

Unity 自走棋游戏

设计与开发总结

受刀塔自走棋影响，我也萌生了设计一款自走棋类的游戏想法，想从零开始在 Unity 里设计一款 3D 自走棋类游戏。因为只有我一个人制作，所以我先从单人模式出发。这个文档主要就是记录我开发过程中的思路和总结。

游戏可以从这个链接下载：<https://github.com/zbmsnj1/CWV2>

Contents

1. 自走棋类游戏分析	2
1.1 游戏起源	2
1.2 游戏核心玩法	2
2. Unity 自走棋设计思路	3
2.1 游戏场景	3
2.2 游戏 UI	4
2.2.1 游戏信息显示界面	4
2.2.2 刷新棋子界面	6
2.2.3 棋子信息显示区域	7
2.2.4 羁绊显示区域	7
2.3 游戏 AI	8
2.3.1 棋子动作控制	8
2.3.2 棋子自动寻路	11
2.4 棋子的技能与羁绊	12
2.4.1 羁绊	12
2.4.2 棋子技能	12
2.5 怪物与关卡	15
2.6 数值与数据	15
2.6.1 数据的导入	15
2.6.2 DOTA 自走棋 经济系统/人口规则 拆解	16
2.6.3 Unity 自走棋 经济系统/人口规则	17
2.6.4 战斗数值的平衡	17
2.7 游戏循环逻辑	17
2.8 新手引导	17

1. 自走棋类游戏分析

1.1 游戏起源

自走棋最早火爆于 DOTA2 创意工坊，是基于 DOTA2 的引擎设计出的一款多人战略类棋牌游戏。由于其玩法简单-----只需要选择并放置棋子到棋盘上，策略多变-----有 55 类棋子，23 个种族，玩家可以自由组合受到了很多人的喜欢。当初最火的时候 DOTA2 服务器经常卡死，人们笑称 DOTA2 为自走棋启动器。随后各个厂商开始研制自己的自走棋类游戏，出现了《云顶之弈》、《多多自走棋》、《逆水寒》“豪侠战棋”、《王者模拟战》、《赤潮自走棋》等等。**我创作 Unity 的自走棋，主要参考的是刀塔自走棋。**

1.2 游戏核心玩法

- 首先在于自走这两字。有些游戏需要特别高端的操作，这会是用户群体变窄，而自走属性直接不给你战斗中操作的机会，解放双手，唯一的操作就是购买，放置棋子，所以才会深受男女老少各类人群喜爱，用户群体很广。这背后的技术需求就是 AI，不需要多么复杂的 AI，核心就棋子寻找离自己最近的敌人，进入攻击范围，开始打斗或者放技能，敌人死亡目标锁定为下一个离自己最近的敌人，如此循环。背后的设计思路就是，要逻辑严谨的设计棋子的每一个状态，比如，走路，战斗，胜利，死亡等等，如果给每个状态都设计相应的动作，那么游戏战斗画面会看起来更加有感觉，但是当棋子多起来，任务量很大。
- 其次，游戏的核心在于棋子，更准确的说是棋子的组合，阵容选择的多变。设计的核心在于让每个棋子都有独特的技能，而且在和其他棋子一同出现场上时有特殊效果（比如攻击增加，护盾，吸血等等）。所以设计每个棋子的技能，还有棋子间的羁绊效果是游戏好玩的核心。
- 最后，棋子的刷新购买机制，以及游戏的经济系统。自走棋有意思的点在于，你赢了你就会累积金钱，而金钱到达一定数量（50）后，可以获得利息，这样就让玩家开发出来了新玩法：游戏前期可以一直输，快速累积金钱早点获利，然后快速升级人口，来增加获得高级棋子的概率。这使游戏深度更加丰富，不光光要考虑阵容的搭配，还要考虑金钱的分配，因为想要的阵容得花大量金钱才能刷新出来。这背后主要设计就是刷新棋子的概率设计，经济系统设计。

