Experiências no Ensino de Projeto de Jogos Digitais

Luís Naito Mendes Bezerra Luciano Silva Ismar Frango Silveira
Thiago Hitoshi Kuma Carlos Fernando de Araújo Jr

Universidade Cruzeiro do Sul, Centro de Ciências Exatas e Tecnológicas, São Paulo, Brasil

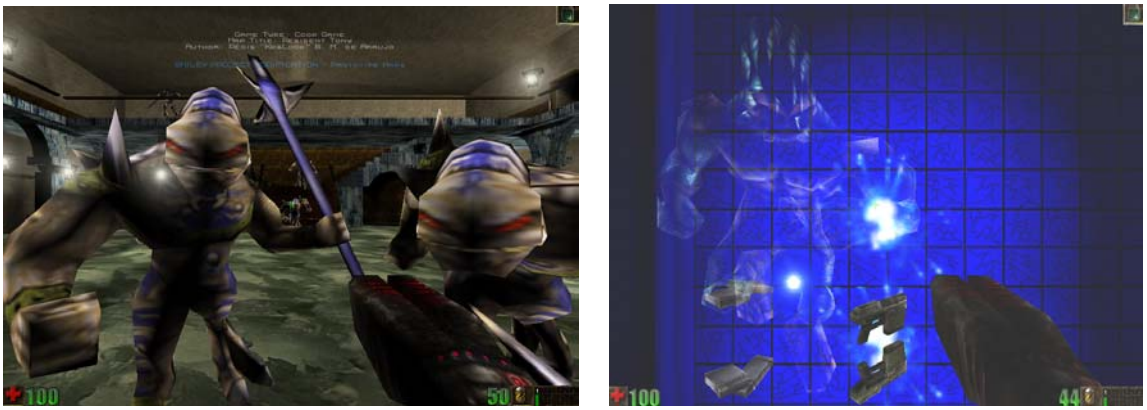

Figura 1:Projetos de Jogos elaborados pelos alunos de 2º semestre de 2005 do CST de Jogos Digitais da UNICSUL.

Resumo

Este artigo descreve uma experiência com um Curso Superior Tecnológico em Design de Jogos Digitais, cujo eixo norteador é o processo de projeto de jogos com complexidades crescente durante o curso. Neste contexto, é necessário verificar a questão das técnicas e princípios de engenharia de software aplicados ao processo de produção de jogos. Será mostrado como o currículo foi elaborado de maneira a seguir as fases de do processo de engenharia de software, permitindo assim aos alunos construírem seus próprios portfólios a partir de projetos de jogos desenvolvidos ao longo do curso.

Palavras-chave: projeto de jogos, engenharia de software aplicada a jogos, currículo

Abstract

This paper describes an experience in a Digital Games Design Undergraduate Course, whose core axis is the game design process with an increasing growing complexity during the course. In this context, it is mandatory to verify the question of software engineering techniques and principles applied to game production process. It will be shown how curriculum was elaborated in order to follow the software engineering process phases, thus allowing students to have a set game projects to build their own portfolios during the course.

Keywords: game project, software engineering applied to games, curriculum

Contatos:
{luis.naito,ismar.silveira,luciano.silva,
thiago.kuma,carlos.araujo}@unicsul.br

1. Introdução

O sucesso de jogos eletrônicos em computadores pessoais e videogames fazem com que atualmente tal forma de entretenimento seja também utilizada em dispositivos mais simples, como telefones celulares.
Apesar do sucesso na área industrial/comercial, o estudo sobre jogos de computador é um campo pouco explorado na área acadêmica. Esta situação era criada por uma série de fatores como a falta de instituições de pesquisa interessadas no tema, a escassez de livros e congressos, bem como de especialistas no meio acadêmico. Porém, este cenário começa a mudar. De fato, pode-se dizer que poucas são as áreas da computação onde é possível se explorar tantos domínios do conhecimento quanto à de jogos. No campo da computação, podemos citar como exemplo às de computação gráfica, inteligência artificial, computação musical, redes de computadores, engenharia de software, dentre outros.

