

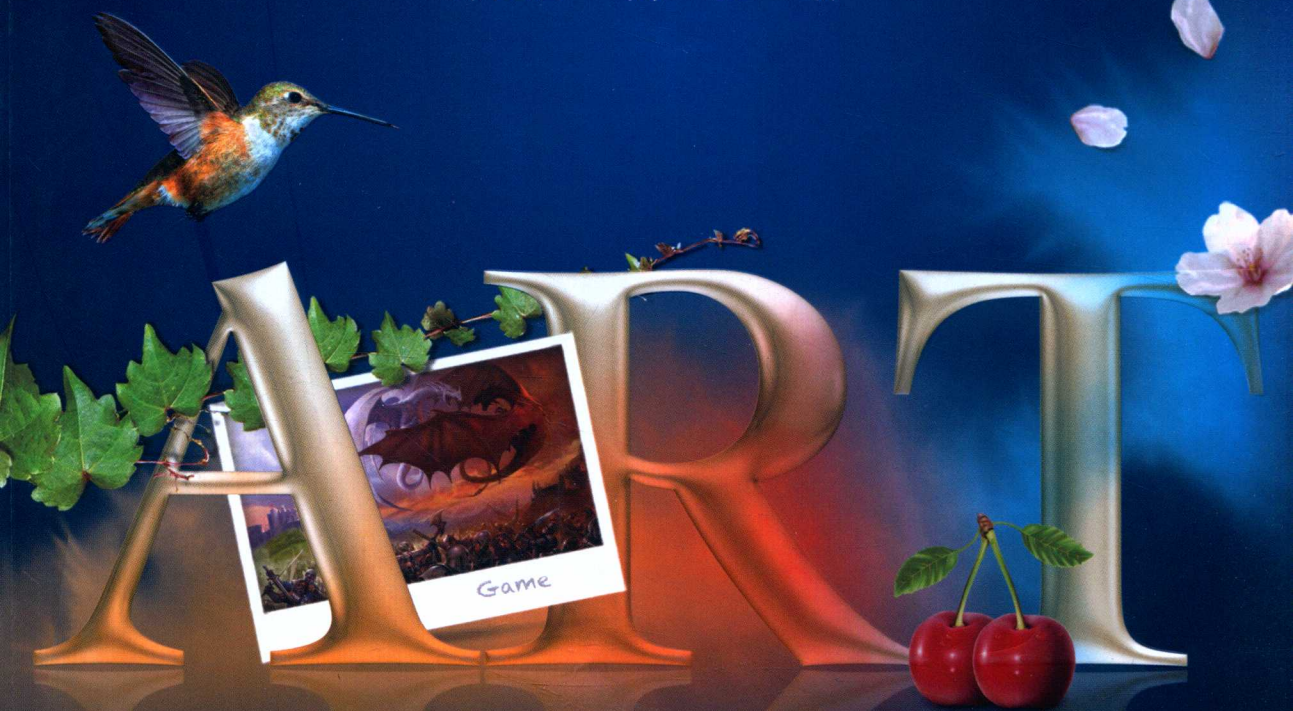


普通高等教育“十二五”规划教材·数字媒体技术

游戏程序设计基础

Game Programming Foundation

杨长强 高莹 编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十二五”规划教材·数字媒体技术

游戏程序设计基础

杨长强 高莹 编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统介绍了 Windows API 二维开发、Direct 3D 三维游戏开发和 Unity 游戏开发相关概念及实现技术。本书由浅入深地介绍了各部分的环境搭建和开发方法；涵盖内容全面，完整地讲解了二维游戏、Direct 3D 游戏和利用 Unity 开发游戏的相关内容；同时注重理论与实践的结合，对于每一章讲解的技术环节都有对应的实现示例。本书配有 PPT、源代码等。

本书在内容安排上非常适合于本科教学，对游戏开发感兴趣的初学者自学时也可以使用本书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