2. Unity 自走棋设计思路

2.1 游戏场景

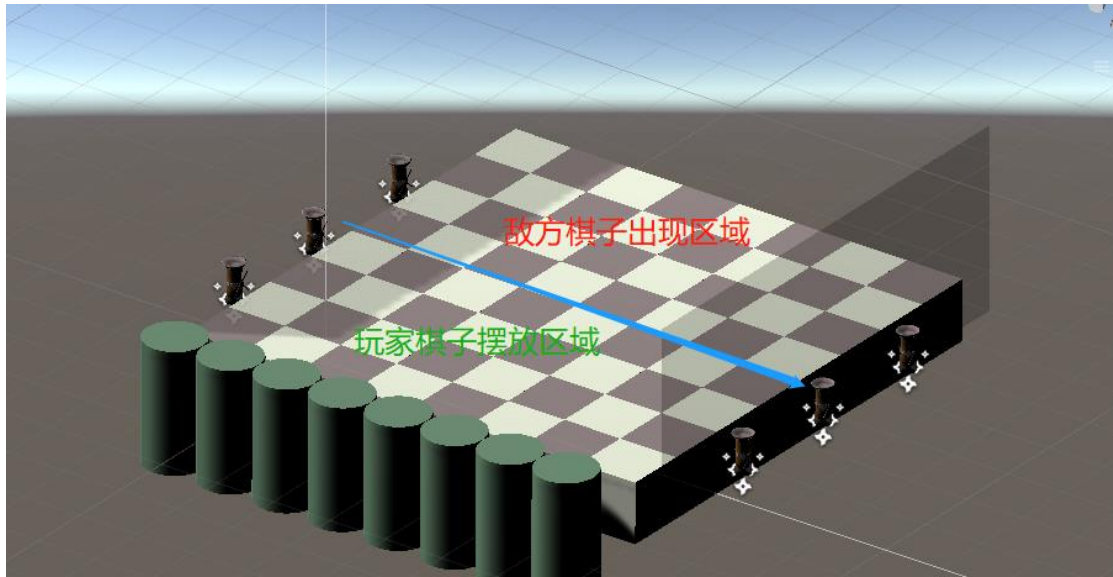


图 2.1A 游戏场景

如图所示，游戏场景很简单，分为两部分。

1. 8 根圆柱组成的高台，可以把购买的棋子放在上面备用。每个圆柱上最多放置一枚棋子。
2. 8x8 个立方体组成的平台，是棋子战斗场景。分为两个区域，左边的玩家区域和右边的敌方棋子出现区域。两边各有三根烟花燃放器。