De acordo com Fuks [2002], ao se projetar e desenvolver software, é necessária uma notação comum entre os envolvidos para que eles possam documentar e debater sobre o projeto do sistema. Esta notação comum reduz a chance de má interpretação, que pode acarretar mudanças na fase de implementação, normalmente associadas a altos custos. Segundo Perucia et al [2005], o processo de desenvolvimento de jogos apresenta uma série de etapas, detalhadas a seguir.

A primeira etapa é a de reunião criativa: fase em que as idéias do jogo são expostas e discutidas por todos. Cada idéia é discutida abrangendo originalidade, inovação, público-alvo, plataforma e possibilidades de mercado. Após o processo de reunião criativa, já se tem o projeto que deve ser desenvolvido. Inicia-se, então, o processo de *game design*.

A segunda fase é a de *Game design*, que é o processo onde são descritas as características principais do jogo como jogabilidade (*gameplay*), controles, interfaces, personagens, armas, golpes, inimigos, fases e todos os aspectos gerais do projeto. Durante essa fase, é elaborado o ***design document***, um documento que descreve as características citadas anteriormente em detalhes. O *design document* funciona como um roteiro de cinema. Com base nessas informações, os artistas irão criar o visual e os programadores desenvolver a interatividade do produto.

Durante a fase de *game design*, os artistas começam a trabalhar na concepção dos personagens e na identificação visual do jogo. Também nessa fase, os programadores fazem a modelagem do software e implementam a estrutura de classes do produto.
Com o *design document* pronto, o gerente de projeto já pode ter noção do tamanho do produto e quanto tempo será necessário para concluí-lo.

Já o *level design* é um mapa geral com os desafios e missões que o jogador deve cumprir para vencer a fase. Enquanto o *level design* esta sendo desenvolvido, os artistas e programadores trabalham em características mais gerais do projeto que não são específicas a cada fase. Com o *level design* concluído, os envolvidos já podem dedicar-se à produção das fases.

A partir dessa fase, são produzidas versões intermediárias do produto, que permite ao gerente um acompanhamento da evolução do produto. Essas versões intermediárias também possibilitam detectar *bugs* que surgem durante o desenvolvimento. Esses *bugs* são corrigidos para a próxima versão, o que acaba por aumentar a estabilidade do jogo até a sua versão final.

Assim, o produto evolui de versão em versão até atingir a versão Beta. A versão Beta já contém todas as fases do jogo e toda a interatividade. Quando essa versão é concluída, inicia-se o ciclo de testes para detecção de bugs e coleta de sugestões que podem melhorar o produto. O ciclo de testes gera refinamentos tanto na programação quanto na arte do jogo para que este se torne um produto mais atrativo ao consumidor. Esses refinamentos continuam até a conclusão da versão *Gold*, que é o produto final.

A figura 2 a seguir ilustra as etapas necessárias para o desenvolvimento de um jogo.

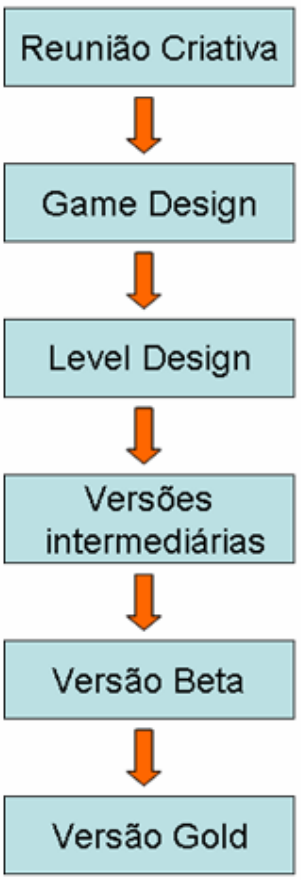

Figura 2: Etapas para o desenvolvimento de um jogo

2. Trabalhos relacionados

Diversos autores vêm adaptando os processos clássicos de Engenharia de Software para o desenvolvimento de jogos. [Flynt e Salem, 2004], [Rucker, 2002].