游戏程序设计基础/杨长强，高莹编. —北京：电子工业出版社，2015.12

ISBN 978-7-121-27425-1

I. ①游… II. ①杨… ②高… III. ①游戏—程序设计—高等学校—教材 IV. ①TS952.83

中国版本图书馆 CIP 数据核字（2015）第 246571 号

策划编辑：任欢欢

责任编辑：任欢欢

印 刷：涿州市京南印刷厂

装 订：涿州市京南印刷厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1 092 1/16 印张：17 字数：435.2 千字

版 次：2015 年 12 月第 1 版

印 次：2015 年 12 月第 1 次印刷

定 价：39.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

◇ 前 言 ◇

计算机相关专业的学生在学习游戏程序开发前，一般已经学习过数据结构、操作系统和面向对象程序开发等专业课程，有些学生还学过计算机图形学等课程，但没有游戏程序开发系统方面的知识。目前市场上适合本科教学的游戏开发教材匮乏。

本书从游戏开发的角度，以理论和实践相结合的方式，系统介绍了计算机游戏开发方法。目前市场上的计算机游戏主要分为二维游戏和三维游戏，针对这样的情况，教材内容安排如下：

第1章介绍了计算机游戏的相关概念和分类。

第2章~第7章讲解利用 Windows API 函数完成二维游戏制作，内容包括 Windows 程序窗口生成和消息机制，贴图，交互，碰撞等内容。

第8章~第13章讲解利用 Direct 3D 进行三维游戏的制作，内容包括三维游戏的基本原理，Direct 3D 的安装和初始化过程，Direct 3D 图形绘制基础、变换、光照和纹理映射等。

第14章~第19章讲解 Unity 进行游戏开发的方法。内容主要从初学者入门的角度去讲解 Unity 游戏开发，无论有无编程经验都可以阅读本部分，从而快速掌握如何使用 Unity 引擎制作 3D 游戏。内容包括 Unity 安装方法，Unity 程序开发框架和编辑器的使用，资源和游戏对象，脚本程序和用户界面以及动画系统等。

本书有如下几个鲜明特点：

1. 由浅入深

一般本科生在刚刚完成一、二年级的相关课程学习后，实际动手进行程序开发的能力还未形成，本教材充分考虑到这种普遍存在的情况，在二维游戏、Direct 3D 游戏开发和 Unity 游戏开发前，细致地进行了环境搭建的讲解。

2. 内容全面

本教材全面介绍了 Windows 二维游戏和三维游戏的开发方法，既讲解了利用 Windows API 开发二维游戏，Direct 3D 开发三维游戏这些底层游戏开发方法，也详细讲解了利用目前最流行的一款游戏引擎 Unity 开发游戏的方法。

3. 提供框架程序

在讲解二维游戏、Direct 3D 游戏和 Unity 游戏开发时，对于每一部分内容，首先详细介绍了框架程序的构建过程，以后的内容在框架程序的基础上完成，降低了学生的学习和开发难度。

4. 理论讲解与实践密切结合

在每一章的讲解中，首先对相关理论进行介绍，然后详细讲解相关函数，最后在对应

【游戏程序设计基础】

的程序框架内，利用本章学到的函数完成相关功能的演示。

由于本课程的独特的特点，教材没有安排课后习题，学生只需在程序框架内独立完成教材相关代码的演示，就能掌握相关章节内容。

本教材的教学讲义已在教学实践中使用，所有教学内容、案例均在教学实践中使用。全部内容的课堂教学 54 学时，实验 20 学时，综合实训 24 学时。

本教材根据教学讲义改编而成，教学讲义编写过程中参考了大量游戏开发的书籍和资料，包括荣钦科技出版的《Visual C++ 游戏编程》教材，毛星云编写的《Windows 游戏编程之从零开始》，《Unity 4.X 从入门到精通》和《Unity 开发网站》等，在此表示衷心感谢。

