar a passagem de conhecimento para um indivíduo ou grupo de indivíduos. Desenvolvidos em *software*, são feitos para integrar a si mesmos conhecimentos que serão adquiridos por aqueles que os utilizam. Dessa forma, quando ocorre a interação, o usuário consegue assimilar o conhecimento passado pelo sistema e posteriormente aplicá-lo (ver Figura 1).

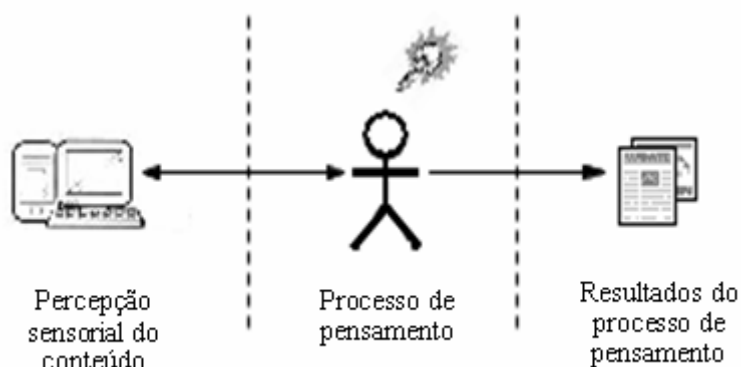


Figura 1: Demonstração de aprendizagem através de uma Sistema de Aprendizagem

Durante o desenvolvimento de um sistema desse tipo se deve atentar para as capacidades de interpretação do utilizador. Entendendo isso é possível alcançar o objetivo: ajudar no aprendizado. Sem este cuidado não há certeza de que o resultado final do uso deste artifício será o esperado. Observando este problema é indicado avaliar, a partir de uma observação do público-alvo, qual o meio que tem melhores chances de sucesso em sua aplicação. Pode-se, por exemplo, criar um sistema hipermídia de aprendizagem (Rezende, 2005). Um outro tipo são os jogos sérios, tratados no projeto e conceituados na próxima parte do artigo.

2.2. Conceituando Jogos Sérios

Devido ao grande sucesso que os videogames ganharam logo se pensou em aproveitar esta popularidade, não só entre os jovens, para a criação de ferramentas de suporte a aprendizagem. Estas são conhecidas como Jogos Sérios, onde o mundo do game é mesclado com o conteúdo educacional desejado. Assim, aproveitando-se das características inerentes este tipo de aplicativo, como a interatividade, o usuário pode absorver o conhecimento de forma quase que inconsciente.

Durante o desenrolar da interação o jogador acaba desenvolvendo as habilidades necessárias para alcançar o objetivo estabelecido pelo enredo do jogo. E paralelamente, também aprende e exercita o conteúdo que se deseja passar. Devido à participação ativa do jogador ajuda a trazê-lo mais próximo do assunto a ser "estudado", o que também contribui muito facilitar o aprendizado. Quando o usuário fica mais próximo do problema ele pode entendê-lo melhor e, com isso, resolvê-lo de maneira melhor (Debate, 2006).

Para que a interação com o jogo sério resulte em um bom aprendizado é importante garantir que o usuário não terá problemas em sua relação com o aplicativo. Desta maneira, a usabilidade é uma grande aliada durante o desenvolvimento, como será visto na seção a seguir.

3. USABILIDADE: MELHORANDO RELAÇÃO COM O USUÁRIO

Um produto, seja *software* ou não, precisa ter uma boa comunicação com seu usuário para alcançar seu objetivo satisfatoriamente. A usabilidade se trata exatamente disso, um bom jeito para defini-la seria: é uma característica desenvolvida e/ou aplicada durante o desenvolvimento de um produto, com a intenção de fornecer a máxima facilidade, eficiência, eficácia e conforto ao usuário.