图 2.1B 回合胜利烟花效果展示

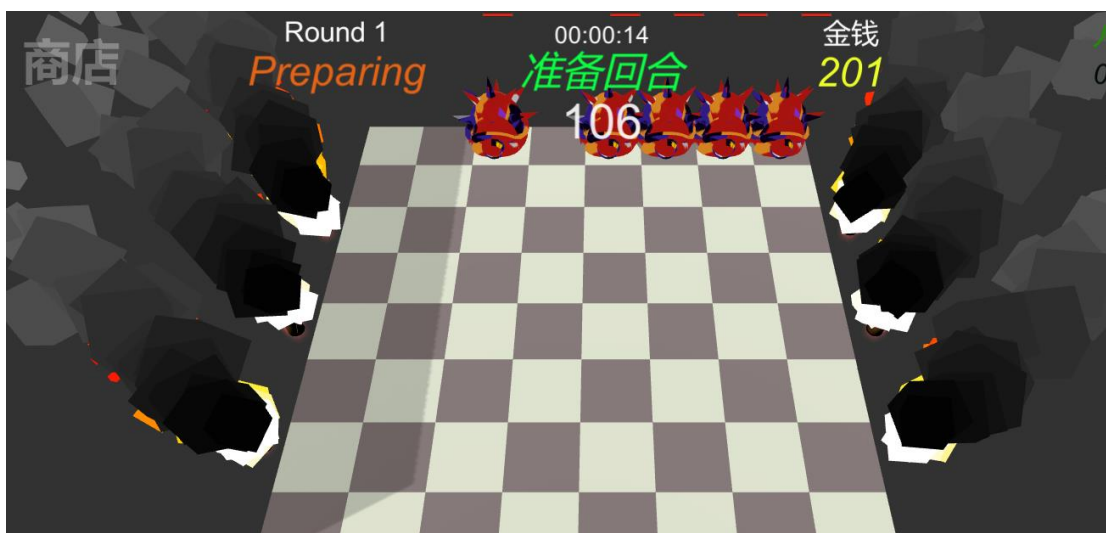


图 2.1C 回合失败烟花效果展示

设置烟花的主要目的，是为了给玩家每回合结果予以反馈，使玩家游戏体验更好。两个烟花特效均从 Unity Asset Store 中免费获得。

2.2 游戏 UI

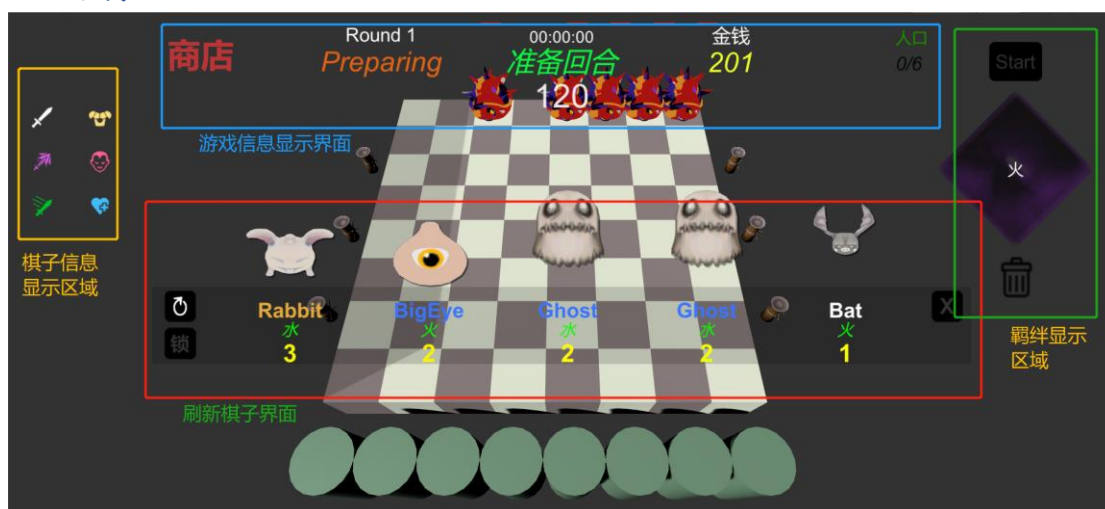


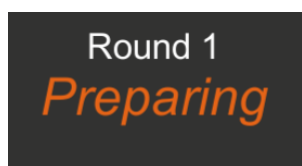
图 2.2A UI 界面

这个游戏战斗是自动的，所有主要交互就在 UI 上，所以 UI 很关键。如上图所示，UI 主要分为四大模块。

2.2.1 游戏信息显示界面



商店 (button)：用来开启/关闭刷新棋子界面，红色代表已开启，灰色代表关闭。



回合数显示：1,2,3.....下面红字可以用来显示每回合关键 boss 名字



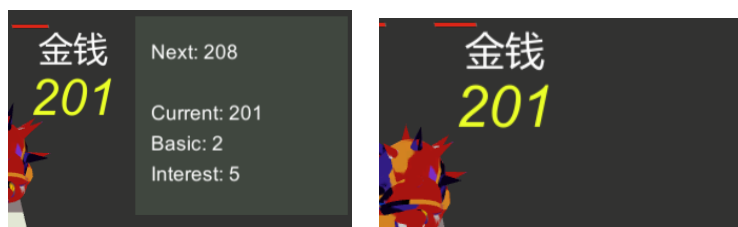
回合状态：分为三类

准备阶段-----选人放置阶段，玩家可以刷新，购买，并随意放置棋子，可以不断变换棋子摆放位置，调整阵型，时间较长；

战斗预告阶段-----只有一秒，玩家不可再选中棋子改变棋子位置了，是预留给系统记录棋子位置，等下回合重新开始时，自动把棋子归还到原来位置；

战斗回合-----玩家不可控制场上棋子了，只能等待棋子自动战斗到结束。但可以刷新购买棋子。时间较长。

在上面有白色小字显示开始游戏总时间，下面大号白字显示每回合各个状态剩余时间，为倒计时。

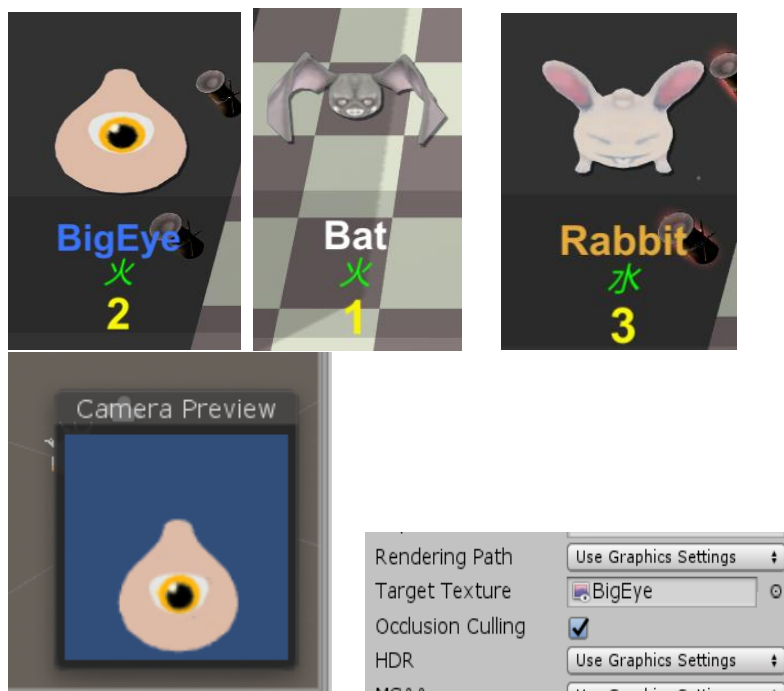


现有金钱信息 (button)：显示现有金钱，同时是一个按钮，用 event trigger 检测鼠标，如果鼠标放上去，就会弹出下回合将要获得的金钱数目，利息等信息，鼠标移开自动消失。



人口信息：右边数字显示场上最大可以放置棋子数量，左边显示已经放置在场上的棋子数量，如果达到最大值，在战斗预告阶段，会随机选择一名棋子弹回圆柱体备用区域，如果备用区域棋子 8 个满了，就会自动卖掉。

