



# INTRODUCTION TO COMPUTER 3D GAME DEVELOPMENT

**Nature of Games & Course Overview**

潘茂林, [panml@mail.sysu.edu.cn](mailto:panml@mail.sysu.edu.cn)

中山大学·软件学院

# 课程须知：课程核心知识

○ 指导教师：潘茂林

○ 课程核心内容：

- 3D 游戏编程基础（不仅是 Unity）
- 面向对象的游戏设计方法（不只是 Programming）
- 3D 游戏架构与框架（不限定引擎）
- 以用户为中心的游戏设计原理
- 基于大数据的游戏分析与评估

○ 如何获得作业，教材等相关资源

- <https://pmlpml.github.io/unity3d-learning/>



# 课程须知：预备知识与教材

## ○ 前置知识需求

- 熟悉 c 或 c++，最好了解 c#
- 最好了解脚本与描述语言，如：lua,, json, xml
- 知道软件设计，如：UML、设计模式
- 知识 AI 算法， 如：A\*

## ○ 教材

- Unity: <http://unity3d.com/cn/unity>
- 教学案例：参考上届同学的博客

## ○ 游戏设计参考书

- Adams 《游戏设计基础》
- Tracy Fullerton 《游戏设计梦工厂》



# 目录

- 游戏的本质（What is a Game）？
- 游戏的历史与未来
- 游戏的应用与分类
- 视屏游戏的原理与引擎架构
- 游戏的开发流程
- 游戏创意表达
- 课程学习方法与考核



# 游戏? (WHAT IS A GAME?)

○ ?



- Game Theory
  - 游戏理论?



## 游戏原理（博弈理论）

### ○ 案例：“囚徒困境”（prisoner's dilemma）

**警察与小偷的游戏：**某团伙两小偷（A，B）被警察叔叔抓了，警察叔叔隔离审问，并告知“坦白从宽，抗拒从严”。小偷只能选择“坦白”或“抗拒”，后果如下表。请问，两个小偷各自的选择是什么？

A\B	抗拒	坦白
抗拒		
坦白	0:10	7:7

A\B=1:1 表示A判1年，B判1年。



# 游戏中的概念 (CONCEPT OF GAME)

- 玩家、对手 (player)
- 规则 (rule)
  - 行为集合 (action set)
  - 结果、报酬 (payoff)
- 策略与决策 (policy & decision making)
  - 目标 (goals)
  - 有意义的选择、决策 (meaningful choice)
  - 有限理性玩家 (limited-rational player)
- 结局 (outcome)
  - 纳什均衡 (Nash Equilibrium)
  - 帕累托最优 (Pareto Optimality)



# 信息与决策

## (INFORMATION & MAKING DECISION)

- 先决知识（玩家个体差异）
  - 不由游戏设计者设定的知识与技能，玩家的智力、体力等。所以游戏必须设定一些前提，如围棋的段位
- 合作、共谋（cooperative）
  - 私下合作。例如，4 人玩桥牌。其中 2 人经常使眼色
- 完全与不完全信息（complete vs. incomplete）
  - 玩家获取的信息可以预测对手的决策。
- 完美与不完美信息（perfect vs. imperfect）
  - 玩家看到所有对手的行为历史，但不一定知道决策依据
- 对称与不对称（symmetric game）
  - 玩家决策不依赖私有信息





# 游戏的定义

- Adams: *Fundamentals of Game Design*
  - A **game** is a form of *interactive* entertainment where *players* must overcome *challenges*, by taking *actions* that are governed by *rules*, in order to meet a *victory condition*.
- Salen & Zimmerman: *Rules of Play*
  - A **game** is a system in which *players* engage in *artificial conflict*, defined by *rules*, that results in a *quantifiable outcome*.



# 游戏要素 (GAME ELEMENTS)

## ○ 游戏目标

- 测天意、展示能力、训练、交友、消磨时间、赚钱等等
- Adams: 满足玩家潜在的欲望（成就感）

## ○ 基本元素

- Players, Rules, Challenges, Outcome

## ○ 扩展元素

- 场景设定与故事 (Setting/Promise & Story)
  - ◆ 角色 (Characters)
  - ◆ 冲突 (Conflict)
- 玩法 (Game Play)
- 美术、音效
- ...