O Rational Unified Process (RUP) [2001], ou o processo de engenharia de software RUP é um processo que oferece uma abordagem baseada em disciplinas, tarefas e responsabilidades para uma organização de desenvolvimento, tornado-se mais bem aplicado quando um projeto em específico é o foco. A particularização de um processo para um projeto garante a produção do software com alta qualidade.

Quando Probasco [2001] propõe uma análise dos dez aspectos essenciais do RUP elege:

1. Desenvolvimento da Visão
2. Elaboração de um Plano
3. Identificação e Mitigação dos Riscos
4. Atribuição e Acompanhamento
5. Exame do Caso de Negócio
6. Concepção de uma Arquitetura de Componentes
7. Construção e Teste Incremental do Produto
8. Verificação e Avaliação dos Resultados
9. Gestão e Controle das Alterações
10. Fornecimento de Suporte ao Utilizador

Desenvolvimento da Visão

A visão que o usuário ou o cliente tem do produto a ser desenvolvido é especificada no nível das principais necessidades dos envolvidos e recursos do sistema, isto é analisando o problema, compreendendo as necessidades dos usuários chaves e gestão dos requisitos à medida que se vão alterando.
Em termos de desenvolvimento de jogos, segundo as definições de Perucia et al [2005], a reunião criativa é base para uma visão, lembrando que o desenvolvimento de jogos em geral a necessidade do usuário parte da pressuposição dos elementos da equipe e mesmo um jogo sobre encomenda essa necessidade não é clara, e a equipe acaba direcionando o projeto através de versões de demonstração.

Elaboração de um Plano

Conhecido também de SDP - *Software Development Plan*, incorpora todas as informações para se gerir um projeto, portanto é atualizado constantemente.

Além da organização, a gestão dos requisitos é a principal tarefa do plano, nas primeiras fases o plano deve constar à escolha de plataforma, engines, ferramentas e bibliotecas. Uma versão é necessária para validar tecnologias deparadas com os requisitos.

Identificação e Mitigação dos Riscos

Ao iniciar um projeto essas tarefas levam a identificação e combate dos aspectos de maior risco, sendo estes itens de grande importância no Plano de desenvolvimento.

Mitigar riscos em desenvolvimento de jogos fatalmente remetem a limitações e requisitos:

- Limitações de plataforma, por exemplo, dispositivos móveis e seu poder de processamento.
- Relação requisitos dos jogos e a tecnologia para alcançá-la, por exemplo. *engines* e bibliotecas.
- Relação entre jogabilidade e questões sobre Interface Humano Computador

Atribuição e Acompanhamento

A análise contínua de resultados e objetivos serve de base para gerir aspectos técnicos e riscos. Nas fases iniciais o sucesso do jogo representa um trabalho de qualidade principalmente no *level design*. Nas fases intermediárias as questões de desenvolvimento e testes das versões Beta [Perucia et al 2005], sendo a finalização baseada em testes finais de aceitação e desempenho.

Entre os principais aspectos a serem observados, ou acompanhados, destacam-se ():

- Personagens (características)
- Física
- Inteligência Artificial
- Elementos de navegação (telas e fluxo)
- Arte (2D, 3D, texturas e animações)
- Som (efeitos sonoros e música)

Exame do Caso de Negócio

Pontualmente neste aspecto a determinação de justificativa de investimento no projeto é o seu âmago. À medida que o projeto avança são avaliados os pontos de retorno do investimento (ROI – *Return of Investment*).

O investimento em produção, teste e distribuição de jogos torna esse aspecto complexo, necessitando o apoio principalmente de profissionais com conhecimento empírico, pois manobras podem ser adotadas, em um projeto de jogo elaborado para celulares, a adoção de uma internacionalização do jogo trouxe a sua viabilidade.