2.2.2 刷新棋子界面



棋子显示 (button)：刷新界面每次一共刷新五个棋子，每个棋子都会在显示棋子的样子，棋子名字，棋子属性，棋子价格。棋子名字的颜色分为白色，蓝色，和橙色，对应棋子价格不同。关键在于在显示棋子时，要显示动态，必须给每个棋子挂上一个 camera，然后输出 target texture，然后把 target texture 放入 UI Raw image 即可显示动态棋子。棋子显示还是一个透明的 button，当你鼠标点中后，会购买棋子，并实例化棋子在 8 个圆柱体上从左往右空闲的位置，如果位置已买，则使 button 失效，不能购买棋子。

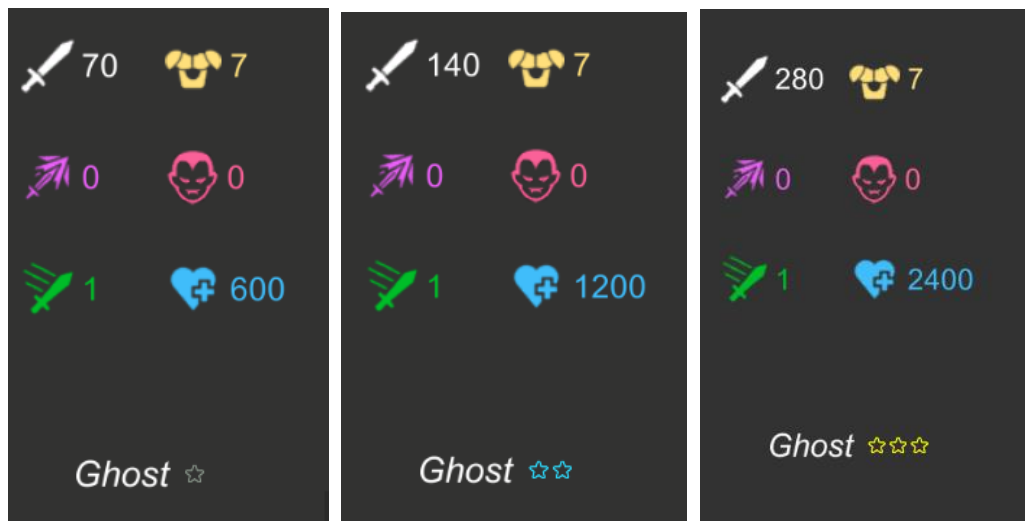


刷新 (button)：可以花费 2 金币，重新刷新 5 个棋子，次数不限，有钱就可以刷新，当金钱小于 2 时，令 button 失效变红。

锁定/解锁 (button)：当遇到想要的棋子，但是没钱购买时，可以按下“锁”，棋子在下一回合开始就不会自动刷新。同时令刷新 button 失效变红。解锁可以解除锁定状态。

关闭刷新棋子界面 (button)：关闭棋子刷新界面。

2.2.3 棋子信息显示区域



当玩家选中一枚棋子，在此区域从上往下，依次显示棋子攻击，护甲，暴击率，吸血率，攻击速度和生命值。其中攻击，护甲，生命值为 int，暴击率，吸血率，攻击速度为 float。

最下面为棋子名称，和棋子目前的星级，星级提升通过三个相同棋子合成，最多升到三星。

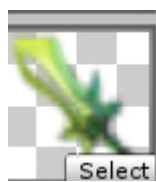
2.2.4 羁绊显示区域



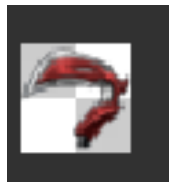
羁绊总共分四种，水，木，火，土。后面章节会详细讲。当在场一名对应属性棋子，图标就会亮起，当场上有两名相同属性的不同棋子时，就会显示图标光圈，光圈为呼吸灯类型，慢慢变亮然后再慢慢变暗，一直循环。中间的白字显示玩家目前选中棋子的属性。