## 案例研究（信息与决策）

	投壶	井字棋	围棋	桥牌	四人麻将	飞行棋
先验条件	喝酒	小孩		段位		
共谋风险?	X			√		
完全信息?	√*			X		
完美信息?	√			√		
对称信息?	√			X		

X: 否

√: 是

√\*: 应该是，但满足先验条件就不是了



# 视频游戏 (VIDEO GAMES) 历史 (1)

- Early history (television technologies)
  - 1949–1950, Charley Adama, "Bouncing Ball", MIT, Whirlwind computer
  - 1952, A.S. Douglas, "tic-tac-toe", University of Cambridge, EDSAC computer
- 1950s–1960s (university mainframe computers)
  - 1961, Steve Russell, "Spacewar!", MIT, PDP-1
- 1970s (the first coin-operated video game, home console)
  - 1971, "Galaxy Game", Stanford University, PDP-11
  - 1972, Agnavox, circuit logic, TV
- 1977–1983 (microprocessors)
- Golden age of video arcade games (1978–1986)
- Home computer games (late 1970s–early 1980s)



## 视频游戏历史 (2)

### ○ 1980's

- Genre (题材) innovation
  - Adventure game, Beat 'em up, cinematic platformer,
  - Role-Playing Games, Fighting games, hack and slash,
  - Interactive movies, Shooter, Racing, Survival horror,
  - Vehicle simulation, Visual novel .....

### ○ 1990's (high performance PC, network)

- 3D, “3D Realms”
- MMORPG, “MUD”
- Real-time strategy, “Warcraft”
- First-person shooters, “Quake”



## 视频游戏历史 (3)

- 2000s (mobile game)
- 2010s (touchscreen, 3D controller)
- And now!!!
- AR 游戏
- VR 游戏