Concepção de uma Arquitetura de Componentes

O RUP permite através de suas atividades elaborar, conceber, desenvolver e validar arquiteturas. Na disciplina de Plataforma para Jogos (curso de Tecnologia em Design de Jogos Digitais da Universidade Cruzeiro do Sul), são classificadas arquiteturas baseadas na divisão de gerenciadores:

- Entrada
- Inteligência Artificial
- Multiplayer
- Principal
- Gráfico
- Som
- Objetos

Cada gerenciador é parte da arquitetura e internamente é concebida por componentes. Segundo RUP [2001] componente é qualquer parte do sistema não trivial, independente e substituível.

A composição da arquitetura deve estar embasada nas definições de requisitos e decisões de riscos, principalmente quanto a atender requisitos funcionais e não funcionais de cada elemento.

Construção e Teste Incremental do Produto

A codificação, construção e teste dos componentes do sistema, de forma incremental, ao longo do ciclo de vida do projeto, devem em termos de jogos, a obediência de cada um dos pontos de verificação das versões Beta.

Verificação e Avaliação dos Resultados

Demonstra resultados de uma iteração, ou seja, versões. Esse aspecto além de base para a Mitigação de Riscos, define nas iterações iniciais de edificação de jogos a adoção de frameworks, engines e bibliotecas.

Gestão e Controle das Alterações

As versões Betas propostas por Perucia trazem garantia de fidelidade de aceitação dos usuários finais, um retorno de melhorias e modificações de requisitos. Esse controle de alterações no desenvolvimento de jogos não cumpre seu papel a não ser que os profissionais líderes do projeto saibam que novas versões de jogos sobrevivem de novidades que em geral nascem de itens contidos nesse controle de alterações.

Fornecimento de Suporte ao Utilizador

A conclusão e entrega de produto, que no caso é um jogo, remete a criação de tutorias de recursos de jogos, demonstrações e suporte.
Pacotes de correção e inclusão de características mostram-se de grande valia para jogos, tornando-os um patamar de novidade por mais tempo.

3. Ensinando Projeto de Jogos

A Figura 3 apresenta o Mapa Conceitual [Novak, 1998] das disciplinas do curso.

