

游戏开发 技术掠影

目录（白话）

- ❖ 先闲聊
- ❖ 再布置作业
- ❖ 然后讲一些作业里要用到的东西
- ❖ 再然后讲一些作业里用不到，但比较有意思的东西
- ❖ 接下来再讲一些作业里要用到的东西
- ❖ 再接下来重申作业
- ❖ 最后闲聊到下课

目录（学术）

- ❖ 基于区块的游戏（必修）
- ❖ 3D图形引擎（选修）
- ❖ 游戏中的物理（选修）
- ❖ 嵌入式脚本（必修）



show一个游戏



Effect引擎

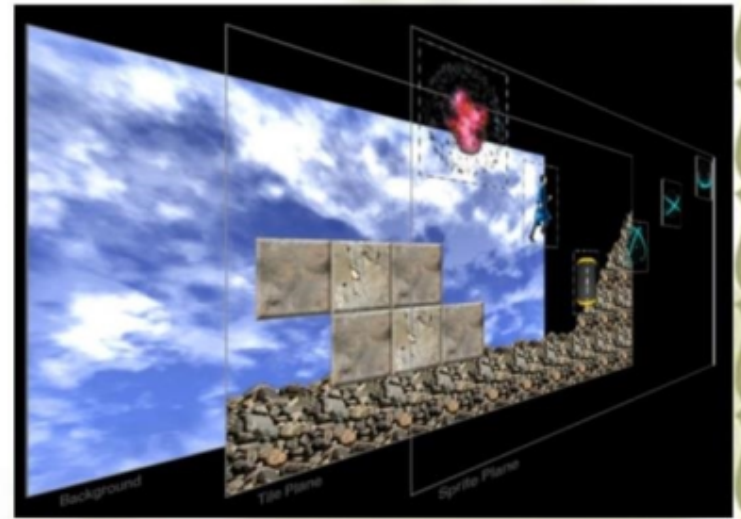
[http:///effect/](http://effect/)

- ❖ **精灵**：主角、敌人、其他活动物体。带透明、可移动、需要碰撞检测
- ❖ **区块**：用来“拼图”的基本单元，一般是矩形或者六边形



层

- ❖ 举例：背景、区块、精灵、HUD
- ❖ z序
- ❖ 时差卷 轴
- ❖ 硬件支持
- ❖ 补充知识：透明
 - ☞ GIF：调色板中的透明色
 - ☞ PNG：alpha通道







Merchandise
Jordain - Produce
Merchandise
Nimras - Produce
Merchandise
Frolan - Elephant Rider
Spirit of Order message

652+609 28087+13 80/80

ACTIVE



None

Spells

WIZARD



Center

POWER



None

Research

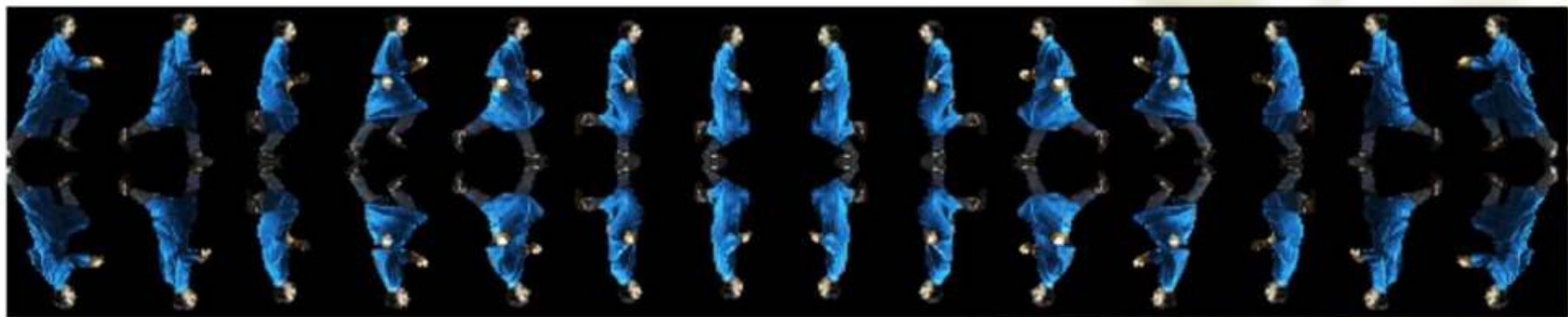


Surface Undergr. Shadow



精灵

- ❖ 可以有多个animation（比如走、跑、跳、射击、死亡）
- ❖ 每个animation包含多个帧（frame）
- ❖ 只绘制基本图像，借助图像变换满足更复杂的需求



精灵的控制逻辑

- ❖ 时钟信号、定时器事件
 - ⌚ 固定间隔：精灵的“射击”动作正处于哪一帧？
 - ⌚ 连续时间：子弹移动了多少距离？
 - ⌚ FPS = Frame Per Second
- ❖ 控制逻辑是定时器事件触发的，因此执行时间一定要短。耗时操作可以开辟线程或者分而治之
- ❖ 思考：屏外精灵如何处理？

区块

- ❖ 游戏中的每个关卡维护一个区块集合（**Tileset**），其中所有区块的大小必须完全一致。



- ❖ 区块也可以有控制逻辑（地面、墙、特殊物品），但更常见的处理方式是创建一个不可见的区块层，用碰撞检测的方式把可见区块和控制逻辑联系起来。



练习：指出精灵、区块和层



碰撞检测

- ❖ 基本几何元素：点、线、矩形
 - ☞ 复杂图像：用多个矩形近似
 - ☞ 更复杂的图像：Pixel-perfect算法
- ❖ 对象：主角+敌人、主角+前景区块...
- ❖ 应根据对象的当前位置和预定偏移量来检测，而不能只根据新位置检测。想一想，为什么（提示：逻辑是定时器驱动的）
- ❖ 如果物体是solid的，处理碰撞之前要做位置修正，确保它永远不与其他物体重叠

有限状态自动机 (FSM)

请把状态转移补充完整

如何让移动更有真实感？比如加速度、惯性、摩擦力

如何把下落与跳跃统一起来？

为什么不把“被攻击”设为一种状态？

怎样区分开站立攻击和空中攻击？





(现在开始跑题…)

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/806232225154010150>