



COMMANDER ASSEMBLER: UMA FÁBRICA DE
JOGOS DE RPG TÁTICOS UTILIZANDO
LINGUAGENS DE DOMÍNIO ESPECÍFICO

PROPOSTA DE TRABALHO DE GRADUAÇÃO

Aluno: Eduardo Wagner Marques de Almeida (ewma@cin.ufpe.br)

Orientador: André Luís de Medeiros Santos (alms@cin.ufpe.br)

14 de março de 2008

Contexto

A indústria de jogos atualmente é uma das mais lucrativas do mundo, movimentando um volume de aproximadamente 7.4 bilhões de dólares e mais de 240 milhões de unidades vendidas em 2006(1), tendo um crescimento de mais de 280% nos últimos 10 anos. O desempenho da indústria de jogos é equiparável as outras grandes indústrias de entretenimento, como a de cinema e música, que movimentaram 9.49 e 11.51 bilhões de dólares respectivamente em 2006 (2) (3).

Existem diversos tipos de jogos no mercado, embora não seja unanimidade, os jogos podem ser divididos em gêneros. A divisão em gêneros proposta por Apperley (4) leva em consideração o *gameplay*, e a interação do jogador, ao invés do aspecto visual dos jogos. Tomando como referência essa divisão podemos citar alguns gêneros como, Ação, Aventura, Esportes, *Role-Playing* e Estratégia. Ainda segundo Apperley é comum um jogo se enquadrar em mais de um gênero dada à entropia dos diversos jogos.

Os jogos de estratégia diferem dos outros gêneros por serem derivados dos jogos de estratégia de tabuleiro. Normalmente obrigam o jogador a pensar, muito mais do que nos outros gêneros. Eles consistem principalmente na tomada de decisões cuidadosas e precisas, além de um planejamento minucioso. Podem ser baseados em turnos (*Turn-Based Strategy* - TBS) ou em tempo real (*Real-Time Strategy* – RTS), além de poderem ser de estratégia ou táticos. Os jogos de estratégia põem sobre os cuidados do jogador o gerenciamento de recursos, comercio, construções, diplomacia, unidades militares entre outros. Os jogos táticos por sua vez focam principalmente no combate.

Os *role-playing games* (RPG), são oriundos dos jogos RPGs de caneta e papel, como o famoso *Dungeons & Dragons*(5). Nesse estilo de jogo, o jogador foca na evolução do personagem durante o desenrolar da narrativa do jogo. Essa narrativa, muitas vezes bastante complexa, é afetada pelas decisões prévias do jogador durante o decorrer do jogo, podendo ser uma narrativa linear ou não-linear. A não linearidade traz uma grande complexidade e aumenta ainda mais a imersão do jogador no ambiente da narrativa do jogo.

Nesse contexto, existe o subgênero de RPG Tático. Neste nicho, os jogos possuem elementos do gênero de RPG, principalmente, a narrativa densa e o conceito de evolução dos personagens, inseridos num ambiente de jogo tático, podendo ser em turnos ou tempo real.

Segundo a ESA(1) em 2006, 2,7% dos jogos vendidos para consoles eram de estratégia e 9,5% eram *role-playing*, aproximadamente 12,2% do mercado de consoles. Enquanto que para PC, 35,4% eram de estratégia e 13,9% eram *role-playing*, o que corresponde a 49,3% desse mercado.

Apesar desse grande faturamento e da alta demanda, o desenvolvimento de jogos, como o de software em geral, apresenta uma dependência muito grande no trabalho quase que artesanal de empregados altamente qualificados em tarefas intensivamente manuais (6).

Com o aumento na demanda de software e no constante crescimento da pressão do mercado para a redução de custo e tempo, muitos especialistas pregam o fim do paradigma de desenvolvimento de software atual baseado no trabalho artesanal, em que cada projeto é trabalhado do início até o fim com pouco ou nenhum reuso entre eles. Este paradigma será substituído por um novo, baseado na manufatura. Com isso projetos de uma mesma família, isto é, projetos que variam possuindo muitos requisitos em comum, podem ser rapidamente montados por componentes reusáveis, possivelmente produzidos por múltiplos fornecedores, com automação de pequenas tarefas (7).

Esse novo paradigma de manufatura, ou fábrica de software, usa de *frameworks*, ferramentas, modelos e linguagens para automatizar ao máximo o desenvolvimento de software. O foco desse trabalho encontra-se na criação de uma fábrica de software.