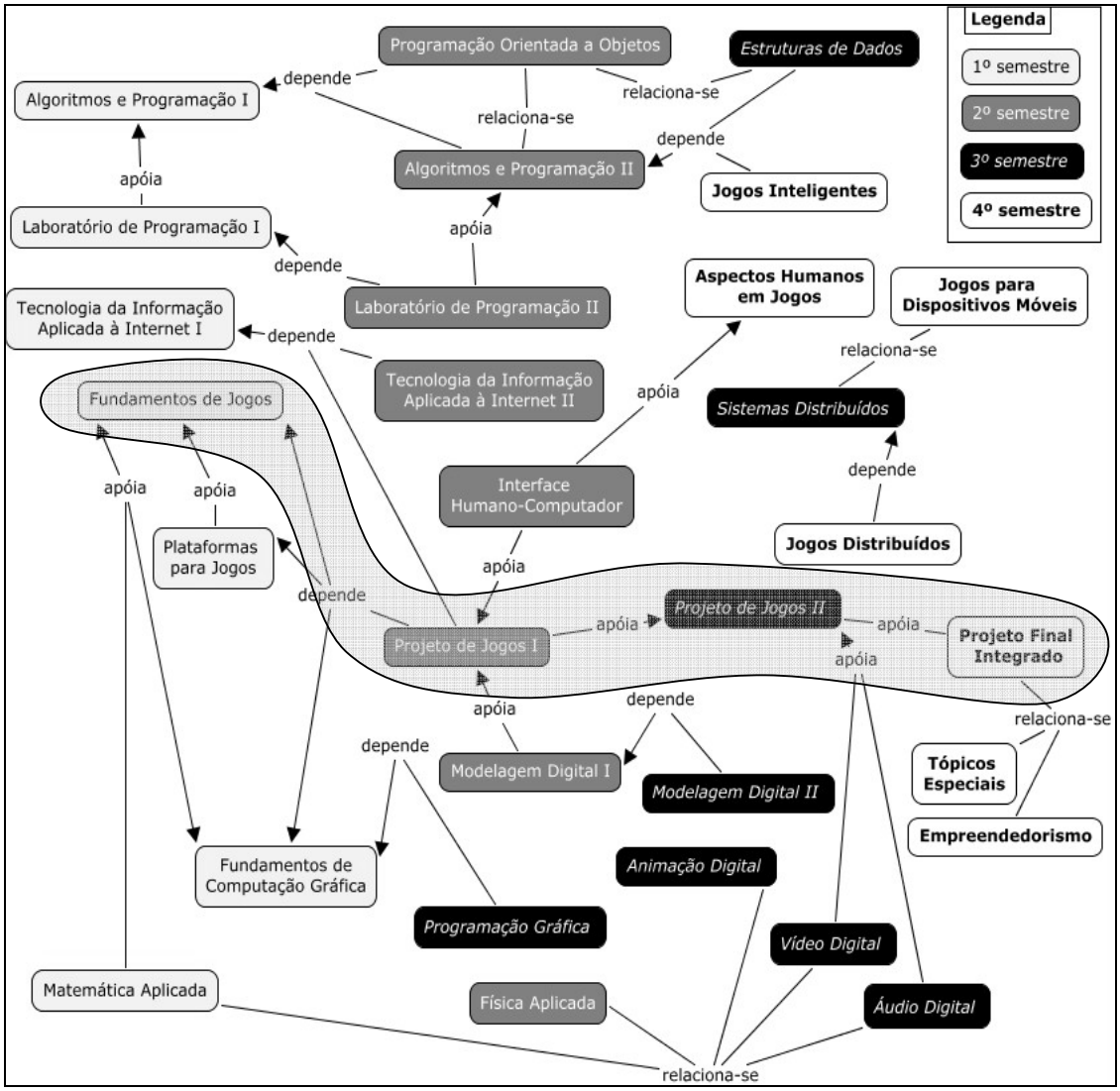

Figura 3: Mapa conceitual do Curso Superior de Tecnologia em Jogos Digitais da UNICSUL, destacando as disciplinas específicas relacionadas a Projeto de Jogos

O curso superior de Tecnologia em Design de Jogos Digitais da Universidade Cruzeiro do Sul foi concebido no ano de 2004 por uma comissão de professores interessados na implantação de um curso inovador que viesse a atender às expectativas de um mercado emergente e contemplasse os anseios da comunidade acadêmica e da administração superior da Universidade. Ao oferecer um curso com características tão inovadoras quanto específicas, a Universidade está cumprindo o seu papel ao contribuir para o desenvolvimento tecnológico nacional.

O curso tem duração de quatro semestres letivos (2 anos), dos quais os dois primeiros semestres são dedicados aos fundamentos de criação de jogos e os últimos, a tópicos mais avançados. O curso tem como um de seus pilares o desenvolvimento de projetos integrados em cada semestre, com ordem crescente de complexidade e tamanho de equipes. Tal abordagem possibilita relacionar de maneira mais efetiva as disciplinas de um determinado semestre, bem como colocar os alunos nas mais diversas situações de trabalho em equipe.

No primeiro semestre, a disciplina "Fundamentos de Jogos" apresenta fundamentos da tipologia de jogos e conceitos relacionados, como *storyboards*, *storytelling*, definição de personagens e itens, *level design*, mecânica do jogo, áudio e fundamentos do processo de teste. Na implantação inicial (1º. Semestre de 2005) e na atual (1º. Semestre de 2006) utilizou-se a ferramenta 3D Game Studio™[Conitec, 2005].

