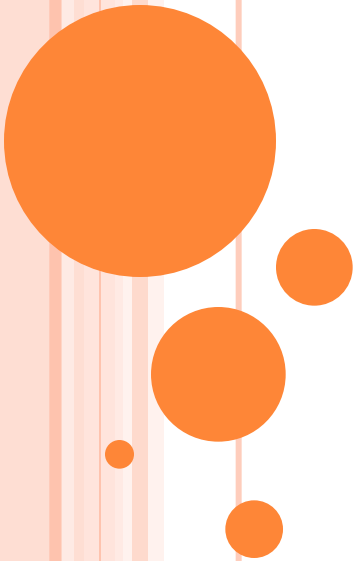
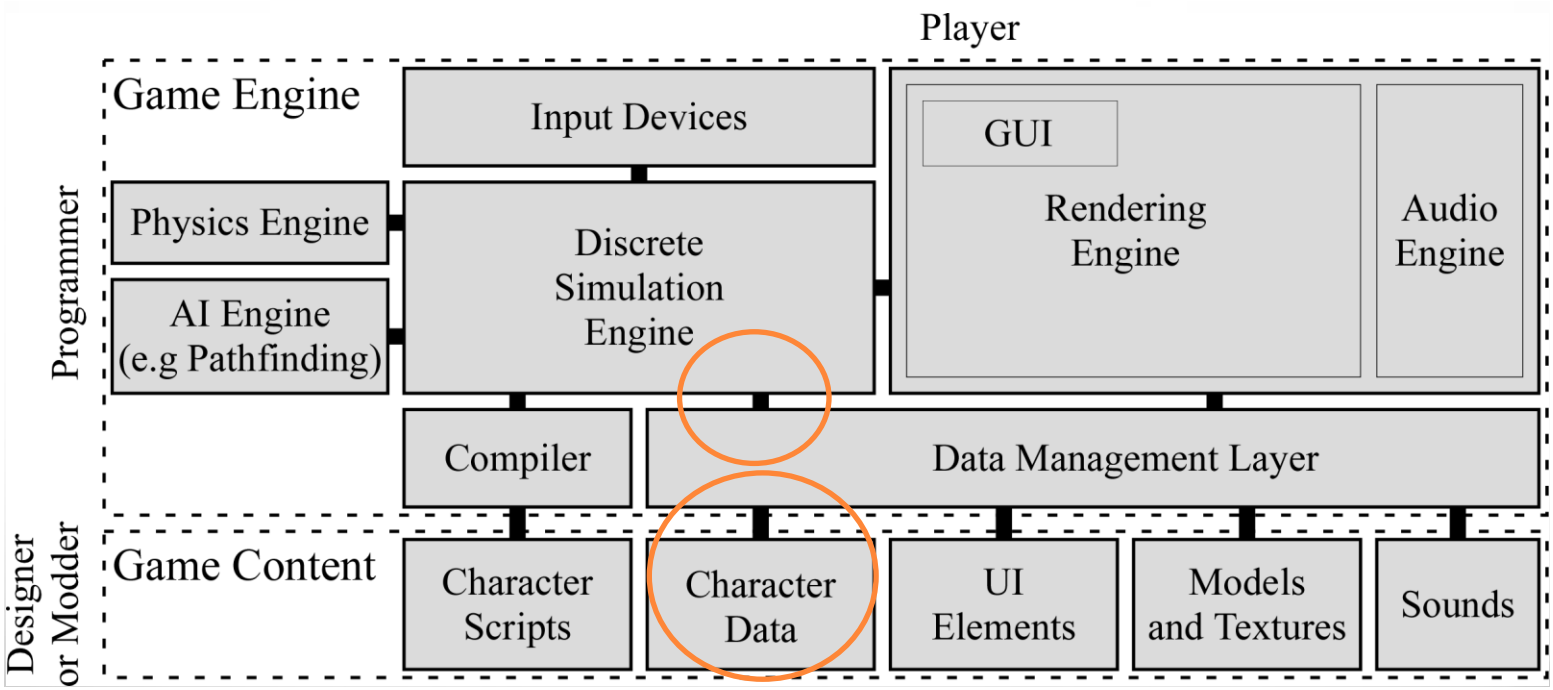


