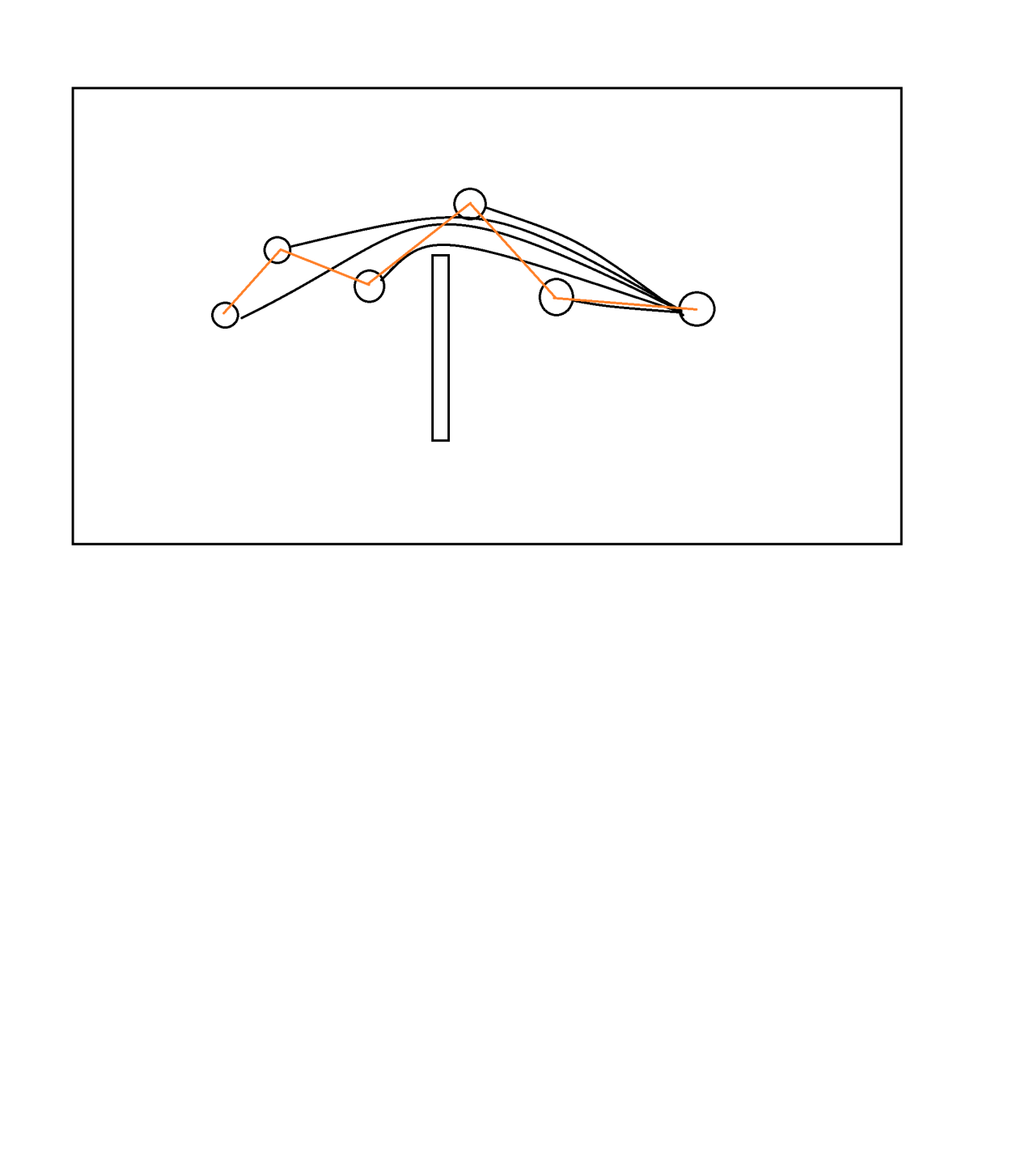
**怪物寻路重复路径修改方案**

**1.目的**

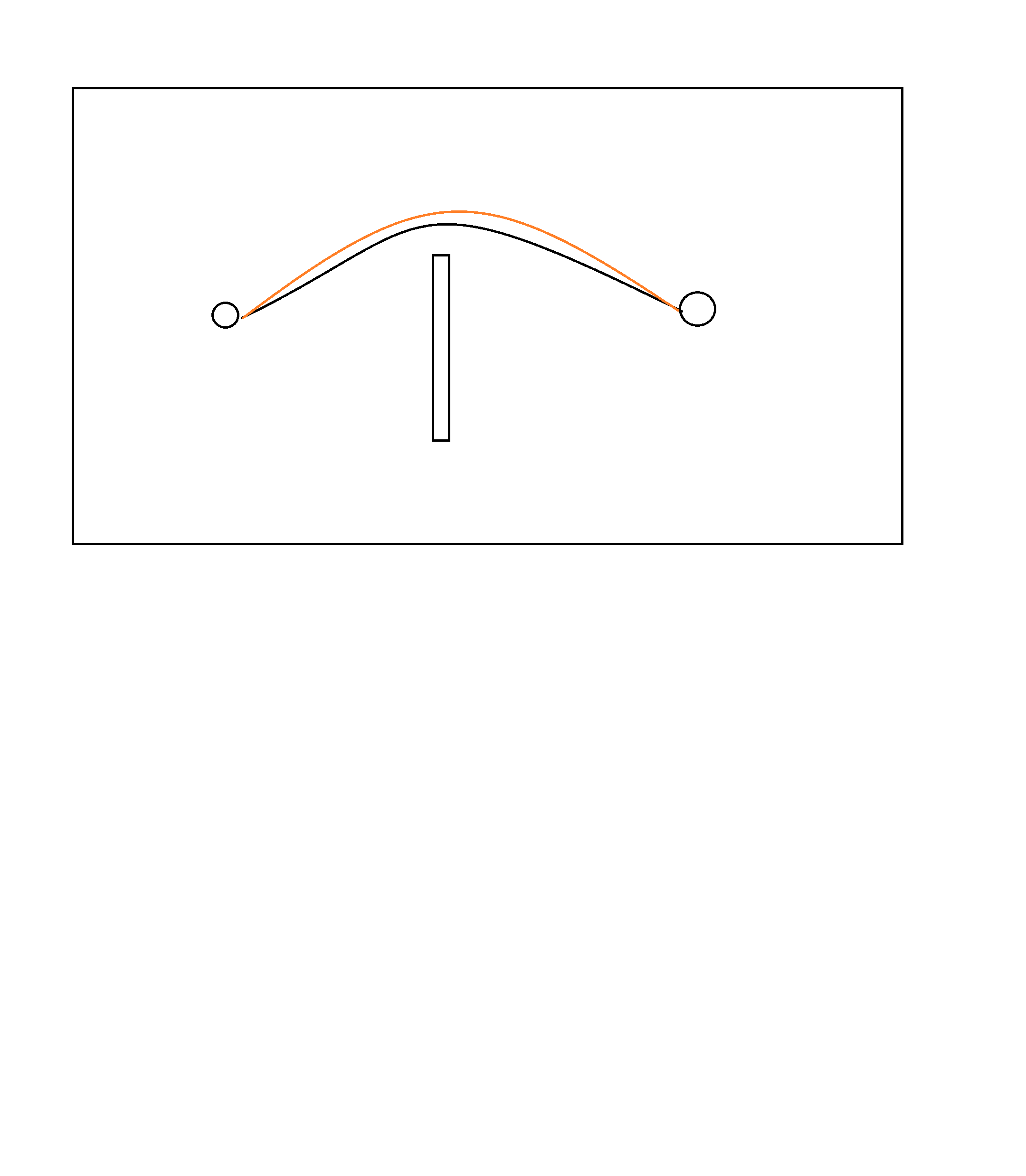
主要是多个怪物存在时，寻路都是最佳路径，导致路线重复，出现怪物排队出发。

1. **解决方案一：**
2. 步骤如下：
3. 计算出最佳路径，目标点的选取在玩家位置为圆心半径为1（可修改）的圆表面上。
4. 选择最佳路径的一个点，然后进行位置偏移
5. 移动到位置偏移后的点，然后重复a.b.c三个步骤，一直到玩家附近。
6. 如图所示