O projeto foi realizado por cada aluno individualmente. A Figura 4 exibe alguns *screenshots* dos resultados obtidos:

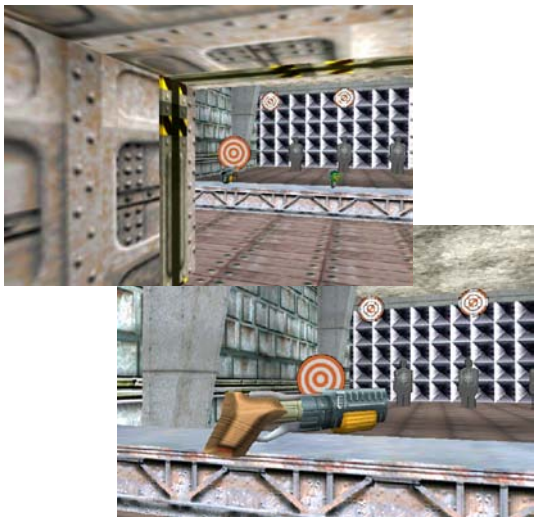

Figura 4: *Screenshots* do primeiro projeto do curso

Para obter maior controle da interdisciplinaridade, foram incluídas somente as disciplinas Fundamentos de Jogos e Fundamentos de Computação Gráfica. A primeira disciplina, conforme discutido anteriormente, tinha o objetivo de fornecer uma visão geral do processo de projeto e desenvolvimento de jogos digitais, enquanto que a segunda disciplina fornecia os conceitos básicos para modelagem, *rendering* e animação de pequenas cenas. Vários modelos foram fornecidos pela disciplina Fundamentos de Jogos para que os alunos precisassem apenas compor a cena de acordo com o projeto.

No segundo semestre do curso, como os alunos já possuem uma visão geral do processo de produção de jogos, inicia-se o estudo mais aprofundado das tarefas iniciais deste processos. A disciplina "Projeto de Jogos de I" apresenta os fundamentos de Engenharia de Software aplicados a jogos. Neste semestre, é desenvolvido um projeto integrado, no qual os alunos decidem o planejamento de um jogo, sob tema e tipologia orientados pelo professor de "Projeto de Jogos I". Este projeto tem como requisitos principais a utilização de estruturas mais complexas de modelagem, efeitos de dinâmica e interfaces mais elaboradas, de maneira a incentivar sistematicamente estudos interdisciplinares.

A disciplina aglutinadora para interdisciplinaridade foi Projeto de Jogos I, que aplica técnicas convencionais de Engenharia de Software às fases de *game design*, *level design*, implementação e testes. Na primeira versão da disciplina, os alunos foram divididos em grupos formados por três alunos e foi proposto o desenvolvimento de um jogo no gênero FPS (*First-Person Shooter*) que atendesse aos seguintes requisitos:

- recursos mais sofisticados de modelagem como controle de malhas por rigging, disponibilizados pela disciplina de Modelagem Digital I;
- recursos de dinâmica, cuja teoria é preparada pela disciplina de Física Aplicada;
- recursos de interação humano-computador, apresentados pela disciplina Interface Humano-Computador;
- forte documentação de projeto e níveis, bem controle gerenciamento do processo produtivo de software, abordados na disciplina Projeto de Jogos I.

Ao final do semestre, os alunos apresentaram os projetos desenvolvidos através de uma sessão pública, onde cada membro da equipe comentou e destacou as dificuldades e facilidades de trabalho na equipe. O desenvolvimento foi realizado utilizando-se o *game engine* Unreal Tournament 2004 [Epic Games, 2006], com modelagem de personagens efetuada em Maya [Autodesk, 2006]. Além da apresentação visual do jogo, foram entregues as documentações de projeto e de níveis, bem como o cronograma de desenvolvimento.

Após a finalização de todas as apresentações, foram eleitos os melhores jogos das equipes, com premiação formada por livros de projeto e desenvolvimento de jogos. Os votos da eleição contemplavam os alunos da disciplina e professores envolvidos nas respectivas disciplinas participantes do projeto interdisciplinar. O jogo vencedor era formado de um pequeno castelo invadido por alienígenas. O objetivo deste jogo era alcançar uma pequena chave localizada no andar superior do castelo, que constitua a segunda fase da estória. Na Figura 5, a seguir, é mostrado um screenshot do primeiro lugar da votação.

