1. 数学和物理

坐标系 xyz

函数和方程 y因变量 x自变量 单变量函数 图像

指数和幂 f(x)=a^x 开方根 1/3

化简，两边同增 上下同乘 乘上根号

三角学 射线 线段 矢量 角 三角形 勾股定理 arctan

矢量 加减 乘 长短 单位矢量 分解成xy方向 点乘与夹角uy=x1x2+y1y2 =长度积cos

物理 时间 距离 质量 位置 速度 加速度

二、状态驱动 fsm 方法多 速度块 分解状态 行为影响状态

追踪状态 躲避状态 找到目标 逃跑 找掩体 fifa rts移动 巡逻 沿路径前进状态

1、方法 if else switch

状态变换表 当前状态 条件 转换为状态

逃跑状态-如果安全-巡逻状态 攻击状态-血量低-逃跑 巡逻-发现敌人-追击

每一步都检查当前状态

2、状态机自行转换状态 把当前状态抽象 ，直接执行行为接口 所有状态都自带通用行为，通过行为转换当前状态（条件不满足则不转换） currentstate.move(){ } 不同的状态对象带同样的接口，然后重写各个动作，动作可以转换currentstate指向的状态 外部则只用currentstate的接口

当前状态：行走 接口： 行走()-行走 跳跃()-跳起 卧倒()-卧倒 攻击()-禁止攻击

状态进入动作 退出动作 进入和退出是同时发生的，从一个转到另一个 状态对象应该是单例吗？（只一次，全局可访问，静态）

全局状态和状态翻转 保存当前状态，可以随时恢复到之前的状态 ，什么时候有这种需求？操作取消？ 延伸成状态队列如何？

可以继续封装一个状态管理层，使用函数来调用和控制状态

状态对象的细化，设计方法：进入状态，离开状态 新状态.进入（） 不是好设计，还是用“当前状态”比较好

可以把行为封装一下 current\_state.execute(action) current\_state.undo(action) action可以是闭包函数（回调） 这是状态模式和

2.8 状态的消息驱动 发送事件 处理事件 一个消息是一个数据结构（数组或者结构或者对象）：发送者，接收者，消息信息（属性）

利用语言已有的消息机制，实现状态机（拥有状态的独立对象）之间通信。。。常见的点击，键盘事件，以及其他任何事件和自定义事件（通知）， 按下按钮->触发回调处理函数->函数中调用 行为->行为改变**当前状态**

**2.9**

Ai 根据状态判断下一步应该做的行为 行为改变状态，但是自动行动的ai也应该根据状态判断自己该如何行为

状态-条件-转换

继续抽象封装？把条件也封装进去？ Current\_state.start(){if(条件)move() }

我之前的学习是正确的吗？ 状态 行为 条件是这样组织的吗？

条件的逻辑判断应该写在哪？ 状态对象是单纯的行为封装，还是应该把状态转换的条件判断也封装进去？

二、自治智能体 自动判断 动态障碍， 事前不知道的处理

行动选择：选定目标，制定计划 到哪去，怎么做

操控 执行

移动

交通工具对象 操控 行为 靠近 抵达 追逐 逃避 漫游 避开 逃避

插入（两个之间）（中点或者其他位置） 掩护，拦球

隐藏，躲在障碍物后

路径跟随

保持距离跟随

组行为（分离）

聚集

群集（凸现行为）

6.操控行为组合

操控力

操控行为的合力

不同行为的开启关闭

截短

抖动

7、不能重叠 碰撞检测？ 检测快撞到了就 避开？

8、大量对象的碰撞检测 空间分区 把对象加入到空间数组

9、不同行为同时进行之间的冲突 接近和远离行为 有个判断边界，会在边界发生抖动，应该使该变化均匀 平滑

1. 体育 足球

所有都是对象 球场，球门\*2 球 球队-4\*2个球员+2个守门

可以用树来组织对象的关系 类似显示对象树

父子层级（不是继承意义上的父子）addChild（） 获取子对象（根据名字或者id）

也也可以用对象属性 =对象 球场.球队.队员

重新开赛， 记录球员初始位置，恢复初始值

球队 所有球员 获得带球者 设置接球者 获得最接近球的球员getclosest 接应队员 接应点计算

球员.队伍 .是否带球 .walk() .run() .shoot() .skill() .get()拦球 setrreciver 传球pass(player) 射向接应点 是否可射门

球 ball.fly(speed) ball.scored() .reback() 回归原位 .speeddown() getx setx 未来位置 .isFly 没被持有