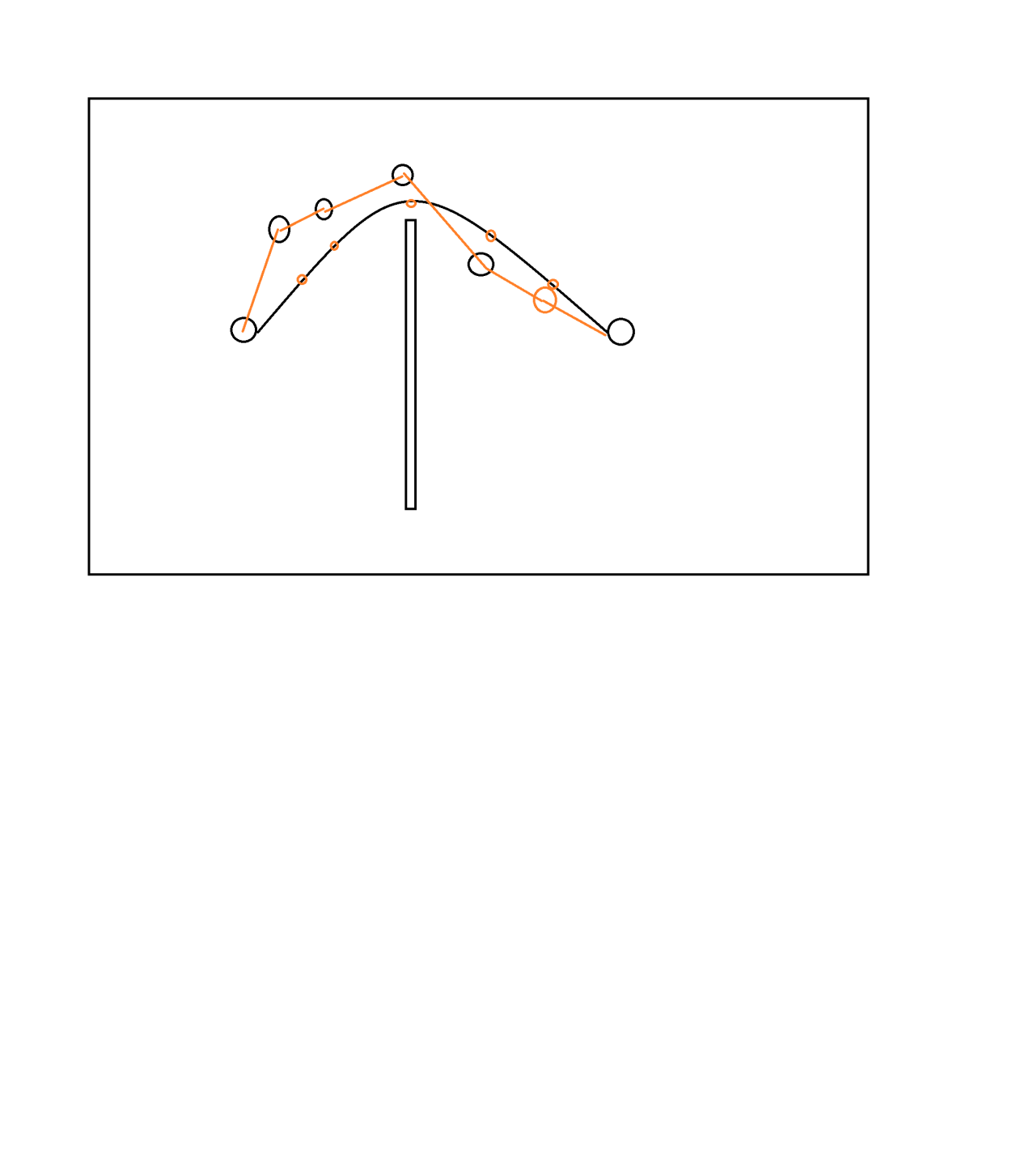
其中圆就是怪物，每一条黑线就是当前怪物当前位置到目的地的最佳路径。橙色的线就是本次解决方案的实际运行路径。

1. 最后的效果待测
2. 缺点
3. 怪物的路线看起来是乱走的。
4. 计算量大
5. **解决方案二**
6. 方案二的步骤：
7. 找出最佳路径，目标点的选取在玩家位置为圆心半径为1（可修改）的圆表面上。
8. 给最佳路径整体加一个偏移值，使路径发生改变
9. 按照新的路径前进
10. 如图所示



其中的圆就是怪物，黑线是计算出的最佳路径，橙色的线就是加偏移值得到的最新的路径。

1. 最后的效果待测
2. **解决方案三**
3. 解决方案三的步骤：
4. 找出最佳路径，目标点的选取在玩家位置为圆心半径为1（可修改）的圆表面上。
5. 选取最佳路径的几个点，然后对这些点进行随机的偏移，最后把这几个点连接起来得到新的路径。
6. 如图所示

