

# 骰碰骰

在酒馆牌桌上弹射骰子的四人  
策略原型

项目四 · Unreal Engine 5.8

2026.09.24 – 10.07



# 目录

|    |                                                    |         |
|----|----------------------------------------------------|---------|
| 01 | <b>项目简介与设计动机</b><br>这是什么游戏，为什么做它                   | 03 – 06 |
| 02 | <b>研究参考、设计问题与世界观</b><br>参考与取舍；核心问题：骰子的随机性；酒馆世界观    | 07 – 14 |
| 03 | <b>核心玩法</b><br>游戏框架、实机流程、核心循环、四种行动与计分              | 15 – 24 |
| 04 | <b>关卡设计、场景迭代与镜头</b><br>桌面路线、酒馆化地形、场景迭代、画面打磨与镜头跟随   | 25 – 36 |
| 05 | <b>策略系统与玩法体验</b><br>位置、点数、地形与时机的取舍；策略案例与结算演出       | 37 – 45 |
| 06 | <b>试玩迭代与 AI 角色</b><br>31 条试玩反馈与优化表；AI 决策、角色表现与你的双手 | 46 – 62 |
| 07 | <b>验证与反思</b><br>验证范围、反思与下一步、工作范围                   | 63 – 67 |
| 附录 | <b>技术实现与蓝图复现</b><br>模块分工与 19 张蓝图                   | 68 – 90 |

# 01

## 项目简介与设计动机

这是一个什么样的游戏？  
我为什么想做它？

### 本章内容

- 04 项目概况：酒馆牌桌上的四方骰子对抗
- 05 设计动机：让骰子的运气变成可以经营的局面
- 06 体验目标：三个设计支柱

# 项目概况：酒馆牌桌上的四方骰子对抗



座位视角实机：入座后等待开局，三位对手在各自座位前

## 一句话介绍

四人围坐酒馆长桌，轮流把骰子弹进得分区。骰子会一直留在桌上，点数、位置和地形在三个回合里不断被改写。

## 项目信息

类型：物理弹射 × 回合策略  
对局：1 名玩家 + 3 名 AI 对手  
引擎：UE 5.8 (C++ / Slate)  
周期：2026-09-24 至 10-07

# 设计动机：让骰子的运气变成可以经营的局面

## 出发点

骰子游戏的乐趣来自运气，挫败也来自运气：投出去是几就是几，玩家只能等着。  
我想做一个运气依然有分量、输赢却取决于判断的桌边游戏——像在酒馆里和几位老熟客赌一局：手上要有准头，心里要算计别人。

试玩中的一句话（原话精简）

“对抗性要发挥出来，而不是大家各自弄自己的。”

### ① 随机要有分量，但不决定胜负

保留物理翻滚的惊喜，同时让方向、力度、位置和事后干预都掌握在玩家手里。

### ② 每一手都和别人有关

骰子留在桌上互相阻挡与碰撞，改点、倒酒都会影响对手。

### ③ 规则长在场景里

酒渍、破毯、木挡板就是地形；对手是有名字、有脾气的常客，不靠说明书解释规则。

# 体验目标：三个设计支柱

## 手感

出手由玩家掌控

### 落到设计上

- 拖拽距离就是力度，方向随手
- 0–65° 仰角：贴地、绕行或高抛
- 轨迹只预告到第一次接触
- 碰撞传力调到“撞得动别人”

### 怎么检验

能否稳定地把骰子打进区、把对手撞出区

## 取舍

每一手都有代价

### 落到设计上

- 投骰、拨动、出牌、倒酒只能选一
- 卡牌每张一次，倒酒每局一次
- 规则每回合轮换，旧位置会贬值
- 最后一手可以改写结算

### 怎么检验

玩家能否说出“这一手为什么这样选”

## 氛围

坐在酒馆的桌边

### 落到设计上

- 座位视角与入座开场
- 三位有名字、有脾气的对手
- 酒客围观，关键时刻有反应
- 环境声与每回合轮换的酒馆曲

### 怎么检验

玩家是否会留意对手的反应并据此判断

# 02

## 研究参考、设计问题与世界观

参考了什么？  
要解决的核心问题是什么？  
游戏发生在怎样的世界里？

### 本章内容

- 08 灵感来源：桌边对局与骰子弹射
- 09 灵感来源：站位策略与行动反馈
- 10 同类研究：取什么，不取什么
- 11 核心设计问题：骰子的随机性
- 12 解法：把随机拆成四层
- 13 世界观：雨夜酒馆里的一局骰碰骰
- 14 世界观如何服务玩法：酒馆里的每样东西都是规则

# 灵感来源：桌边对局与骰子弹射

## 《地狱黑杰克》 Black Jacket



参考特点：围桌对局，用特殊牌干预局面。  
项目转化：酒馆桌面与不同对手营造氛围。四张一次性牌改点或保护骰子，出牌与投骰共用一手。

## Tumblin' Dice 骰子弹射桌游



参考特点：弹射骰子，朝上点数乘落点倍率。  
项目转化：骰子跨回合留桌，同一站位的收益随计分规则变化；碰撞同时改变点数和后续局面。

# 灵感来源：站位策略与行动反馈

## 冰壶：位置也能创造价值

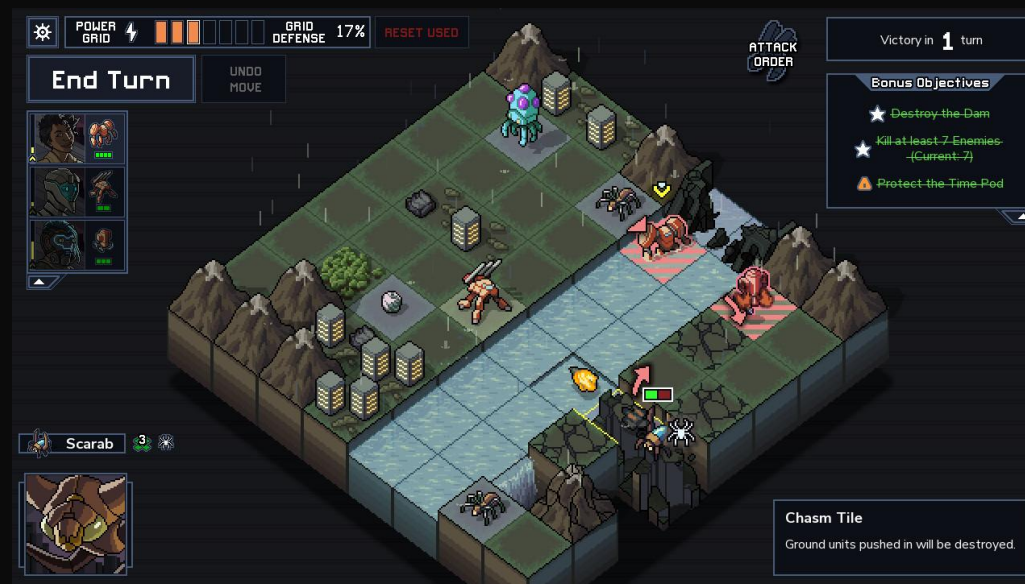


图源：Curl Aberdeen 官方入门页

参考特点：占位、保护与击打具有不同目的。

项目转化：低分骰也可挡住路线。中央争夺区、侧向绕行和高抛越障，让玩家在保位与进攻之间取舍。

## Into the Breach：行动前读懂局面



图源：Subset Games 官方 Steam 页面

参考特点：行动前显示敌人的攻击意图。

项目转化：显示当前分、下一轮规则和合法目标，行动后反馈得失。物理结果仍有不确定性。

# 同类研究：取什么，不取什么



原则：先让单局对抗成立，再叠加系统；随机风、无预告地形等会降低结果可读性的设计暂不采用。

# 核心设计问题：骰子的随机性

## 试玩中暴露的问题

“落下是几就是几，除非取回重投，没法改变。”

试玩反馈

想留住区里的高分骰，改点却只能拆掉自己的局面。

试玩反馈：“这是核心矛盾”

高抛撞了几次看不清，点数像是“突然变了”。

机制研究

## 设计判断

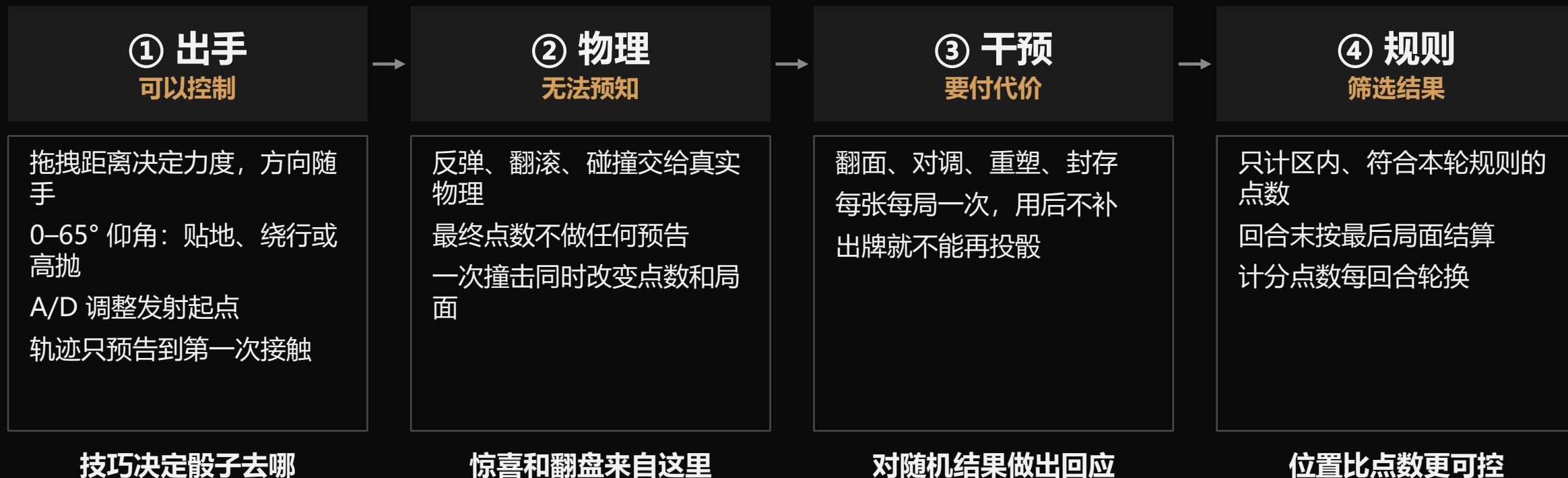
随机本身不是问题——它制造惊喜和翻盘。问题是玩家面对随机无事可做。

目标不是消除随机，而是给它一个结构：出手前能控制，结果出来后能干预，最后由规则筛选。

## 试过、又撤回的方案



# 解法：把随机拆成四层



## 随机制造局面，选择决定胜负。

AI 遵守同一结构：未知翻滚按期望值估算，重塑不偷看随机结果，也从不修改真实物理。

# 世界观：雨夜酒馆里的一局骰碰骰



美术基调：仿中世纪北欧酒馆，参考《猎魔人》的时代感——烛光、木梁、羊皮纸。当前实机俯瞰。

## 设定

北地边陲的一间老酒馆。每逢雨夜，常客们就围着那张磨旧的绿毯长桌玩“骰碰骰”——骰子碰骰子：三巡过后，谁留在灯下得分区里的点数最多，谁就赢走今晚的酒钱。

## 桌边的三位常客

Hob（橙）：慢性子老熟客，占住中间就不挪窝  
Aldric（紫）：爱押大注的布商，出手激进  
Brann（绿）：退役佣兵，嘴上抱怨，只守不攻

# 世界观如何服务玩法：酒馆里的每样东西都是规则

## 洒在桌上的酒渍

规则：低摩擦区：摩擦 0.06（桌毯 0.45）

目的：滑得更远，也更容易出界，考验力度

## 磨破的桌毯

规则：高摩擦区：摩擦 1.15，不能倒酒

目的：可靠的停骰点与缓冲落点

## 钉在桌上的木挡板

规则：截断直线路线的短墙

目的：逼出贴地绕行与高抛两种路线

## 举杯倒酒

规则：每局一次的行动：主动制造酒渍

目的：地形从固定变成玩家的选择



酒馆里的东西 → 在玩法里是什么 → 设计目的

## 三位常客

规则：三种 AI 风格：控区 / 激进 / 保守

目的：对手的性格就是可以读出的打法

## 围观的酒客

规则：对关键时刻鼓掌、惊呼、起哄

目的：告诉玩家“刚才发生了大事”

## 桌沿的立卡、铜币与沙漏

规则：分数、累计与轮到谁都放进场景

目的：少用 2D 面板，信息不出戏

转折点（试玩反馈）：冰面、矮墙与酒馆主题割裂，于是按主题换成酒馆里本来就有的东西，功能不变。

# 03

## 核心玩法

一局怎么进行？  
玩家每一手在决定什么？

### 本章内容

- 16 游戏框架：一局三回合，每方每回合一手
- 17 游戏流程：一局实机 16 步
- 18 核心循环：读局面 → 选一手 → 出手 → 看结果
- 19 四种行动：同一手，四种改变局面的方式
- 20 三回合计分规则
- 21 当前分与累计分
- 22 拖拽投骰与力度
- 23 四张有限卡牌
- 24 倒酒的选择与限制