上面 **Start (button)** 可以直接跳过准备阶段，使准备阶段倒计时归零。

下面**垃圾桶 (button)** 是删除棋子的，按下去之后，鼠标就会从原来图标（左）变成下图（右），然后点击想要删除的棋子，即可删除棋子并返还一定金钱。



原本光标



删除光标

2.3 游戏 AI

2.3.1 棋子动作控制

游戏的 AI 比较简单，所以我们使用状态机（FSM）足够应对，这里我们使用的双层 FSM。下图为第一层。

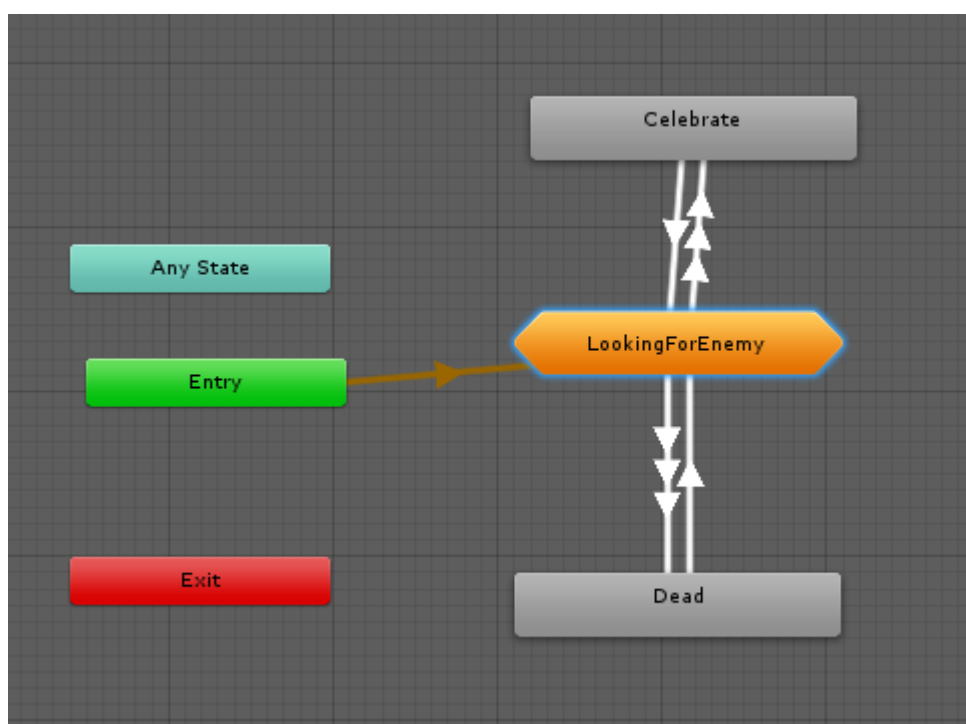


图 2.3A 双层 FSM 第一层

第一层有三个状态，**寻找敌人 (LookingForEnemy)**，死亡 (Dead)，庆祝 (Celebrate)。其中寻找敌人包含第二层三个子状态，其本身没有动作。

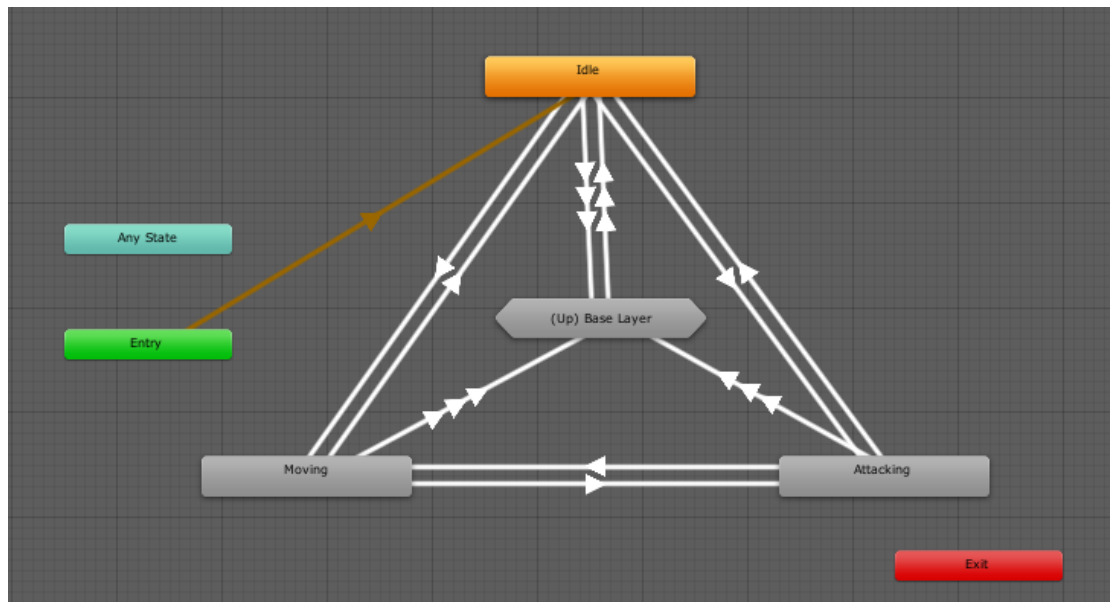
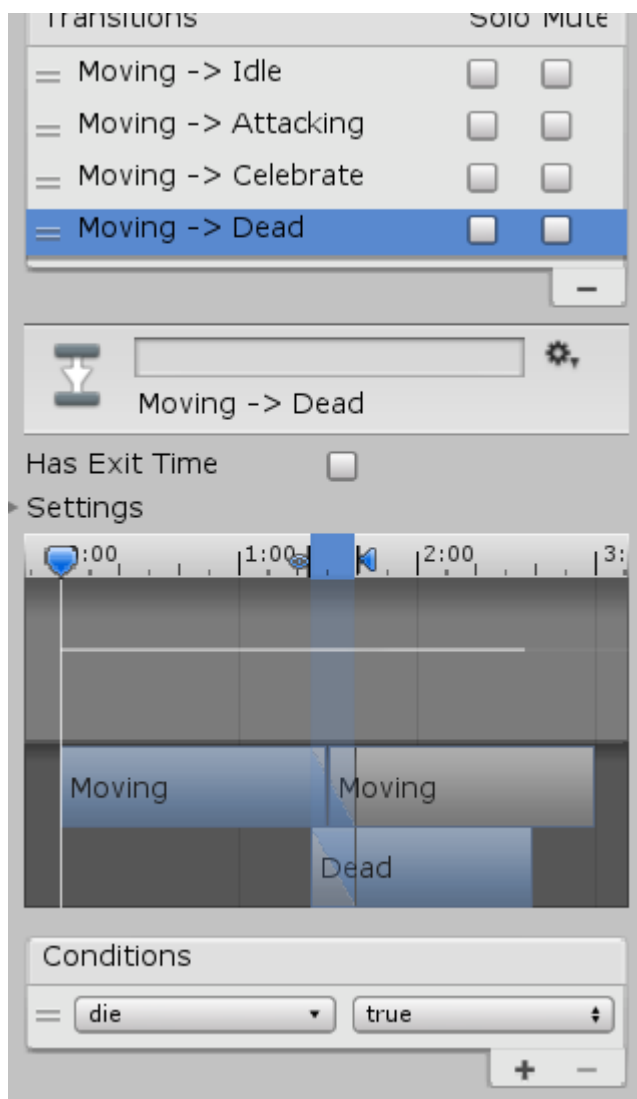