Para garantir a aplicação deste conceito o projeto está se guiando pelas heurísticas da usabilidade estabelecidas por Jacob Nielsen (Zambalde, 2005), listadas e definidas a seguir:

- *Feedback*: sistema deve retornar uma resposta ao usuário de acordo com a atividade realizada, de forma a demonstrar sobre como sua ação foi interpretada e que ação foi/será realizada em resposta;
- Falar a linguagem do usuário: deve ser utilizado um linguajar correspondente com o utilizado por quem utilizará o produto/*software*;
- Saídas claramente marcadas: disponibilizar a opção de desfazer uma ação realizada sempre que possível;
- Consistência: uma determinada função realizada no sistema em um determinado tipo de situação sempre calhará em um mesmo resultado. Além disso, também abrange o conceito de que agrupar funções semelhantes juntas e dar nomes intuitivos as funções e atalhos;
- Prevenir erros: não deve haver ações possíveis ao usuário que possam causar um erro do sistema;
- Minimizar sobrecarga de memória: produto/aplicativo não deve exigir que quem o usa lembre-se de informações em demasia para poder usá-lo;
- Atalhos: o sistema deve oferecer atalhos para aumentar a eficiência durante sua operação;
- Diálogos simples e naturais: conteúdos de diálogos devem ser simples o suficiente para serem entendidas rapidamente, sem que seja necessário esforço para entendê-las;
- Boas mensagens de erro: as mensagens de erro devem dizer de forma direta e simples qual problema ocorreu, o provável motivo e uma solução se possível;
- Ajuda e documentação: o sistema deve ser acompanhado de uma documentação que o descreva, e também possuir tópicos de ajuda, que se possível devem estar presentes também no sistema.
- Além disso, o durante o projeto, foram realizadas entrevistas com o público-alvo para colher opiniões de características que deveriam ser incluídas no desenvolvimento, de forma a garantir que o sistema atenda suas necessidades.

3.1. Validação da usabilidade

Para garantir que o projeto foi desenvolvido com atenção e aplicação dos conceitos de usabilidade, será realizada uma avaliação deste conceito. A seguir serão citadas três formas de testes que podem ser utilizadas individualmente ou em conjunto umas com as outras.

Inspeção de Usabilidade

Realizada por avaliadores, especializados de preferência, em cima de aspectos da interface relacionados à usabilidade, para localizar problemas e se possível opinar sobre uma possível forma de saná-los.

Há quatro formas de fazê-la:

1. Avaliação Heurística: avaliadores verificam a interface segundo lista de princípios básicos de usabilidade – citados anteriormente;
2. Revisão de *Guidelines*¹ (linhas guias): análise de acordo com uma lista de regras/conceitos definidos para o projeto;
3. Inspeção de consistência: verifica solidez quanto à fundamentação de cores, formatos de E/S, *layout* etc.;
4. Percorso Cognitivo: simulação do usuário realizando suas tarefas típicas enquanto usa a interface.

Avaliação heurística é a preferida por ser mais rápida, fácil e barata que as outras. E por estas razões também será utilizada no projeto.

Teste de Usabilidade

Realizado com o auxílio do usuário, onde é definido um plano de testes contendo o que se deseja avaliar. Neste teste pretende-se avaliar a interface através de um questionário que focaliza o que foi definido no plano de testes. Durante o uso do aplicativo poderão ser encontrados problemas que não estão presentes nas perguntas da avaliação. Estas devem ser anotadas a parte, para que junto ao questionário sejam entregues aos avaliadores que acompanham o usuário durante o teste.

O teste de usabilidade, em geral, é realizado para:

- Ajudar no processo de desenvolvimento: quando é realizado durante o desenvolvimento, pois se forem identificados problemas com a usabilidade do produto avaliado, estes podem ser corrigidos;
- Avaliar a qualidade da interface: quando da finalização do desenvolvimento, a interface é testada para avaliar se está de acordo com os padrões, análises heurísticas ou *guidelines*, para satisfazer usuários e expectativas do projeto.