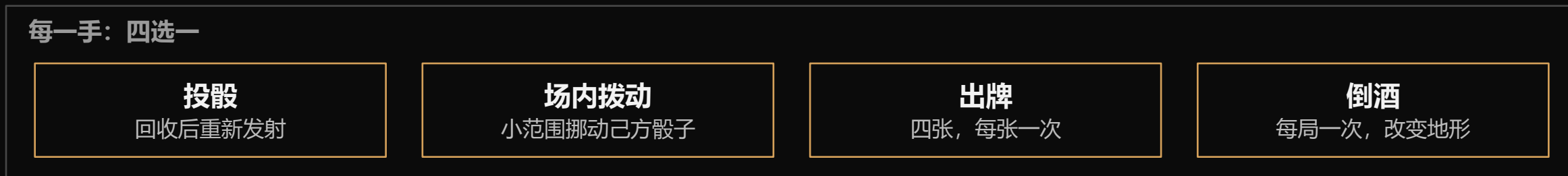
# 游戏框架：一局三回合，每方每回合一手



## 每个回合



## 每一手：四选一



设计重点：骰子不会被收走，每一手的结果都留给后面的人；规则每回合轮换，同一个位置的价值会变。

# 项目四 · 游戏流程

16 步 · 3 回合 · 4 人同桌

骰碰骰 (DiceClash) 实机一局：入座 → 开局投放 → 三回合行动、出牌与结算 → 终局领奖台



**入座开场**  
俯瞰酒馆后推镜入座，点击才开始



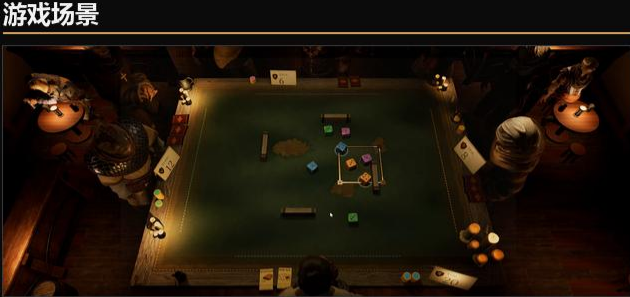
**三位常客问候**  
Hob、Aldric、Brann 依次开口



**开局投放**  
拖拽定方向与力度，铜色刻度环



**对手出手**  
镜头转向行动方，Brann 掷出绿骰



**酒馆与牌桌** 酒客与观战者围着牌桌



**翻面特写**  
镜头拉近，蓝骰 1 → 6



**打出「翻面」**  
卡牌浮到桌心，随后化沙消散



**选择行动**  
投骰 / 出牌 / 倒酒，手牌扇形升起



**开局落定**  
每人两颗骰子留在桌上，不计分



**桌边常客** 三位对手各有外观、性格与待机



**对手也出牌**  
Hob 打出翻面，走同一套演出



**回合结算**  
分数化成铜币，飞进桌沿硬币堆



**第一回合领奖台**  
四人上台，按名次与性格播放待机



**打出「对调」**  
先点自己的骰子，再点附近的敌骰



**骰子特写** 染色骨骰、黄铜得分框、粗麻破损区



**终局领奖台**  
你站上木桶举手，喊出胜利台词



**最后一手**  
镜头跟着最后一颗骰子推进



**第三回合**  
得分区换了位置，立卡记着总分

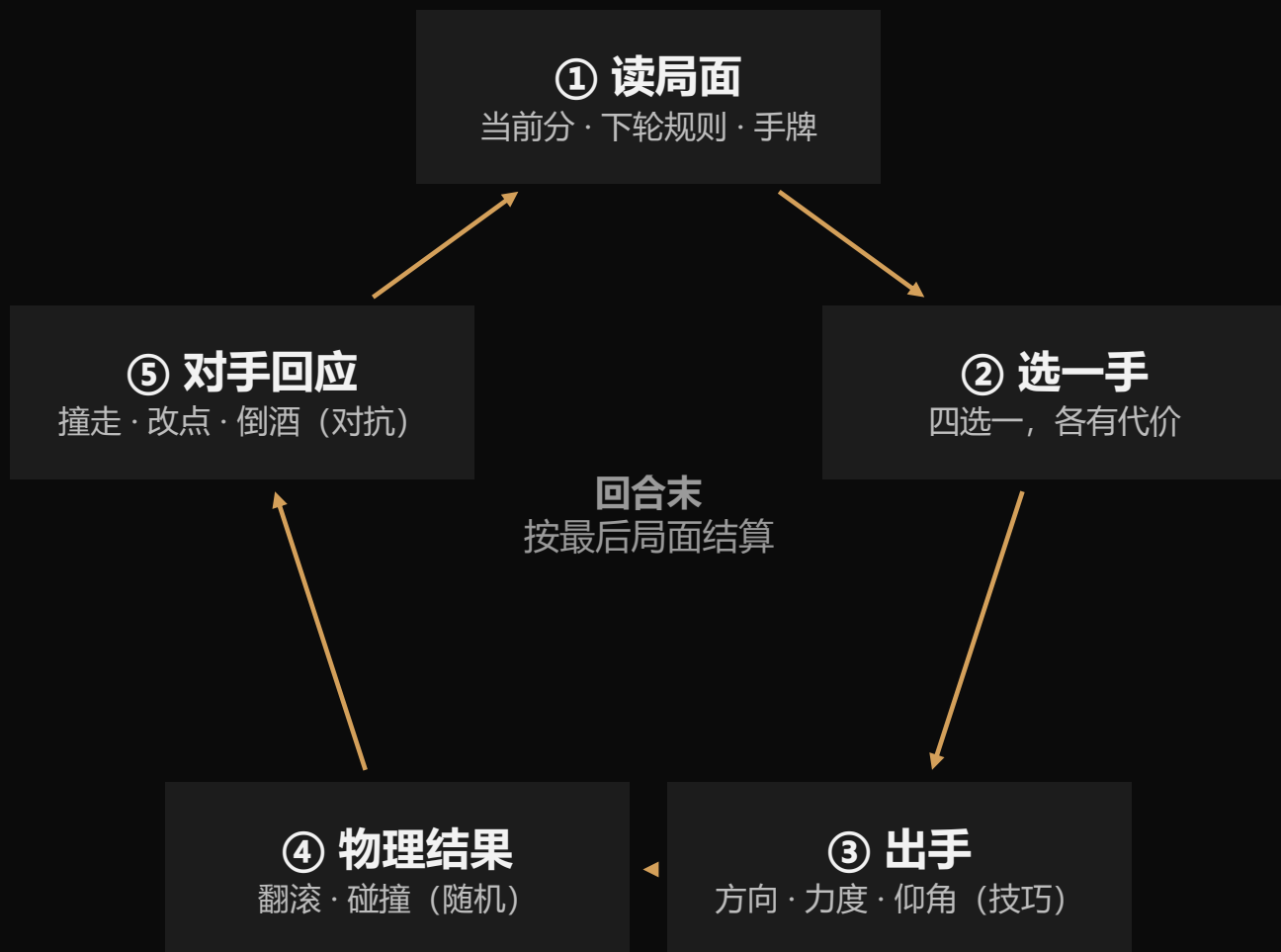


**对调生效**  
蓝骰 1 → 5, Brann 5 → 1



**领奖台舞台** 独立小舞台实时渲染，按名次站位

# 核心循环：读局面 → 选一手 → 出手 → 看结果



## 每一环的设计意图

读局面：信息尽量放在桌上——立卡上的分数、对手面前扣着的牌、下一轮得分区的虚线框。

选一手：四种行动只能选一个，选择本身就是策略。

出手：拖拽决定方向和力度，轨迹只预告到第一次接触。

物理结果：翻滚和最终点数不预告，惊喜留在这里。

对手回应：三位对手按各自性格改写局面，桌边酒客跟着起哄。

骰子留在桌上，所以每一手同时是进攻、防守，也是留给下一个人的局面。

# 四种行动：同一手，四种改变局面的方式



## 投骰（回收重投）

改变什么

位置与点数，由物理决定

代价与限制

拆掉原来的站位；出界骰必须先回收

设计目的

主要得分手段，技巧与运气并存



## 场内拨动

改变什么

小范围挪动己方骰子

代价与限制

同样消耗一手，仍会带动翻滚

设计目的

不拆局面的微调



## 出牌

改变什么

只改点数，位置不变

代价与限制

四张各一次、用后不补；出牌即结束本手

设计目的

给随机结果一个回应



## 倒酒

改变什么

地形：制造低摩擦的酒渍

代价与限制

每局一次；不能盖住破毯、挡板与桌外

设计目的

改变之后所有人的路线

硬规则（试玩中确立）：每一手只能做一件事——出牌或投骰二选一，不额外赠送行动。卡牌和倒酒是有限资源，什么时候用就是策略。

# 三回合计分规则

## 开局

区域内有效点数

不计分

计分方式

只完成投放

## 第 1 回合

区域内有效点数



计分方式

朝上点数  $\times 2$

## 第 2 回合

区域内有效点数



计分方式

朝上点数  $\times 2$

## 第 3 回合

区域内有效点数



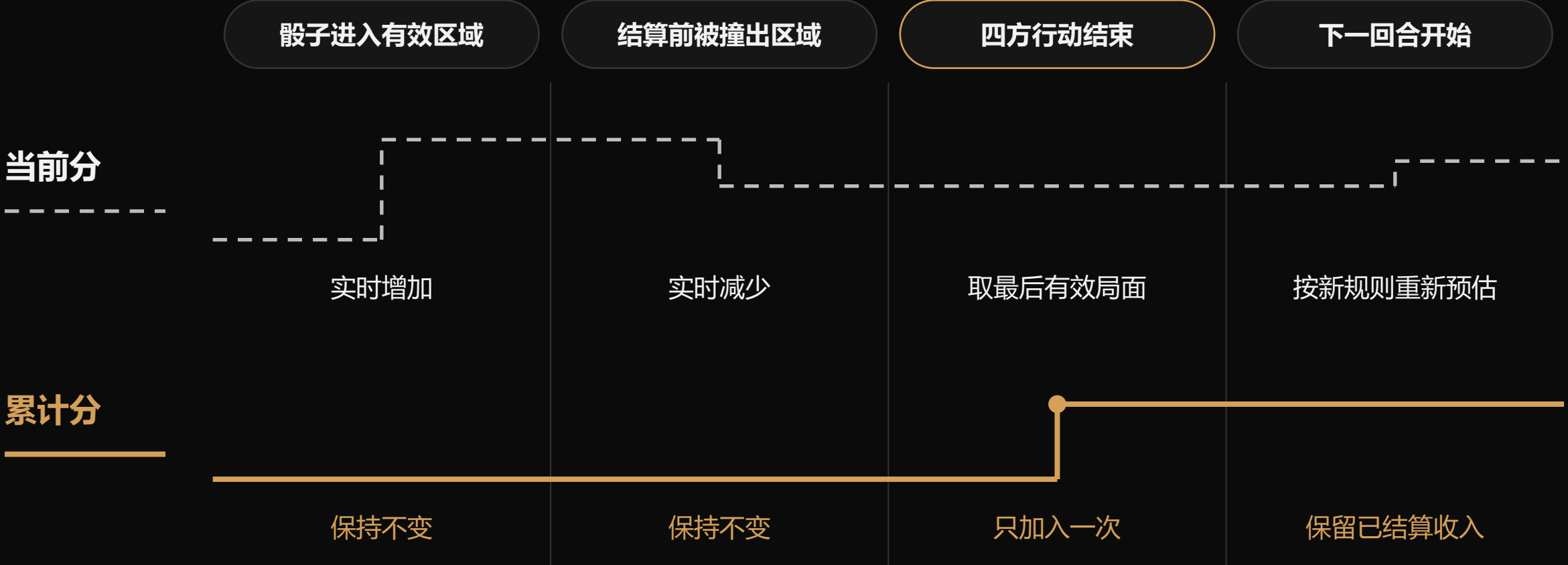
计分方式

朝上点数  $\times 2$

区域外 / 不符合规则 • 全部 • 0 分, 击出不额外加分

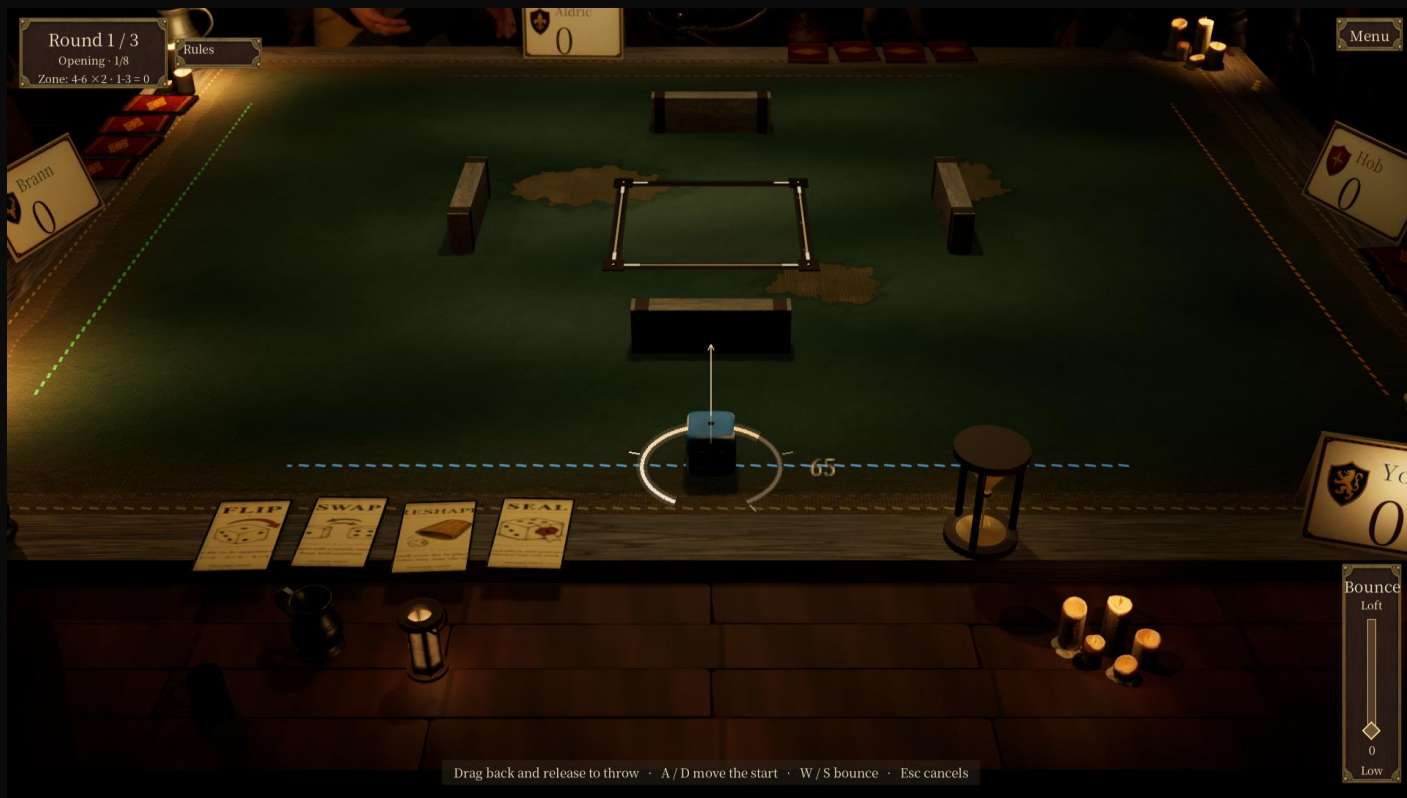
骰子中心落在边界内即视为入区。未投放、出界和手持重投骰子不计分。

# 当前分与累计分



设计目的：进入区域代表暂时优势，直到回合末才成为确定收入。

# 拖拽投骰与力度



座位视角实机：瞄准机位，骰子脚下的铜色力度环

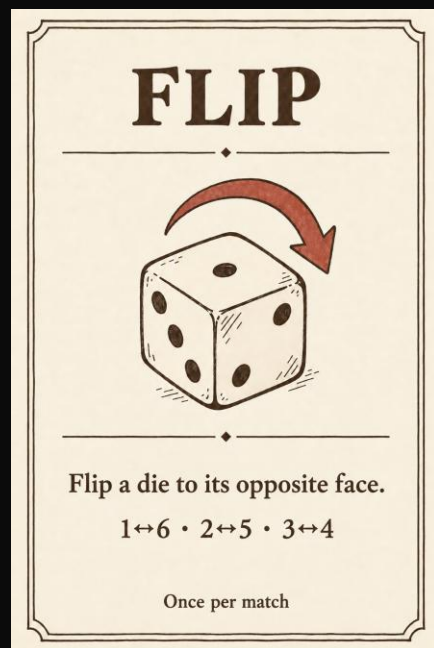
## 拖拽决定方向与力度

按住发射骰向后拖动，松手投出。骰子脚下的透视铜色刻度环与数字显示当前力度，底部力度条淡化为辅助。

## 起点与弹跳可调整

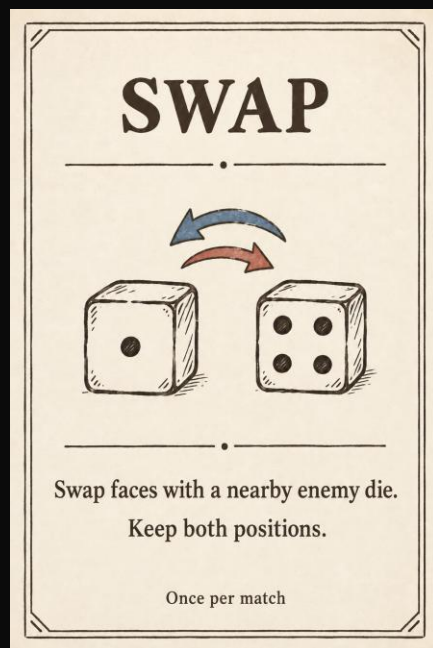
A/D 键移动起点，侧边滑条调整弹跳，取消不消耗行动。  
瞄准时镜头压低到骰子后方，画面下方是你面前的木板边桌；拖拽中镜头冻结。

