Game Engines

http://en.wikipedia.org/wiki/Game_engine
http://www.gamecareerguide.com/features/529/what_is_a_game_.php

What is Game Engine?

- A game
  - all the content (models, animations, sounds, AI, and physics) which are called 'assets'
  - the code required specifically to make that game work, like the AI, or how the controls work
- Game Engine
  - Is a software system designed for the creation and development of video games
  - Is Not game specific technology
  - Reusable software components within different games

Game Engine History

- Prior to game engines, games were typically written as singular entities. Very little could be reused between games.
- The term "game engine" arose in the mid-1990s.
- Developing teams of *Doom* and *Quake* started to separate game-specific rules and data from basic concepts like collision detection and game entity.
- Later games, such as *Quake III Arena* and *Unreal* were designed with this approach in mind, with the engine and content developed separately.
- The practice of licensing such technology has proved to be a useful auxiliary revenue stream for some game developers. (e.g. Unreal Engine) At the very least, reusable engines make developing game sequels faster and easier, which is a valuable advantage in the competitive computer game industry.
- Modern game engines (e.g Unity) are some of the most complex applications written. The continued refinement of game engines has created a strong separation between rendering, scripting, artwork, and level design.

What do Game Engines offer?

- Offer reusable components that can be manipulated to bring a game to life.
  - Rendering (loading, displaying, and animating models)
  - Collision detection between objects
  - Physics (simulation of physics systems such as rigid body dynamics)
  - Input, graphical user interfaces
  - Multimedia (e.g. sound)
  - AI
  - Networking
  - …

Why use a Game Engine?

- Game engines provide all the core functionality needed, right out of the box, to develop a game application while reducing costs, complexities, and time-to-market — all critical factors in the highly competitive video game industry.
- In addition to reusable software components, in many cases game engines provide a suite of visual development tools. These tools are generally provided in an IDE to enable simplified, rapid development of games in a data-driven manner.

Types of Game Engines

- *Roll-your-own game engines (lowest level)*
  - use publicly available application interfaces, such as APIs likeDirectX, OpenGL, to create their own engines.
  - may use other libraries, both commercial and open source, to help them along the way (e.g. physics libraries like Havok and ODE, scene graph libraries like OpenSceneGraph, and GUI libraries like AntTweakBar)
  - Advantage: give programmers the greatest amount of flexibility, letting them pick and choose the components they want and integrating them exactly how they want
  - Disadvantages:
    - Take the longest amount of time to build
    - Programmers frequently will have to build the tool chain from scratch, since they can rarely rely on all these libraries to work together straight out of the box. This makes rolling your own engine less attractive to most game developers, even the professional ones.

Types of Game Engines (2)

- *Mostly-ready game engines (mid level).*
  - Most game engines to be "mostly ready".
  - They are ready for prime time right out of the box, with rendering, input, GUI, physics and so on.
  - Many of them have mature tool chains so you don't have to roll your own.
  - E.g
    - OGRE and Genesis3D, which are both open source
    - Low priced engines like Torque, and high priced ones such as Unreal, id Tech, and Gamebryo.
  - They still require a bit of programming to get them up and running into a complete game. (some scripting or sometimes even low-level coding to get a real game working)
  - Mostly-ready game engines are a bit more limiting than roll-your-own engines and are frequently optimized for the general case.

Types of Game Engines (3)

- *Point-and-click engines (highest level).*
  - They include a full tool chain that allows you to point and click your way to creating a game.
  - They are becoming more and more common these days. (eg. GameMaker, Torque Game Builder, and Unity3D)
  - They are built to be as friendly as possible, and are made to require as little coding as possible.
  - Advantages:
    - The best thing about these engines is that they allow you to work quickly, and play your games quickly, without too muck work. If you're just starting out in game design, you could do worse than these tools.
  - Disadvantages:
    - The problem with many point-and-click engines is that they can be extremely limiting.
    - Many do one or two types or genres of game well, or one or two types of graphics modes. This is not to say they're useless. Even faced with the restrictions of these tools, it's possible to make highly creative games or even find creative ways around those restrictions.

Notable Commercial Engines

- Popular commercial engines are:
  - **Unity** personal (free), plus, pro, enterprise
  - CryENGINE 3
  - Source
  - Unreal Engine 3
  - Game Maker Studio
  - id Tech 4
  - GEM 2
  - …

  Also see https://en.wikipedia.org/wiki/List_of_game_engines.

Notable Open Source Engines

- Popular open source engines are:
  - Crystal Space - A free portal based graphics engine
  - Delta3D - An open source engine specifically created by the Navy, especially for training applications
  - Game Blender - A sub-application of Blender
  - Irrlicht - A simple free open source 3d engine
  - The Nebula Device - A powerful engine used in many commercial games
  - Ogre3D - A well known LGPL graphics engine
  - Open Dynamics Engine - A cross platform physics engine used in many commercial games
  - Panda3D - Full featured Python/C++ 3D engine
  - Wolfenstein 3D engine till id Tech 3 - Those engines have all been released as open source

  Also see https://en.wikipedia.org/wiki/List_of_game_engines.