《神灵选拔赛》

支持 AI 与自定义角色的回合制对战游戏

选题与需求分析报告

作者：邓博誉

一、背景意义

自从计算机进入普通民众的视野以来，利用计算机来制作和游玩电子游戏一直是很流行的娱乐活动。从最原始的“Pong”乒乓球游戏，到现在风靡全球的《塞尔达传说-荒野之息》，电子游戏始终在软件领域起着举足轻重的作用。其中，作为回合制的战斗模型，在上世纪 80 年代便开始采用，由于其对新玩家的友好度高、上手难度高、紧张感小和变化多样等特点，适合多种玩家，一直被多种 RPG（角色扮演类游戏）和 AVG（冒险类游戏）所青睐。

虽然回合制战斗看似简单，但在模型开发过程中，又有很多挑战。例如，在《精灵宝可梦》系列游戏中，其游戏特色决定平均每一年都会有数十以至于上百个战斗角色和技能加入其对战系统，如果其回合制战斗模型的设计在可拓展性的方面有所欠缺，那么每一次版本更新的工作量会是巨大的。此外，当越来越多的玩家开始抛弃上世纪末流行的硬核游戏时，越来越多的玩家加入了大量使用回合制战斗的休闲游戏阵营。同时，大部分游戏的回合制战斗的 AI 模型过于粗糙，为了增加挑战性，只能使用数值方法，塑造出一个强大的敌人来让玩家挑战，这样一来，回合制游戏玩家便少了许多与敌人有来有回，斗智斗勇的快感。此外，在 RPG 和 AVG 玩家群体中，不乏富有想象力，希望能在游戏中控制一个由自己所塑造的人物的青少年，而目前市面上的绝大多数游戏，只支持在人物外形和音效等方面予以定制。至于数值模型的玩家自定义功能方面，则出于平衡性问题和程序拓展性问题而一直没有在成品游戏中出现。

本项目的目标即为尝试解决此问题。项目将使用 C++语言制作一个支持单人对战 AI 和双人对战的回合制战斗系统，战斗系统的复杂程度应达到或超过《精灵宝可梦-Let’s go》的复杂程度。角色应有攻击、防御、速度、生命值、魔法值等属性值；技能应至少包括以下类型：持续&延时攻击类、持续&延时恢复类、角色/场地状态变化类；此外，角色和场地应可附加不同的状态，如：角色的毒状态会使角色在回合结束时损失生命值；场地的暴雨状态会使场地内所有角色的速度下降等……此外，新的角色、技能和状态应均可以通过添加文件/在设定文件中增添代码的方式快速地加入源程序，不应需要改变原有程序代码。

此外，模型应有强大的战斗 AI，其职能应接近或超越玩家的游戏水平。利用此 AI 制作单人游戏模式，并利用 AI 的双手互搏来调整新添加的人物/技能/状态的数值，使游戏不同角色的 AI 互搏战斗胜率维持在 50%左右，以保持在添加自定义对象后游戏的平衡性。

通过实现以上需求，本模型可在目前的回合制战斗游戏中“AI 强度不够”和“无数值自定义功能”的问题中做出探索，有一定的实践意义。

二、同类软件调研分析

回合制战斗游戏种类繁多，此处例举几个知名游戏的战斗系统予以介绍和分析：

· 1、《精灵宝可梦》系列：

精灵宝可梦系列是回合制战斗的创始者之一，其特点在于：角色繁多且更新速度快，战斗系统复杂，数据庞大，目前已有逾 800 个战斗角色、数百种特性、数百种道具和上千种技能。这样庞大的设定体系显然需要拓展性极强的核心架构。但其缺点也很明显，即战斗模式单一，由于角色速度只是决定先手方，使得战斗缺少不确定性带来的激情，且其游戏内置 AI 过于简单。另外，虽然技能种类很多，但雷同很多，多样性较差。

· 2、《仙剑奇侠传》系列：

《仙剑》系列游戏的战斗系统是目前最为常见的回合制游戏形式之一。其特点在于将角色的攻击、魔法、特技、防御等动作分开处理，分别对应消耗不同的点数，技能多样性比口袋妖怪要好，且可以多对多对战。但此类游戏角色较少（数十个），且每一代都会重新制作，所以不需要过多地考虑拓展性问题，且其战斗 AI 也一样过于羸弱，只能依靠在数值上加强敌人来为玩家带来挑战。