# 四张有限卡牌



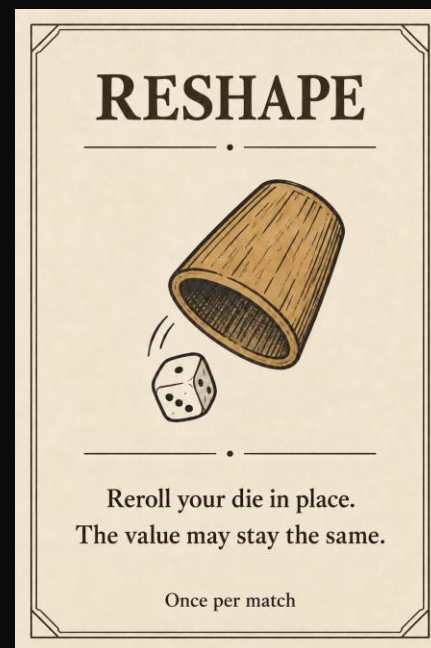
## 翻面

任意未封存骰子变为对面点数



## 对调

己方与 220 单位内敌方交换点数



## 重塑

己方原地随机变为 1-6 点



## 封存

到己方下次行动前免疫卡牌改点

每方各一张，用后不补。出牌结束本方行动，不能随后再投骰。

# 倒酒的选择与限制



早期版本实机：倒酒选点

## 一次行动改变地面

点击近侧酒杯或使用 E 键选点。  
每方每局一次，酒渍保留到比赛结束。

## 反馈先说明是否可用

范围不能覆盖破损、挡板或桌外。非法点击和取消不扣次数，也不消耗行动。

# 04

## 关卡设计、场景迭代与镜头

桌面怎样提供路线选择？  
场景如何从抽象走到酒馆？  
镜头怎样兼顾瞄准与全局？

### 本章内容

- 26 关卡与玩法迭代总览
- 27 普通界面的第一轮迭代
- 28 高度与障碍：增加可选择的路线
- 29 酒馆主题如何改变关卡元素
- 30 桌面布局：从规则摆放到路线设计
- 31 倒酒：把固定地形变成玩家行动
- 32 关卡布局：每块区域如何影响选择
- 33 酒馆环境的迭代
- 34 画面打磨：从调试几何到酒馆里的实物
- 35 座位视角：入座开场与四种机位
- 36 镜头跟随：什么时候动，动多少