图 2.3B 双层 FSM 第二层

第二层为空闲（Idle）、移动（Moving）、攻击（Attacking）。



每个状态的转换由 bool 变量控制，而各个 bool 变量在脚本中赋值。

状态装换的逻辑为，刚开始棋子处于**空闲状态**，然后战斗开始后，棋子朝离自己最近的敌人走去，变成**移动状态**，然后当棋子进入攻击范围，棋子变成**攻击状态**，当棋子杀完当前敌人，由**攻击状态**又转变回**空闲状态**，当棋子被杀死，变成**死亡状态**，当所有敌对棋子死亡，棋子进入**庆祝状态**。



每个状态都对应一个模型动作，因为每个棋子状态机相同，所以用 override 控制器即可。

2.3.2 棋子自动寻路



图 2.3C 棋子战斗画面

规则： 棋子在所有敌人中寻找直线距离最近的一个敌人，然后朝那名敌人走去。遇到队友或者其他敌人挡在身前，会绕过去。直到与目标敌人的直线距离 \leq 棋子攻击范围，则停下并进入攻击状态。

这个棋盘其实没有任何障碍物，本不需要自动寻路。但是有种情况，可能会需要寻路算法，就是当棋子前方有友方棋子挡在面前，此时棋子应该绕过友方棋子，才显得更智能一点。我尝试了很多种方法，比如给棋子加碰撞体刚体，当与友方碰撞，就旋转一个角度，走出去一段距离，再继续寻找最近的敌人。但是效果都很一般，都显得不自然。最终我用来 unity 自带的 NavMeshAgent 自动寻路，才完美解决这个问题。棋子 Agent 半径设置为 0.4，都不需要增加碰撞体，棋子也不会互相重叠在一起，并且完美解决友军棋子挡在身前的问题。但是又出现了个小 bug，当棋子在攻击状态，攻击其他棋子时，会不停原地旋转。最终查阅各种资料，通过设置 `updateRotation` 和 `isStopped` 完美解决。如下面两行代码所示，只需在战斗状态的棋子如下设置即可解决棋子旋转 bug，然后在移动状态，调回到默认值即可。

```
transform.GetComponent<NavMeshAgent>().updateRotation = false;  
transform.GetComponent<NavMeshAgent>().isStopped = true;
```

2.4 棋子的技能与羁绊

2.4.1 羁绊

总共分五类

- 双火：所有棋子攻击力提升 20%
- 双水：所有棋子生命值提高 200 点
- 双土：所有棋子护甲提高 10 点
- 双木：所有棋子攻击速度提高 15%
- 火水木土：所有棋子暴击几率提高 5%，降低敌人护甲 5 点

双火指的是火属性不同棋子两个，如果是两个一样的火棋子，则不会触发。双水等同理。上面所有数值不一定是最终数值。

2.4.2 棋子技能

- Bat（已实现）



火属性：价值 1，远程，攻击范围 1 星 5, 2 星 6, 3 星 7

技能：吸血光环（给所有友方棋子提供吸血百分比），1 星 10%，2 星 15%，3 星 20%

特点：辅助队友吸血，拥有全游戏远的攻击范围

- BigEye（已实现）

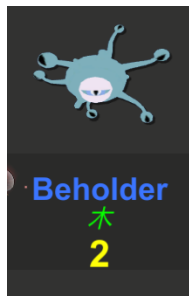


火属性：价值 3，远程，攻击范围 3

技能：1 星，攻击敌人伤害倍率为 1.2 倍；2 星，可以产生分裂攻击，但是主攻击伤害降低，主攻击倍率 0.8，副攻击倍率 0.6；3 星，可以产生两个分裂攻击，主攻击倍率 0.8，副攻击倍率 0.6，0.6

