后端

1. 写一个Warrior类: 属性值: 队伍 (wTeam), 种类 (wType), 生命值 (wLife), 攻击力(wAttack), 速度(wSpeed), 攻击速度(wASpeed), mCD(距离下一次攻击的回合数), aCD(距离下一次攻击的回合数)
2. 派生四个类: Knight, Archer, DefenceTower (没错, 防御塔也可以视为一个不会移动的对象), Base 属性值待定
3. 定轴: 朴素想法是y = x, y=0粘合x=a, y=a粘合x=0, a随地图大小d等比例变化, d可以先定一个值, 之后再调, 而后划分轴作为位置的刻度
4. 确定防御塔, 主塔, 兵营(仅用于操作, 不那么重要)的位置, 随d等比例变化
5. 时间轴:

   我们采用一个priority_queue作为**指令队列**, 所有的操作都转换成元组类型的命令入队, 依次完成

   回合数采用60turns/sec与gamezero默认帧数一致以增加连续性
6. 命令的形式化 (所有的外部输入(e.g. 鼠标操作)都转化成该形式:

   格式

   turnID side moveType additionalInfo

   e.g. 21 0 2 3

   具体标号在实现时确定并写在代码注释中
7. 对象的存储

   可以每方用3个list来存各条路线上本方士兵的情况, 我们可以认为士兵的个数不太多, 故$O(n^2)$的遍历是可以接受的
8. 行动 每轮都按照所写顺序依次执行

   造兵: 在点击后创造一个命令, 该命令也在回合最初执行, 其在x回合后 (x需要确定, 是一个不太长的时间) 在选定的list内创建一个指定的士兵对象, 将其append到对应路径的list中

   行动: 对双方的list进行遍历, 查找每个对象是否会和其余对象发生交互

   攻击: 检测aCD, 若aCD=0则重置aCD, 再遍历敌方列表判断是否存在攻击范围内的敌方对象, 攻击通过遍历敌方list找到满足如下要求的敌方对象: 距离最近 -> 血量最低 -> 随机选 (血量一样的话). 双方的攻击同时进行, 溢出伤害暂且认为会自动失效 (略不合理)

   若本回合未攻击: 先检测mCD是否为0, 若为0则重置mCD再遍历列表看前面的格子是否被己方人员占满(即有三个己方单位, 因为未攻击, 不可能有敌方单位阻拦), 若没满则可以选择前进

   死亡: 生命值$\leqslant$ 0即死亡, 移出所在的list并给敌方金钱奖励

   若对方主塔死亡则判定游戏结束
9. 金钱机制: 游戏开始时置0, 上限10coins, 生成速度1coin/s
10. 联网: 敌方的指令用socket传输json数据置入指令队列, 在本地计算得到结果

前端:

1. 画地图, 对象
2. 金钱机制的显示
3. 画选项, 将点击选项转换成指令置入指令队列
4. 对象行动的绘制(遍历list来显示各对象目前情况)

以下为摘录:

地图暂定为正方形，玩家操作面板可以放在两侧

格 — 确定边长后可作为长度单位，直接将格作为全图的位置信息，例如格（5，5）是否有单位等 (仅用于绘制)

每一条路设置3格宽，长度为10来格左右 (10来格不能体现连续性, 个人认为至少要100格)

角色血量<0时立即死亡，图标消失，更新格子数据