· 3、《Mother》系列：

《Mother》系列是上世纪较为小众的 RPG 游戏系列，但近年来越来越受到追捧，其原因之一为其回合制对战游戏的特点。其角色在受到伤害后生命值不会立即扣除，而是有一个逐渐减少的过程，如果角色受到了致命伤害，而在生命值减少至 0 之前对其采取回复手段的话，就可以避免死亡。这是一个将回合制战斗方式与快节奏的战斗联系起来的优秀案例。其缺点与以上两款游戏相似：智慧不高的 AI 和无法自定义角色。

此项目要开发的软件，会综合《精灵宝可梦》和《仙剑奇侠传》的优点，兼具《宝可梦》的可拓展性和《仙剑》的多样性，同时由于时间因素，会舍弃一些不影响设计思路的功能，如《宝可梦》的道具和特性系统。另外，此项目会将《Mother》系列独特的战斗机制融入其中，让角色的“速度”属性变成更刺激的“根据概率额外进行一回合”的设定，以让玩家在较为平淡的回合制战斗系统中也体会到赌博心理和不确定性带来的紧张感和刺激感。最重要的是，本程序会加入强大的 AI 系统，并设计游戏自动调平衡功能，以支撑可拓展性衍生出的自定义角色功能。

三、详细功能设计

3.1 模型设计

游戏中的主要元素有：角色、场地、技能、角色状态、场地状态，下面分别介绍。

3.1.1 角色

角色即为游戏中由玩家或 AI 操作的单位。角色具有以下属性：

1，C_NAME：该角色的名称，用于战报打印和回合方识别；

2，HP：即生命值，生命值<=0 时，即判该方失败；

3，MP：即魔法值，释放大多数技能会消耗魔法值，剩余魔法值低于技能需求时，即无法释放；

4，ATK：即攻击力，决定角色造成伤害大小的数值之一；

5，DEF：即防御力，决定角色受到伤害大小的数值之一；

6，SPD：即速度，决定角色此回合获得额外行动机会概率的数值之一；

7，STAT_L：状态列表，角色身上所附加的状态记录在此；

8，MOVE_L：技能列表，角色可以使用的技能记录在此；

9，MAX_HP：角色的生命值上限，也是角色的初始生命值；

9，MAX_MP：角色的魔法值上限，也是角色的初始魔法值；

3.1.2 技能

技能为玩家选择让角色进行的所有行动的总称。技能有如下属性：

1，M_NAME：技能的名称，主要用于提供给用户以识别

2，M_INFO：技能的简介，在用户选择技能后，会打印出来提供给用户；

3，S_DH：对自身的生命值造成的变化，正加负减；

4，S_DM：对自身的魔法值造成的变化，正加负减；

5，S_DA：对自身的攻击力造成的变化，正加负减；

6，S_DD：对自身的防御力造成的变化，正加负减；

7，S_DS：对自身的速度造成的变化，正加负减；

8，S_RMSTAT：移除自身的指定状态；

9，O_DH：对对手的生命值造成的变化，正加负减；

10，O_DM：对对手的魔法值造成的变化，正加负减；

11，O_DA：对对手的攻击力造成的变化，正加负减；

12，O_DD：对对手的防御力造成的变化，正加负减；

13，O_DS：对对手的速度造成的变化，正加负减；

14，O_RMSTAT：移除对手的指定状态；

15，M_ATK：技能的攻击力，与释放者的攻击力 ATK 和被攻击者的防御力 DEF 组成伤害公式，公式暂定为：

$$DAMAGE = max(ATK * M\_ATK / DEF, 1)$$

16，M_ASTAT：技能增加的状态

17，SELF_TARGET：布尔量，技能是否以自己为目标，作用于 M_ATK 和 M_ASTAT 参数的目标。

18，COM_MOVE：组合技能，默认为 void，若不是，则可以是一个组合技能。释放该技能之后立即释放此组合技能。

19，M_ID：技能 ID，作为技能的唯一识别号

3.1.3 状态

状态是本游戏玩法多样性的核心。状态被技能附加在角色的状态列表中，可以通过技能消除或等持续时间为 0 后自动消除。状态有不同的初始持续时间，每一次经过一回合的回合结束阶段，其持续时间-1，同时发动其当前回合的特定功能（例：可以很轻易地实现一个“毒疗”状态，持续 2 回合，第一回合扣除生命值，第二回合增加生命值和攻击力）。这就需要其在每一回合结束时，经过一不同的状态对象特有的“刷新函数”重新设置其各项属性值。这个“刷新函数”即为此回合制对战游戏的重要特点之一。

状态有如下属性：

1，E_NAME：状态的名称，主要用来提供给用户以识别；

2，E_INFO：状态的简介，在用户选择显示角色详细信息后，会打印给用户；

3，E_DH：对自身的生命值造成的变化，正加负减；

4，E_DM：对自身的魔法值造成的变化，正加负减；

5，E_DA：对自身的攻击力造成的变化，正加负减；

6，E_DD：对自身的防御力造成的变化，正加负减；

7，E_DS：对自身的速度造成的变化，正加负减；

8，E_RMSTAT：移除自身的指定状态；

9，INI_T：初始持续时间，即状态被施加时的持续时间；

10，N_T：当前持续时间

11，REF_F()：状态的刷新函数，激活时 N_T--，且根据需求刷新状态对象其他属性的值。

12，E_POS：布尔量，是否是正面状态?