特点：分裂攻击，打击多个目标

- Beholder (待实现)



木属性：价值 2，远程，攻击范围 4

技能（持续 2 秒，冷却 3 秒）：偷取正在攻击的敌方攻击速度，并根据偷取的攻击速度增加自身攻击速度，并有几率增加双倍效果。1 星，偷取敌人 10% 攻速，40% 概率双倍收益，2 星，偷取 15 攻速，50% 概率双倍，3 星，偷取 20% 攻速，60% 概率双倍。

特点：偷取攻速，概率大幅提高攻速，看运气的棋子。

- Chest (已实现)



木属性：价值 3，近战，攻击范围 1

技能：暴击伤害改变，1 星 300%，2 星 450%，3 星 650%

特点：大幅提高输出，也是看运气的棋子。

- Turtle (已实现)



土属性：价值 2，近战，攻击范围 1

技能：反弹敌人普通攻击的伤害，1 星反弹 20%，二星反弹 30%，三星反弹 40%

特点：前排，反弹敌人伤害。

- Slime（待实现）



土属性：价值 4，远程，攻击范围 4

技能（持续 3 秒，冷却 5 秒）：给距离最近的一名友方棋子套盾，减免 50%伤害，并增加攻击力，1 星增加 15%攻击力，2 星增加 30%攻击力，3 星增加 45%攻击力。

特点：价格最高的棋子，辅助能力极强，有盾有加攻，所以稀有很难刷到。

- Ghost（待实现）

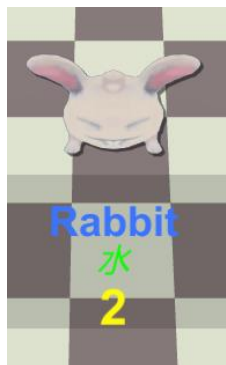


水属性：价值 1，近战，攻击范围 1

技能：普通攻击有概率增加金币，1 星 5%概率增加 1 金币，2 星 10%概率增加 1 金币，3 星 20%概率增加 1 金币，并且棋子在场上时，每回合结束收益增加 10%（取整 int，最低增加 1 金币，最高增加 10 金币）。

特点：前期收益很高，可以增快金钱获得速度，积累资源。

- Rabbit（待实现）



水属性：价值 2，近战，攻击范围 1

技能（持续 2 秒，冷却 5 秒）：嘲讽周围敌人，强迫敌人攻击自己，并在持续时间结束回复自身血量，1 星回血 30%，2 星回血 40%，3 星回血 50%。
特点：嘲讽敌人，持续回血。

2.5 怪物与关卡

待实现

2.6 数值与数据

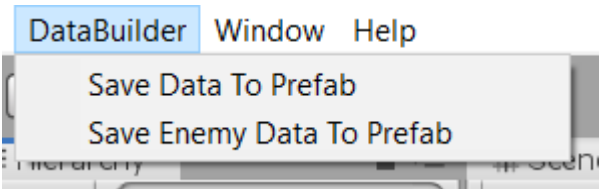
2.6.1 数据的导入

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ChessNe	ID	Star	Price	RaceClass	Max_HP	BaseAtt	Atk_spe	Armor	Baoji	LifeStea	HP_Reg	Atk_Range	
2	Slime	1	1	4	土	700	50	1	5	0.05	0	0	4	
3	Slime	2	2	8	土	1400	100	1	5	0.05	0	0	4	
4	Slime	3	3	24	土	2800	200	1	5	0.05	0	0	4	
5	Turtle	4	1	2	土	750	60	1.1	7	0.1	0	0	1	
6	Turtle	5	2	4	土	1500	120	1.1	7	0.1	0	0	1	
7	Turtle	6	3	12	土	3000	240	1.1	7	0.1	0	0	1	
8	Ghost	7	1	1	水	600	70	1	7	0.1	0	0	1	
9	Ghost	8	2	2	水	1200	140	1	7	0.1	0	0	1	
10	Ghost	9	3	6	水	2400	280	1	7	0.1	0	0	1	
11	Rabbit	10	1	2	水	900	70	1.5	3	0.1	0	0	1	
12	Rabbit	11	2	4	水	1800	140	1.5	3	0.1	0	0	1	
13	Rabbit	12	3	12	水	3600	280	1.5	3	0.1	0	0	1	
14	Bat	13	1	1	火	600	60	1.5	3	0.05	0	0	5	
15	Bat	14	2	2	火	1200	120	1.5	3	0.05	0	0	6	
16	Bat	15	3	6	火	2400	240	1.5	3	0.05	0	0	7	
17	BigEye	16	1	3	火	750	55	1.1	7	0.05	0	0	3	
18	BigEye	17	2	6	火	1500	110	1.1	7	0.05	0	0	3	
19	BigEye	18	3	18	火	3000	220	1.1	7	0.05	0	0	3	
20	Beholde	19	1	2	木	600	100	1.5	3	0.05	0	0	4	
21	Beholde	20	2	4	木	1200	200	1.5	3	0.05	0	0	4	
22	Beholde	21	3	12	木	2400	400	1.5	3	0.05	0	0	4	
23	Chest	22	1	3	木	650	80	1.5	3	0.1	0	0	1	
24	Chest	23	2	6	木	1300	160	1.5	3	0.1	0	0	1	
25	Chest	24	3	18	木	2600	320	1.5	3	0.1	0	0	1	