Objetivos

Este trabalho pretende realizar a construção de uma fábrica de software para o domínio de jogos de RPG táticos. Apresentando o passo a passo, as dificuldades encontradas, e as ferramentas utilizadas nesta construção.

No final do projeto, um jogo de RPG tático será criado, usando a ferramenta, como estudo de caso.

Para construção da ferramenta, batizada de *Commander Assembler*, serão utilizadas técnicas de Modelagem de Domínio Específico (*Domain-Specific Modeling - DSM*) para criação da especificação da fábrica (*Schema*), e posterior criação de uma linguagem de domínio específico (*Domain-Specific Language – DSL*) para otimizar o esforço no uso da ferramenta (8).

Para a criação da DSL será utilizado o plug-in para o Visual Studio, que permite a criação de DSLs visuais.

A fábrica deverá consumir um mini-motor de jogos de RPG táticos que também será construído baseado na modelagem conceitual do domínio. Esse motor rodará utilizando o *framework* XNA. Assim os jogos criados pela ferramenta podem ser facilmente portados para Xbox ou PC.

Durante a análise do domínio será utilizada a técnica de *feature modeling* (9) para identificar pontos de variabilidade, que deverão ser incorporados a DSL, e pontos comuns, que serão incorporados a *engine*.

O relatório final do projeto possuirá seis capítulos. No primeiro capítulo conterà uma introdução, contexto e objetivos do projeto. O segundo capítulo apresentará a construção da DSM, discriminando a metodologia empregada e o seu passo a passo. O capítulo 3 entrará em detalhes sobre a construção da DSL visual. O capítulo 4 mostrará a criação da *mini-engine* a ser consumida pela DSL. No capítulo 5 apresentaremos a criação de um jogo de RPG tático utilizando a *Commander Assembler*. Finalmente, o capítulo 6 abordará a conclusão do projeto e trabalhos futuros.

Cronograma

Atividade	Março	Abril	Maio	Junho
Levantamento Bibliográfico e estudo do material	█			
Análise do domínio		█		
Desenvolvimento da DSL		█		█
Desenvolvimento da Engine		█	█	
Realização do Estudo de Caso				█
Escrita do Relatório Final		█	█	█
Preparação da Defesa				█
Defesa				█

Referências

1. **ESA.** Essential Facts About the Computer and Video Game Industry. *ESA - Entertainment Software Association*. [Online] 2006. [Citado em: 14 de Março de 2008.] <http://www.theesa.com/archives/files/ESA-EF%202007.pdf>.
2. **MPA.** U.S. Entertainment Industry: 2006 Market Statistics. *MPA - Motion Picture Association*. [Online] 2006. [Citado em: 14 de Março de 2008.] <http://www.mpa.org/USEntertainmentIndustryMarketStats.pdf>.
3. **RIAA.** *2006 U.S. Manufacturers' Unit Shipments and Value Charter*. [Online] Recording Industry Association of America, 2006. [Citado em: 14 de Março de 2008.] <http://76.74.24.142/6BC7251F-5E09-5359-8EBD-948C37FB6AE8.pdf>.
4. **Apperley, Thomas H.** Genre and game studies: Toward a critical approach to video games genre. *SIMULATION & GAMING*. Março 2006, 2006, Vol. 37, 1.
5. **Arneson, Gary Gygax and Dave.** *Dungeon and Dragons*. s.l. : TSR.
6. *Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools*. **Greenfield, Jack e Short, Keith**. Anaheim, CA, USA : ACM New York, NY, USA, 2003, Conference on Object Oriented Programming Systems Languages and Applications. ISBN:1-58113-751-6.
7. **Greenfield, Jack e Short, Keith.** Moving to Software Factory. *Visual Studio Team System*:. [Online] Micorsoft Corporation, 2004. [Citado em: 14 de Março de 2008.] <http://blogs.msdn.com/askburton/articles/232021.aspx>.
8. *Notable design patterns for domain-specific languages*. **Spinellis, Diomidis**. 1, Karlovasi, Grécia : Journal of Systems and Software, 2000, Vol. 56.
9. *Towards a More Precise Definition of Feature Models*. **Riebisch, Matthias**. Darmstadt, Alemanha : s.n., 2003. European Conference on Object Oriented Programming - ECOOP.

Assinaturas

André Luís de Medeiros Santos (Orientador)

Eduardo Wagner M. de Almeida (Aluno)