# 游戏应用

## ○ 庞大的产业

- 近十年来产业规模成倍增长，已超过传统的电影产业
- 有统计数据，美国青年每天花在游戏上的时间超过20分钟。游戏广泛运行于PC, 手机, 互联网

## ○ 热门的行业

- 娱乐市场、体育市场
- 教育市场（驾驶、生产线操作、救助等真实场景模拟）
- 电子商务、旅游导览等市场
- 医疗服务市场

## ○ 广阔的就业

- 良好的薪酬
- 策划、开发、测试、运营、游戏大数据分析



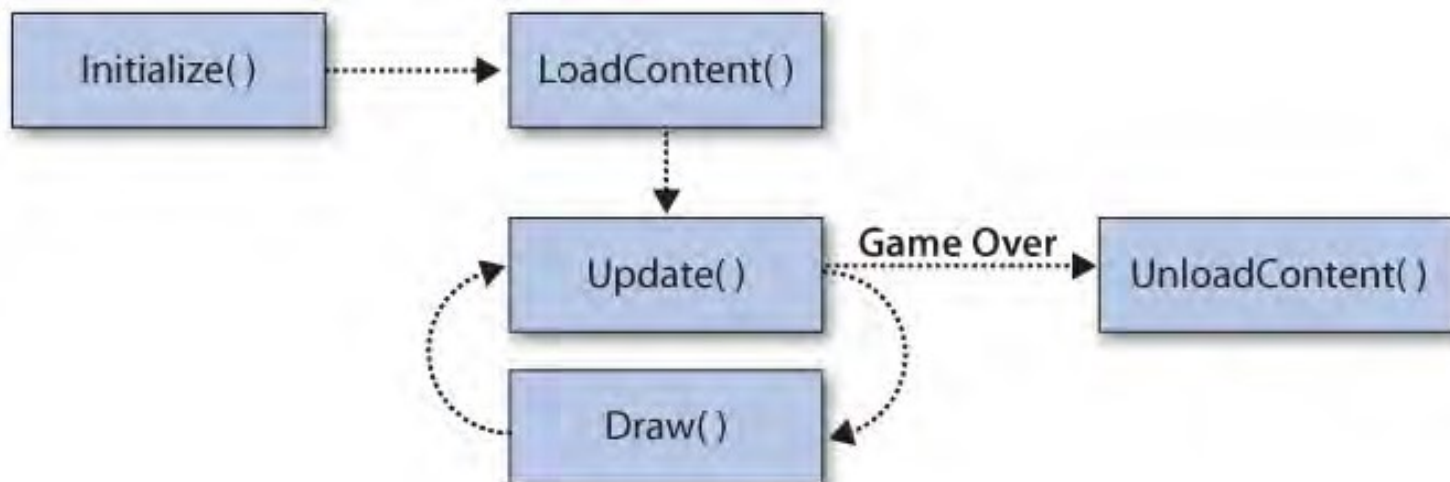
# 游戏的分类

- 交互技术(interaction)
  - 2d, 卷轴-2d, 2.5d, 3d, AR,VR
- 题材 (genre/type)
  - Puzzle, Shoot, Escape, Adventure, RPG ... ..
- 平台 (platform)
  - PC, 游戏机, 手机, 页游, Flash游戏 ... ..
- 玩家 (players)
  - 单人, 双人, 多人, 组队, 人/机
- 视角 (perspective)
  - 第一人称, 第三人称
- 网络 (internet)
  - 离线, 在线





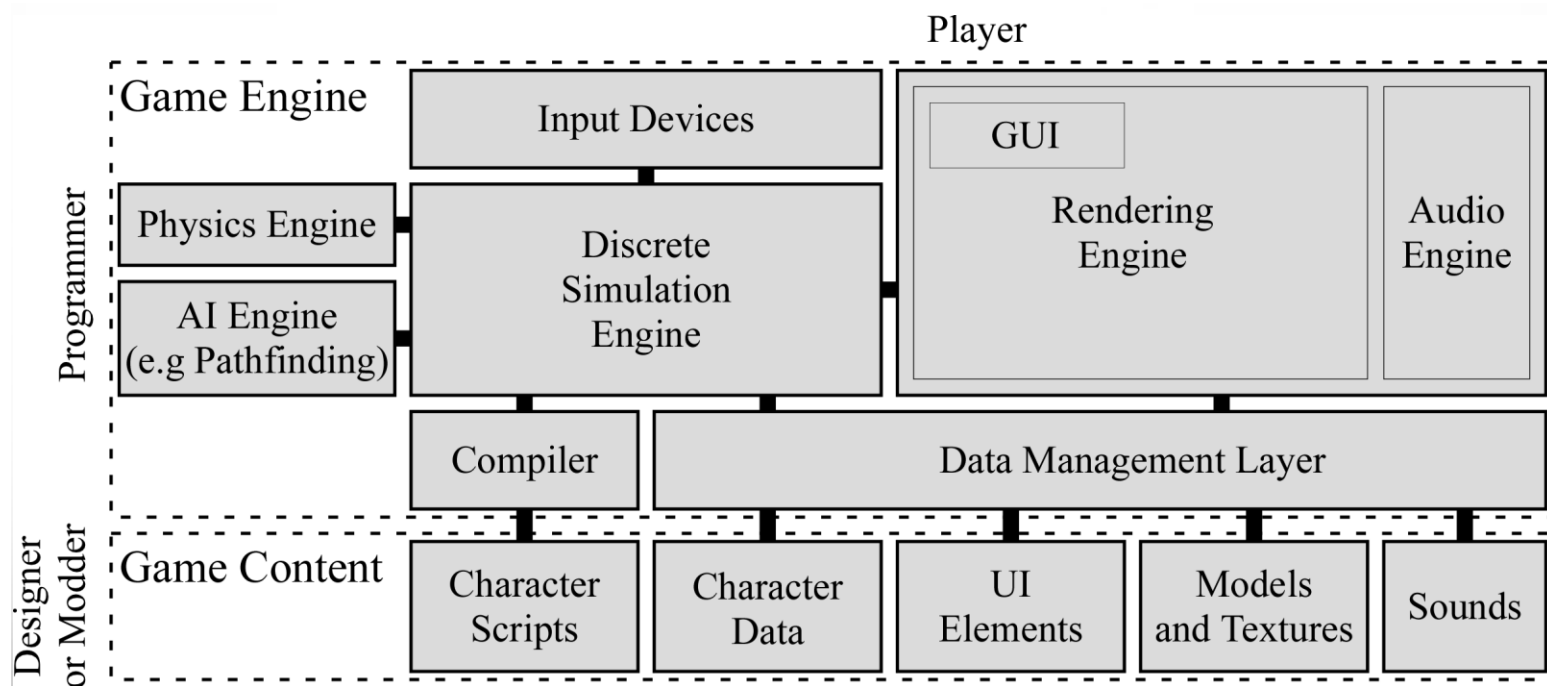
# 视频游戏编程原理



1. 初始化 – 加载资源
2. Update – Draw 循环



# 游戏引擎架构 (1)



## 游戏引擎架构 (2)

### ○ 玩家

- 输入设备 (...), 输出设备 (...)

