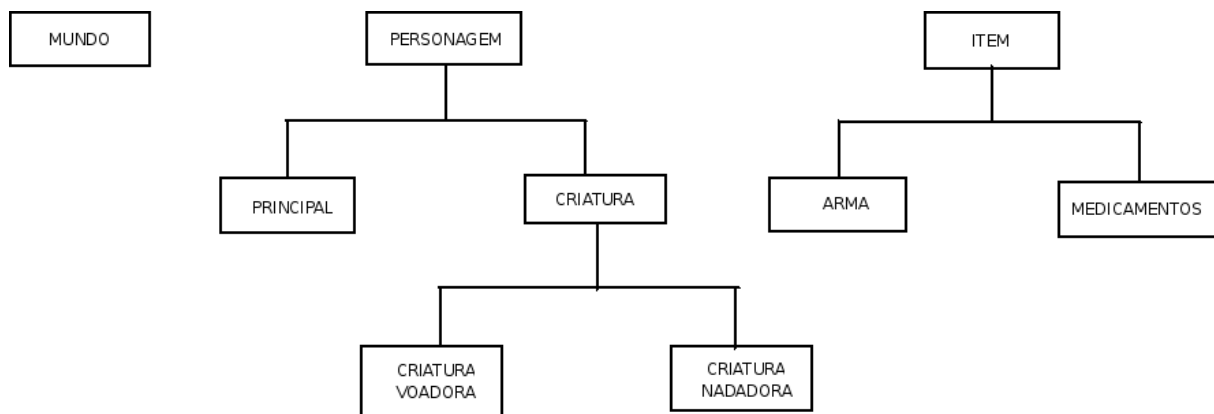


Aplicando o conceito de orientação a objetos (OOP) no desenvolvimento de games

Se você começou a programar utilizando o conceito de programação estruturada, assim como eu, pode ser que tenha dificuldade em aplicar o conceito de OOP para resolver problemas e modelar seu software. Eu tive essa dificuldade, porém, depois de algumas pesquisas na internet e em livros da minha biblioteca, as coisas ficaram mais claras.

Você pode não ter notado, mas, todos os games simulam alguma coisa seja do mundo real, seja de um mundo idealizado. Os games de esportes e simuladores não deixam dúvidas pois possuem elementos do mundo real como jogadores, carros, aviões, trens, etc. Os games de aventura e estratégia normalmente acontecem num mundo criado e possuem pessoas, animais, monstros, armas, equipamentos, veículos, etc. Até mesmo os games arcade são simulações de mundos pois possuem monstros, aliens, armas, naves, etc. Ou seja, todo game pode ser encarado como uma simulação de um mundo. Sendo uma simulação de um mundo o game possui **objetos** (pessoas, criaturas, armas, veículos, etc). Podemos dizer que os objetos de um game são os objetos do seu mundo e considerando o game como uma coleção de objetos podemos aplicar a OOP ao seu desenvolvimento.

Para clarear as coisas vamos a um exemplo. Digamos que você está criando um game de aventura. Podemos imaginar um objeto como sendo o próprio mundo, o qual poderia armazenar informações como o mapa do mundo, o tempo, as condições atmosféricas, etc. Todos os outros objetos poderiam conter informações acerca da sua posição no mundo. Estas poderiam ser, por exemplo, coordenadas cartesianas do tipo (x,y). O personagem principal poderia ser outro objeto e armazenar, além das suas coordenadas (x,y), informações como pontos de vida e coisas coletadas durante o jogo (armas, medicamentos, chaves, etc). Estes itens coletados pelo personagem principal também poderiam ser objetos. Por exemplo, as armas poderiam ser objetos que armazenam informações sobre a quantidade de munição existente e danos causados ao inimigo. As criaturas inimigas também poderiam ser objetos que armazenam sua posição e informações sobre seu aspecto visual, suas habilidades, sua força, etc. Aqui podemos imaginar que todas as criaturas compartilhariam algumas características básicas e outras características seriam específicas. Por exemplo, todas as criaturas teriam força e uma arma, mas somente um tipo específico de criatura poderia voar e somente outro tipo específico de criatura poderia nadar. Daí concluímos que podemos ter o objeto “criatura” e dois objetos descendentes deste que seriam o objeto “criatura voadora” e o objeto “criatura nadadora”. Abaixo podemos ver um exemplo da organização básica das classes de nosso game de aventura.



Utilizando esta abordagem podemos dizer que cada objeto é responsável por si mesmo. O game propriamente dito não precisa se preocupar com o que cada objeto está fazendo. Ele só precisa oferecer uma tela onde os objetos serão desenhados e um “game loop”. Os próprios objetos fazem suas atualizações a cada iteração do loop. Para as criaturas controladas pela AI estas atualizações poderiam ser a direção a seguir e se devem atirar ou fugir. Para o personagem principal estas atualizações significam executar operações baseadas nas entradas do usuário. O ponto principal da idéia é que os objetos são entidades independentes que sabem cuidar de si mesmos.

O melhor fruto colhido da abordagem OOP no desenvolvimento de games é a reutilização de código. Digamos que você concluiu seu game de aventura e agora resolveu criar um game semelhante, porém, ambientado no espaço com naves espaciais em vez de criaturas e personagens. Em vez de começar do zero, você poderia pegar os objetos desenvolvidos no primeiro game e utilizá-los neste novo game. Mesmo que não utilizasse todas as classes desenvolvidas no primeiro game, você poderia utilizar algumas e desenvolver apenas as subclasses necessárias para o novo game.

Assim, concluímos que a OOP é uma excelente abordagem para o desenvolvimento de games. Espero ter esclarecido um pouco o conceito para você. Qualquer dúvida a mais me manda um email.

Um grande abraço e bons estudos.

Samuel Dias Neto

samuel@samueldiasneto.com

www.samueldiasneto.com