Teste de Comunicabilidade

Este teste é baseado na Engenharia Semiótica², verifica se a interface transmite as intenções e princípios de interação que guiaram o projetista. Utiliza-se um conjunto de interjeições comuns ao usuário para que ele demonstre em que situação ocorre uma quebra na comunicação com o sistema. Por exemplo, pode pedir para o usuário preencher um questionário com uma lista de componentes que se deseja avaliar e pedir para este colocar uma expressão correspondente ao lado, desta forma, quando não acha o componente “X” da lista ele colocaria uma expressão do tipo: “Cadê?”.

Através deste teste, pode ter uma visão melhor de como o usuário encara a interface do sistema, conseguindo, com isso, identificar novas dúvidas que possam surgir durante a interação com o sistema.

¹ *Guidelines*: Conjunto de regras e/ou conceitos que pretendem regulamentar o desenvolvimento/realização de uma tarefa, nesse caso um projeto de *software*.

² Engenharia Semiótica: Baseada na semiótica, ou seja, estudo dos signos/sinais. Ajuda a entender como os indivíduos interpretam as mensagens do sistema.

3.2. Usabilidade aplicada a jogos

Conhecida como **jogabilidade** utiliza os mesmo conceitos aplicados a interface de sistemas comuns, guardando as suas proporções. Para que o uso dos conceitos de usabilidade seja bem aplicado deve-se dividi-la em:

- Interface: forma de comunicação (gráficos) do jogo;
- Mecânica de jogo: regras que regem como deve funcionar o “mundo do jogo”;
- *Game play*: ocorre quando o jogador interage com a mecânica do jogo e outros jogadores.

A interface deve receber uma atenção especial, já que ela é um dos principais fatores que determinam o sucesso ou não de um jogo. Para que a interface faça sucesso ela deve ser atrativa, motivacional e entreter o jogador ao máximo. Para garantir isto é preciso que os testes de usabilidade sejam bem aplicados e conduzidos de forma a documentar os resultados obtidos e correções que precisarem ser realizadas.

4. CARACTERÍSTICAS DO PROCESSO DE DESENVOLVIMENTO

4.1. Metodologias de desenvolvimento

Para o pronto sucesso do desenvolvimento de um projeto de *software* é necessário a aplicação de um processo de desenvolvimento, ou seja, um conjunto de praticas adotadas durante o projeto para garantir que o objetivo seja atingido com eficiência e eficácia. Os jogos eletrônicos não fogem a regra.

Para estes tipos de sistemas há cinco fases gerais que são definidas e gerenciadas por uma metodologia aplicada:

- Concepção: nesta fase o “grupo” que deseja desenvolver o jogo verifica a viabilidade do desenvolvimento do projeto, sendo esta operacional, técnica, econômica e de cronograma. (Barros, 2007);
- Pré-produção: são definidos os elementos necessários para a construção do jogo, assim como detalhes da dinâmica e detalhes técnicos;
- Produção: onde ocorre a feitura do sistema, incluindo os recursos gráficos e de som necessários;
- Pós-produção: fase que engloba realização de testes do produto da fase anterior e correções de erros que sejam encontrados. Por fim, faz-se um documento *post mortem* do projeto, como forma de analisar os pontos positivos e negativos durante o desenvolvimento. (Carvalho, 2006);
- Pós-lançamento: onde se realizam atividades como, mais versões, expansões etc.

Durante o projeto acadêmico de pesquisa a metodologia adotada incluirá as fases de pré-produção, produção e pós-produção. Isso, pois estas são as partes praticas cabíveis de aplicação, devido a seu caráter acadêmico.

4.2. Metodologias existentes para desenvolvimento de jogos eletrônicos computacionais

Antes de apresentar a metodologia de desenvolvimento que é utilizada no projeto, é válido citar algumas das metodologias existentes para desenvolvimento de jogos eletrônicos.

Game Waterfall Process (GWP)

Baseada na metodologia cascata (*waterfall*) para desenvolvimento de *software*, que adota um perfil em fases lineares, isto é cada fase se sucede de outra. Por causa desta característica, uma mudança nos requisitos pode causar grandes problemas.