◇ 目 录 ◇

第 1 章 游戏程序设计概述	1
1.1 计算机游戏的概念	1
1.2 计算机游戏的分类	2
1.2.1 角色扮演游戏	2
1.2.2 动作类游戏	3
1.2.3 实时策略游戏	3
1.2.4 第一视觉射击游戏	3
1.2.5 模拟游戏	4
1.2.6 体育类游戏	4
1.3 设计游戏的要素	4
1.3.1 策划	5
1.3.2 程序	5
1.3.3 美术	5
1.3.4 音乐	6
1.4 计算机游戏的发展趋势	6
1.4.1 沉浸感	6
1.4.2 交互性	6
1.4.3 国际化	7
思考题	7
第 2 章 Win32 应用程序	8
2.1 Visual Studio 2010 中 Win32 应用程序创建过程	8
2.2 Windows 主函数	11
2.3 窗口建立过程	12
2.4 Windows 消息循环	14
2.5 窗口过程函数	15
2.6 Win32 应用程序示例	16
2.7 游戏程序框架	18
思考题	21
第 3 章 Windows 绘图函数	22
3.1 屏幕绘图的相关概念	22
3.1.1 窗口和视口	22

3.1.2	GDI 坐标系	23
3.2	画笔、画刷与文字	25
3.2.1	GDI 对象的建立	25
3.2.2	GDI 对象的选用与删除	27
3.2.3	GDI 示例	27
3.3	GDI 绘图函数	31
3.3.1	点线函数	31
3.3.2	形状函数	32
3.3.3	填充函数	33
	思考题	33
第 4 章	游戏中的角色与场景	34
4.1	位图显示	34
4.2	镂空贴图	37
4.2.1	使用 BitBlt() 中的参数 Raster 完成镂空贴图	38
4.2.2	使用 TransparentBlt() 函数完成镂空贴图	39
4.3	地图显示	40
4.3.1	平面拼接地图	40
4.3.2	斜角拼接地图	43
	思考题	47
第 5 章	动画	48
5.1	使用定时器完成游戏动画	48
5.1.1	建立定时器	48
5.1.2	删除定时器	49
5.1.3	示例	49
5.2	利用消息循环完成游戏动画	51
5.2.1	利用消息循环完成动画原理	51
5.2.2	示例	52
	思考题	54
第 6 章	键盘与鼠标交互	55
6.1	Windows 键盘消息处理	55
6.1.1	Windows 键盘概述	55
6.1.2	键盘消息处理	56
6.1.3	键盘交互程序示例	57
6.2	Windows 鼠标消息处理	61
6.2.1	鼠标消息的处理方式	61
6.2.2	示例	63
6.2.3	相关函数的讲解	69

思考题	70
第 7 章 运动与碰撞检测	71
7.1 运动	71
7.1.1 匀速直线运动	71
7.1.2 变速运动	75
7.2 碰撞检测	77
7.2.1 以物体框架来检测碰撞	77
7.2.2 用颜色来检测碰撞	79
7.3 粒子系统	83
思考题	83
第 8 章 3D 游戏概述	84
8.1 3D 坐标系及转换	84
8.2 模型对象的建立	85
8.3 视图变换	85
8.3.1 平移变换	86
8.3.2 旋转变换	86
8.3.3 缩放变换	87
8.4 投影变换	87
8.5 3D 游戏的开发手段	88
思考题	89
第 9 章 Direct 3D 简介	90
9.1 Direct 3D 的体系结构	90
9.1.1 Direct 3D 的绘制流程	90
9.1.2 Direct 3D 绘制程序框架图	91
9.2 Direct 3D 开发环境配置	92
9.3 Direct 3D 初始化	94
9.3.1 创建 Direct 3D 接口对象	95
9.3.2 获取设备的硬件信息	95
9.3.3 填充 D3DPRESENT_PARAMETERS 结构体	96
9.3.4 IDirect 3D 设备接口的创建	98
9.4 Direct 3D 渲染	99
9.4.1 清屏操作	99
9.4.2 绘制	100
9.4.3 翻转显示	100
9.4.4 Direct 3D 的渲染过程	101
9.5 Direct 3D 中二维文本的绘制	102
9.6 Direct 3D 框架程序	104

