

Extensão do Framework XNA para Customização de Elementos de Jogos

Luan Brenelli, Luciano Silva

Laboratório de Processamento Gráfico, Faculdade de Computação e Informática
Universidade Mackenzie, 01302-907, Rua da Consolação, 930, São Paulo(SP)

luanbrenelli@msn.com, luciano.silva@mackenzie.br

Abstract. *Customization of games is an interactive feature in modern games, which lets you define custom elements to each player. The customization can be seen in different genres, for example, gender fighting (Soul Calibur), race (Mod Nation Racers), platform (Little Big Planet), shooting (Army of Two) and sports (FIFA 2011). The elements that can be customized in a game depend on the genre they belong to. In this context, this work extends the XNA framework to provide a generic mechanism for customization of elements of a game.*

Resumo. *Customização de jogos é um recurso de interatividade nos jogos modernos, que permite definir elementos personalizados para cada jogador. A customização pode ser vista em diferentes gêneros como, por exemplo, o gênero luta (Soul Calibur), corrida (Mod Nation Racers), plataforma (Little Big Planet), tiro (Army of Two) e esportes (FIFA 2011). Os elementos que podem ser customizados em um jogo dependem muito do gênero a que pertencem. Neste contexto, este trabalho visa a dotar o framework XNA de um mecanismo genérico para customização de elementos de um jogo.*

1. Introdução

Customização refere-se a um conjunto de recursos dentro de um jogo que possibilitam uma interação diferenciada a partir de personalização de elementos gráficos, de áudio, de condições de simulação, dentre outros. O processo de customização aparece em vários jogos modernos e apresenta formas diferenciadas em função do gênero, como aquelas encontradas em Soul Calibur (gênero de luta), Mod Nation Racers (corrida), Little Big Planet (plataforma), Army of Two (tiro) e Fifa 2011 (esportes).

Apesar da grande variedade de jogos que utilizam recursos de customização, os game engines ainda carecem de recursos que facilitem a programação da funcionalidade de customização de elementos de um jogo. Herbelin, Crig e Brooks [2008], por exemplo, propõem um esquema genérico de customização que vai além do game engine e atinge o *hardware* do jogo. Scacchi [2010] expõe uma técnica de customização conhecida como *mod* e faz uma análise completa de seus tipos. Porém, assim como outras técnicas de customização, o foco é uma customização prévia e que não pode ser alterada com o jogo em andamento.

Neste contexto, este trabalho apresenta uma extensão do game engine XNA 4.0 [Microsoft, 2011] para suportar customização dinâmica de diversos elementos dentro do engine: sprites, modelos, fontes e áudio. A extensão é baseada no padrão de projeto Decorator [Gamma *et al.*, 1994] e permite a inclusão ou exclusão de customizações aos elementos do jogo.

2. Extensão do *Framework* XNA 4.0

Para cada componente de jogo do XNA 4.0, foi aplicado o padrão Decorator. A título de exemplo, o diagrama abaixo mostra a aplicação do padrão para o componente Model, responsável por carregar objetos tridimensionais no XNA 4.0:

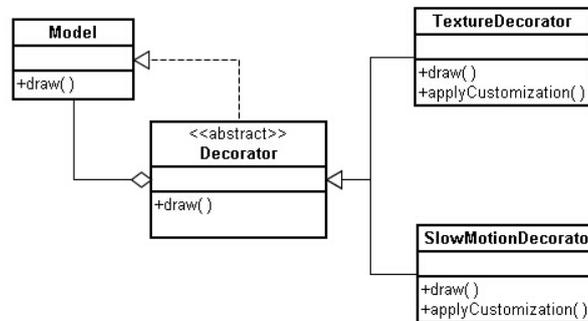


Figura 1: Aplicação do padrão Decorator no componente Model.

A customização é definida pelas extensões de Decorator, que contêm as informações específicas de customização do modelo. O código abaixo ilustra a definição de uma customização para alterar a textura e efeito de slow motion para um modelo 3D:

```

Model m = Content.load<Model>("modelo.fbx");
Decorator d = new TextureDecorator(m);
d = new SlowMotionDecorator(d);
  
```

A aplicação dos decoradores depende de sua natureza. Por exemplo, um TextureDecorator irá atuar somente na chamada do método draw(...), que atualiza a tela do XNA. Já um decorador do tipo SlowMotionDecorator pode atuar tanto dentro do método update(...), que atualiza o estado do jogo, quanto dentro do método draw(...). A figura abaixo mostra um exemplo de modelo 3D e o resultado da aplicação do decorador TextureDecorator:



Figura 2: Modelo 3D (esquerda) e resultado da aplicação do decorador (direita).

Referências

- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. New York: Addison-Wesley, 1994.
- Heberlin, B.; Ciger, J.; Brooks, A.L. Customization of gaming technology and prototyping of rehabilitation applications. In: *Proceedings of ICDVRAT 2008*, Reading, UK, 2008, pp. 211-218.
- Microsoft (2011). Microsoft XNA. Disponível em <http://msdn.microsoft.com/xna>. Acesso em 30/11/2011.
- Scacchi, W. (2010). *Computer Game Mods, Modders, Modding, and the Mod Scene*. Disponível em: <http://www.ics.uci.edu/~wscacchi/GameLab/ModSquad-Scacchi.pdf>. Acesso em: 10/11/2011.