

Game Engines

Edward Seim

February 8, 2015

Game Engines are the engines that are reused to make games work. Abstractly, they are the component that allows designers, scripters, and programmers focus on the job of creating a game and making it unique, rather than having to worry about the inner workings of the software and worry about compatibility on platforms and devices.

Games in the 1990's proved the viability of reusable base components in games rather than building a new one for every game. Doom (1993) was the most representative of this movement. In the beginning, game engines would be made by each game company, thereby tailoring them to their needs in the games they wanted to create. Nowadays, making in-house engines is so costly that typically a company will specialize in either games or game engines. The companies that make the game engines will then sell the rights to their engines to other companies that will actually make games with them.

Probably the most important decision to make when first thinking about making a game is whether it will be 2D or 3D. The decision alone can help you decide on better game engines better suited to each task. You can then narrow down the list of game engines by which additional features you will need to compile the game mechanics you want. Choosing 2D over 3D generally means you'll get a simpler design process as well as easier to make. A very important thing to recognize is that since 2D is not particularly realistic, it allows the making of games with worlds that do not even try to be realistic. In fact if done correctly they can embrace this difference to make truly new and original

game play. 3D games on the other hand while costing more to make, with more difficult control schemes, and having to deal with camera problems, have their advantages. Apart from the fact that its more realistic, you can have more background and add more depth to a game. Done correctly, the graphics can be much more enticing than the typical 2D game and therefore is more likely to immerse you into the game, though this also demands more powerful platforms and well optimized game engines. Because of the larger world possibilities, you will typically get more playtime out of a 3D game than a 2D game.

Based on that 2D/3D choice comes the decision as to the lighting that you'll want for your game. 2D presents itself with a much wider variety of lighting techniques as a result of the world variety you can have. In general the choices are based in how the textures in your game adapt to the world. A good place to start is will they illuminate themselves or have to be lit to be seen. An example would be something like a pool of some radioactive substance glowing or just a pillar in a room. Then the question becomes will particular objects light up those around it, like say a candle or light bulb. In general, the technical ways that lighting is carried out is via texture modification and texture mapping. The combination of these gives 2D objects depth and texture along with the demonstration of the world you create. Different game engines will handle these differently, or may even provide you options as to the lighting scheme you want for your game. 3D games are fairly universal in their lighting schemes. It either is lit everywhere where every object looks the same whether its blocked or hidden, or its based on shadow casting and night which has a more realistic feeling. The more realistic drive to make 3D games more realistic makes it much more limited in its lighting schemes.

Delving into examples of game engines, two of the most noteworthy ones are Unity and Unreal Engine. Both are well documented, with extensive manuals about the software and its features and their implementations. Unreal Engine also provides a wide variety of example projects which allow to quickly jump off point for anyone trying to get started on their game. An important thing

to note is that Unity has DRM (Digital Rights Management) on its software and the game you will create, so they can control use of their software. Unreal Engine has no DRM. Since I do not use a pc, I can't speak to the quality of either of their software packages. On mac, I found the Unity software to feel more polished and in general much higher quality than the Unreal offerings. As we ended up using Unreal Engine for this class, I can say that their mac software does not have that mac feel and smoothness that I get with all my favorite Apple and 3rd Party apps.