Introduction to Computer 3D Game Development

Data Driven Design（DDD）

潘茂林，panml@mail.sysu.edu.cn
中山大学·软件学院

游戏引擎架构


Player
Game Engine
Input Devices
GUI
Rendering Engine
Audio Engine
Physics Engine
Discrete Simulation Engine
AI Engine (e.g Pathfinding)
Programmer
Compiler
Data Management Layer
Designer or Modder
Game Content
Character Scripts
Character Data
UI Elements
Models and Textures
Sounds

目录

- Data Driven Design（DDD）基础
  - DDD 动机
  - 数据驱动的设计
  - 内容生产
- Unity 对 DDD 支持
  - Unity 游戏部署结构
  - 序列化技术
  - 数据读写
  - 案例研究

- 面向对象的编程思考
  - 无

Data Driven Design（DDD）基础
（1）DDD动机 – 发布游戏

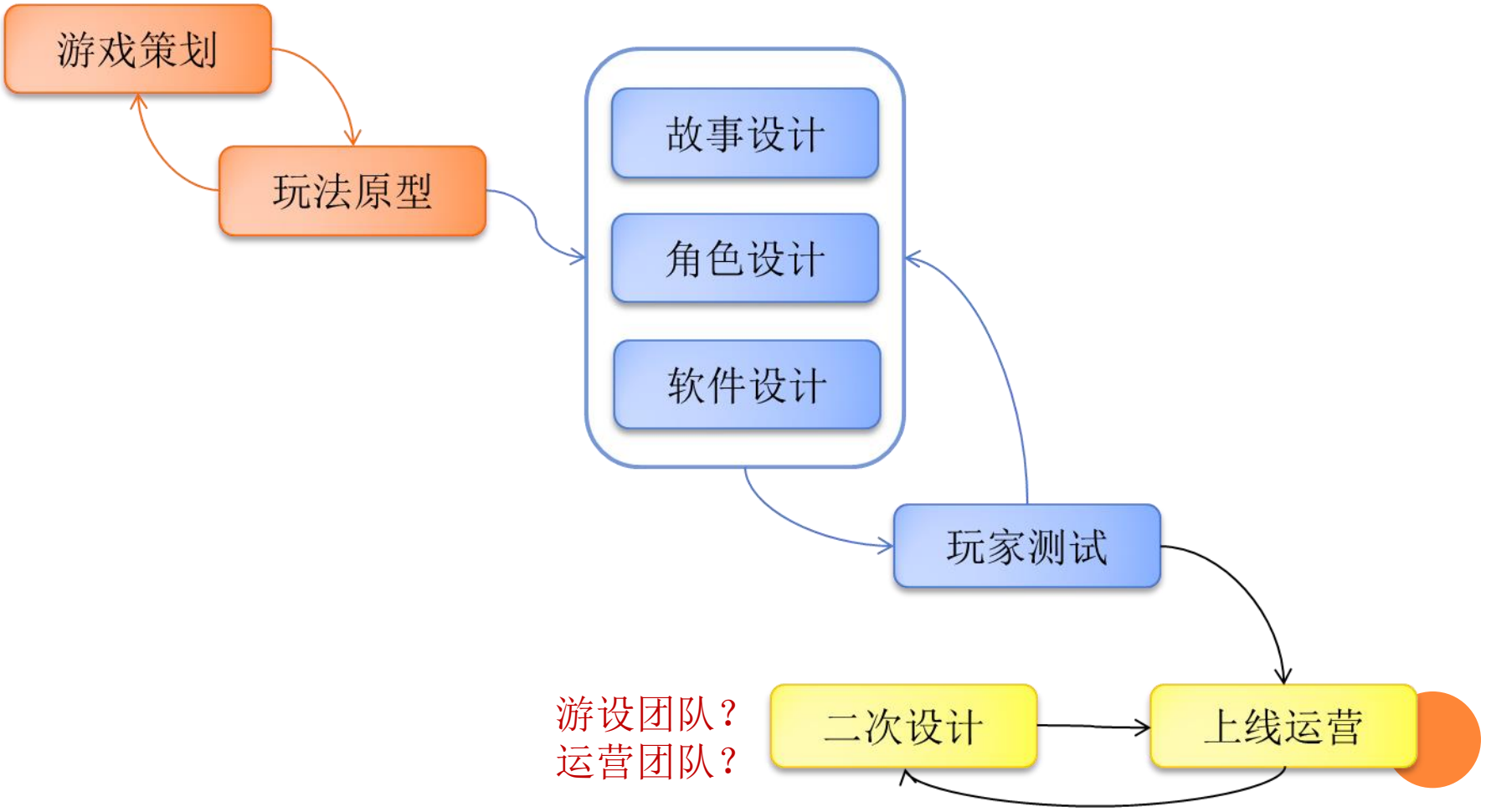
- 准备
  - 创建一个空项目，例如 app
  - Asset Store -> Unity Essentials / Sample / Survival Shooter 选导入
- 发布
  - 菜单 File -> Build Setting ... 或 Build & Run
  - 选择你需要的平台，选择平台参数，如果没有对应平台的发布包则显示按钮 Open Download Page。非 windows 平台还需要相关 SDK 开发包。
  - 平台选择有一个按钮 Player Settings ... 其实是“开发者设置”
  - 选择场景。如果多个要设置 0 号场景作为启动场景
  - 选择 Build，创建一个 out 子目录
  - 得到一个 exe 例如 shooter.exe，shooter_Data