思考题	110
第 10 章 Direct 3D 图形绘制基础	111
10.1 以顶点缓存为数据源的图形绘制	112
10.1.1 基础知识	112
10.1.2 在 Direct 3D 编程中使用顶点缓存的四个步骤	113
10.2 顶点缓存程序示例	120
10.3 以索引缓存为数据源的图形绘制	123
10.4 索引缓存程序示例	128
10.5 Direct 3D 内置几何体概述	132
10.5.1 立方体的创建	132
10.5.2 圆柱体的创建	133
10.5.3 2D 多边形的创建	134
10.5.4 球体创建	134
10.5.5 圆环的创建	135
10.5.6 茶壶的创建	135
思考题	136
第 11 章 Direct 3D 变换	137
11.1 视图变换	137
11.2 投影变换	141
11.3 视口变换	142
11.4 Direct 3D 变换示例	143
11.5 Direct 3D 固定功能渲染流水线概述	148
思考题	149
第 12 章 Direct 3D 光照与材质	150
12.1 光照类型	150
12.1.1 环境光	150
12.1.2 漫反射光	151
12.1.3 镜面反射光	151
12.1.4 自发光	152
12.2 光源类型	152
12.2.1 点光源	154
12.2.2 方向光源	154
12.2.3 聚光灯光源	155
12.3 材质	155
12.4 灯光与材质示例	157
思考题	162

第 13 章 Direct 3D 纹理映射	163
13.1 纹理映射的概念	163
13.2 Direct 3D 中纹理映射的实现方法	163
13.2.1 纹理坐标的定义	163
13.2.2 顶点坐标与纹理坐标的对应	164
13.2.3 纹理的创建	165
13.2.4 纹理的启用	166
13.3 纹理绘制示例	168
思考题	173
第 14 章 游戏引擎	174
14.1 什么是游戏引擎	174
14.2 目前比较流行的几款主流引擎	175
14.3 Unity 游戏引擎简介	176
14.4 Unity 下载与安装	178
思考题	180
第 15 章 Unity 程序开发框架和编辑器使用	181
15.1 Unity 程序开发框架、工程和应用以及场景的关系	181
15.2 工程的创建和导入	182
15.3 Unity 编辑器介绍	184
15.3.1 官方资源导入方法	184
15.3.2 场景中的 6 个视图	185
15.3.3 编辑器界面设置	186
15.3.4 Unity 编辑器——Project（项目视图）	186
15.3.5 Unity 编辑器——Hierarchy（层次视图）	188
15.3.6 Unity 编辑器——Inspector（检视视图）	190
15.3.7 Unity 编辑器——Scene（场景视图）	192
15.3.8 Unity 编辑器——Game（游戏视图）	194
15.3.9 Unity 编辑器——Console（控制台视图）	195
思考题	195
第 16 章 资源和游戏对象	196
16.1 Unity 资源	196
16.1.1 场景、资源、游戏对象、组件间的关系	196
16.1.2 内部资源创建	198
16.1.3 外部资源导入	199
16.1.4 Unity 中预设的创建	200
16.1.5 Unity 中图片、模型和音频、视频的支持	200

16.2	Unity 常用组件介绍	202
16.2.1	Transform (变换组件)	202
16.2.2	Camera (摄像机组件)	203
16.2.3	Lights (光源)	205
16.3	常用物理引擎组件	206
16.3.1	Rigidbody (刚体组件)	206
16.3.2	Collider (碰撞器组件)	207
	思考题	208
第 17 章	Unity 脚本程序基础	209
17.1	什么是脚本程序	209
17.2	Unity 脚本编辑器	210
17.3	Unity 脚本的创建与编辑	211
17.3.1	Script (脚本) 创建	211
17.3.2	编辑脚本程序	212
17.3.3	常用事件函数	214
17.3.4	游戏对象和组件访问	215
	思考题	217
第 18 章	Unity GUI 图形用户界面	218
18.1	UGUI 的基本介绍	218
18.2	UGUI 的创建和基本操作	219
18.3	UGUI 实例演示	220
18.3.1	GUI 之 Button 和 Text	221
18.3.2	GUI 之 Toggle 应用	222
18.3.3	GUI 的 Image 和 Scrollbar 应用	224
18.3.4	通过 Button 调用其他场景	225
18.3.5	Slider 与游戏对象	226
18.4	打包与发布	227
	思考题	228
第 19 章	Mecanim 动画系统	229
19.1	Mecanim 动画系统及其优势	229
19.2	Mecanim 工作流程	230
19.3	人形角色动画讲解	230
	思考题	238
第 20 章	游戏开发实例——奔跑的轮胎	239
	思考题	260
	参考文献	261