# 关卡与玩法迭代总览



早期杯垫、区外基础分与击出奖励，均不等于最终规则。沉浸感改造只改表现层，规则与 AI 决策不变。

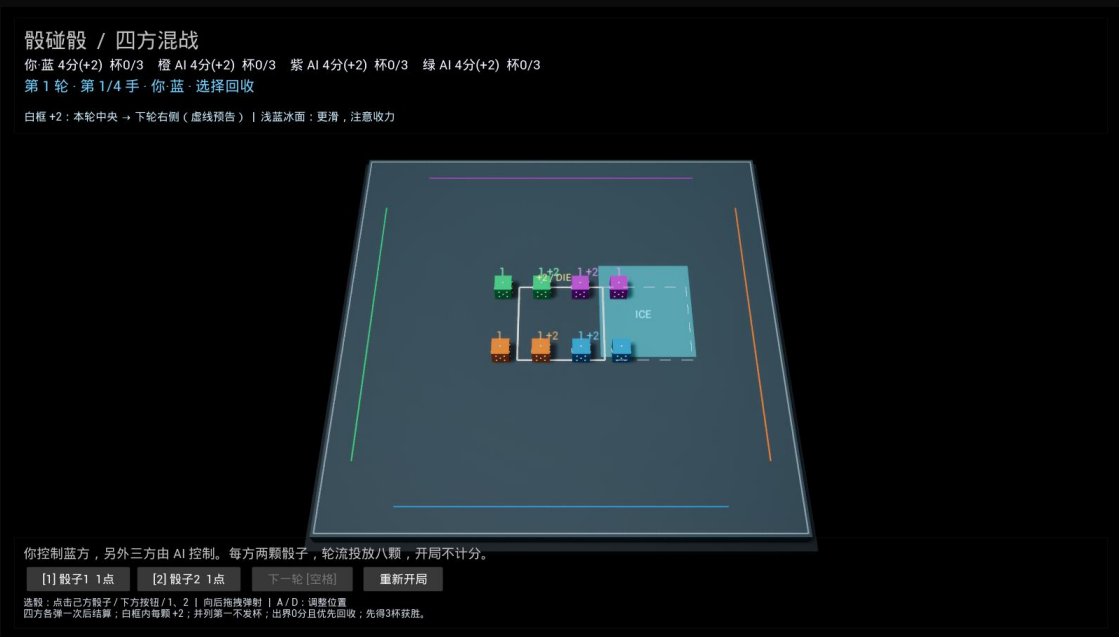
# 普通界面的第一轮迭代

## 两方原型：先验证物理闭环



两方 + 中央加分区，依据早期备份重运行

## 四方原型：加入位置争夺



四方 + 固定冰面，依据早期备份重运行

最初无酒馆，先实现投骰、回收、停稳计分与跨轮保留。试玩反馈“玩法单一、AI太笨”后，增加得分区、三个对手和低摩擦冰面。

# 高度与障碍：增加可选择的路线



## 问题：平面路线重复

原型主要沿桌面直线互撞。玩家提出“高射炮”机制，希望骰子能越过障碍。

## 改动：贴地、绕行与高抛

加入0-65°仰角与首段轨迹预览。冰面移到外围，中央保留争夺空间，高低与奇偶规则交替。

外围冰面与四道矮墙，依据当时的备份重运行

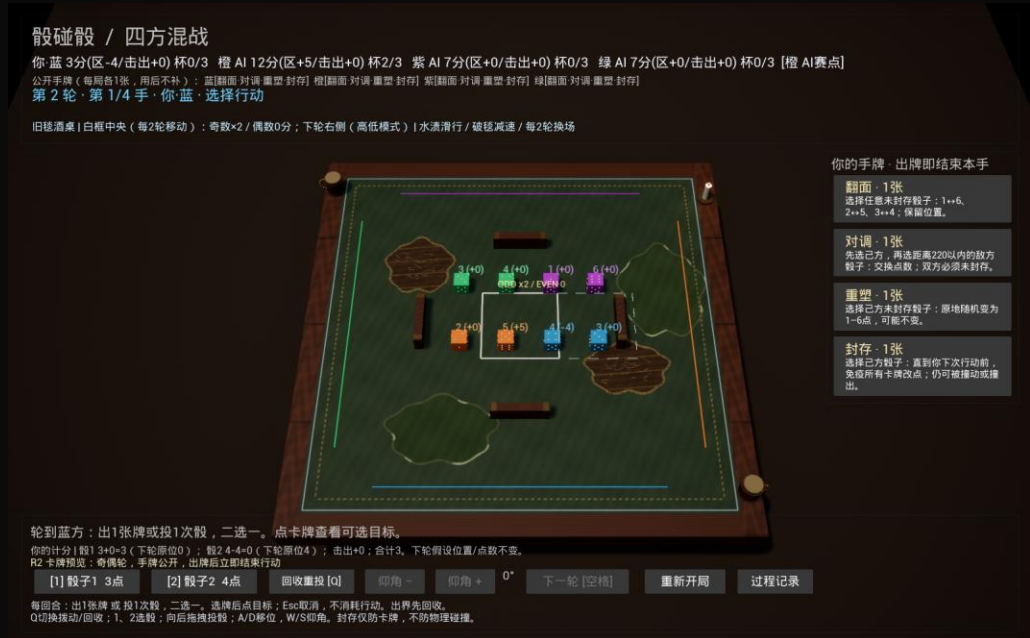
# 酒馆主题如何改变关卡元素

## 抽象地形：冰面与矮墙



同一历史规则备份的重运行画面

## 酒馆桌面：酒渍、破毯与木挡板

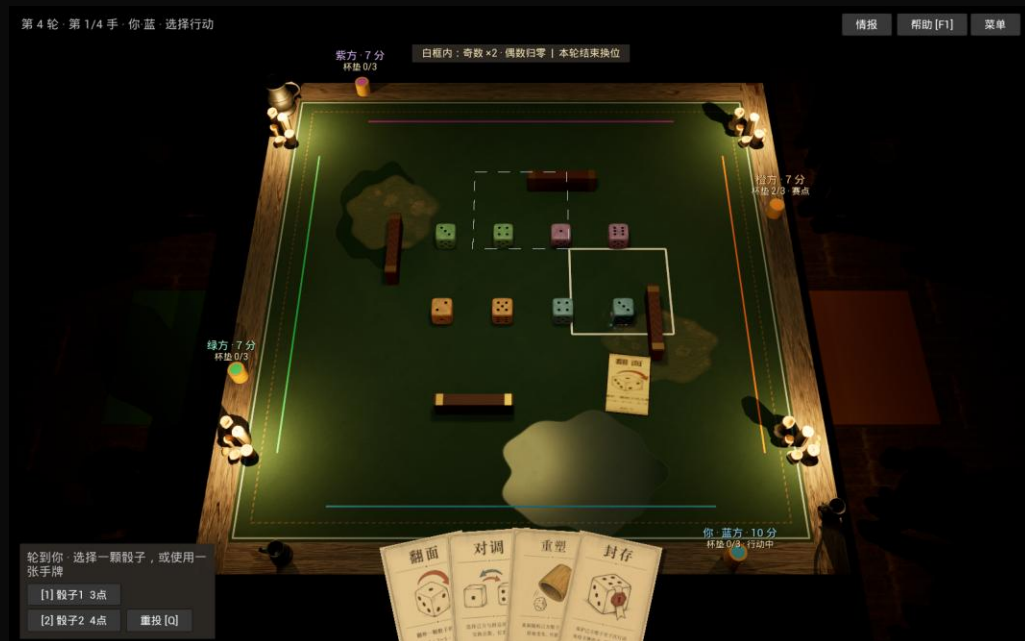


酒馆桌面初版，原始历史截图

保留低摩擦、减速与阻挡的玩法功能，换成可理解酒馆物件。得分区改为两轮后移动，减少每轮重新学习局面的负担。

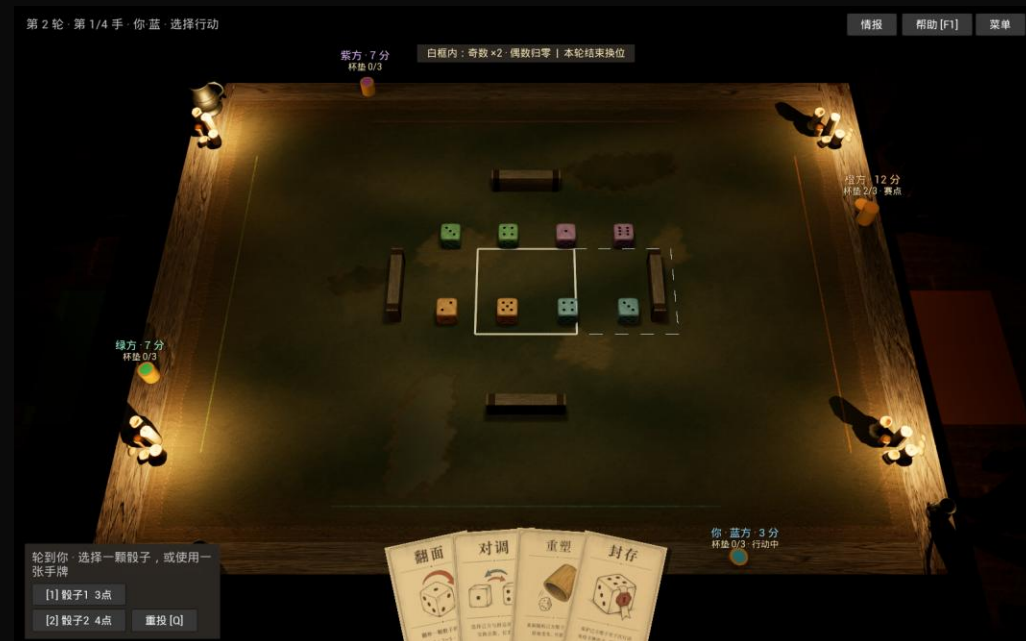
# 桌面布局：从规则摆放到路线设计

## 早期湿区与破损区



历史截图：分布仍偏规整

## 加宽桌面与不对称区域

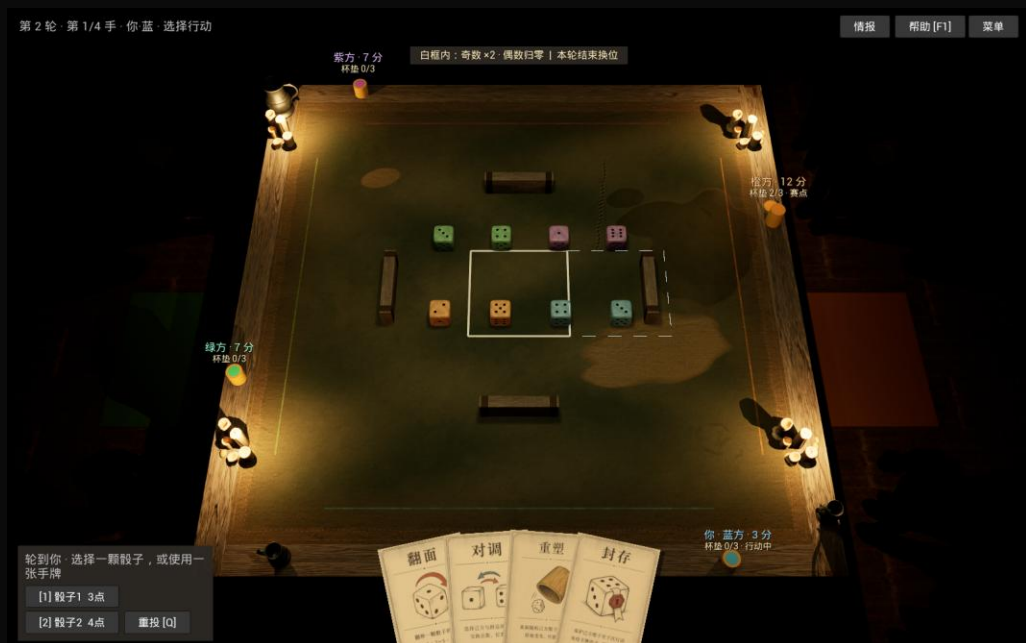


历史截图：依据绕板、冲撞、停骰与占区重排

按反馈将桌面横向加宽1/3，错开区域大小与轮廓，保留普通桌面通道。布局围绕投骰路线与风险选择展开，做旧材质则强化酒馆氛围。

# 倒酒：把固定地形变成玩家行动

之前：预设湿滑区域



历史截图，湿区由场地预设

之后：主动选择倒酒位置



当前英文实机：选点预览与一次行动代价

移除固定湿区，改为每方每局一次倒酒。湿区持续到本局结束，不能覆盖破损处、挡板或桌外；改地形与投骰、出牌竞争同一手行动。

## 关卡布局：每块区域如何影响选择



编号对应当前实机地形；四方从各自桌边发射。

## 1 中央争夺区

符合当轮规则才得分。前两轮居中，第三轮换位，原有站位需要重新评估。

## 2 短挡板

截断直接路线。贴地绕行与高抛越障需要不同的力度和落点。

### 3 破损减速区

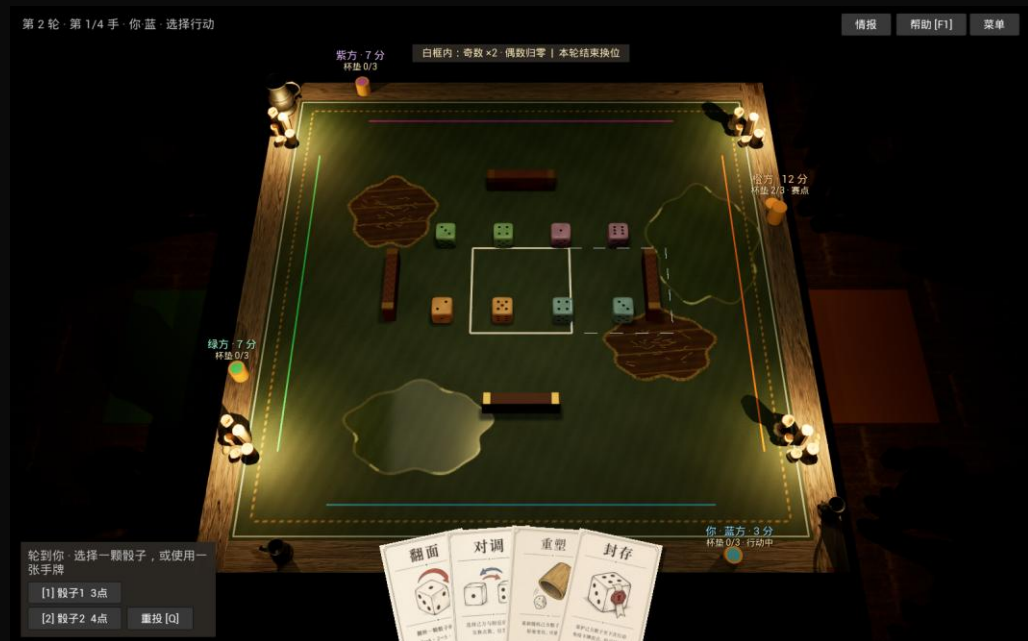
降低滑行距离，提供停骰机会；同一位置不能倒酒。

## 4 普通绕行通道

保留进区与侧向攻击路线。合法位置可倒酒，湿区持续到本局结束。

# 酒馆环境的迭代

## 早期桌面与背景



历史搭建阶段

## 酒馆营业中：酒客、观战者与声音



当前实机俯瞰：吧台、侧桌与桌边共 12 名酒客各忙各的

保留桌面玩法结构，逐步整合木地板、吧台、酒杯、角色与照明；再加入免费动捕驱动的酒客、7 层环境声与每回合轮换的酒馆曲，主桌仍是视觉焦点。

# 画面打磨：从调试几何到酒馆里的实物

瞄准视角 · 改前



座位视角 · 改前



出牌特写 · 改前



瞄准视角 · 现在



座位视角 · 现在



出牌特写 · 现在



问题：得分区框、出手线和光圈是画在屏幕上的线，像调试几何，骰子也挡不住；瞄准时画面下方三分之一是黑的。  
改动：黄铜框、缝线、染色骨骰；标记放到桌面上、被骰子遮挡；瞄准下方是你的边桌；补光、调色、景深、薄雾。

# 座位视角：入座开场与四种机位

## 开场：俯瞰酒馆



入座开场第一帧，当前实机

## 推镜入座，等待开局



镜头落到座位，点击后才进入瞄准

座位（自动取景到最远对手头顶）、瞄准（压低到骰子后方）、跟随（跟最快的骰子，不越过桌心）、结算（原俯视相机）。拖拽时镜头冻结，因为瞄准靠把鼠标反投影到桌面。

# 镜头跟随：什么时候动，动多少



## 01 选择、等待

座位视角，自动取景：自己的桌沿到最远对手的头顶  
像真的坐在桌边，同时看得到三位对手



## 02 瞄准

压低到骰子后方，260 的拖拽范围都在画面内  
拖拽距离就是力度，必须看得见

冻结

跟随

## 03 拖拽中

冻结不动  
瞄准靠鼠标反投影到桌面，镜头一动手感就漂

## 04 出手后

跟最快的骰子，只跟一部分，不越过桌心  
看清碰撞，又不丢掉全局

停 1 秒

## 05 骰子停下

先停 1 秒“看结果”，再转向下一位  
试玩反馈“节奏太快”后加入



## 06 对手回合

留在座位，略转向正在出手的人  
知道谁在出手，但不离开自己的位置



## 07 出牌

拉近目标骰子看它翻面，再回到座位  
看清对手对哪颗骰子做了什么



## 08 回合结算

等演出放完，切回俯视相机  
看全桌：铜币飞进各家硬币堆，再弹出领奖台

所有参数可用 dc.Cam.\* 实时调整；自动检查时不启用新镜头，规则结果不受影响。

# 05

## 策略系统与玩法体验

玩家在什么之间取舍？  
怎样让一手好棋被看懂？

### 本章内容

- 38 策略空间：位置、点数、地形、时机
- 39 被撤回的方案与卡牌的引入
- 40 卡牌选择与目标反馈
- 41 行动顺序与最后一手
- 42 策略案例：最后一手，保位还是撞击
- 43 策略案例：翻面后结算，再看下一轮
- 44 回合结算：等演出放完，再把分数“发”出来
- 45 桌面内的界面：立卡、牌堆、沙漏、力度环

# 策略空间：位置、点数、地形、时机

## 位置·抢、挡、撞

选择：进区占分；用低分骰挡住路线；把对手撞出得分区

代价：进攻可能打散自己的站位；第三回合区域换位

## 点数·用牌改写

选择：翻面、对调、重塑、封存

代价：每张只有一次；出牌就不能投骰；封存挡不住物理撞击

## 地形·倒一杯酒

选择：在对手的必经之路上制造酒渍

代价：每局一次；酒渍对自己同样有效；不能盖住破毯与挡板

## 时机·谁最后出手

选择：最后一手可以在结算前改写局面

代价：先手每回合轮换；先出手的布局要经受三次干扰

规则每回合轮换（4-6 → 奇数 → 4-6 且区域换位）：上一回合的好位置，下一回合可能不再得分，留在桌上的骰子要重新经营。

# 被撤回的方案与卡牌的引入

试过

## 三种地图与场内拨动

玩家希望在原桌上实时变化，而非增加新地图。拨动仍有位移与随机风险，不能充分解决保住好位置后改点的问题。



撤回

## 多地图与直接调点

恢复同一桌面的打磨方向。  
“ $\pm 1$ 调点”停留在讨论阶段，玩家认为过于生硬，没有作为最终玩法落地。



采用

## 有限卡牌与行动二选一

翻面、对调、重塑、封存提供不同干预方式。每张每局一次，出牌或投骰共用一手，不额外赠送行动。

设计重点：在保留站位的同时经营点数，并让“移动”与“改点”产生机会成本。

# 卡牌选择与目标反馈



座位视角实机：轮到你选择时，扇形手牌从桌沿升起

## 桌沿上的牌与扇形手牌

平时你的手牌正面朝上摆在桌沿；轮到你选择时扇形手牌升起，悬停前移。选牌后提示合法目标与技能限制。

## 改点与移动分开

翻面、对调与重塑只改点数，保留位置。封存后仍可被撞动、翻面或击出。

# 行动顺序与最后一手



座位视角实机：对手回合，镜头略转向行动方，桌沿的沙漏已传到他面前

## 轮到谁，看桌沿的沙漏

换人时沙漏被拿起、越过桌面传给下一位，翻过来重新流沙（约 30 秒，只计时不限时）。回合牌写第几手与本回合的计分规则。

## 最后一手影响最终收入

最后一手仍可把对方撞出得分区。出牌演出放完、镜头回到桌面后才统一结算。

# 策略案例：最后一手，保位还是撞击



第二回合最后一手。早期版本实机局部，固定讲解局面。

**读局面：自己6分，对手10分**

区内蓝骰3点得6分，橙骰5点得10分。奇数翻倍，偶数0分。现在轮到蓝方，行动后立即结算。

**比较：移动有风险，改点保留位置**

撞击：可把橙骰移出区，也可能打散自己的站位。

翻面：橙骰5变2，位置不变，当前10分变0分。

封存：无法消除橙方10分，且仍可被物理撞动。

**选择翻面，保住己方6分。**

**代价：消耗一手和本局一次 Flip。**

# 策略案例：翻面后结算，再看下一轮



实际执行 Flip 并完成回合末结算（早期版本的账本界面），固定讲解局面。

## 结果：保位，改变对手收益

橙骰5点变2点，蓝骰仍是3点。两颗骰子的位置不变，蓝方本轮收入6分，橙方收入0分。

| 本轮结算  | 蓝方 | 橙方 |
|-------|----|----|
| 之前累计  | 8  | 12 |
| 本轮收入  | +6 | +0 |
| 结算后累计 | 14 | 12 |