Outro ponto a ser observado, é sua orientação à documentação, ou seja, após cada fase um documento é gerado. Isso dá um bom suporte ao desenvolvimento, sendo que com isto é possível ter uma visão mais clara do objetivo do projeto e do que está sendo desenvolvido. Entretanto pode causar também o atraso do projeto, por a equipe estar ocupada gerando documentação e deixando de lado outras atividades.

Extreme Game Development (XGD)

Também baseada em uma metodologia existente, o *Extreme Programming* (XP) é adaptada a multidisciplinaridade das equipes de desenvolvimento de jogos. Por isso a maioria das suas características foi herdada. Características como: comunicação valorizada, simplicidade, *feedback*, coragem, respeito.

Por ter um caráter interativo, comporta bem mudanças nos requisitos do sistema o que proporciona bastante flexibilidade ao desenvolvimento. Podemos citar sua escassa produção de documentos de suporte ao desenvolvimento que pode causar algum tipo de inconsistências durante o projeto.

Game Unified Process (GUP)

Criada com o intuito de unir práticas da metodologia XP e RUP³ misturando as interações curtas da primeira metodologia com as longas da segunda, de forma a fazer uma correspondência com as necessidades do time multidisciplinar de desenvolvimento. Apesar de ser nova já é citada em diversos trabalhos acadêmicos e sites de referência do assunto.

Assim como a XGD, a GUP, por ser interativa tem uma boa flexibilidade, no entanto, devido a também ter semelhanças a RUP, baseia seu desenvolvimento em documentação que descreve os processos do sistema. Isso garante um rumo certo ao projeto e que não ocorram incongruências entre as ações do grupo envolvido.

4.3. Processo de desenvolvimento utilizado no projeto

O processo que será utilizado, e também que será produzido, no projeto envolve três das cinco fases de produção de um jogo, segundo a metodologia de desenvolvimento citada anteriormente: pré-produção, produção e pós-produção.

Durante a fase de pré-produção, realiza-se uma pesquisa com uma porção do público-alvo para coleta de requisitos, através de um questionário baseado em uma pesquisa previa sobre jogabilidade⁴ e usabilidade. Com isso pretende-se ter uma idéia das preferências do grupo de usuários, tanto em relação aos mecanismos de interação do usuário com o jogo, quanto em relação ao estilo do jogo preferido. Realiza a modelagem das interações, ou seja, de que maneira será realizada a interação com o jogador. Além disso, a modelagem da organização interna do sistema também é planejada, de acordo com as regras do paradigma⁵ de programação orientado a objeto e modelado através de UML⁶ (do acrônimo em inglês: *Unified Modeling Language*).

A fase seguinte se adiciona um aspecto interativo e incremental, envolvendo a produção e pós-produção. Na primeira parte há o desenvolvimento de uma porção do jogo para teste (pós-produção), tanto de funcionalidade quanto de usabilidade. Avaliações pelos usuários serão realizadas. Em seguida uma nova porção é gerada e testada individualmente. Se não forem encontrados erros esta é integrada à porção anterior e ocorre um novo teste. Esse processo se seguirá até a finalização, garantindo assim, total compatibilidade com o que foi estabelecido na fase anterior e a concordância do sistema com os conceitos da usabilidade.

Com a utilização desta metodologia pretende-se concluir o projeto de modo a atender suas expectativas iniciais, passando o conhecimento de lógica computacional da melhor maneira possível. Também se quer garantir a correta aplicação das *guidelines* adotadas, o que contribuirá para o aprendizado e conforto do jogador durante a interação com o jogo.

Dessa forma se aumenta a garantia de que o jogo agrada o público-alvo, atendendo o que foi definido na fase de pré-produção, tanto no que diz respeito à preferência dos usuários quanto em relação aos conceitos aplicados ao projeto.

³ RUP: *Rational Unified Process* – metodologia de desenvolvimento famosa e com grande suporte a documentação.