游戏程序设计概述

本章学习要求：

了解计算机游戏的相关概念、分类，游戏设计的要素及发展趋势。

1.1 计算机游戏的概念

游戏在英文中的单词是 Game，意译为“比赛、竞争、游戏”，通过非对抗性的、友好的体力与技巧比赛，使得游戏参与者在体力上得到锻炼，同时在精神上也得到了乐趣。因此游戏可以如下定义：游戏是具有特定行为模式、规则条件、身心娱乐和胜负判定的一种行为表示。

在游戏的定义中，行为模式表示游戏特定的流程模式，这种流程模式是贯穿整个游戏的行为，例如，猜拳游戏中包括剪刀、石头、布三种行为模式。规则条件是游戏参与者必须遵守的规则。身心娱乐是游戏所带来的娱乐性，是游戏的精华所在。一个没有输赢胜负的游戏，仿佛少了它存在的意义，因此可以说，胜负是所有游戏的最终目的。

随着计算机技术的发展，计算机参与进原本人与人之间的游戏，以计算机来代替原来必须由人来承担的角色，这样的游戏称为**计算机游戏**。可以从游戏技术，游戏内容和玩家角度三个方面体会计算机游戏的本质。

从游戏技术上看，计算机游戏是“以计算机系统为平台，通过人机互动形式实现的，能够体现当前计算机技术较高水平的一种新形式的娱乐方式”。计算机游戏的实现技术基础是计算机技术。

从游戏内容看，计算机游戏是一个让玩家追求某种目标，并且让玩家可以获得某种“胜利”体验的娱乐性文化产品。计算机游戏具有丰富而独特的表现力，能表现出许多鲜明生动的艺术形象，从这个意义上说，计算机游戏和戏剧、电影一样，是一种综合性艺术，并且是融合了技术的、更高层次的综合艺术。有人把计算机游戏称为继绘画、雕刻、建筑、音乐、诗歌（文学）、舞蹈、戏剧、电影（影视艺术）之后的人类历史上的第 9 种艺术。计算机游戏的艺术性体现在世界观、剧情、人物、规则上，再加上表现这些的媒体（音乐和画面）。计算机游戏建构了一个虚拟的世界，游戏的创作人员要使得这个虚拟的世界具有“价值”，才能被玩家接受，因此要创作出这个世界中的历史、各种力量和它们的均衡、善恶准则，甚至要创作出具有特定人文特色的风俗习惯等，然后再在这个虚拟世界加入玩家的

角色，制定好相关的游戏规则。

从玩家角度看，计算机游戏能够提供其他艺术形式无法提供或无法满足的内容。例如，可以为玩家提供挑战的机会和场所；由于计算机游戏具有虚拟的社会性，可以给玩家提供与朋友进行交流的机会；提供玩家独处的经历；提供满足感；提供情感的体验；提供幻想等。

1.2 计算机游戏的分类

1.2.1 角色扮演游戏

RPG (Role-Playing Game, 角色扮演游戏) 通常以科幻故事或历史题材为背景来构造完整的故事情节，玩家在游戏中扮演一个或多个角色，通过练级或发展剧情来完成游戏。目前流行的 RPG 分为回合制和即时制，回合制游戏注意剧情推动游戏进程，强调故事性，代表作有《仙剑奇侠传》、《最终幻想》等；即时制游戏的规则性强，故事会带有奇幻色彩，代表作有《暗黑破坏神》。