13，E_NEG：布尔量，是否是负面状态?

14，E_ID：状态 ID，作为状态的唯一识别号。

3.1.3 场地

场地拥有其专属的状态，同时也负责记录自开局至当前的所有游戏数据，包括每回合的角色状态、技能选择等。场地状态为两个角色状态的组合和场地状态施加者的记录。每回合结束时，在角色状态触发作用之后，即触发场地状态作用。

3.1.4 相关算法和公式

游戏的 AI 暂定使用 Monte-Carlo 随机搜索算法+多臂老虎机模型，其打分公式为：

$$SCORE = A * (SELF\_HP - OPPO\_HP) + B * (SELF\_MP - OPPO\_MP)$$

其中 A、B 为待定系数，需要在游戏 α 测试时确定。

除了攻击造成的伤害使用 3.1.2 节第 15 项提到的公式外，其余均使用简单的加减法进行。当人机对战时，由于人类的计算能力不如机器，所以在随机搜索树算法的打分过程中可能需要适量加入扰动，以提高用户的胜率，具体应该加入多大程度的扰动，还需要测试后决定。在角色平衡性调整功能中，不加入扰动。

角色平衡性调整过程，在一次次的 AI 互搏后，使用模拟退火算法调整角色的 ATK、DEF 和 SPD，三项应同时同比例增减，不能过分增强或削弱某一参数，以求保留自定义角色属性值特色。如果项目开发尚有空余时间，可尝试加入技能/状态的平衡性自动调整。

3.2 游戏阶段

该游戏将被设计有如下几个阶段：

1，选项阶段：用户可通过输入命令来选择进入单人模式、双人模式或角色调平衡模式。（选项应用 1、2、3……编号，由计算机打印出来，用户输入 1，2，3……即可确认，下面所有的选择均如此设计。）

2，游戏开始：在单人模式中，用户首先选择自己的角色，然后选择电脑对手的角色。在双人模式中，A、B 两用户依次选定自己的角色。调平衡模式中，用户选择想要调平衡的角色。

3，游戏阶段：若用户选择了单人或双人模式，则游戏开始，用户根据系统提示选择动作，直至一方生命值变成 0 时，游戏结束。在单人模式中，用户需根据计算机提示，在每一个回合开始之前输入想要使用的技能编号，电脑应根据 AI 算法自动选择对其最优的技

能释放。在双人模式中，计算机应提示该由哪一个角色控制的玩家输入行动，两个玩家根据提示轮流输入指令，直至一方生命值变成 0，游戏结束。

4，角色调平衡阶段：在此阶段，玩家选择角色后，计算机应自动将该角色与其余角色使用 AI 进行互搏，计算出胜率。随后根据胜率增加/减少被选择的角色的基本属性，随后继续进行 AI 互搏，直至胜率在对战所有其他角色时，其胜率均值接近 50%。

3.3 游戏设定

游戏设定，即为游戏中角色、技能等对象的建立。虽然本游戏将支持快速添加新的角色、技能和状态，但作为成品游戏，此游戏在完成时将包括 4 个角色、20 个技能，10 种状态以供演示使用。暂定如下：

3.3.1 角色设定

C_NAME	Irrawa	Mew	Rosie	Asibi
MAX_HP	1200	1000	900	1100
MAX_MP	800	1000	1000	1200
ATK	125	100	90	110
DEF	100	100	90	110
SPD	80	100	120	100
MOVE_LIST	1，2，3，4，5	6，7，8，9，10	11，12，13，14，15	16，17，18，19，20