Figura 5: *Screenshot* do primeiro lugar do concurso

Desta forma, ao final do primeiro ano, tem-se um aluno com uma visão geral do processo de desenvolvimento de jogos, sua dificuldade e seus requisitos iniciais, além de habilidades essenciais de programação. Para levar este aluno a situações mais complexas, no segundo ano do curso lhe são apresentados detalhes mais refinados de produção e aplicados os recursos de programação nos mais variados aspectos de implementação.

No terceiro semestre, a disciplina "Projeto de Jogos II" prossegue com os conceitos essenciais de Engenharia de Software, com ênfase maior a testes e qualidade. Nela, são discutidos os vários tipos e técnicas de testes, bem como padrões de qualidade para processos de desenvolvimento.

Neste semestre, o projeto consistiu na produção de um jogo estilo arcade totalmente programado no game engine Ogre3D [2006], disponível sob LGPL. Além das usuais exigências de Engenharia de Software e Modelagem, grande parte do foco do projeto é voltado para implementação eficiente dos documentos de projeto e de níveis.

Para este projeto, cada grupo formado por quatro alunos recebeu o mesmo projeto de jogo e de níveis de complexidade média, denominado "Pegue o Cara!" e, como tarefa inicial de desenvolvimento, realizaram as seguintes tarefas:

1. Análise do documento de jogo
2. Análise do documento de níveis
3. Elaboração de cronograma de desenvolvimento e atribuição de tarefas

As tarefas incluem modelagem, animação, produção de áudio e programação em Ogre3D. Foi exigido que cada grupo apresentasse, também, um esquema de rotatividade nestas tarefas, de tal forma que todos os elementos da equipe pudessem experimentar as dificuldades de cada uma delas.

O jogo "Pegue o Cara!" é formado por duas equipes de jogadores, onde cada uma delas tem o objetivo de capturar um determinado elemento do jogo ("o Cara") e jogá-lo dentro de uma lixeira. O rascunho inicial de cada jogador e "do Cara", bem como seus movimentos, foram fornecidos aos alunos e são reproduzidos na Figura 6.

Figura 6. Rascunhos e movimentos básicos do jogo "Pegue o Cara!"

As tarefas básicas de modelagem residiram inicialmente em escolher qual a técnica de modelagem mais adequada a ser utilizada. Como o *engine* Ogre3D suporta de maneira muito eficiente modelos poligonais, decidiu-se pela utilização da bem-conhecida técnica de *Low-Polygon*. Personagens foram modelados segundo esta técnica e passaram, posteriormente, pela fase de *rigging* (esqueletamento) para composição dos *frames* de movimentação. Para que os modelos gerados pudessem ser testados na ferramenta Ogre3D, foram necessárias conversões para o formato MESH, o que permitiu aos alunos experimentar os problemas decorrentes de conversão tridimensional.

Partes das equipes responsáveis por esta tarefa inicial de modelagem faziam testes sistemáticos na ferramenta Ogre3D para verificar o peso dos modelos numa execução simples contra um terreno poligonal. Esta estratégia permitiu reforçar a técnica de prototipação rápida, apresentada no semestre anterior.

Concomitantemente à tarefa de modelagem dos personagens, a outra parte da equipe iniciou a tarefa de geração dos cenários nos níveis. O jogo é formado por três níveis, onde o nível de dificuldade aumenta pelo posicionamento das lixeiras em locais de difícil acesso e várias barreiras. O esquema básico do primeiro de três níveis do jogo "Pegue o Cara" é mostrado na Figura 7:

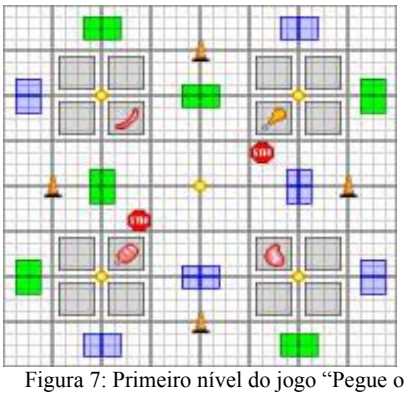

Figura 7: Primeiro nível do jogo "Pegue o Cara!"