⁴ Resultado do processo de interação do jogador com o jogo.

⁵ Paradigma: o paradigma de programação determina a visão que o programador terá sobre a organização e execução do programa.

⁶ UML: linguagem de modelagem, que permite que desenvolvedores visualizem o programa e sua estrutura através de diagramas padronizados.

4.4. Comparação entre metodologias

Aqui é exibida uma tabela como forma de comparação entre a metodologia utilizada no projeto e as demais metodologias apresentadas anteriormente. São abordadas as seguintes características:

- Suporte a documentação: qualidade da documentação produzida para apoiar o desenvolvimento;
- Flexibilidade: trata-se da capacidade de resposta a mudança no ambiente e/ou requisitos;
- Comunicação: capacidade de integrar a equipe, de forma a informar sobre andamento e necessidades do projeto;
- Valor: trata-se da capacidade de gerar valor durante o processo de desenvolvimento;
- Testes: trata-se da relevância da aplicação dos testes, isto é, se os resultados serão válidos para um desenvolvimento eficiente e eficaz do sistema.

Processos de desenvolvimento	Características do Processo de Desenvolvimento				
	Suporte a documentação	Flexibilidade	Comunicação	Valor	Testes
GWP	✓✓✓	✓	✓	✓✓	✓
XGP	✓✓	✓✓✓✓	✓✓✓✓	✓✓✓✓	✓✓✓
GUP	✓✓✓✓	✓✓✓	✓✓✓✓	✓✓✓✓	✓✓✓✓
Processo utilizado no projeto	✓✓✓	✓✓✓	✓✓✓	✓✓✓	✓✓✓
Legenda: ✓: Ruim ✓✓: Regular ✓✓✓: Bom ✓✓✓✓: Excelente					

Tabela 1: Comparação entre metodologias

4.5. Allegro, a biblioteca C/C++

Amplamente utilizada por programadores de jogos iniciante é uma biblioteca de C++ de código aberto que oferece grande facilidade de programação, devido suas rotinas para trabalhar com gráficos, sons, entradas do usuário. Além disso, outra razão que levou a escolha desta para foi o fato de haver bastantes referências e com isso seria fácil achar ajuda caso preciso (*Game Development Community Network*, 2007).

Uma boa característica que é agregada ao projeto utilizando Allegro é de que após sua conclusão, será fácil fazer uma ampliação ou modificação do mesmo para que possa atender mais disciplinas, possivelmente da área de computação.

5. PRÁTICA DO DESENVOLVIMENTO

Seguindo o processo para a produção do jogo sério do presente trabalho, primeiramente foi realizada uma coleta de dados com um grupo de indivíduos do público-alvo. Isso foi feito através da aplicação de um questionário, cujas respostas eram direcionadas para possibilitar extração de detalhes necessários para compreender questões relevantes, já citadas neste artigo. A partir dos resultados dos questionários foi desenvolvido um roteiro para o jogo, descrição das interações básicas, o estilo (RPG de aventura) e os requisitos básicos do projeto.

Com estas características definidas, deu-se início à modelagem. Primeiramente foi planejada como seria a interação entre o jogador e o jogo, observando-se os conceitos de usabilidade. Em seguida, foram feitos os

diagramas, que definem funcionalidades e a organização lógica do aplicativo. Para que desta forma evite equívocos durante sua construção.

Com a estrutura estabelecida, foi iniciada a programação do jogo, para isso foram selecionadas duas formas possíveis, um engenho de desenvolvimento⁷ ou programação em C++ com Allegro. Inicialmente se tentou utilizar o engenho de desenvolvimento. Após algum tempo de uso se percebeu que a ferramenta não iria atender às necessidades do projeto. Então se partiu para a segunda opção, mais flexível em relação à outra (ver Figura 2).



Figura 2: Imagens do jogo desenvolvido utilizando Allegro.