球门 边界坐标 球柱坐标（反弹）

状态机 球队状态：防守 进攻 准备开球 给球员发送消息（执行函数）可用监听器，也可直接a.func()

球员状态： 等待 接球 踢球 带球 追球 回位 接应 使用消息系统，接到球

状态进入

子对象位置

2、如何让npc变傻？ 一、第一次是完美的，之后慢慢产生偏差 2、随机偏差 3、和正确结果本身就有偏差的算法 比如在方形范围拦截圆形

五、图 节点 边 树

导航图 所有可以访问的位置 顺序移动或者来回移动 行走轨迹可以沿直线也可以稍偏差，或者随机偏差， 但是目标点固定

依赖图 资源管理 资源的先决条件判断 从当前拥有的兵种推断所有的建筑。技术

根据依赖图设计建造顺序，最短路径

状态图 状态图 状态转换的路径 连续状态，可以有回路

**状态空间**

图的实现

搜索算法 盲目搜索 深度优先

Dfs优化 魔方还原 限制深度的搜索，1，2，3，4，5，6

广度优先

基于开销（权） 的搜索 最短路径 多条短边<一条长边

边放松 当前最优路径

最短路径树

迪 算法

改进-A星算法 欧几里得启发因子 曼哈顿距离启发银子

六、脚本语言 舞台逻辑

Ai逻辑

复活位置

地图编辑器

七、 枪战游戏

弹药 种类

1、触发器 地图暗雷 地图触发提示， 比如尸体 拉杆触发门 触发器区域 声音通知

**碰撞检测让谁进行： 雷碰撞人还是人碰撞雷？**

2、运动ai 武器控制ai

Ai 实现，制定策略 目标-子目标 评估目标和状态的适宜度，调整当前目标

移动

路径规划

感知：视觉 听觉 枪声提示目标位置给ai 脚步声 隐身 实时游戏不一定需要

黑暗中看不到 阻挡视线

发现尸体 听到战斗声

短期记忆，记住躲藏掩体的位置 记忆图，记忆表 绕过掩体或者攻击掩体，保持攻击状态 x秒内没感觉到对方的身影，就删除记忆

目标选择 ：被杀次数 朝向 攻击范围

八、路径规划

导航图

基于格子

可视点（POV） 各点

导航网 多边形 连接 多边形内任何点都可以走

只能沿着边走：用粗粒的导航图

Rts/rpg 战旗 npc按照格子移动 对着目标点 可能产生别扭的路径，建议使用路径平滑算法

细粒度的图：非常多节点 如何创建

洪水填充算法 ：种子节点

空间分割： 就近查询图最短路径，不用遍历所有路径

单元空间分割（格子分割） 四叉树 bsp

节点加入到格子？ 只算格子内的？

路径规划，无障碍（连通节点） 离角色最近 离目标最近 最低成本路径

多个同样目标， 其中选一个最短的 ，迪算法更好

节点式路径，边式路径 到达点开始行为，这样就需要执行-取消 还是在边范围进行行为

路径平滑算法 避免过多弯曲

点的连接一般是 左右上下 斜方向 可以只连相邻 也可以远连接 边太多也不好，只要是连通就行

1. 预先计算查询表，在还没开始的时候就算一个路径 每个节点到其他节点的最短路径

spt最短路径树 需要用的时候直接取出来

2、预先算成本 成本表 最短成本

时间片路径搜索，切换cpu资源（线程方式或者其他方式比如协程）

分层路径寻找

NPC困境 被墙卡住，被走廊卡住 撞墙无法转向 被其他角色挤开

应该设置时间检测 ：如果路径一段时间没走通，就重新规划路径

十、目标驱动

行为不是状态，而是目标的集合 原子目标 组合目标

从抽象的目标 递归分解成一层层子目标

每次思考，检查游戏状态，选择一个目标，试图实现，成功或者失败，或者改变策略

子目标需要依次满足 抽象目标，变为具体的行为

目标队列？ 子目标 目标对象

目标状态：闲置 激活 完成 失败

目标评分

命令队列

十一、模糊逻辑 解释语言规则 模糊化 在0-5m之间

普通集合 集合运算符

模糊集合 隶属函数 可以是三角或者梯形 y轴是隶属度 不同的范围之间有交错

比如平均 笨 聪明 有三个范围0-80 70-100 90-250 三个三角，然后交错看隶属度

没有很明确的分界

模糊语言变量 iq={笨，平均，聪明}