

Podcast

Disciplina: Design patterns e gerência de configuração: do projeto ao controle de versões.

Título do tema: Integrando design patterns: Gerência de configuração em projetos de software

Autoria: Arthur Gonçalves Ferreira

Leitura crítica: Marco Ikuro Hisatomi

Olá, ouvintes! No podcast de hoje nós vamos conversar um pouco sobre propostas de projetos de desenvolvimento de jogos eletrônicos com integração de padrões de projetos GoF.

Você deve estar se perguntando, padrões de projeto GoF podem ser aplicados em projetos de desenvolvimento de jogos? A resposta é muito clara, com toda certeza os padrões de projeto GoF podem e devem ser aplicados em projetos de desenvolvimento de jogos eletrônicos. A justificativa para isso é que existem diversos jogos com problemas praticamente idênticos, que podem ser implementados com um padrão de projeto GoF. Neste caso aplica-se o conceito de reutilização de código, que é justamente a ideia dos padrões de projetos. Vamos conhecer algumas propostas de projetos de desenvolvimento de jogos eletrônicos com a integração de padrões de projeto GoF.

A primeira proposta de projeto está relacionada no desenvolvimento de jogos eletrônicos onde um personagem precisa atravessar vários cenários para atingir um determinado objetivo. Os cenários são ambientes diversos como, por exemplo, campos abertos, labirintos e castelos. Os jogos que estão dentro desse contexto são basicamente jogos de RPG e jogos de tiro. Um jogo dentro desse contexto, pode até ser diferente em relação a cenário e visual, mas a ideia e a estrutura do código são parecidas, o que significa que podemos utilizar um padrão de projeto GoF para tornar a ideia e o Código mais simples.

Para este tipo de projeto de desenvolvimento de jogos eletrônicos, onde existem personagens que precisam passar por cenários, para alcançar um objetivo, é muito utilizado o **padrão de projeto GoF de criação abstract factory**. Este padrão de projeto, faz com que todos os métodos que realizam a geração de cenários, estejam diretamente relacionados a classe principal do jogo e acessem a classe abstrata responsável por criar os cenários.

A segunda proposta de projeto está relacionada no desenvolvimento de jogos eletrônicos que estão relacionados ao gênero de jogos de luta. Alguns jogos desse tipo possuem um problema bem específico, a maioria dos inimigos enfrentados possuem um visual bem parecido. Observamos isso, por exemplo, em um jogo conhecido como “gangue de rua”. Praticamente todos os adversários possuem o mesmo tipo de roupa, mesmos movimentos ou até o mesmo tipo de movimento, o que de fato muda é a cor do personagem, o nome e outros pequenos detalhes.

Projetos de desenvolvimento de jogos do gênero de luta, encontram esse tipo de problema, porque criar personagens com uma grande variedade de características é um processo que demanda muito tempo e pode acarretar também na grande capacidade de espaço por parte do código do jogo ocupado no armazenamento de disco. Para auxiliar na resolução deste problema podemos utilizar o **padrão de projeto GoF de criação builder**.

O padrão builder, auxilia na criação de personagens em um jogo de luta, criando condições mais favoráveis para a criação de personagens variados, a partir de um personagem mestre, que conterá diversas características que serão distribuídas em diferentes personagens.

Você pode se perguntar, como ficaria em relação um jogo do gênero de luta voltado para dispositivos com recursos limitados como tablet e celulares? Nestes casos ter personagens com a mesmas características é um procedimento normal a ser adotado, porém existe a necessidade de se resolver o problema de duplicação de código para cada personagem. Imagine que exista vários personagens que carregam os mesmos atributos, teríamos uma grande quantidade de código duplicado. Como resolver? Utilizando o padrão de projeto GoF de criação prototype.

A terceira proposta de desenvolvimento de jogo eletrônico está relacionada justamente a jogos do gênero de luta voltados para dispositivos com poucos recursos e que precisam utilizar a ideia de duplicação de personagens com as mesmas características, mas que precisam resolver o problema da duplicação excessiva de código. Com o padrão de criação prototype é criado um personagem central com algumas características que servirá como um espelho para os outros personagens parecidos com a ideia do personagem mestre utilizado no padrão builder, a diferença é que cada personagem clone possui um método que constrói de forma aleatória suas características.

E por falar em desenvolvimento de jogos eletrônicos para diferentes plataformas, é claro que não podemos deixar de falar sobre uma proposta de desenvolvimento de jogos que consiga rodar em várias plataformas. Tentar adaptar um jogo para que possa ser jogado em mais de um dispositivo pode ser visto como um grande problema, porque cada plataforma possui suas características como por exemplo, tamanho de tela e memória. Para resolver este problema é muito comum utilizar o padrão de projeto GoF estrutural adapter.

O padrão adapter cria um método único de comunicação para qualquer dispositivo. O adapter cria ainda uma classe capaz de desativar todas as funcionalidades consideradas desnecessárias para o dispositivo que está sendo utilizado para rodar o jogo.

Os padrões de projetos GoF são muito utilizados no desenvolvimento de jogos eletrônicos, independente do dispositivo. Com eles, podemos realizar desenvolver diversos projetos reutilizando ideias já existentes. Espero que você tenha aprendido sobre a utilização desses padrões em projetos no

desenvolvimento de jogos e que tenha ajudado em ideias de projetos que você possa realizar.

Este foi nosso podcast de hoje! Até a próxima