RPG 强调对人生的模拟，如果说飞行模拟类、体育类、动作类等类型的游戏都是对现有的某项人类活动的再现与模拟的话，那么 RPG 体现的则是对整个人生的模拟，因此 RPG 所构造的情感世界是所有类型的游戏中最为强大的，能带给我们深刻的体验感，这种体验感来源于每个人内心深处对人生的感悟和迷茫、无奈与苛求、失意与希望，这些体验可以在 RPG 所构造的虚拟人生的情感世界中得到共鸣。

RPG 可以用一个三维坐标系来定位，如图 1-1 所示。构成 RPG 的艺术性、故事性和交互性三大特性分别构成该坐标系的三个轴，若把每个坐标轴的最大坐标值定为 1，那么坐标点 $(0,0,1)$ 代表纯粹的艺术作品，如视觉艺术和音乐作品等； $(1,0,0)$ 点代表完全的操作性活动，如体育运动； $(0,1,0)$ 点则代表故事情节及其纯线性的展现和播放，如电影剧本、VCD 和录像带。而 RPG 则位于点 (x, y, z) ，其中 $0 < x < 1$ ， $0 < y < 1$ ， $0 < z < 1$ 。而不同类型的 RPG，在这个三维空间所处坐标不同。偏重交互性的，其 x 值较大；偏重故事性的，其 y 值较大。需要指出的是： x, y, z 的值都不能为 0，因为构成 RPG 的三大特性缺一不可。

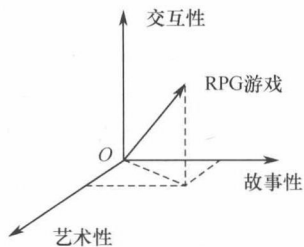


图 1-1 RPG 游戏定位

在艺术性方面，RPG 借助于多媒体视听技术，综合利用了美术、动画、音乐、音效、

文学、戏剧等多种艺术娱乐表达形式。在故事性方面，RPG 和电影的关系密切，其“情节”是由“剧本”严格限定的，也就是单线发展的。与被动欣赏的电影不同的是 RPG 给游戏者提供了与游戏交互的能力，玩家在游戏中有一定的主动性，但这种主动性改变不了原本设定的故事情节，RPG 模型一般由各个事件顺序链接而成，在每个事件中玩家具有交互能力。

1.2.2 动作类游戏

ACT (Action Game, 动作类游戏) 最初是家用游戏机上最流行的游戏类型，游戏画面以横卷轴方式展开，玩家要从左向右一关一关地闯过去，最终打倒大魔头。

目前流行的动作游戏通常为第一视角类游戏，该类游戏中，角色对手都被武装起来，游戏通过角色视觉变化发展。动作游戏通常有多个玩家同时在线，游戏对手往往是人控制，而不是计算机操作的，想在游戏中获胜需要快速的反应、良好的手眼协调能力，并且要熟悉游戏中的武器装备。也有一些动作游戏是按照第三视角完成的，玩家可以看到自己的游戏角色以及游戏角色所处的虚拟世界。

ACT 的代表作包括《超级玛丽》，目前三维 ACT 游戏典型的例子为《古墓丽影》系列。

1.2.3 实时策略游戏

在 RTS (Real-Time Strategy Game, 实时策略游戏) 中，游戏双方往往都是人，实时是指游戏一方在生产、制造和布阵的同时，另一方也在进行各种操作。当一方停下来时，敌方不会停下来，这样双方时刻在进行生产和制造的竞争，使得游戏更加紧张刺激。策略从广义角度讲是通过资源采集、生产、后勤、开拓和战争来振兴本种族或国家，战胜其他种族或国家；从狭义角度讲是指用来击败敌人的各种军事指挥手段。许多计算机技术的出现是从 RTS 中提出来的，比如寻径算法、人工智能、指令序列等，这些技术后来又应用到其他类型的游戏中。

大多数 RTS 的游戏规则都遵循“采集—生产—进攻”三部曲原则。即通过对几种资源的采集和利用，来构建基地或城市，生产武器，组建军队，然后向敌方发起进攻。RTS 最重要的两个要素是资源管理和狭义的战争策略。