Data Driven Design（DDD）基础
（1）DDD动机 – 发布游戏

- 观察结果
  - 生成了哪些目录、文件
  - 哪个文件最大？
- 玩一下游戏

https://docs.unity3d.com/Manual/PublishingBuilds.html

Data Driven Design（DDD）基础
（1）DDD动机 – 应对游戏的变化

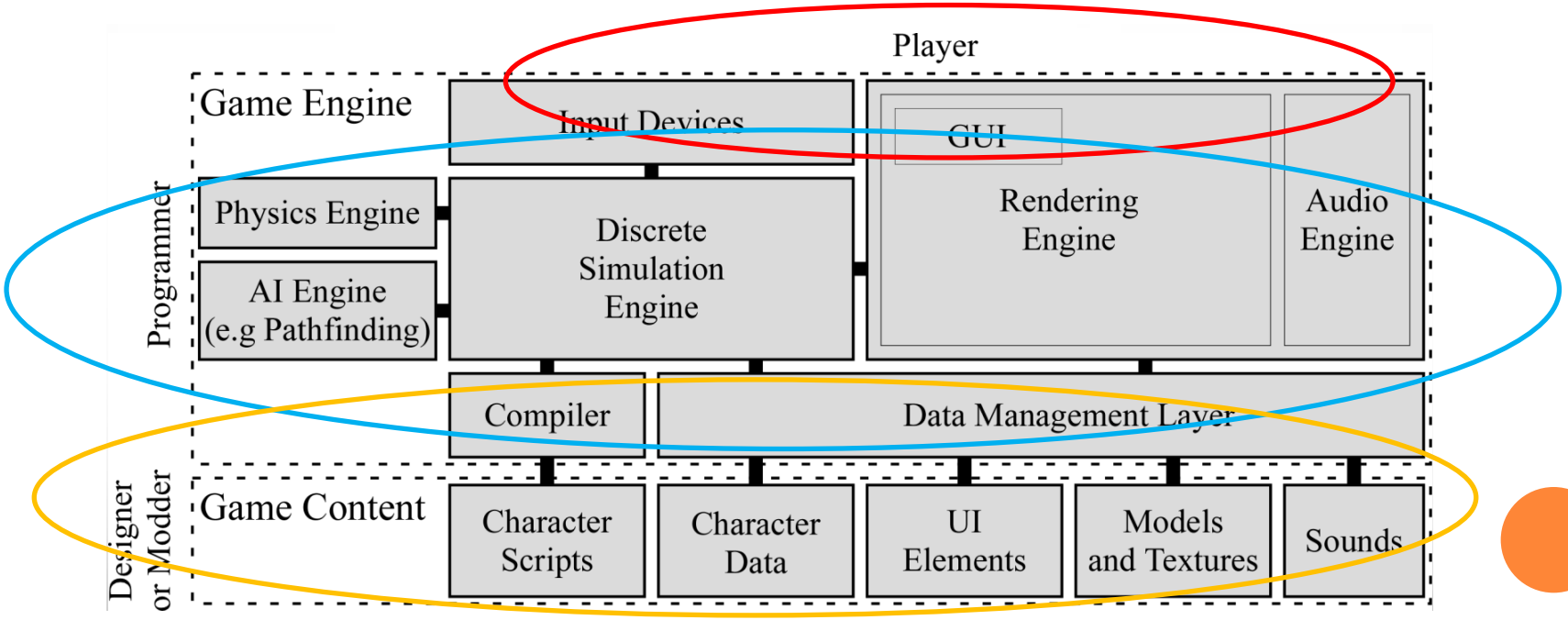


Data Driven Design（DDD）基础
（1） DDD动机 – 游戏基本构成

- 游戏引擎
  - 定义游戏的空间；
  - 游戏对象控制算法；
  - 基本行为与玩法
- 规则与机制
  - 游戏参数
  - 障碍与挑战类型
  - 各种能力
- 用户接口
  - 用户交互元素
  - 交互手段
- 内容与挑战
  - 模型
  - 素材
  - 动画
  - 语音、视频
  - ……

Data Driven Design（DDD）基础
（1）DDD动机 – 游戏进化的要素

- 程序员：创建游戏引擎；关注技术性开发
- 设计师：创建游戏内容；包括艺术内容，行为内容
- 建模者：修改游戏内容；
- 玩家：与游戏交互；



Data Driven Design（DDD）基础
（1）DDD动机 – 游戏进化的用户感知