注：以上数值均为初期设计数据，未经平衡性调整，具体数值需在后期测试后确定。

3.3.2 技能设定

ID	技能介绍	ID	技能介绍
1	ATK:65, MP_COST:80	11	ATK:200, Cost SELF_HP:100, 吸血（将造成伤害的 50%恢复成自身生命值）
2	ATK:20, MP_COST:100, add stat: CHOKE to opponent	12	ATK:10, MP_COST:45, add STATIC_OVERLOAD stat to self.
3	ATK:30, MP_COST:150, add stat: AQUA_BLAST to opponent	13	MP_COST:50, 随机复制一个对手的技能并施放，该技能不再额外消耗魔法。
4	MP_COST:50, add stat: NAYAD_BREEZE to field	14	MP_COST:75, add SPIRITIFIED stat to self
5	MP_COST:20, randomly remove NEG stats on self.	15	MP_COST:25, Cost SELF_HP:50, SELF_DEF += 20, SELF_ATK += 10
6	MP_COST:75, add stat NETHER_CIRCUIT to self.	16	MP_COST:20, 使游戏状态回溯至 3 回合之前（自身 MP 在回溯后消耗）,若目前回合数小 1 于 3，则技能无效。
7	MP_COST:25, self ATK += 15	17	MP_COST:50，add stat NIGHTMARE to opponent
8	MP_COST:25, ATK:30, add TOXIC stat to opponent	18	MP_COST:50，add stat LOTUSLAND to self
9	MP_COST:25, add POISON_AURA to field.	19	MP_COST:10，跳过下一个回合
10	MP_COST:150, 从对手身上移除 POISON 或 TOXIC 状态，立即对其造成该状态在接下来 5 回合内会造成的伤害；若未能移除这些状态，则固定造成 30 伤害。	20	ATK:500，只有游戏进行 50 回合后才有效

3.3.3 状态设定

名称	类型	说明
CHOKE	NEG	持续 3 回合*，减少 15 速度
AQUA_BLAST	NEG	持续 2 回合，若该状态自然消失，消失时造成 ATK=300 的伤害，计算攻防。
NAYAD_BREEZE	FIELD	持续 5 回合，施加者每回合恢复 20HP 并获得速度增益**，初始增益为 5，每回合+=5，最高 100
NETHER_CIRCUIT	POS	持续 2 回合，施加者 ATK 和 DEF 获得 20 增益，且 POISON 或 TOXIC 对其造成的伤害会转而恢复等量生命值
TOXIC	NEG	持续 3 回合，只要此状态不消失，其每回合造成的伤害遵循以下公式： DAMAGE = 10 * 1.3^n 其中 n 为持续回合数
POISON_AURA	FIELD	持续 5 回合，每回合为场上双方添加 POISON 状态。
POISONED	NEG	持续 3 回合，每回合造成 30 伤害
STATIC_OVERLOAD	POS	持续 5 回合，每当角色受到直接技能伤害时，对对手造成该伤害的 50%的伤害。
SPIRIFIED	POS	持续 1 回合，DEF 获得 3 倍增益。（收到的伤害降为 1/4.
NIGHTMARE	NEG	持续 2 回合，施加回合结束时造成 200 伤害，状态消失时恢复等量生命值
LOTUS_LAND	POS	持续 2 回合，施加回合结束时恢复 200 生命值，状态消失时损失等量生命值

注：

*：状态的持续时间可以被刷新，但其余数值在状态消失前不会刷新。当一个还未消失的状态被再次添加时，该状态的持续时间会刷新到初始值，其余的一切均不变。比如“NAYAD_BREEZE”状态，若角色每 4 回合为自己添加一次该状态，则其速度增益会持续上升至上限值 100。

**：“增益”和“削弱”与状态共存，会随着状态的消失而消失。技能和状态造成的“属性值上升”和“属性值下降”则通常不会消失。比如，技能 7 的“self ATK += 15”会直接加到角色属性值上，不会消失。

3.3.4 战斗及清算流程

回合战斗的流程如下:

1，玩家 1 选择角色技能使用;

2，玩家 2 或 AI 选择游戏技能使用 ;

3，双方角色判断速度，速度快的一方优先行动，若速度相同则随机决定。这里假设 A 角色比 B 角色快;

4，A 角色行动;

5，A 角色行动技能清算（包括扣除技能魔法值消耗，扣除造成伤害的生命值，添加状态至相应列表等，下同）;

5，判断是否分出胜负（若是，结束游戏，显示结果，下同）;

6，判断 A 是否获得额外回合，获得额外回合的概率计算公式为:

$$P = \min(\max([(SPD\_A/SPD\_B) - 1]/2, 0), 0.9)$$

若获得额外回合，则回到步骤 1（注意，额外回合不会造成状态的清算）;

7，B 角色行动;

8，B 角色行动技能清算;

9，判断是否分出胜负;

10，A 角色状态清算（各状态持续时间-1，若持续时间<0 则从状态列表移除状态，下同）;

11，判断是否分出胜负;

12，B 角色状态清算;

13，判断是否分出胜负;

14，场地状态清算;

15，判断是否分出胜负;

16，回到步骤 1.