RTS 代表作包括“星际争霸”(Starcraft)，“魔兽争霸”(Warcraft) 和“帝国时代”(Age of Empires)。

1.2.4 第一视觉射击游戏

FPS (First-Person Shooter, 第一视觉射击游戏) 起源于早期苹果机上的迷宫游戏和游戏机上的 ACT，通过引入第一视角和三维图形使得游戏的表现力得到极大提高。首先是置入感的提高：三维世界和第一视角的应用使得玩家第一次能够感到他们“面对”着一个真实的三维世界。其次是交互性的提高：三维地图使得玩家们摆脱了 ACT 由一个路线前进的

限制，玩家可以沿多种路径到达终点，更增加了搜索前行的乐趣和不确定性。NPC (Non-Player Character) 概念的提出使得玩家所要面对的对手有了一定的智能和适应性，不再是像 ACT 中的敌人那样定点定时出现。

FPS 的要素有以下三点：

(1) 三维关卡 (3D Level)：关卡这个要素是从 ACT 游戏中继承的。整个游戏由一系列关卡组成。每个关卡有自己独特的三维场景。

(2) 任务 (Mission)：任务就是玩家在一个关卡里要完成的使命和要达到的目的。不管是夺取旗帜也好，还是安置炸弹也好，达到这个目的才能通过这一关，进入下一关。任务可以是相互嵌套的，比如一个主任务的完成需要先完成几个子任务。

(3) NPC：NPC 是具有一定智能和适应性的，不是由玩家来控制的敌方或者友方。游戏业人工智能技术的应用最初就是从 NPC 上发展起来的。NPC 要阻碍 (敌方) 或者帮助 (友方) 玩家完成任务。

《Quake》(雷神之锤) 系列和《Unreal》(虚幻) 系列是 FPS 的代表作，《Half-Life》被认为是 FPS 的突破性作品。

目前 FPS 的发展趋势是：第一，提高 NPC 的人工智能，引入小组机制，使得 FPS 不仅仅是疯狂扫射和冲锋，而有了战术配合。第二，被 FPS 长期忽视的故事性也越来越得到强化。第三，联网对战使得 FPS 更加刺激。受到空前好评的《Half-life》(半条命) 就代表了这三种趋势。

1.2.5 模拟游戏

SIM (Simulation Game, 模拟游戏) 包括策略模拟类游戏，如《三国志》系列，恋爱模拟类游戏，如《心跳回忆》系列。模拟类游戏的共有特征是复杂的数字式管理，如在《三国志》中的各城市各武将的数值，在《心跳回忆》中有各种各样表达人物状态的数字，因此模拟游戏实际上是用一个非常粗糙的数学模型和数字式管理来实现的。

1.2.6 体育类游戏

SRT (Sports Game, 体育类游戏) 涵盖三个层次：管理、战术、技能。技能方面指单纯的模拟某项运动，比如赛车游戏；战术是指足球的团队配合和排兵布阵等；管理，包括俱乐部的管理和球员的培训。这类游戏有《FIFA》系列足球游戏。

以上介绍的几种游戏类型的划分并不严格，因为游戏制作时会根据实际需求运用各种技术，游戏类型的分类是一种习惯和约定俗成，其实用性大于理论性，游戏类型的主要作用是有利于业界人士之间和业界和玩家之间的沟通。

1.3 设计游戏的要素

游戏设计包括四个要素，即策划、程序、美术和音乐。

1.3.1 策划

游戏策划制定出整个游戏的规划、流程与系统。策划人员必须编写出一系列的策划书供其他游戏参与人员阅读。通常，策划人员所要做的工作可以归纳为下列几点：

- (1) 游戏规划：游戏制作前的资料收集与环境规划。
- (2) 架构设计：设计游戏的主要架构、系统与主题定义。
- (3) 流程控制：绘制游戏流程与控制进度规划。
- (4) 脚本制作：编写故事脚本。
- (5) 人物设置：设置人物属性及特性。
- (6) 剧情导入：把故事剧情导入引擎中。
- (7) 场景分配：场景规划与分配。