- 游戏变化是常态
  - 代码中存在没有测试到的缺陷
  - 游戏内容不美观
  - 上市时间限制，部分代码与内容没时间制作
  - 逢年过级不定期退出优惠活动

- 基于用户感知的进化策略
  - 少让玩家感知升级（质量差的标准）
  - 定期升级（负责的表现）

Data Driven Design（DDD）基础
（2）数据驱动的设计

- 游戏引擎与数据分离
  - 游戏开发者除了游戏引擎开发必要代码
  - 规则、用户接口、内容都交给设计者、建模者、用户自己设计
  - 游戏引擎外无代码

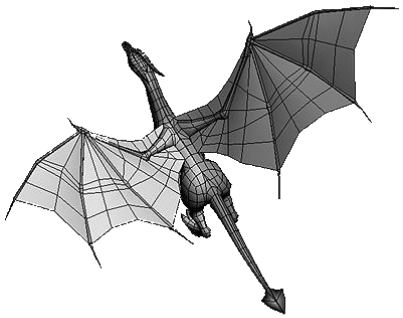
- 典型应用
  - 运营者可以锁定/开放部分关卡、功能模块
  - 建模者可以修改游戏形象，甚至推出新的人物
  - 设计者可以提供新的关卡、地图等内容
  - 部分资源可以延迟加载，已满足手机游戏 Size 的限制

Data Driven Design（DDD）基础
（2）数据驱动设计流程

- 少数程序员开始编程（原型）
- 程序员开始构建“内容流程 / content pipeline”
  - 艺术家和设计者的生产力工具
  - 数据导入游戏的查看、测试环境
- 提升艺术家和设计师的效率
  - 聚焦内容创作
- 在线自动更新
  - 在线更新代码（几乎无法静默安装，应用市场审批周期长）
  - 在线更新数据

