

Reflexão 01: Princípios de Projeto OO

Os principais pontos em se usar um engine para jogos são as funcionalidades tipicamente fornecida as quais são: uma engine gráfica para renderizar gráficos 2D, uma physic engine para simular a física ou simplesmente para fazer detecção de colisão, suporte a animação, sons, inteligência artificial, networking, gerência de memória, gerência de arquivos, gerência de linha de execução e suporte a grafos de cenas e entidades. Os princípios de POO (Projeto Orientado á Objetos) auxiliam na criação dessas funcionalidades por meio de práticas importantes e essenciais que são próprias desse paradigma como os módulos, a reutilização de pacotes, o baixo acoplamento entre classes, inversão de dependência e os princípios de SOLID, todos esses em conjunto ajudam a criar uma engine com estabilidade, que permite refatoração do seu código e é aberto para futuras implementações que venham alterá-las / melhorá-las. Além disso, ajudando não somente na via engine-desenvolvedor, mas também na via engine-hardware, já que uma engine é estabelecida tendo como base o auxílio da criação de jogos para desenvolvedores independentemente do hardware no qual está sendo implementado e visado, ou seja, retira do desenvolvedor a necessidade de conhecer a fundo a plataforma - alvo, por esta causa uma engine necessita ser bem estruturada não somente para atender as necessidades do desenvolvedor mas também o contato com o hardware da plataforma - alvo.

Um exemplo de uma engine de jogos com princípios de POO seria a *DimensioneX Multiplayer Engine*, em termos simples, essa engine é focado no desenvolvimento de jogos multiplayer no browser, semelhante aos jogos da rede social *Facebook*, voltado para iniciantes. Um dos pontos chave dela é que ela não é uma engine gráfica, por ser browser, contudo ela faz com o cliente seja uma combinação de HTML, JavaScript gerados no server, portanto as cenas gráficas não são nada menos do que pilhas de imagens estáticas e ícones providos pelo programador. E por fim, todas as outras interações são feitas através de uma linguagem de script orientada a objetos.

Em vista disso, podemos observar muitos usos e aplicações onde os conceitos de POO, primeiramente é possível ver a necessidade de uma engine que possa suportar diversas plataformas já que browser diferentes requerem diferem tipos de contato, como por exemplo as diferenças entre Safari, Chrome e Mozilla Firefox. Contudo, não são somente divergências como explicado acima, grande parte desse código precisa ser gerado de maneira genérica já que seu uso (display, cálculos e entre outros) são compartilhadas entre esses navegadores de igual forma, como os seus códigos auto gerados em HTML e JavaScript. Para que tudo isso ocorra, é possível observar a partir de uma análise rasa como feita acima o uso de padrões de projetos, o uso dos conceitos de SOLID, o uso do princípio de Demeter e outros.