1.3.2 程序

程序是实现游戏的载体，在策划书中的内容必须利用程序来加以组合成形。程序由框架师和程序员完成，框架师必须了解策划人员的构想计划，根据他们的想法与理念，将设计转化成实际游戏，框架师还要具备拆解策划书的能力，将分解出来的游戏功能分配给程序员编写，在程序员完成相关代码后，将各个代码正确应用到系统中，以达到策划人员所要求的画面或功能。

程序设计人员所要做的工作，可以归纳为下列几点：

- (1) 编写游戏功能：编写策划书上的各类游戏功能，包括编写各种编辑器工具。
- (2) 游戏引擎制作：制作游戏核心，而核心程序足以应付游戏中发生的所有事件及图形管理。
- (3) 合并程序代码：将分散编写的程序代码加以结合。
- (4) 程序代码除错：在游戏的制作后期，程序人员可以开始处理错误程序代码，及重复进行侦错的动作。

1.3.3 美术

对于玩家而言，最直接接触他们的就是游戏中的画面，在玩家尚未真正操作游戏的时候，可能会先被游戏中的绚丽画面所吸引，进而去玩这款游戏，因此优秀的美术人员是非常重要的。美术人员所要做的工作，我们可以将它归纳为下列几点：

- (1) 人物设计：不论是 2D 还是 3D 的游戏，美术人员必须根据策划人员所规划的设置，设计与绘制游戏中所有需要的登场人物。
- (2) 场景绘制：在 2D 游戏中，美术人员要一张张地画出游戏所需要的场景图案；在 3D 游戏中，美术人员必须绘制出场景中所有必须要使用到的场景对象，以提供给地图编辑人员使用。
- (3) 界面绘制：除了游戏场景与人物之外，还有一种经常在游戏中所看见的画面，那

就是使用界面。这种用户界面就是让玩家可以与游戏引擎做直接沟通的画面。美术人员要把亲和性与方便性作为设计用户界面的原则。

(4) 动画制作：游戏中少不了会有几个串场的动画，美术人员会根据策划书的需求制作出音效十足的动画。

1.3.4 音乐

音乐的衬托能够增强游戏的娱乐性，音乐制作人员需要根据具体的场景和情节做出游戏中所需要用到的音效与相关的背景音乐。

1.4 计算机游戏的发展趋势

计算机游戏的发展与计算机技术的发展密不可分，主要依托虚拟现实、人工智能、计算机图形学、计算机视觉和交互技术等学科的发展。

从玩家的角度来看，计算机游戏的发展主要归纳为以下三个方面。

1.4.1 沉浸感

所谓沉浸感，就是沉浸在计算机构建的虚拟环境中。为了使玩家获得这种感觉，首先要在计算机中构建并渲染出与现实世界尽可能相同的环境，另外尽量减少现实世界对玩家的干扰。

1.4.2 交互性

游戏作为一种娱乐形式，第一要实现置入感，构建一个虚拟的游戏世界；第二要提供交互性，使得玩家可以和这个虚拟的游戏世界进行交流。交互性体现在两方面：交互手段、交互的可能反应。我们用一个简单的例子来说明这两个概念。

假定在一个游戏中，游戏人物需要从山崖上摘一朵花献给自己的女友。首先我们遇到的问题是如何让玩家操纵游戏人物去摘花。可以有以下4种设计方案。

最简单的一种设计：当玩家操纵游戏人物走到距花朵的一段距离内时，弹出一个窗口，显示提示“花朵在附近，是否摘下？”，提示下方有“是”和“否”两个按钮。玩家可以在“是”的按钮上点一下以摘下花朵。

复杂一点的设计，可以让玩家操纵游戏人物移动到花的旁边，然后改变屏幕光标的形状，改成手的形状，同时显示一个信息条：“在花朵上按鼠标按钮即可将其摘下！”这时候，玩家按一下鼠标，花朵被摘下来了。

更复杂的设计，是类似于《黑与白》中的姿势识别与控制系统：玩家所控制的人物走到花朵的旁边，玩家移动鼠标使得屏幕光标位于花朵上方，这时屏幕光标改变成一只张开的手的形状，当玩家在花朵上按下鼠标按键后，屏幕光标显示一只收拢的手的形状，玩家