Game Engines are genre specific

First-person shooters typically focus on technologies such as:

- efficient rendering of large 3D virtual worlds;
- a responsive camera control/aiming mechanic;
- high-fidelity animations of the player's virtual arms and weapons;
- a wide range of powerful handheld weaponry;
- a forgiving player character motion and collision model, which often gives these games a "floaty" feel;
- high-fidelity animations and artificial intelligence for the non-player characters (NPCs)—the player's enemies and allies; and
- small-scale online multiplayer capabilities (typically supporting up to 64 simultaneous players), and the ubiquitous "death match" gameplay mode.

Racing game

- Various "tricks" are used when rendering distant background elements, such as employing two-dimensional cards for trees, hills and mountains.
- The track is often broken down into relatively simple two-dimensional regions called "sectors." These data structures are used to optimize rendering and visibility determination, to aid in artificial intelligence and path finding for non-human-controlled vehicles, and to solve many other technical problems.
- The camera typically follows behind the vehicle for a third-person perspective, or is sometimes situated inside the cockpit first-person style.

- When the track involves tunnels and other "tight" spaces, a good deal of effort is often put into ensuring that the camera does not collide with background geometry.

Platformer Game Engine

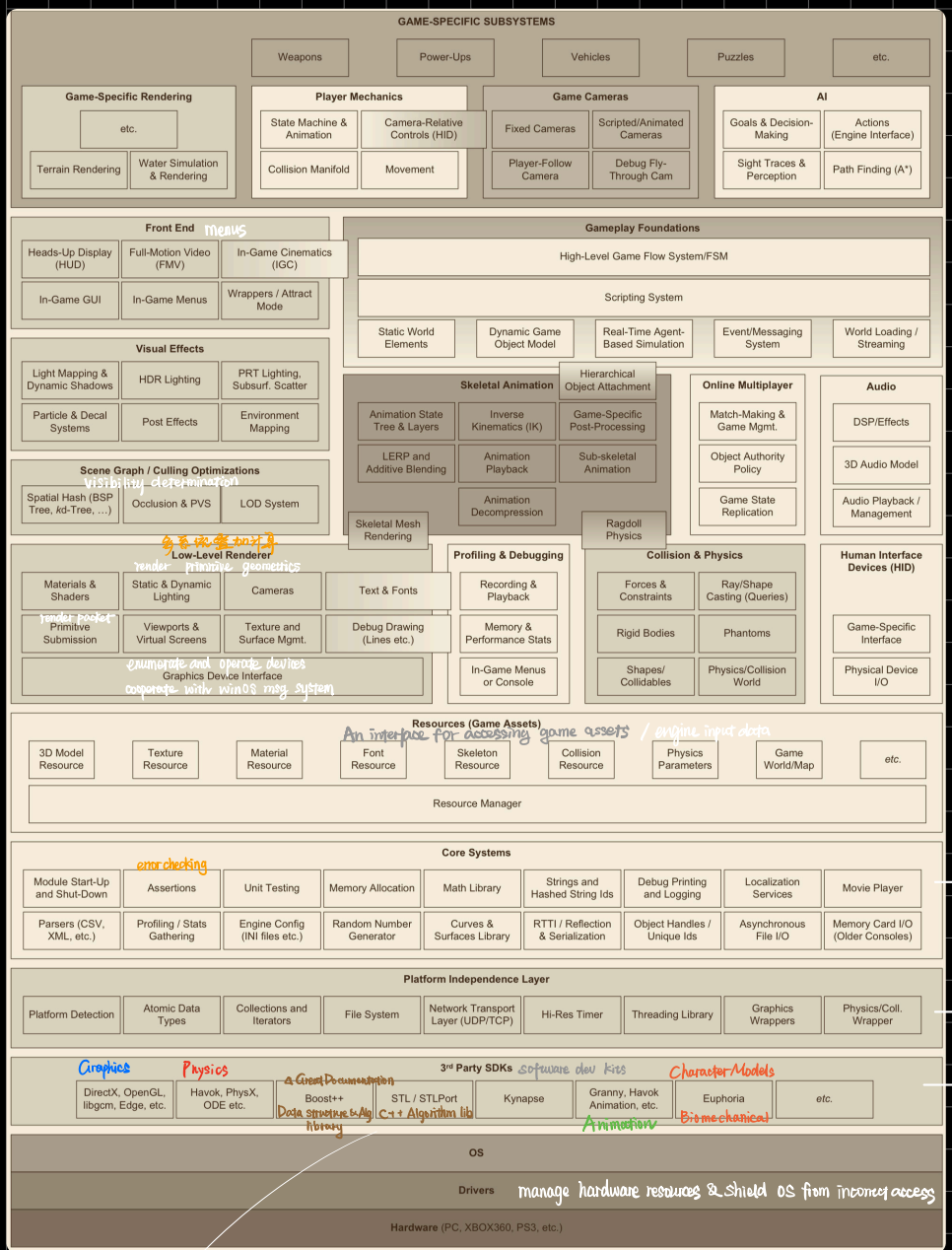
- moving platforms, ladders, ropes, trellises and other interesting locomotion modes;
- puzzle-like environmental elements;
- a third-person "follow camera" which stays focused on the player character and whose rotation is typically controlled by the human player via the right joypad stick (on a console) or the mouse (on a PC—note that while there are a number of popular third-person shooters on a PC, the platformer genre exists almost exclusively on consoles); and
- a complex camera collision system for ensuring that the view point never "clips" through background geometry or dynamic foreground objects.

RTS grid layout of the world

Some other common practices in RTS games include the following techniques:

- Each unit is relatively low-res, so that the game can support large numbers of them on-screen at once.
- Height-field terrain is usually the canvas upon which the game is designed and played.
- The player is often allowed to build new structures on the terrain in addition to deploying his or her forces.
- User interaction is typically via single-click and area-based selection of units, plus menus or toolbars containing commands, equipment, unit types, building types, etc.

Game Engine Structure 从底层到顶层，无circular dependency



useful facilities

detect & deal with variance among diffent platforms

Data Structure / Algorithms

STL — okay for PC because of handful memory manage enviornment

Not completely depend on for performance concern

使用可预期&有限内存分配

STL Memory Allocation might not work well on console, as malloc / new / free works slowly meanwhile. because console usually has smaller virtual memory, cache miss is very expensive hence most games has own memory allocator