Data Driven Design（DDD）基础
（2）内容生产--流程

Artist　　Data Format　　Software Code



COLLADA

Data Driven Design（DDD）基础
（2）内容生产--工具

- 关卡编辑器（level editor）
  - 创建内容与挑战
  - 定义游戏世界的物体与属性
  - 调整参数（物理，难度等等）

- 脚本语言
  - 定义角色行为
  - 描述规则与事件
  - 描述用户接口
  - 描述性语言（XML，JSON），解释性语言（Lua）

Data Driven Design（DDD）基础
（2）内容生产—关卡编辑案例

Data Driven Design（DDD）基础
（2）内容生产—关卡编辑

- 关卡编辑器
  - 通常与游戏共享代码
  - 关卡编辑可以嵌入游戏中
    - 直接与游戏共享代码
    - 带关卡编辑器的游戏范围窄
  - 独立关卡编辑
    - 用于开发核心玩法
    - 通过DLL或资源包与游戏共享
- 关卡配置
  - 使用 XML， JSON 配置游戏地图、物体、规则
  - 使用便捷工具自动生成配置

Data Driven Design（DDD）基础
（2）内容生产— XML\JSON


```
<?xml version="1.0" encoding="utf-8"?>
<levels>
  <level id="0" name="level0" unlock="1" />
  <level id="1" name="level1" unlock="0" />
  <level id="2" name="level2" unlock="0" />
  <level id="3" name="level3" unlock="0" />
  <level id="4" name="level4" unlock="0" />
  <level id="5" name="level5" unlock="0" />
  <level id="6" name="level6" unlock="0" />
  <level id="7" name="level7" unlock="0" />
  <level id="8" name="level8" unlock="0" />
  <level id="9" name="level9" unlock="0" />
</levels>
```

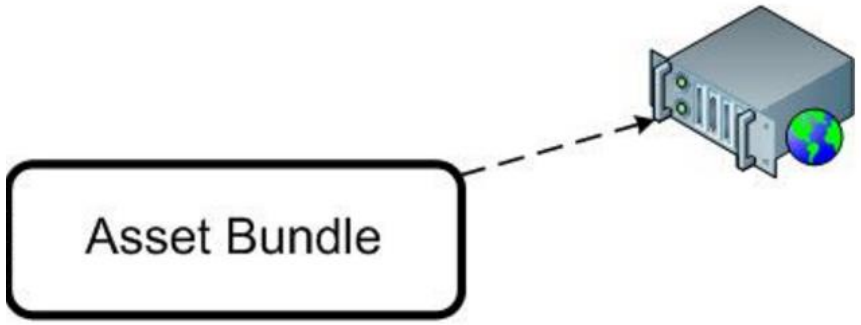
Unity3D游戏开发之快速打造流行的关卡系统 http://blog.csdn.net/qinyuanpei/article/details/46411485

Unity 对 DDD 支持
（1）部署结构

- Unity 程序部署结构
  - 程序划分
    - 代码（cs, dll）；资源（状态机、动画、模型、材料… …）
  - 资源部署
  - Resources
    - 部署在 Resources 目录下，与代码一起打包
    - 可以用 Resource.load 加载
  - AssetBundle
    - 独立压缩的资源包，可以联网下载
    - 用 WWW 类从中心下载， 用 AssetBundle 加载
  - StreamingAssets
    - 部署在StreamingAssets 根目录下的内容
  - PersistentData
    - App 的私有数据目录（数据可清除）

Unity 对 DDD 支持
（1）部署结构

- Asset Bundles



- 资源包业务流程
  - Build AssetBundles
  - Upload AssetBundles to Internet
  - Download AssetBundles at run time
  - Load GameObjects from AssetBundles

A guide to AssetBundles and Resources
https://unity3d.com/cn/learn/tutorials/topics/best-practices/guide-assetbundles-and-resources

Unity 对 DDD 支持
（1）部署结构

- 流式资源
  - 用于存放视频、音乐等流式文件
  - 只读！！！
- 实际存放位置
  - 依赖于操作系统
  - https://docs.unity3d.com/Manual/StreamingAssets.html
- 程序访问点：
  - Application.streamingAssetsPath
  - 访问类 WWW