Durante o desenvolvimento, foi muito observado o conceito de orientação a objetos⁸, visto que com sua utilização alterações e correções no código-fonte⁹ se tornariam mais rápidas e simples. Sendo esta parte do desenvolvimento realizado incrementalmente, onde em cada incremento é feita uma parte do sistema. Ao final de cada porção do jogo sério, foi feito um teste para averiguar sua correspondência com os requisitos. Preocupava-se também em avaliar sua usabilidade. Caso tudo estivesse de acordo, seguia-se para o próximo incremento.

Ao final do projeto realizaram-se testes com indivíduos pertencentes à população de futuros usuários observando o grau de satisfação dos jogadores ao se utilizar o jogo, através de observação e questionários. E, após algum tempo de uso do sistema, verifica-se o impacto que o uso do jogo teve em relação ao desempenho na disciplina de lógica de programação. Para isso, aplicam-se testes que cobre o conteúdo abordado no jogo e realizam-se entrevistas com os usuários do jogo sério.

6. RESULTADOS ALCANÇADOS

Como resultados teóricos do projeto, foram possíveis, primeiramente, compreender como as ferramentas de auxílio a aprendizagem funcionam, em especial os jogos sérios, a forma mais eficaz de desenvolvê-las e aplicá-las. Também foi aprendido como se estrutura a produção de um jogo eletrônico, quais elementos o compõem e quais suas fases e as competências de cada uma. Além disso, entendeu-se como a usabilidade é importante para a construção de um sistema, não só um game, e que sua aplicação contribui drasticamente para a interação com o usuário.

De forma prática, foi feito todo o projeto de interações e funcionalidades do aplicativo, tendo em vista os aspectos de usabilidade. Combinando a isso, uma pesquisa que contribuiu para o entendimento do público-

⁷ Engenho de desenvolvimento: ferramenta que agrega tanto funções para codificação quanto para manipulação de gráficos e sons.

⁸ Orientação a objetos: paradigma de análise, projeto e programação de sistemas de software, que tem como base a interação entre objetos.

⁹ Código-fonte: conjunto de instruções escritas em uma determinada linguagem de programação que são responsáveis pelo funcionamento de um *software*.

alvo. Acrescentando-se a isso o processo de desenvolvimento adotado ao projeto que compreende também a fase de testes, de grande importância ao objetivo final do projeto.

Devido à falta de conhecimento prévio da complexidade e extensão do projeto de um jogo eletrônico computacional, bem como da grande variedade de conceitos envolvidos com um jogo sério, o projeto não conseguiu terminar o protótipo para aplicação e testes com os usuários. Sendo assim, a uma possível continuação do projeto, competiria ao termino do desenvolvimento do projeto, de acordo com o planejamento já feito, e a aplicação dos testes necessários e já mencionados.

7. CONCLUSÕES

O jogo sério pode ser de grande auxílio a aprendizagem. Para alcançar esta meta, o projeto teve o cuidado de atender as fases originais da construção de um jogo eletrônico. Entretanto, estas fases precisam ser adaptadas de forma a atender aos requisitos estabelecidos e conceitos adotados. Entre eles está a aplicação de usabilidade. Com sua integração ao processo haverá um aumento na capacidade de aprender do usuário.

Outro aspecto importante será a realização de testes visando se certificar do respeito às diretrizes do projeto, tanto em relação às necessidades do game quanto à jogabilidade. Sendo os testes realizados com um grupo de usuários do público-alvo.

Uma dificuldade observada sobre o desenvolvimento do projeto é a ausência de referências mais completas e consistentes relativas a metodologias desenvolvimento voltadas para jogos. Isso tornou um pouco mais complicado o trabalho, já que se torna necessária a adaptação de processos existentes. O que aumenta o tempo gasto com as fases iniciais do projeto. Outro fato a que se deve dar atenção é a multidisciplinaridade do desenvolvimento de *games*, que envolve diversas áreas do conhecimento.

Com a conclusão do projeto, além de um protótipo prático, usável e testado de código aberto e uma avaliação sobre a reação dos usuários e o processo de desenvolvimento. Também quer disseminar um novo campo de interesse nas instituições de ensino.