## 下一轮仍需重新布局

蓝方最后行动，收益可立即结算。第三轮区域换位且只计4-6点，留桌骰子需要重新经营。

# 回合结算：等演出放完，再把分数“发”出来



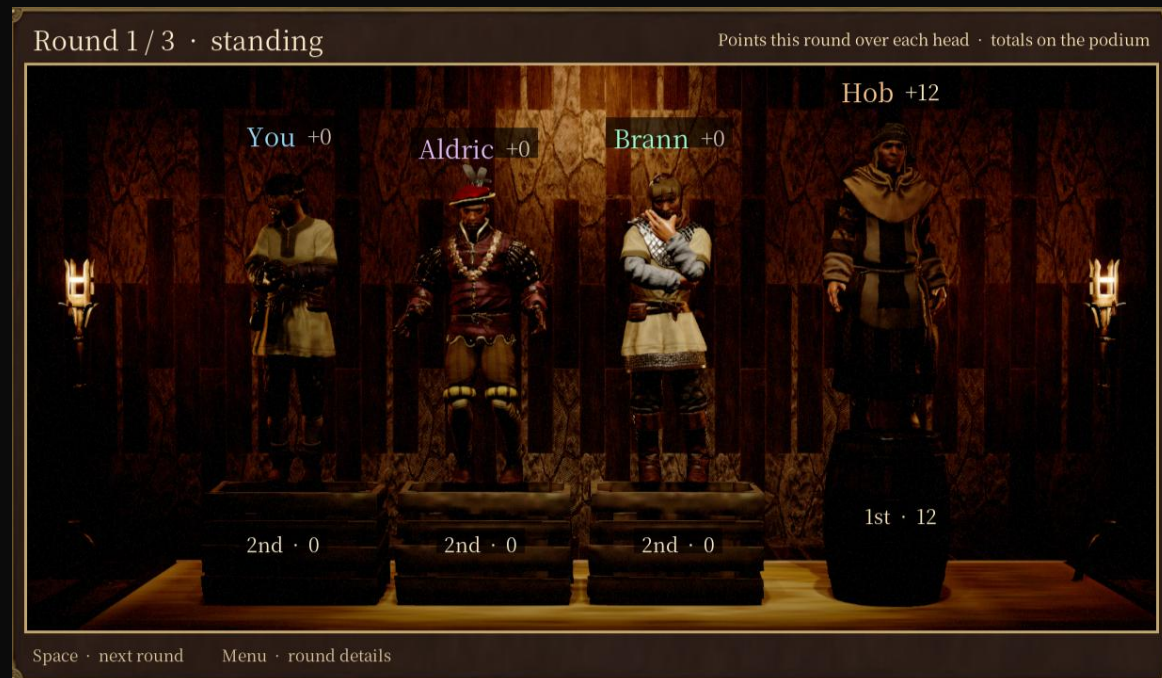
镜头到位后，得分骰子的分数化成铜币，划弧飞向各家的硬币堆



赢家时刻：立卡抬起、脸上的光亮起，对手和观众这时才反应

问题：最后一手的出牌演出还没放完，账本就弹了出来；结算只有一张表。

改动：等演出放完、镜头到位、再停 1.4 秒；计分仍在同一帧入账，演出只决定何时显示。



领奖台：舞台先暗，追光从第 4 名打到第 1 名；四人站在木桶、木箱和地上，第 1 名欢呼、说一句胜利台词，最后一名沮丧。

# 桌面内的界面：立卡、牌堆、沙漏、力度环

## 桌沿立卡、牌堆与沙漏



立卡与桌沿对齐；对手的牌扣着，你的牌正面朝上

## 瞄准时的力度环



骰子脚下的透视铜色刻度环替代小蓝弧

尽量把信息放进场景：分数写在立卡上，手牌数就是面前的牌，轮到谁看桌沿传递的沙漏，力度画在骰子脚下，问候是说话人头旁的气泡。2D 面板只留回合牌与规则提示。

# 06

## 试玩迭代与 AI 角色

试玩反馈如何变成改动？

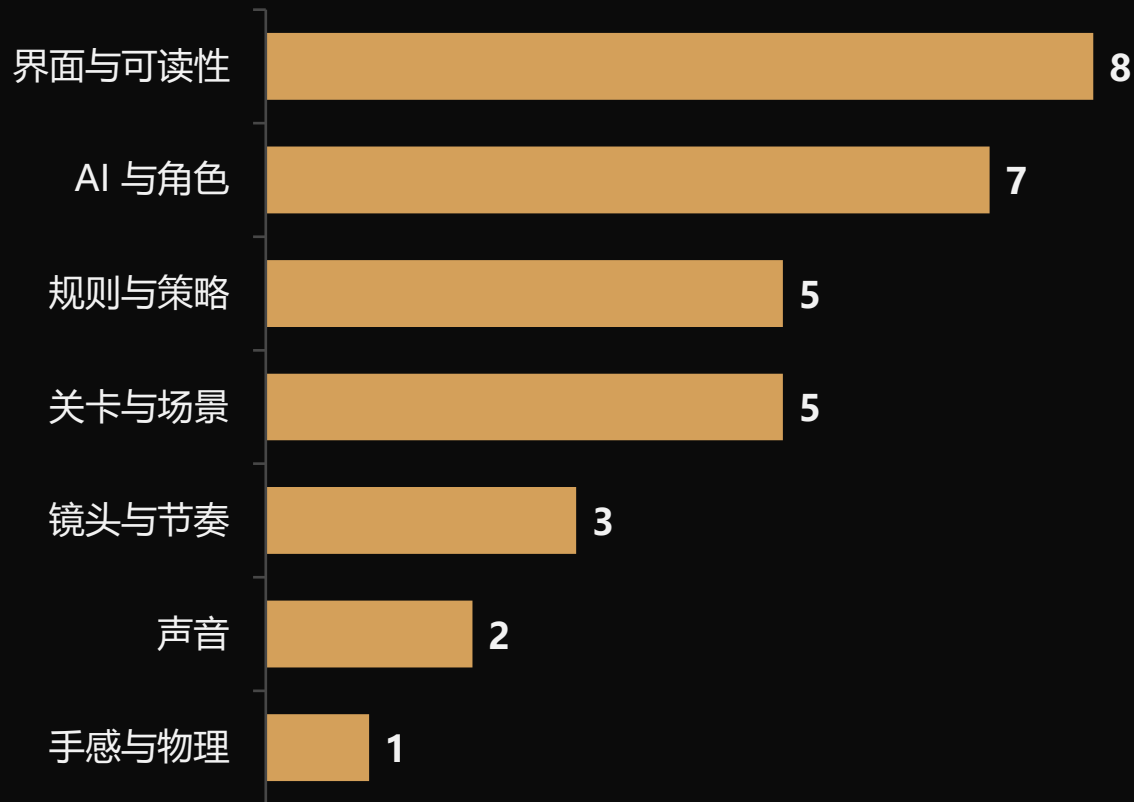
AI 对手怎样既会打、又像人？

### 本章内容

- 47 试玩迭代：31 条反馈，每条都落到了改动
- 48 试玩反馈与优化 ①：玩法与规则
- 49 试玩反馈与优化 ②：场景、界面与反馈
- 50 试玩反馈与优化 ③：界面、镜头与角色
- 51 试玩反馈与优化 ④：画面、结算与玩家
- 52 一次数据驱动的修正：AI 为什么几乎不出牌
- 53 AI 的候选决策
- 54 AI 风格与预测边界
- 55 沉浸感改造：从看棋盘到坐在桌边
- 56 具名对手：名字、性格与开场问候
- 57 对手出手节拍与视线
- 58 出牌演出：升起、化沙、拉近、翻面
- 59 桌边对别人关键时刻的反应
- 60 待机：随局势变化的心情与回合间的小动作
- 61 对手换装：请场景里的酒客上桌
- 62 你也坐在桌边：双手与声音

# 试玩迭代：31 条反馈，每条都落到了改动

反馈按类别（条）



**31 条**

试玩反馈，来自开发记录

**3 轮**

迭代：玩法成形 → 沉浸感改造 → 画面与结算打磨

**5 个**

方案被撤回或替换：多张地图、 $\pm 1$  调点、稳住、杯垫胜利、击出奖励

**11 项**

自动检查：表现层改动后逐行回归，规则结果不变

数据来源：开发对话中项目所有者的实机试玩反馈，以及 Docs 下的文档。目前还没有外部同学试玩数据，见第 7 章“下一步”。

# 试玩反馈与优化 ①：玩法与规则

## 手感与物理

**“用力撞到了也弹不出对手，自己反而出界”**

问题：攻击方承担了大部分动能

改动：降低上抛与旋转、区分骰间反弹：  
85% 力度能撞出目标，自己留桌

## 规则与策略

**“玩法单一，AI 太笨”**

问题：只有互撞，没有目标与路线差异

改动：加入得分区、四方对局与低摩擦冰面

## 关卡与场景

**想让骰子像“高射炮”一样越过障碍**

问题：平面直线投射，路线只有一种

改动：0-65° 仰角、首段轨迹预览与短墙

## 规则与策略

**“杯垫的作用几乎为 0”**

问题：胜利标记不产生决策

改动：先加击出奖励，后改为三回合累计计分

## 规则与策略

**“落下是几就是几，只能取回重投”**

问题：唯一的改点手段会拆掉自己的局面

改动：试过稳住、拨动、多地图后，改为四张  
一次性卡牌

## 规则与策略

**“对抗性要发挥出来，而不是各弄各的”**

问题：各方行动几乎互不影响

改动：硬规则：出牌或投骰二选一；卡牌能改  
对手的点数

## 关卡与场景

**“新加的地图很苍白，没看懂”**

问题：用加内容代替让原桌面变化

改动：撤回多地图；原桌面按回合换区与地形

## 关卡与场景

**“得分区变化太快”**

问题：每回合换位，来不及经营

改动：两个完整回合后才换位

## 关卡与场景

**“冰面改成酒渍，毯子破的地方减速”**

问题：抽象地形与酒馆主题割裂

改动：酒渍 / 破毯 / 木挡板，玩法功能不变

# 试玩反馈与优化 ②：场景、界面与反馈

## 关卡与场景

**“太规整、太对称，像现代赌桌”**

问题：地形按磨损摆放而不是按机制

改动：按路线与风险重新分布、大小不一；做旧材质；桌面加宽 1/3

## 界面与可读性

**“为什么 AI 能调起点，我不能？”**

问题：操作入口不可见

改动：A/D 调起点写进提示，删去多余按钮

## 声音

**“这哪是骰子的声音” “太小声了”**

问题：合成音效缺乏质感

改动：换成 CC0 实录，混音提高约 3 倍

## 界面与可读性

**“卡牌与酒馆割裂，不够做旧”**

问题：视觉语言不统一

改动：泛黄羊皮纸卡面；扇形手牌，悬停前移

## AI 与角色

**“得失分不只要弹 UI，还要音效和角色反应”**

问题：反馈只停在界面层

改动：对手反应动画 + 三套不同声线

## AI 与角色

**“反应太短，切回待机很僵硬”**

问题：时长与过渡不足

改动：动画延长约 1/3、交叉淡出；两套待机轮换

## 界面与可读性

**“UI 内容太多，还挡住了上方的 AI”**

问题：常驻与临时信息没有分层

改动：只常驻行动顺序与规则，其余按需出现

## 界面与可读性

**“根本不知道对手用了什么技能牌”**

问题：出牌展示太小、太快

改动：放大并延长，后升级为桌面出牌演出

## 规则与策略

**“没有骰子在得分区，却有 6 分”**

问题：区外也计基础分，结算时机不清

改动：只计区内、回合末结算、三回合累计

# 试玩反馈与优化 ③：界面、镜头与角色

## 界面与可读性

**“这五个 UI 框太大了”**

问题：信息框挤占桌面

改动：缩小 1/3，重新分配字号与间距

## 镜头与节奏

**“开局直接进入瞄准很突兀”**

问题：缺少进入情境的过渡

改动：入座开场：俯瞰 → 推镜入座 → 问候  
→ 点击开始

## 镜头与节奏

**“思考、投骰、镜头追踪节奏太快”**

问题：骰子停下后没有停顿

改动：先“看结果”1秒再思考；节奏  $\times 1.2$ ；  
镜头转速降低

## AI 与角色

**“对手开局看天花板、乱看”**

问题：视线目标包含其他人的头

改动：视线只看桌上的骰子与得分区

## AI 与角色

**“AI 几乎不用卡牌”**

问题：投骰预测收益只兑现约 27%

改动：按置信度 0.45 折算后再比较：出牌  
8/54 → 24/54

## 声音

**“听不到环境声”**

问题：素材本身很轻，叠加距离衰减

改动：按实测响度分层重配增益

## AI 与角色

**“酒杯和手的位置不对”**

问题：动捕手型与杯子不匹配

改动：每帧按手指弯曲拟合握杯

## AI 与角色

**“旧对手的服装很丑”**

问题：角色缺少辨识度

改动：换成场景里辨识度最高的酒客造型

# 试玩反馈与优化 ④：画面、结算与玩家

## 界面与可读性

### “骰子要合理遮挡边界”

问题：得分区框与光圈是画在屏幕上的叠层

改动：改成桌面上的发光物体，骰子自然挡住

## 界面与可读性

### “沙漏做出真正的流沙效果计时”

问题：回合顺序只是顶部的一条 2D 条

改动：桌沿沙漏：上沙变少、下沙堆高，换人时翻转传递

## 镜头与节奏

### “结算窗口出现得太早”

问题：最后一手的出牌演出还没放完就弹出账本

改动：等演出放完、镜头到位、再停 1.4 秒才出现

## 界面与可读性

### “每回合结算后来一段结算动画”

问题：结算只有一张账本表格

改动：铜币从得分骰子飞进各家硬币堆，再上领奖台

## AI 与角色

### “领奖台要把角色实地放上去，玩家只露双手”

问题：你在桌边没有身体，也没有声音

改动：真人领奖台弹窗；你的双手搭在桌沿、会反应、有配音

# 一次数据驱动的修正：AI 为什么几乎不出牌

## 现象

实际对局中，对手出手 8 次、出牌 0 次；6 局自动对局里也只有 8/54 个对手回合出牌。

## 诊断

AI 拿“最佳一投”的预测和出牌比较。加日志后发现：46 次投骰平均预测提升 16.8，实际只有 4.5，约 27%——乐观的预测总是压过确定的卡牌效果。

## 修正

比较前把投骰的预测提升乘以置信度 0.45；激进的 Aldric  $\times 1.25$ ，保守的 Brann  $\times 0.85$ 。规则与评分公式不变。

## 结果

6 局自动对局中对手出牌 24/54，三位对手都会出牌；重塑、封存仍少见，留作下一步。

0 / 8

改动前：实际对局中对手出手 8 次，出牌 0 次

27%

投骰预测提升的实际兑现率（46 次投骰）

8  $\rightarrow$  24

6 局自动对局、54 个对手回合中的出牌次数（15%  $\rightarrow$  44%）

这是沉浸感改造中唯一一次有意修改 AI 决策；出牌频率可用 `dc.AI.ShotConfidence` 实时调整。

# AI 的候选决策

## 合法行动

—— 优先处理出界，排除敌方骰子和不可用卡牌目标

## 路线采样

—— 比较五条起点、多个目标、四档初始仰角与六档力度

## 候选模拟

—— 近似预测重力、表面摩擦、短墙、碰撞与出界

## 方案比较

—— 权衡自身得分、对手威胁、出界风险与卡牌消耗

## 精调与执行

—— 细调较优方案，由真实 Chaos 物理执行投骰

### 出牌决策

投骰预测收益实测只兑现约 27%，先按置信度 0.45 折算（紫  $\times 1.25$ 、绿  $\times 0.85$ ）再与卡牌效果比较

候选计算分帧执行，目标预算约每帧 4 毫秒。出牌频率的调整是本轮唯一有意改动 AI 决策的地方（六局中对手出牌 8/54  $\rightarrow$  24/54）。

# AI 风格与预测边界



Hob (橙) 居中



Aldric (紫) 更冒险



Brann (绿) 更保守

阵营差异：名字与性格只改显示和演出

## 局势倾向

领先时更重视保分。落后和末回合威胁提高冒险倾向

## 随机结果

未知翻滚按期望估值，重塑不预知实际随机点数

## 物理限制

简化模拟不包含精确立方体接触与完整角速度

## 倒酒策略

预测读取湿区，当前没有 AI 主动倒酒策略

AI 不修改真实骰子的运动结果。难度、胜率和四方平衡仍需要真人试玩。

# 沉浸感改造：从看棋盘到坐在桌边



改造只改表现层：计分、物理、AI 决策一行未改，每项都可关闭；说明见 Docs 下各文档。

# 具名对手：名字、性格与开场问候



座位视角实机：入座后对手依次问候，台词以羊皮纸气泡出现在说话人头旁

## 三个人，三种打法

Hob（橙）慢性子老熟客，控区。  
Aldric（紫）押大注的布商，激进。  
Brann（绿）退役佣兵，保守。  
性格按已有 AI 风格写，AI 本身不改。

## 开场问候

各说一句，用原配音演员的语音：“You needa relax.” “Coin favours the bold.” “Held worse lines than this.” 最后你抬手回一句 “Well met, friend.”，随时可点击跳过。

# 对手出手节拍与视线



座位视角实机：轮到对手时镜头略转向行动方

## 思考、准备、出手

骰子停下后先看结果 1 秒再思考；按性格 1.3–3.3 秒的思考后，骰子划 0.5 秒弧线落到发射点，对手前倾、看向出手方向，在手势节拍上发射。

## 视线只看桌上的东西

脖子 40%、头 60% 转向目标：出牌目标 > 滚动的骰子 > 得分区 > 自己的骰子。三人反应延迟不同，不会同步转头。确定性检查保证节拍开关不改 AI 决策。

# 出牌演出：升起、化沙、拉近、翻面



卡牌从牌堆升起，放大浮到桌面中央，字幕说明谁出了什么



沿发光边缘化作沙粒飘散



镜头拉近目标骰子，翻到新点数 (5 → 4)



回到座位，轮到下一方

对手与玩家同一套演出，约 5.1 秒 (dc.Card.Pace 可调)。只转可见的外壳 VisualShell，计分用的 PipRoot 出牌时已是新点数。四层音效：滑出、小风铃、细沙、骰子落定。