Unity 对 DDD 支持
（1）部署结构

- 用户数据
  - 可以用来存放程序运行数据
  - 可读可写
  - 内容限制 - 无
  - 清除手机缓存文件会一并清理这里的东西
- 程序访问点
  - Application.persistentDataPath
  - 不同平台，具体目录不一样
  - 读类 WWW
  - 写类 StreamWriter

Unity 对 DDD 支持
（1）序列化技术

- 序列化（Serialization）
  - 一个内存对象变为与地址无关的可传输、可存储的数据格式，通常是文本格式
  - 例如：保存一个场景，则需要把场景中对象序列化
- 反序列化
  - 将序列化恢复为一个内存对象。
  - 例如：加载一个场景

- 数据格式：
  - 私有、压缩格式
  - XML，JSON，YAML 描述格式（易于阅读）

Unity 对 DDD 支持（1）序列化技术

- JSON 序列化

- 见课件代码

Unity 对 DDD 支持
（2）数据读写

- PlayerPrefs 类
  - https://docs.unity3d.com/ScriptReference/PlayerPrefs.html
  - 一个 key – value 的持久化**数据库**
  - 存储位置随平台而不同，图 window 放在注册表中
  - 系统会按策略自动保存，一般不用你 save
  - Value 支持 整数、浮点数、字串
- 用途
  - 存放用户、装备、游戏进度等信息
- Key 的管理
  - 为了避免 Key 的字符串散落程序各处，导致无法维护
  - 使用一个 Singleton 类统一读写该类。在SSDirector中
  - 使用常数列表

Unity 对 DDD 支持
（2）数据读写

- WWW 类
  - https://docs.unity3d.com/ScriptReference/WWW.html
  - 支持多协议的流式文件只读类。支持：支持 http://, https:// , file:// 和 jar:file:// 等协议。
  - **协程友好**（后台读，完成会通知）
- 常用方法
  - assetBundle 下载资源并实例化
  - text 下载文本
  - texture 下载图片
  - isDone 检查是否完成
  - bytesDownloaded, progress 检查进度

Unity 对 DDD 支持
（2）数据读写

- 重要的静态方法
  - EscapeURL：Escapes characters in a string to ensure they are URL-friendly.
  - LoadFromCacheOrDownload：Loads an AssetBundle with the specified version number from the cache. If the AssetBundle is not currently cached, it will automatically be downloaded and stored in the cache for future retrieval from local storage.
  - UnEscapeURL：Converts URL-friendly escape sequences back to normal text.

Unity 对 DDD 支持
（2）数据读写

- 使用StreamWriter类写文本
  - http://www.manew.com/3219.html
  - App 唯一可写文件的位置，请使用 Application.persistentDataPath 位置

Unity 对 DDD 支持

实验一：资源转储

- 将互联网存为本地资源，在必要时自动升级
- 代码见课件

Unity 对 DDD 支持
（3）应用案例研究

游戏设计要求：
1. 代码仅有一关
2. 资源可定期更新
3. 在线控制关卡配置

- “找你妹”的数据驱动设计
- 游戏建模
  - 游戏物体：texture，theme，size，tag1，tag2 … …
  - 问题：找 tag 的 数量
- 关卡模型
  - 物体数量 col x row ： 越多越难
  - 问题 #tag / 内容 #tag 的占比 ： 越接近 1 越难
  - 素材平均 tag 数 ： 越高越难
  - 主题类型 theme ： 无难度，避免玩家烦闷（bored）
- 建立可序列化 level-def 类
  - 建立网络素材库，素材描述文件
  - 建立关卡配置文件，在线控制关卡配置

课程小结

- DDD 是游戏设计核心技术
  - DDD的动机与应用
  - Unity 部署结构
  - 常用的类
  - 关卡参数化的方法
  - 内容描述的方法

作业（Lab 12）

- 作业

  （选做，博客形式提交）制作一个 2D 游戏，如“找你妹”，实现数据驱动的设计。