### ○ 程序员

- 离散仿真引擎 (循环定时执行)
- 输入设备、渲染引擎、语音引擎、物理引擎、AI引擎
- 数据管理层 (资源管理层)

### ○ 设计师或建模者

- 脚本、角色数据 (配置数据, 如 XML, JSON等)
- UI元素、3D 模型与贴图、语音

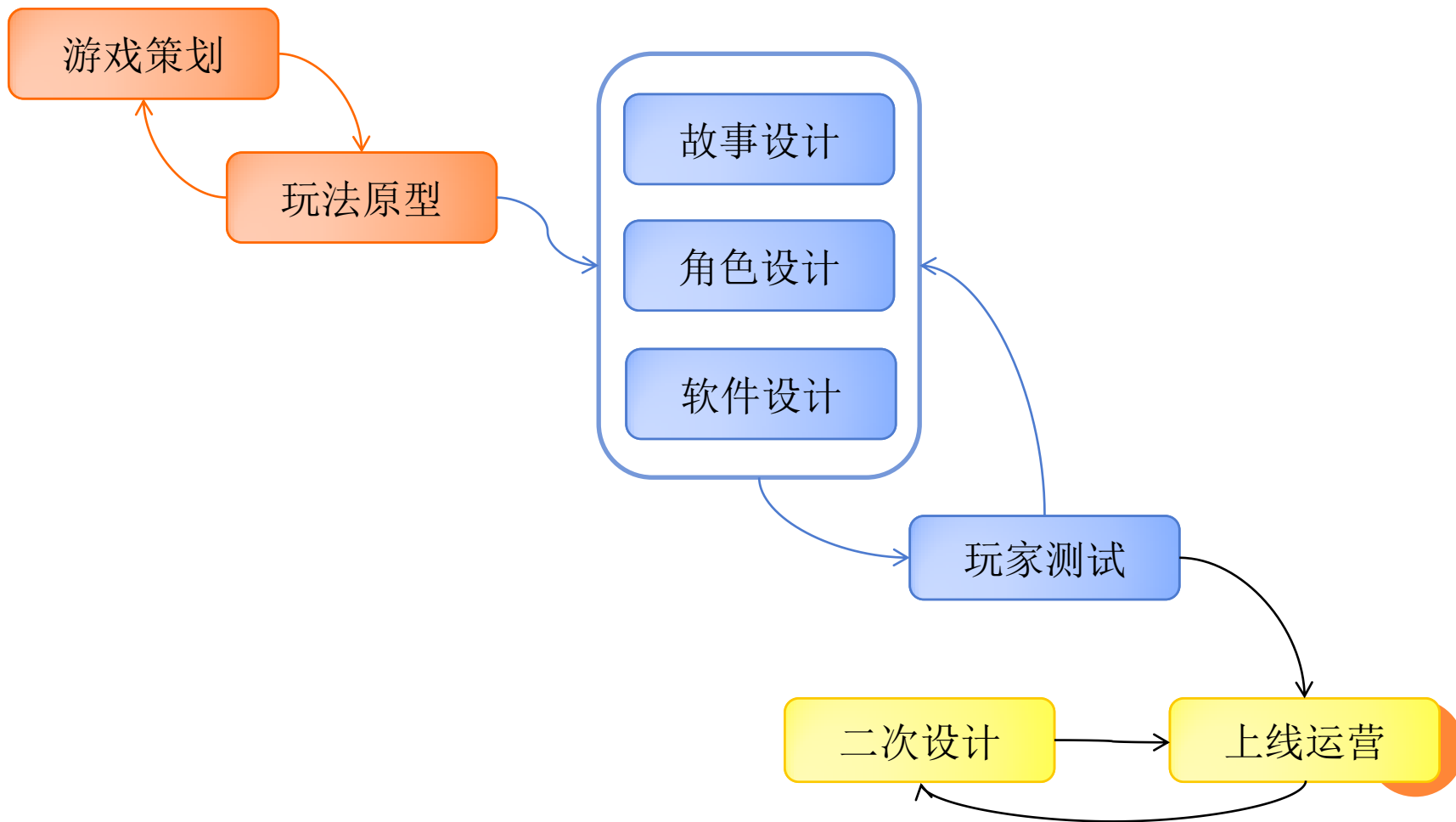


## 游戏引擎架构 (3)

- 数据（资源）与引擎（代码）分离设计必要性
  - 游戏生命产品周期短
  - 游戏升级快
  - 资源需要动态更新（在线升级）
- 数据（资源）分两大类
  - 规则（脚本 + 配置数据）
  - 实体（模型、图片、语音、UI元素）
- 游戏代码开发成本高，周期长
  - 代码要能复用和通用，适应客户需求的变化
  - 设计水平远高于普通程序员水平！！！！