# 桌边对别人关键时刻的反应

击飞·Aldric 探身指着笑



击飞·Hob 双手一抬



大比分·Aldric 酸溜溜地甩手



赢下一回合·左侧观战者鼓掌



大比分·Brann 勉强拍两下



你赢下整局·Aldric 一摊手



每次行动落定后按优先级取一个时刻（赢整局 > 赢回合 > 撞飞骰子 > 一手  $\geq 4$  分）；不在事件里的人各自按性格概率反应、彼此错开零点几秒；每次最多一句该角色的配音，只在没人说话时播。动作全部来自自己导入的群众反应动捕。

# 待机：随局势变化的心情与回合间的小动作

Hob · 平局：放松站着



Hob · 小动作：伸手挠帽子底下



Aldric · 领先：仰头笑



Aldric · 领先：侧头打量桌面



Brann · 落后：低头生闷气



Brann · 小动作：挠头



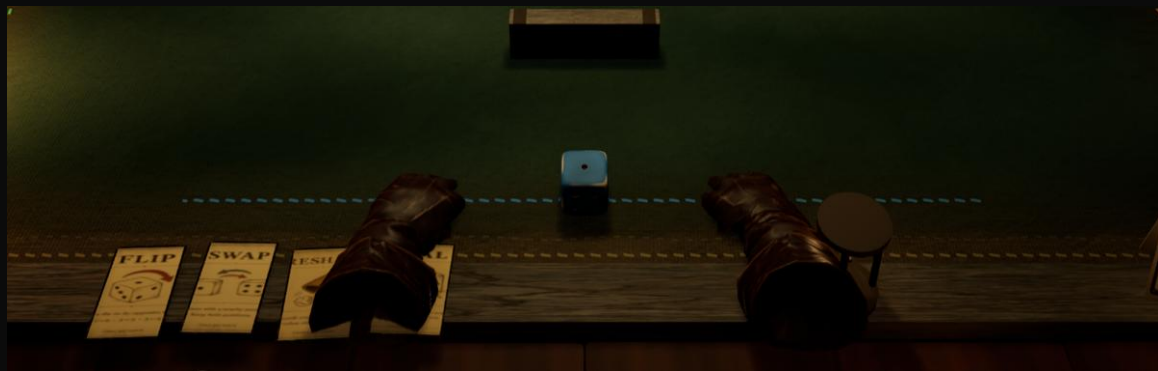
平局 / 领先 ( $\geq 4$  分) / 落后 ( $\geq 6$  分) 三种心情，各有 1-2 段长待机，随机取 12-20 秒一截交叉淡入；没轮到自己时每隔一阵做一个小动作（挠脖子、比划、环顾），一有事 0.35 秒收回。

# 对手换装：请场景里的酒客上桌



上：改前 / 改后座位视角；下：Hob ← 侧桌长袍人（草帽从座位看会挡脸，换成窄檐布帽），Aldric ← 红羽毛帽 + 红宝石链饰，Brann ← 皮帽兜 + 鳞甲老兵。新衣服是跟随部件，复制原部件的骨骼姿势，动画与语音不用重挂。

# 你也坐在桌边：双手与声音



平时：双手搭在面前的桌沿，手指微弯，随呼吸起伏



开场：三位对手问候完，你抬手回一句 “Well met, friend.”

座位镜头就是你的眼睛，所以只画双手：同尺寸的身体只显示手臂，两节骨骼 IK 把手放到桌沿，肘部落到桌沿以下。瞄准时双手不进画面；配音是第四位演员。



赢下回合：抬手鼓掌，喊一声 “YEAH!”



俯视镜镜头里，你和三位对手一样站在桌边

# 07

## 验证与反思

验证了什么？  
还缺什么？下一步做什么？

### 本章内容

- 64 自测与验证范围
- 65 反思：做对了什么，还缺什么
- 66 工作范围与素材来源
- 67 成果

# 自测与验证范围

三回合计分

252

项通过

区域、最后局面、重复结算、并列与重置

可读性状态

175

项通过

行动进度、悬停原因、快照与排名

分数过程反馈

38

项通过

当前分与累计隔离、变化反馈与清理

卡牌与布局

通过

卡牌检查通过，720p / 1080p 目检  
规则路径与画面布局

NPC 节拍一致性

6

项通过

节拍开 / 关时 AI 出牌与出牌的决策签名  
一致

表现层回归

11

项检查逐行一致

每项演出都有开关；关闭时结果与改动前  
相同，Smoke 通过

自动检查不能证明手感、难度与胜率平衡。当前未完成独立打包和专项性能验收。

# 反思：做对了什么，还缺什么

## 做对了什么

- 把随机当设计对象：分四层处理，而不是消除它
- 从试玩问题出发，先做最小版本；多地图、 $\pm 1$  调点、稳住不合适就撤回
- 世界观与规则同源：酒馆物件就是地形，对手性格就是 AI 风格
- 演出全部带开关，11 项自动检查逐行一致，表现层不改规则

## 还不够的地方

- 只有设计者本人的试玩：难度、胜率与四方平衡没有外部数据
- 四方混战里“我进攻、第三方获利”的问题尚未实测
- AI 不会主动倒酒；重塑、封存很少被使用
- 演出拉长了每一手和每回合结算的时长，节奏需要真人检验

## 下一步

- 同学试玩：能否理解“四手后才结算”、能否认出对手出的牌、二选一的理由、首局用时
- 记录出手顺位与胜率，再决定是否调整先手或计分
- 补 AI 倒酒策略，调整重塑与封存的评分
- 骰子放回主人面前；打包与性能验收（领奖台多一次场景渲染）

最大的收获：先找到“玩家为什么不爽”的根因，再用最小的规则改动去回应，而不是继续堆内容。

# 工作范围与素材来源

## 设计与迭代

整理玩法目标、桌面路线、卡牌取舍、计分规则及可读性改进。项目所有者确认设计方向，AI 工具辅助实现与资料整理。

## 美术与声音

环境与角色使用 Nordic Tavern、Modular Medieval NPC 等现有素材（领奖台的木桶、木箱、火把架同属 Nordic Tavern）；酒客与反应动作来自 Anderson Rohr 免费动捕，对手待机来自对白动捕包。卡牌与 UI 装饰包含 ImageGen 生成内容。

## 可核对的来源

ambientCG 纹理、RandomMind 音乐、Kenney 与 Freesound 音效（含落币声），环境声与出牌音效为 CC0。VoiceBosch 为 CC BY-SA 4.0，玩家配音 theuncertainman 为 CC BY 4.0，其他已记录声音为 CC0。

模型资产的购买来源与发布权限需在正式公开作品集 / 试玩包前补齐。此处不主张全部素材为原创。

# 成果



座位视角实机：当前版本

## 已形成可运行的玩法原型

四方投骰、有限卡牌、倒酒、三回合累计与结算演出接入同一对局流程；座位视角、你的双手、具名对手、酒客与声音让它像一局在酒馆里进行的牌局。

# 附录

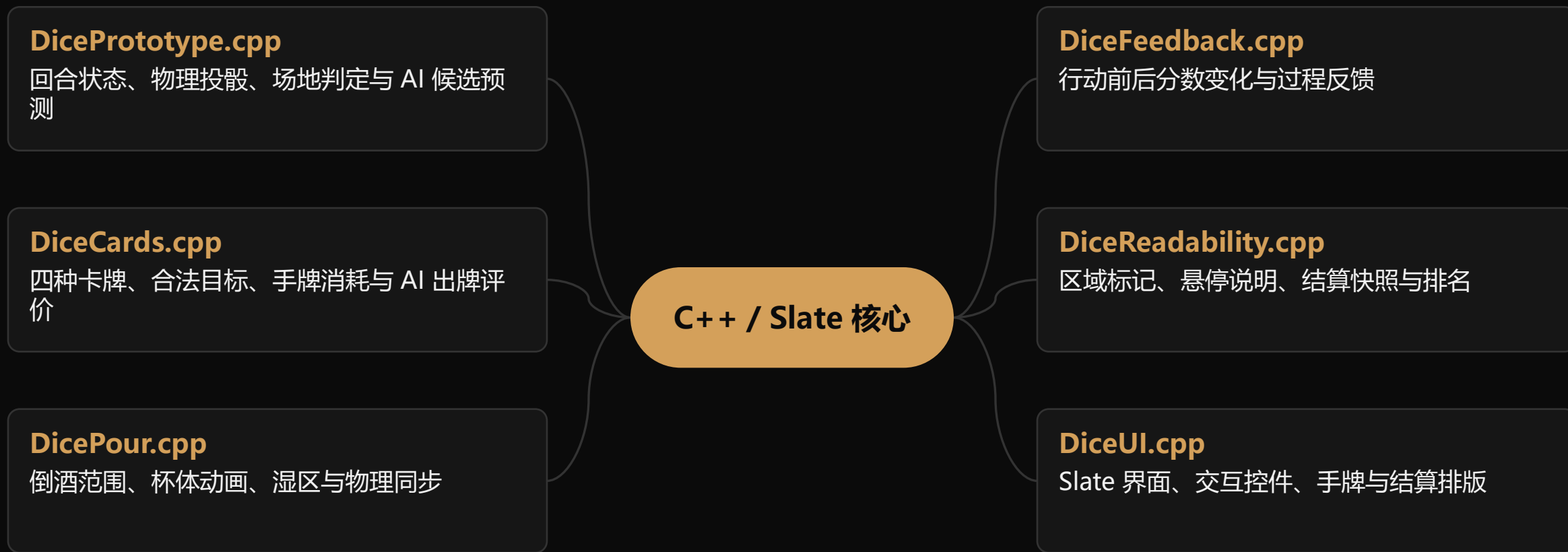
## 技术实现与蓝图复现

核心逻辑在 C++ 中怎样组织；  
再用蓝图节点复现一遍，  
方便不读代码的人对照。

### 本章内容

- 69 核心实现与模块分工
- 70 沉浸感改造新增的模块
- 71 蓝图实现：用节点复现核心逻辑
- 72 蓝图图 ×19：骰子规则、投骰、AI 决策、镜头、出牌演出、换装、待机、桌边反应

