## Linguagem de Programação

Até a década de 90 cada jogo tinha a sua engine, feita para possibilitar a maior eficiência no uso da memória e da unidade de processamento possível, de acordo com as exigências de cada jogo. Um jogo que só usava formas geométricas, por exemplo, não precisava tratar imagens na sua engine. O nível da microeletrônica e da computação já possibilita o uso de engines genéricas, mas o desenvolvimento de uma ainda demanda uma programação muito próxima da máquina.

É por isso que a escolha da linguagem de programação precisa ser feita com cuidado. Considerando o conhecimento da equipe e o propósito do projeto, que era uma engine didática, para influenciar o designe e o desenvolvimento de jogos de acordo com o nosso alcance, as linguagens de programação selecionadas para a análise foram Actionscript, C++, C# e Java, de acordo com a simplicidade, o poder e a compatibilidade de cada uma.

### Estudo Comparativo

O Actionscript é uma linguagem orientada a objetos desenvolvida pela Macromedia. O que no início era uma ferramenta para controlar animações se tornou uma linguagem de script tão complexa que podia ser usada no desenvolvimento de um jogo. Embora essa linguagem ainda seja muito usada no desenvolvimento de jogos de web, o que provocou a decisão contrária a ela foi a expectativa de que o HTML 5 viesse a incorporar o Javascript e, dessa forma, modificar ou inutilizar o Actionscript.

O C# (C Sharp) é uma linguagem multi-paradigma da Microsoft feita para o desenvolvimento de sistemas próprios para a plataforma .NET. C# e Java compartilham a mesma simplicidade na leitura e na codificação, assim como a mesma forma de interpretação e compilação, mas um programa em C# está mais próximo da máquina do que um programa em Java. A escolha parecia feita quando nós entendemos que a ligação do C# com a Microsoft poderia custar a compatibilidade do nosso jogo.

Java é uma linguagem orientada a objetos desenvolvida pela Sun, hoje possuída pela Oracle. A sua fama de espaçosa e pesada não é coerente com a realidade: hoje a linguagem conta com a Compilação na Hora ou *Just in Time Compilation* (*JIT Compilation* ou só *JIT*), para que a sua execução não seja mais interpretada. Mas a sua principal característica é a compatibilidade: a Máquina Virtual do Java ou *Java Virtual Machine* (*JVM*) é uma plataforma virtual que pode ser feita compatível para qualquer plataforma física.

Por mais evoluída que seja a JVM, no entanto, o Java não admite o acesso à máquina necessário para o desenvolvimento de uma Game Engine, senão com o uso do C++, por meio da Interface Nativa do Java ou *Java Native Interface* (*JNI*). Em outras palavras, para usar o Java, nesse caso, nós teríamos que usar o C++. Esta, por sua vez, não é a mais simples na codificação, mas não tem limitação alguma tanto em termos de compatibilidade quanto em termos de acesso à máquina.

### A Escolha: C++

C++ (C Mais Mais ou *C Plus Plus*) é uma linguagem de programação multi-paradigma, com suporte para a programação imperativa e a programação orientada a objetos, de uso geral, desenvolvida por Bjarne Stroustrup, para formar uma camada de orientação a objetos sobre a linguagem de programação C. O C++ possibilita a programação de baixo nível assim como a programação de alto nível e por isso é considerado uma linguagem de programação de nível médio em termos de proximidade da máquina.

Na versão de 2011, com uma óbvia e construtiva influência da Biblioteca Boost, o C++ ganha uma série de características, algumas das quais eu cito: verificação de tipo dinâmica, estruturas de conversão e controle de alto nível, reflexão padronizada, uma biblioteca de computação paralela padronizada, coleta de lixo automática, etc. Com essas mudanças o C++ ganha em simplicidade, sem, contudo, perder a proximidade da máquina de outrora.

### Referência Bibliográfica

<http://www.cplusplus.com/info/description/>

<http://www.stroustrup.com/C++11FAQ.html>