# 游戏开发过程模型 (1)



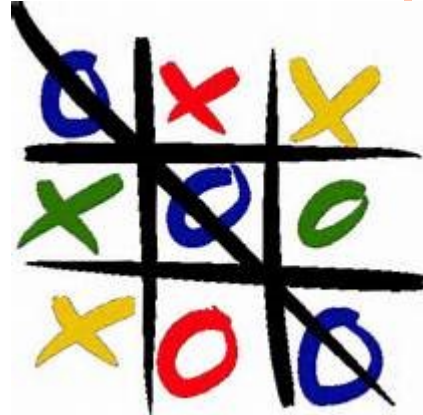
## 游戏开发过程模型（2）

- 游戏是高风险项目，三阶段开发模型的目标：
  - 持续评估风险，降低成本
  - 保持开发、设计持续快速更新
- 以策划为中心的原型循环（一）
  - 投资人、市场等评估核心玩法、人物形象、市场定位等要素，决定是否继续投资
- 以玩家为中心的设计、开发循环（二）
  - 故事设计包括游戏故事结构，角色出场顺序，难度曲线
  - 角色设计包括环境、角色的模型与视觉、听觉艺术效果
  - 软件设计包括架构、模块设计、编码与测试
- 以运营为中心的升级循环（三）
  - 二次设计，指添加、修改资源的升级活动
  - 游戏有20-30%的资源就上线拉！是否继续由市场决定



# 游戏表达与游戏元素

## (TIC-TAC-TOE)



### 目标陈述

这是一个双人游戏，比谁先在3\*3格子上形成连珠。

### 玩法

1. 游戏需要的工具仅为纸和笔，画出3\*3的格子
2. 用O和X然分别代表两个游戏者，轮流在格子里留下自己的标记
3. 直到一方满足胜利条件

### 规则

规则名称	前提条件	动作	结果
投子规则	yours turn & has blanks	选择一个空位投下符号	
结束规则	行、列、对象出现相同符号	结束游戏	该轮玩家胜出
	没空格	游戏结束	平局

# 游戏创意的表达

- 游戏背景（Setting）
  - 通过一个故事描述玩家的人设与使命
- 游戏特色（Features）
  - 3-5 个最吸引玩家的特点
- 玩法（Game-play）
  - 最简洁完成任务的故事，或一组场景图片构成的故事
  - 主要武器、道具的描述
  - 核心的挑战任务的描述
- 技术（Technology）
  - 新颖的人机交互模式（例如：用 Knect 获取用户太极拳动作）
  - 描述潜在有趣的 AI（如：AI 自动评分算法）





# 课程学习方法

- 课程是案例驱动的。
  - 在课程之间，你需要独立完成 6-8 个游戏
  - 具体语言案例实现，可以参考往年同学的博客
- 教学不是 step by step 的
  - 注重对引擎架构中编程部件的理解
  - 注重面向设计方法和中高级语言技巧
- 课程并没有设定 3D 游戏引擎
  - 选用 Unity 5.5 是因为易用，易于入门，自学资料丰富



## 课程项目要求（可选）

### ○ 项目

- 组织一个游戏团队, 开发 **compelling** 和 **creative** 的 3D+VR 视频游戏或应用.

### ○ 团队要求

- 6人或以上, 至少一名游戏策划
- 至少两名游戏编程人员
- 有一个简单的策划稿（1-3页A4）
- （我们努力为你提供相关装备）



## 各种竞赛

- Most games will qualify for Imagine Cup
  - U.S. and World competition in game design
  - <http://www.imaginecup.com/> (World)
  - <http://www.imaginecup.us/> (U.S.)



## 成绩评定

- 课程作业（文档）(80%)
- 期末考试 – Unity 3D 实际操作测试（20%）
- 提供优秀博客（1-10%）
  
- 合计 100-110%