O último semestre representa o ponto culminante do curso: os alunos já possuem uma visão bastante refinada do processo de produção de jogos, como também maturidade elevada para vários aspectos de implementação como recursos gráficos, estruturas de dados e ambiente distribuídos, além de estarem habituados ao trabalho em equipe. Assim, é possível exigir um projeto de complexidade mais elevada, com recursos já estudados nos semestres anteriores e utilização dos recursos a serem desenvolvidos no semestre, bem como utilização de uma equipe de trabalho um pouco maior. A disciplina "Projeto Final Integrado" tem por responsabilidade acompanhar o desenvolvimento do projeto final do curso.

4. Conclusões

Com o presente artigo, foi possível demonstrar como a construção de um currículo tendo como bases norteadoras as fases do processo de engenharia de software permitiu aos alunos a desenvolverem seus próprios projetos de jogos ao final de cada semestre, permitindo aos mesmos a construção de portfólios individuais, o que enriquece o currículo do estudante, aumentando seu potencial de empregabilidade neste nicho de mercado em franca expansão no país.

Vale ressaltar que os resultados aqui apresentados ainda são parciais, visto que a disciplina "Projeto Final Integrado" está sendo oferecida pela primeira vez no 2º semestre de 2006, de forma que os autores não possuem os resultados finais dos projetos dos alunos para uma análise mais conclusiva.

Trabalhos futuros incluem aprimorar o processo já iniciado com novas turmas ingressantes no curso, bem como propor projetos transdisciplinares com outros cursos mantidos pela IES, como Pedagogia, História, Geografia, Matemática e outros que puderem ser contemplados com projetos futuros de jogos.

Referências

ALMEIDA, M. S. O. MGUP: RUP Aplicado a Jogos Móveis. Maringá, 2006.

AUTODESK, INC. "Autodesk Maya". Available from: http://usa.autodesk.com [Accessed 14/8/2006]

CONITEC, INC. "3D Game Studio". Available from: http://www.3dgamestudio.com [Accessed 1/8/2006].

EPIC GAMES INC. "Unreal Engine 2.0 and Edit". Available from: http://udn.epicgames.com [Accessed 1/8/2006]

FLYNT, J. P E SALEM, O. Software Engineering for Game Developers. Sebastopol: PTR Prentice-Hall, 2004.

FUKS, H., RAPOSO. A. B., GEROSA, M. A. "Engenharia de Groupware: Desenvolvimento de Aplicações Colaborativas". *In: XXI Jornada de Atualização em Informática, Anais do XXII Congresso da Sociedade Brasileira de Computação*, v. 2, cap..3, pp. 89-128, 2002.

IBM, RATIONAL SOFTWARE CORP. Avaliable from: http://www-306.ibm.com/software/awdtools/rmc/features/ [Accessed 21/8/2006]

NOVAK, J. D. "Learning, Creating, and Using Knowledge : Concept Maps As Facilitative Tools" *In: Schools and Corporations*. New York: Lawrence Erlbaum Associated Ed, 1998

OGRE3D. "Ogre3D Official Site". Available from: http://www.ogre3d.org [Accessed 01/08/2006].

PERUCIA, A. S. ET AL. *Desenvolvimento de Jogos Eletrônicos – Teoria e Prática.* São Paulo: Ed. Novatec, 2005.

PROBASCO, L. *The Ten Essentials of RUP*. Canada: Rational Software, 2001.

RUCKER, R. Software Engineering and Computer Games. New Jersey: Addison-Wesley, 2002