# 核心实现与模块分工



项目核心使用 C++ 与 Slate；沉浸感改造新增的模块见下一页。之后的蓝图图用节点复现核心逻辑，方便对照阅读；运行时仍是 C++。

# 沉浸感改造新增的模块

## **DiceCameraRig / DiceIntro**

座位视角机位与平滑、入座开场推镜、拖拽冻结

## **DiceNPCBeats**

对手思考 / 准备 / 出手节拍，视线与前倾，节拍一致性检查

## **DiceAudioRoom**

声音分类混音、房间混响、环境声层、音乐轮换与压低

## **DiceCardShow / DiceCardPiles**

出牌演出（浮现、沙化、翻面）与对手牌堆

## **DiceSeatPlacards / DiceCharacters**

桌沿立卡；对手名字、性格与开场问候

## **DiceSpectators / DiceOpponentLooks**

酒客拼装、动捕偏移修正、握杯拟合；对手换装（跟随部件）

## **DiceNPCIdles / DiceAudience**

按局势的待机与小动作；桌边对关键时刻的反应与语音

## **DiceTableLook / DiceTurnGlass**

桌面材质、前景边桌、桌面上的发光标记；传递的流沙沙漏

## **DiceAtmosphere / DiceSettleShow**

脸部补光、调色、景深、薄雾；结算时铜币飞进硬币堆

## **DicePodiumStage / DicePlayerHands**

真人领奖台弹窗（场景捕获）；你的双手（IK）、反应与配音

全部是 AClashGameMode 的成员，按主题拆成 .cpp 文件；每个模块只加表现层调用，自动检查时不启用。

# 蓝图实现：用节点复现核心逻辑



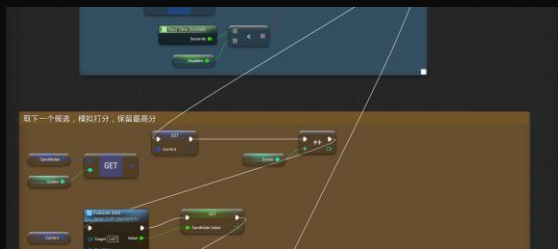
## BP\_DiceRules

DicePrototype.cpp · 骰子规则  
TopFace、DieContribution



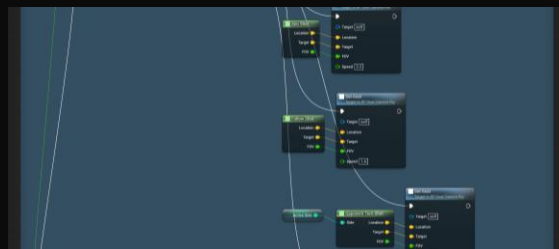
## BP\_DiceLauncher

DicePrototype.cpp · 投骰  
ShotVelocity、Launch、轨迹预览



## BP\_OpponentAI、 S\_ShotCandidate

DicePrototype.cpp · AI (+ DiceCards)  
候选生成与展开、分帧评估、出牌决策



## BP\_SeatCameraRig、 E\_CameraShot

DiceCameraRig.cpp  
机位切换、BlendToGoal、AimShot



## BP\_CardShow

DiceCardShow.cpp  
出牌演出时钟



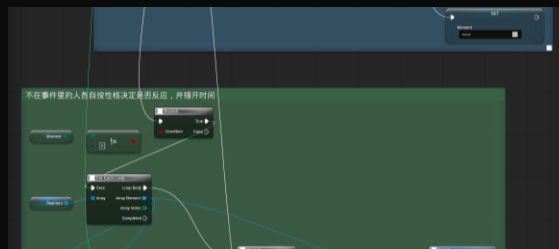
## BP\_OpponentLooks

DiceOpponentLooks.cpp  
ApplyOutfit



## BP\_NPCIdleDirector、 E\_IdleMood

DiceNPCIdles.cpp  
UpdateMood、小动作与打断

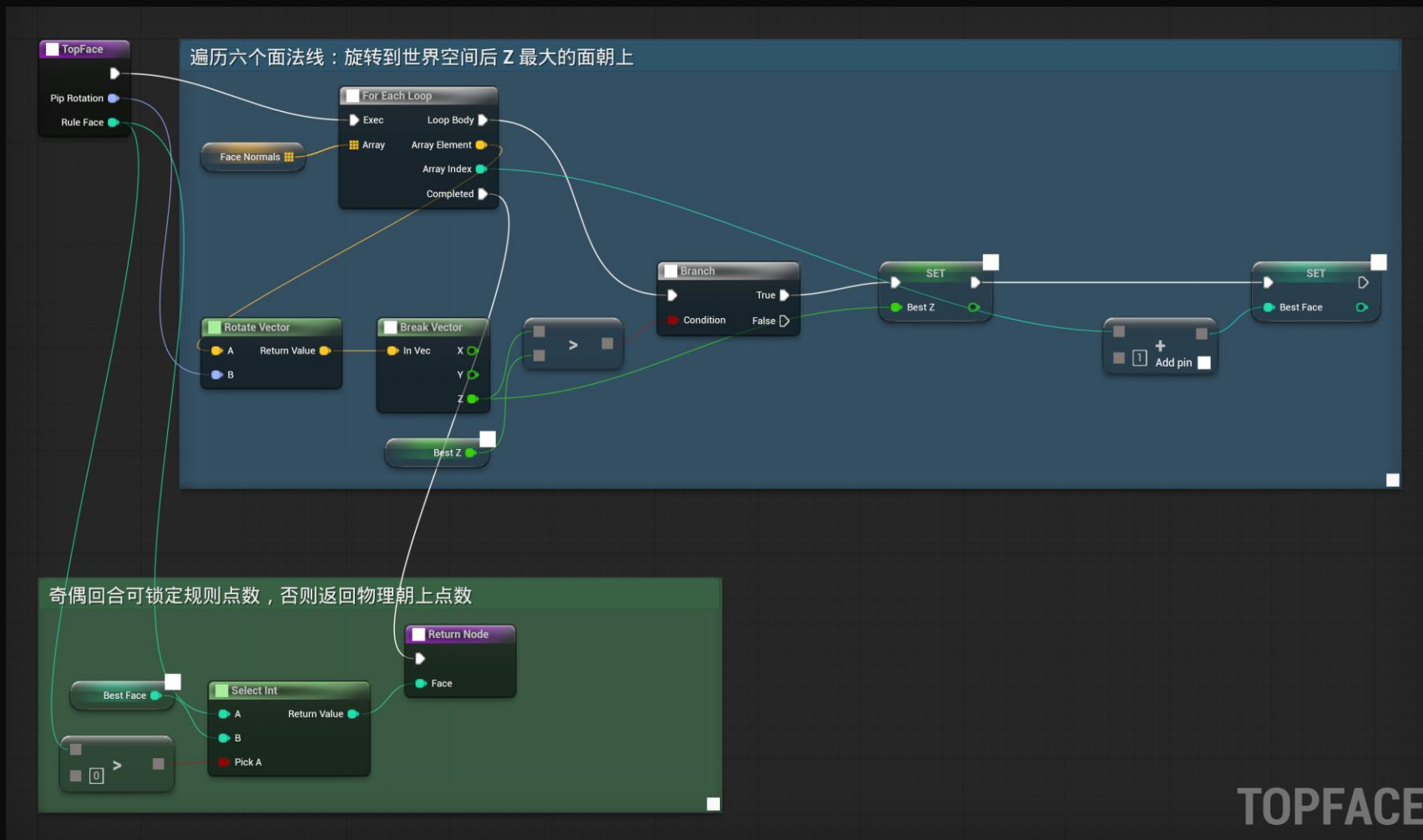


## BP\_TableReactor、 BP\_TableAudience

DiceAudience.cpp  
OnActionSettled、CanReact、React

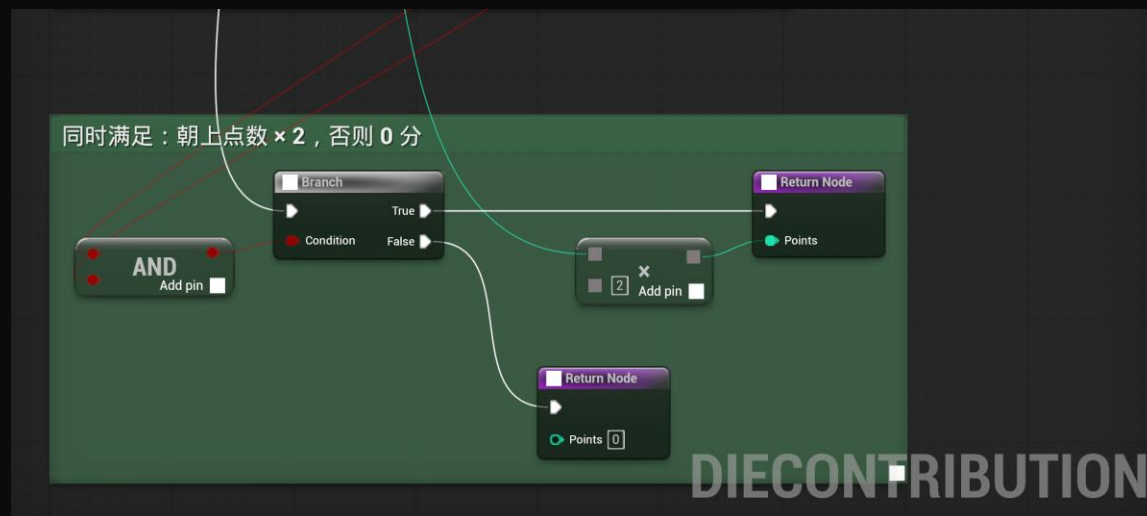
蓝图由脚本在编辑器中生成并编译通过，可在 /Game/DiceClash/BlueprintShowcase 打开查看；运行时仍使用 C++，蓝图只用于展示与对照。

# 蓝图实现：骰子朝上点数 (TopFace)



对应 C++: DicePrototype.cpp · AClashDie::TopFace。六个面法线旋转到世界空间，Z 最大的面朝上；奇偶回合可锁定规则点数。

# 蓝图实现：单颗骰子的贡献分 (DieContribution)



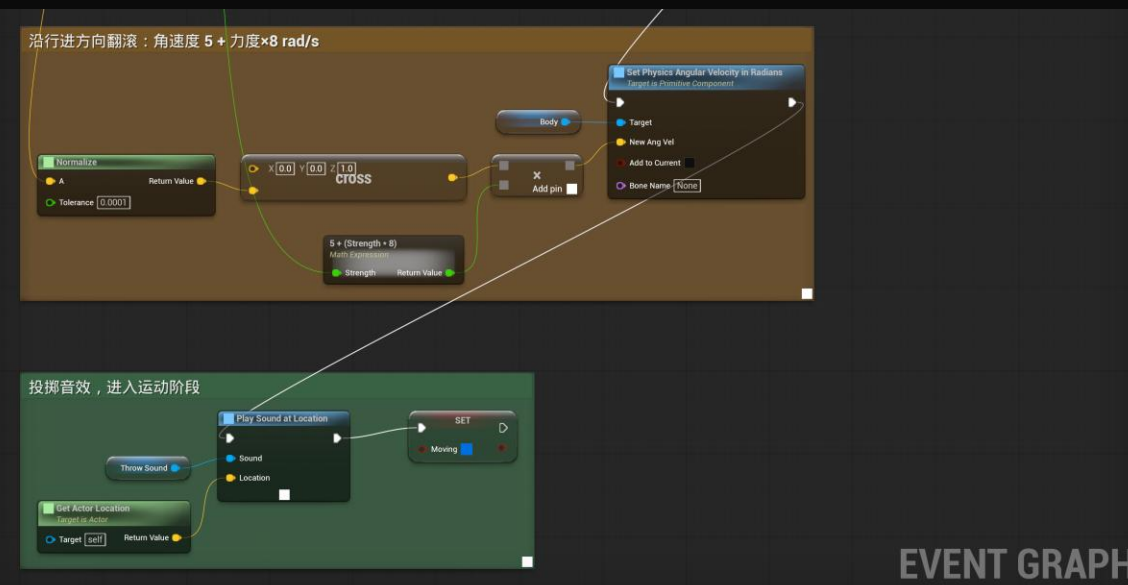
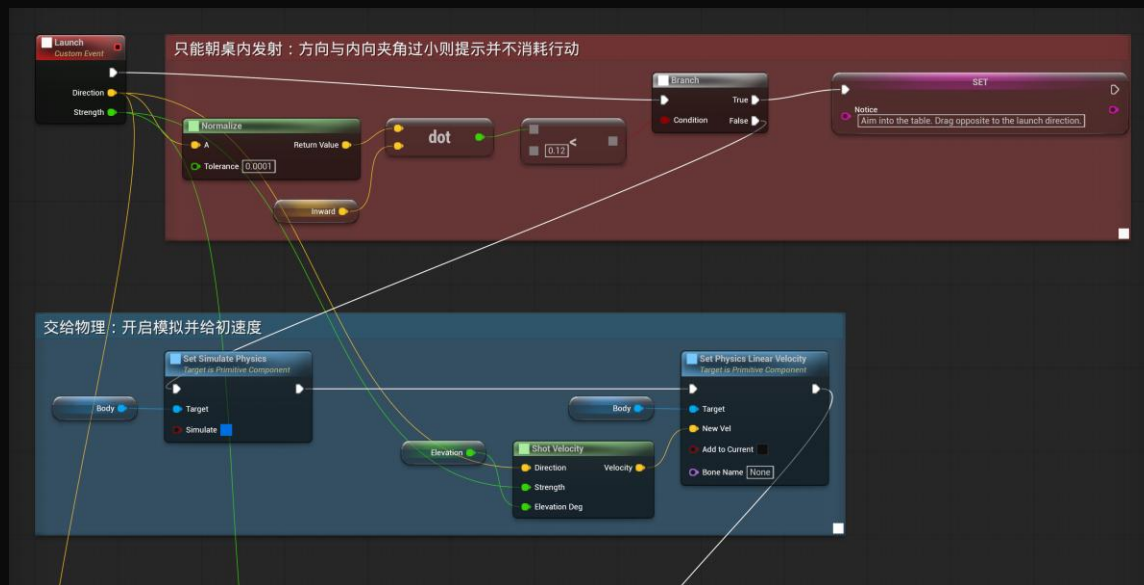
对应 C++: DicePrototype.cpp · AClashGameMode::DieContribution。骰子中心在得分区内且点数符合当轮规则，才按朝上点数  $\times 2$  计分。

# 蓝图实现：投骰初速度 (ShotVelocity)



对应 C++：DicePrototype.cpp · AClashGameMode::ShotVelocity。力度 0-1 映射到速度 180-1050，再按仰角拆成水平与竖直分量。

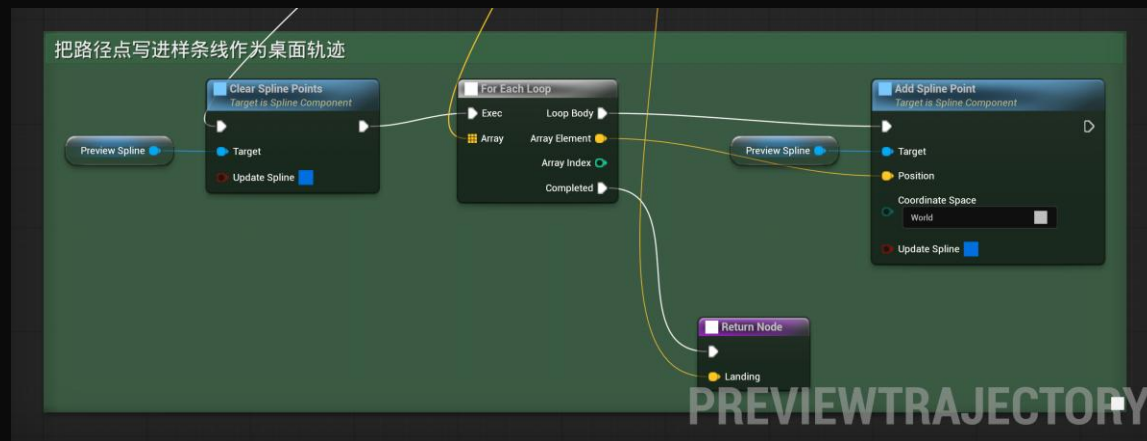
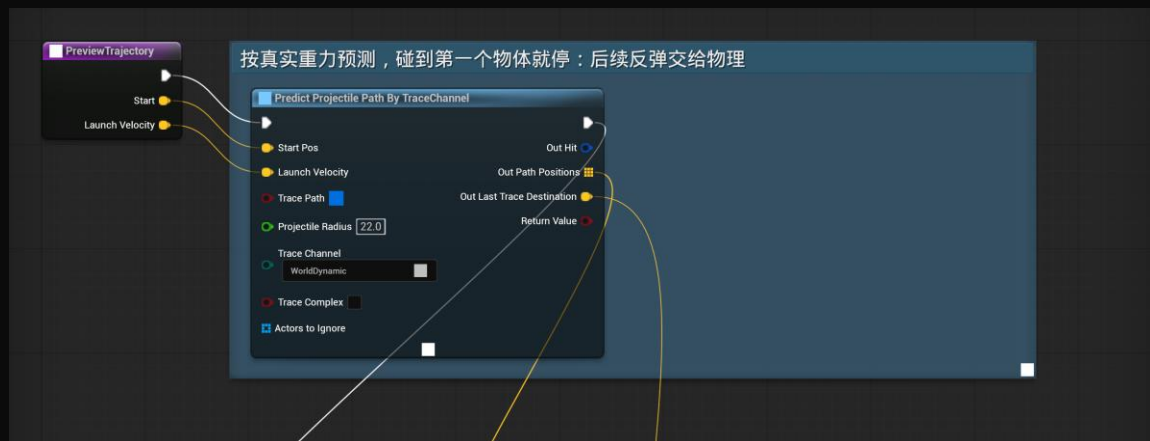
# 蓝图实现：发射事件 (Launch)



EVENT GRAPH

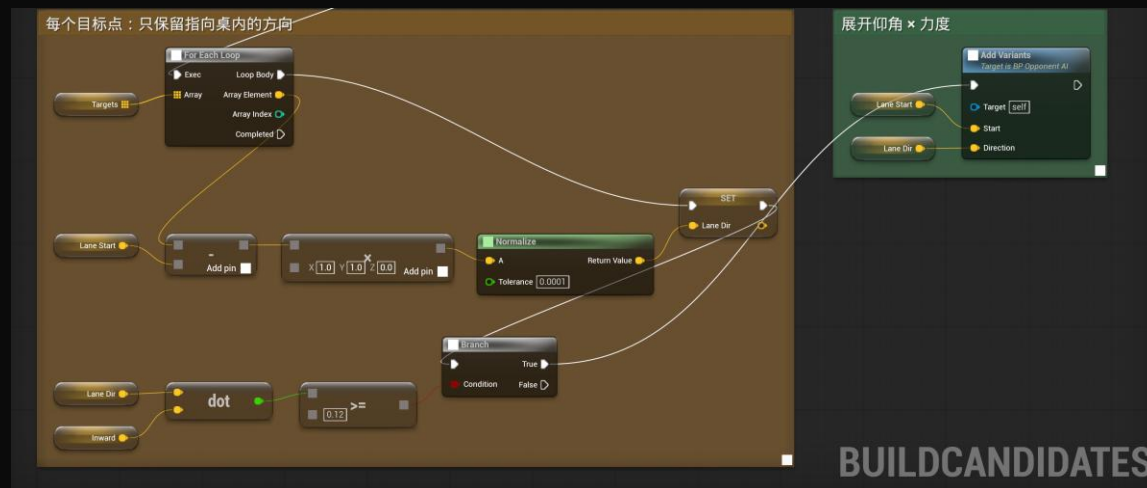
对应 C++：DicePrototype.cpp · AClashGameMode::Launch。方向必须朝桌内；交给物理并沿行进方向翻滚；投掷音效后进入运动阶段。