REFERÊNCIAS

Allegro Wiki – Disponível em: <http://wiki.allegro.cc/Main_Page>

ARAÚJO, A. R. S. **Agile Game Process: Metodologia Ágil para Projetos de Advergimes**. 2006. Trabalho de Graduação – Universidade Federal de Pernambuco, Recife, 2006.

BARBOSA, S. D. J. **Projeto de avaliação de sistemas interativos**. In: Notas de aula. Departamento de Informática – PUC-RIO, Rio de Janeiro, 2002.

BARROS, R. L. B. de. **Análise de Metodologias de Desenvolvimento de Software aplicadas ao Desenvolvimento de Jogos Eletrônicos**. In: Trabalho de Graduação – Departamento de Informática – Universidade Federal de Pernambuco, Recife, 2007.

BARROS, V. T. de O. **Avaliação da interface de um aplicativo computacional através de teste de usabilidade, questionário ergonômico e análise gráfica do design**. In: Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina como requisito parcial para obtenção do grau de Mestre em Engenharia da Produção. Universidade Federal de Santa Catarina, Florianópolis, 2003.

CARVALHO, G. H. P. de. **Um Modelo Preditivo para Desenvolvimento de Jogos de Computador**. In: Trabalho de Graduação – Departamento de Informática – Universidade Federal de Pernambuco, 2006.

CARVALHO, G. H. P. de. **Um Modelo Preditivo para Desenvolvimento de Jogos de Computador**. In: Trabalho de Graduação – Departamento de Informática – Universidade Federal de Pernambuco, 2006.

Debate Lúdico: Jogos Sérios, Disponível em:

<<http://debateludico.blogspot.com/2006/04/jogos-srios.html>> Acesso em abril de 2008.

DEMACHY, T. *Extreme Game Development: Right on Time, Every Time*. Disponível em:
<http://www.gamasutra.com/resource_guide/20030714/demachy_pfv.htm>

FLOOD, K. *Game Unified Process (GUP)*. Disponível em:
<<http://www.gamedev.net/reference/articles/article1940.asp>>

Game Developing Community Network, Disponível em:
<<http://www.allegro.cc/>> Acesso em: setembro de 2007.

Game Developing Community Network. Disponível em:
<<http://www.allegro.cc/>>

GURGEL, I.; ALMEIDA, E. W. M.; ARCOVERDE, R. L.; SULTANUM, N. B.; TEDESCO, P.. **A Importância de Avaliar a Usabilidade dos Jogos: A Experiência do Virtual Team**. In: V Brazilian Symposium on Computer Games and Digital Entertainment, 2006, Recife. DIGITAL PROCEEDINGS of the V Brazilian Symposium on Computer Games and Digital Entertainment, 2006.

KOIVISTO, E. M. I.; KORHONEN, H.. *Mobile Game Playability Heuristics*". In: Forum Nokia. Version 1.0, 2006.

MCGUIRE, R. *Paper Burns: Game Design with Agile Methodologies*. Disponível em:
<http://gamasutra.com/features/20060628/mcguire_01.shtml>

REZENDE, Flávia; BARROS, S. L. S.. **A Hipermídia e a Aprendizagem de Ciências: Exemplos na Área de Física**. Física na Escola, São Paulo, v. v. 6, n. 1, p. 63-68, 2005.

SANTOS, R.; MAIA, F.. **A Importância da Usabilidade de Interfaces para a Qualidade do Aprendizado Mediado pelo Computador**. Brasil.

ZAMBALDE, A. L.; ALVES, R. M.. **Interface homem-máquina e ergonomia**. In: Curso de pós-graduação "Lato Sensu" (Especialização) à distância: administração de sistemas de informação. 1ª ed. Editora da Universidade Federal de Lavras. Fundação de Apoio, Ensino, Pesquisa e Extensão, Lavras – Universidade Federal de Lavras, 2005.