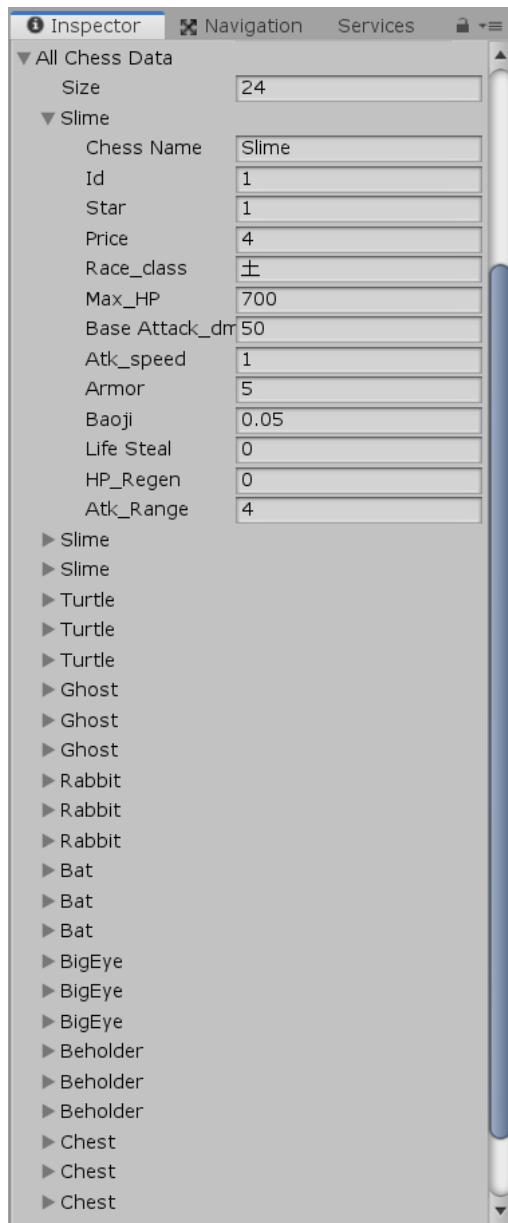
图 2.6A 棋子属性数值表（仅展示，非最终数值）

为了方便修改棋子的属性和调整平衡性，我们用一张 Excel 表格储存所有棋子属性的数值，有名字，ID，星级，价格，属性，血量，攻击等等。

使用 EPPlus 导入 Excel 表格，然后我写了个简单的 Unity 插件，可以把导入的数据，储存在创建好的预制体（Prefab）脚本上。因为预制体上的数据不会长时间保存，所以我们要把写好数据的预制体拉到场景中，作为独立的 GameObject，也就是要 Unpack Prefab Completely。然后我们的游戏就可以实时读取数据了。



上图是数据导入预制体插件，分别把 Excel 中棋子数据导入到预制体，和把 Excel 中怪物数据导入到另一个预制体。（棋子数据在 Excel 表格第一页，怪物数据在 Excel 表格第二页），导入后的数据结构如下图所示。



2.6.2 DOTA 自走棋 经济系统/人口规则 拆解

DOTA 自走棋，人口最大值为 10 人口，人口越多，能刷到稀有棋子的概率越高。金钱的作用有两点，一是刷新棋子，二是升级人口。

先说人口，DOTA 自走棋的人口升级需要经验，人口与经验对照为：2 人口 1 经验，3 人口 2 经验，4 人口 4 经验，5 人口 8 经验，6 人口 16 经验，7 人口 32 经验，8 人口 56 经验，9 人口 88 经验，10 人口 128 经验。

经验的获得有两种途径，每回合自动获得 1 经验，每 5 金钱可以购买 4 经验。

所以前三回合，人口会自动从 1 人口升到 3 人口。后面的回合需要看情况购买经验升级人口。

再说金钱，前 4 回合，每回合基础增加回合数的金钱。第 5 回合开始，固定增加 5 金钱。然后根据现有金钱，每回合结束计算利息，计算公式为总金钱的 10%，最高利息为 5 金钱。

2.6.3 Unity 自走棋 经济系统/人口规则

待实现，初步想法是，最大 4 人口，从第一回合到第四回合，从 1 人口增加到 4 人口。因为我们棋子数目很少，所以要限制人口。但是限制了人口，金钱的作用就大打折扣了，只有刷新棋子，购买棋子的用途了。所以我的想法是，花费 5 金钱可以增加下回合刷新某属性棋子概率。

2.6.4 战斗数值的平衡

待实现

2.7 游戏循环逻辑

已实现，待写

2.8 新手引导

待实现