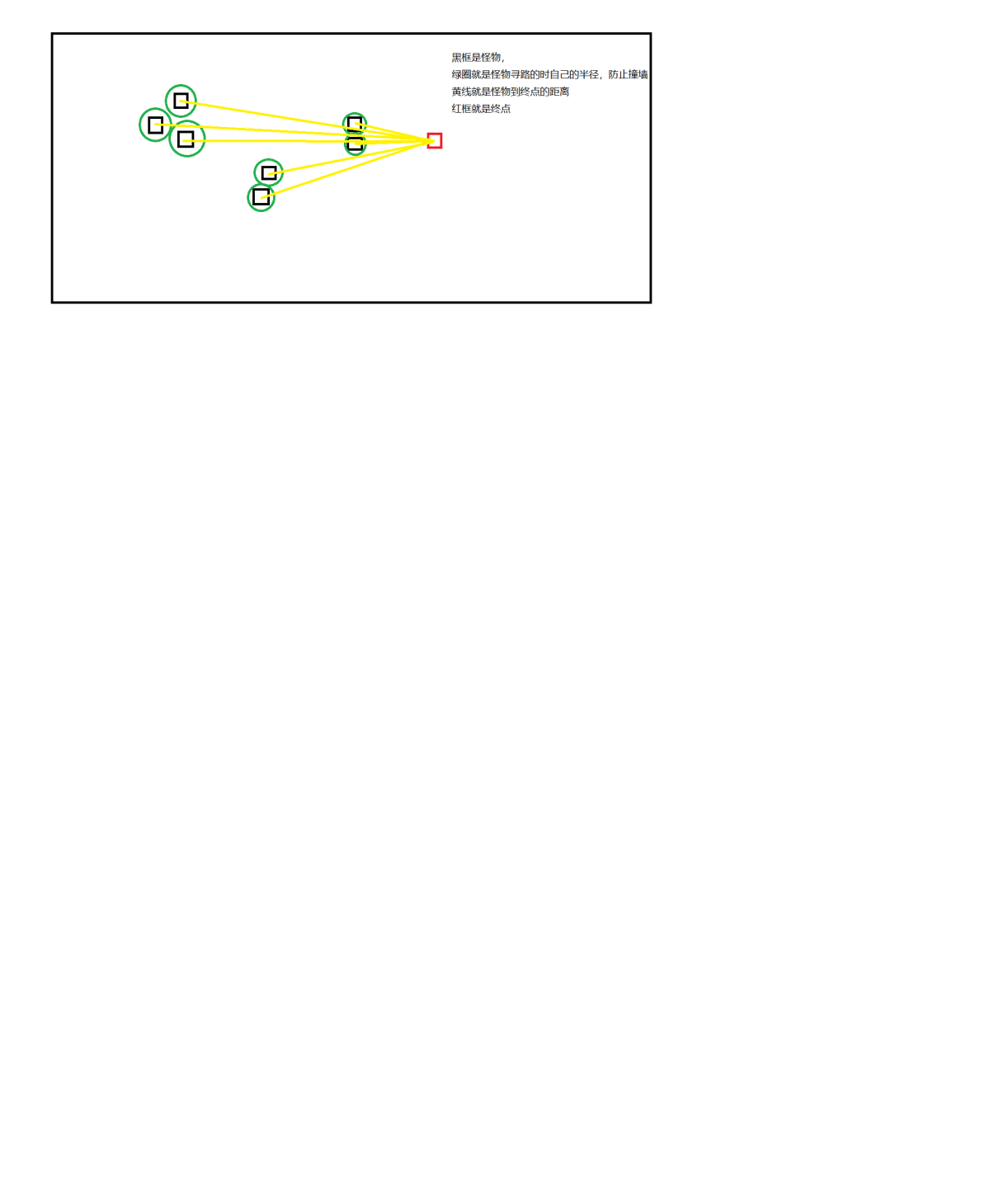


其中的圆就是怪物，黑线是计算出的最佳路径，橙色的线就是加偏移值得到的最新的路径。

1. 最后的效果待测
2. **解决方案四**

先不改怪物的移动路径，通过修改生成点生成的怪物位置和修改怪物的移动速度，以及生成怪物的时间间隔来解决此方案。

1. 解决方案的步骤：
2. 利用unity自带的navmeshagent组件来实现寻路；
3. 通过动态的修改navmeshagent的radius来实现怪物之间的间隔。设置一个最大值和最小值，当怪物离玩家的距离达到一定的范围时，开始减少radius直到最小值，超出范围之后，开始增加radius直到最大值。
4. 除此之外还要修改不同怪物的移动速度。
5. 还有怪物产生的时间间隔。
6. 图示



主要是动态的修改navmeshAgent的radius

1. 效果还行，性能消耗也不高sss