# 蓝图实现：首段轨迹预览 (PreviewTrajectory)



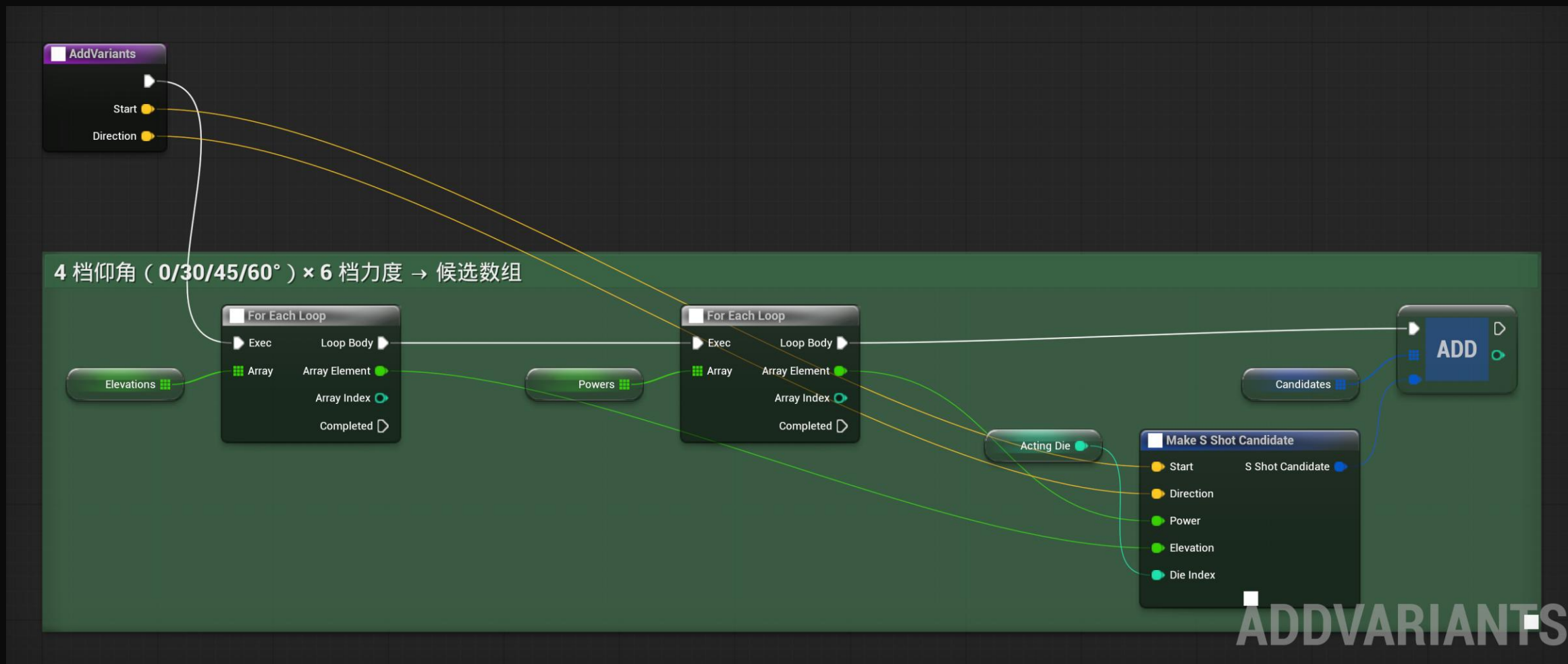
对应 C++：DiceUI.cpp 瞄准 HUD 的轨迹预览。按真实重力预测到第一次接触为止，后续反弹由物理决定。

# 蓝图实现：AI 候选生成 (BuildCandidates)



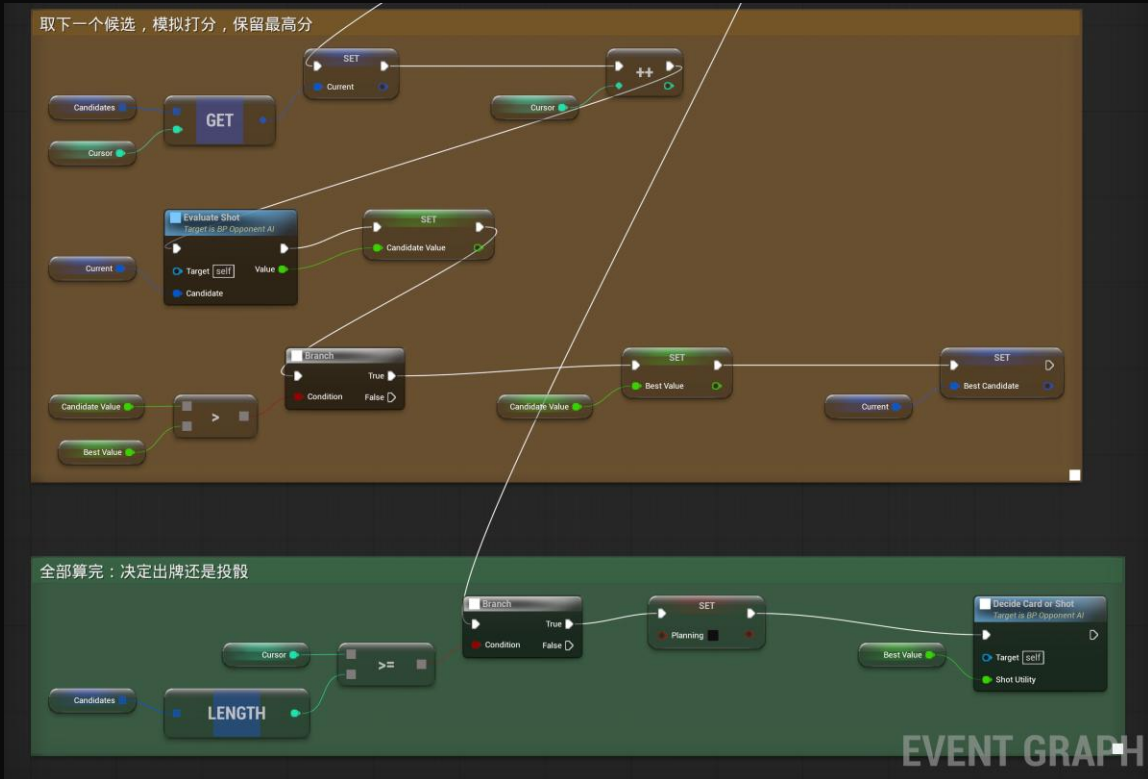
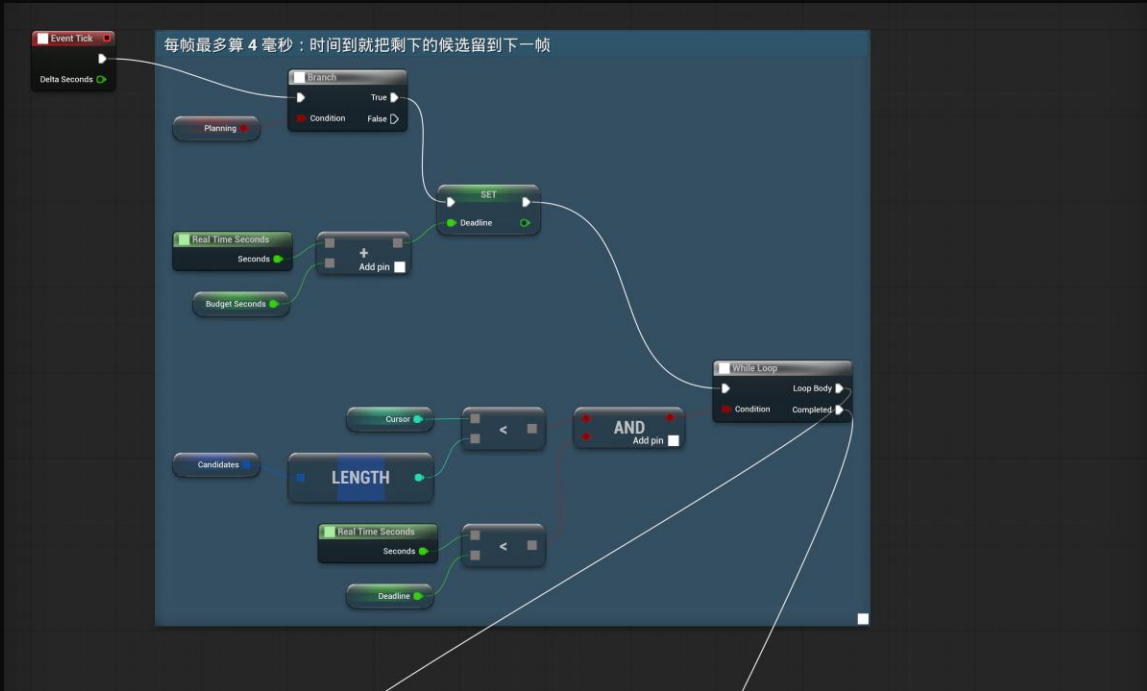
对应 C++：DicePrototype.cpp · PlanOpponentShot。五发射线 × 多个目标点，只保留指向桌内的方向。

# 蓝图实现：AI 候选展开 (AddVariants)



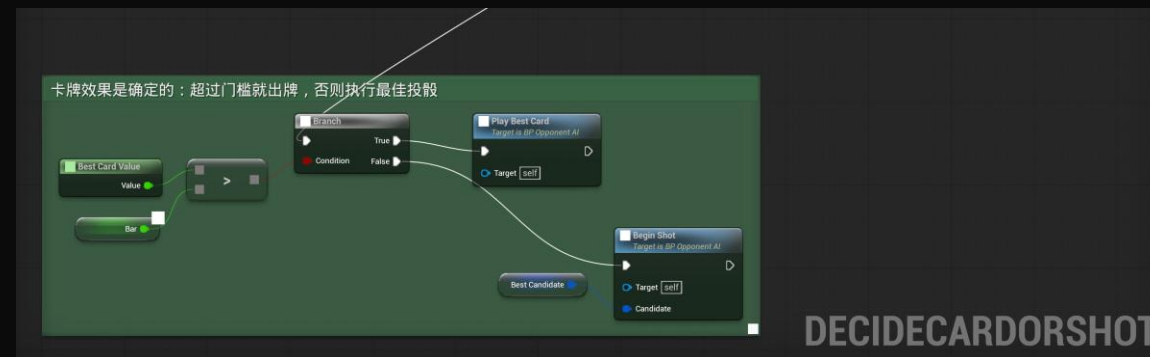
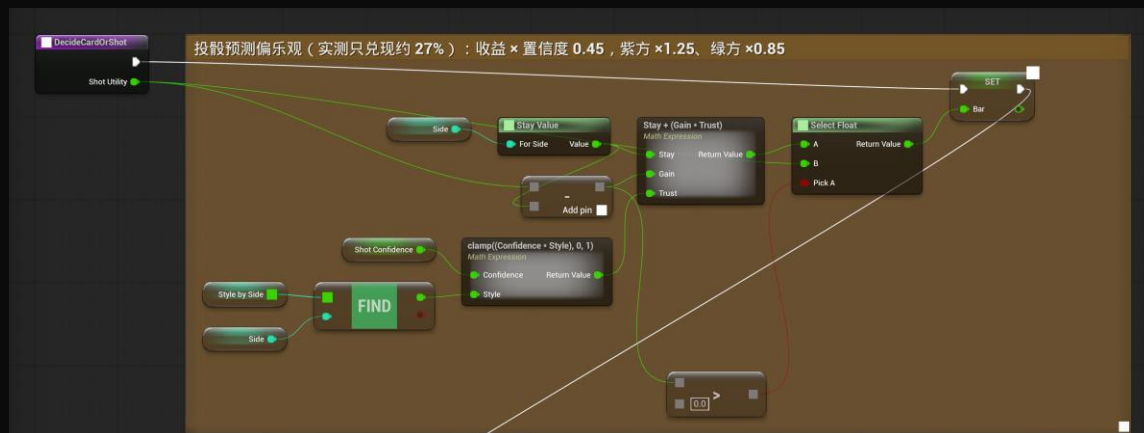
对应 C++：DicePrototype.cpp · PlanOpponentShot。每个方向展开 4 档仰角 × 6 档力度，写入候选数组。

## 蓝图实现：AI 分帧评估 (Event Tick)



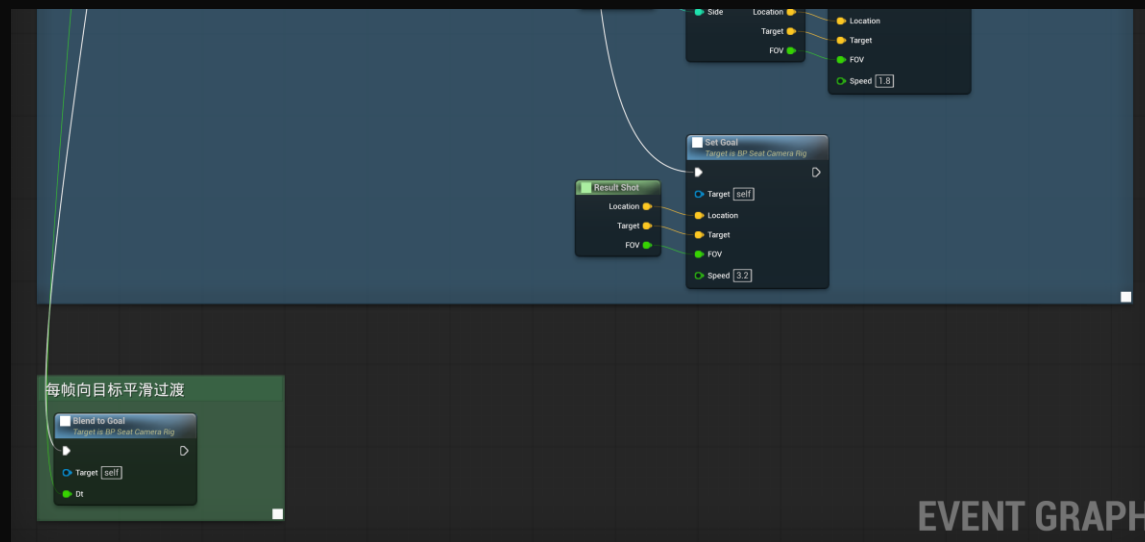
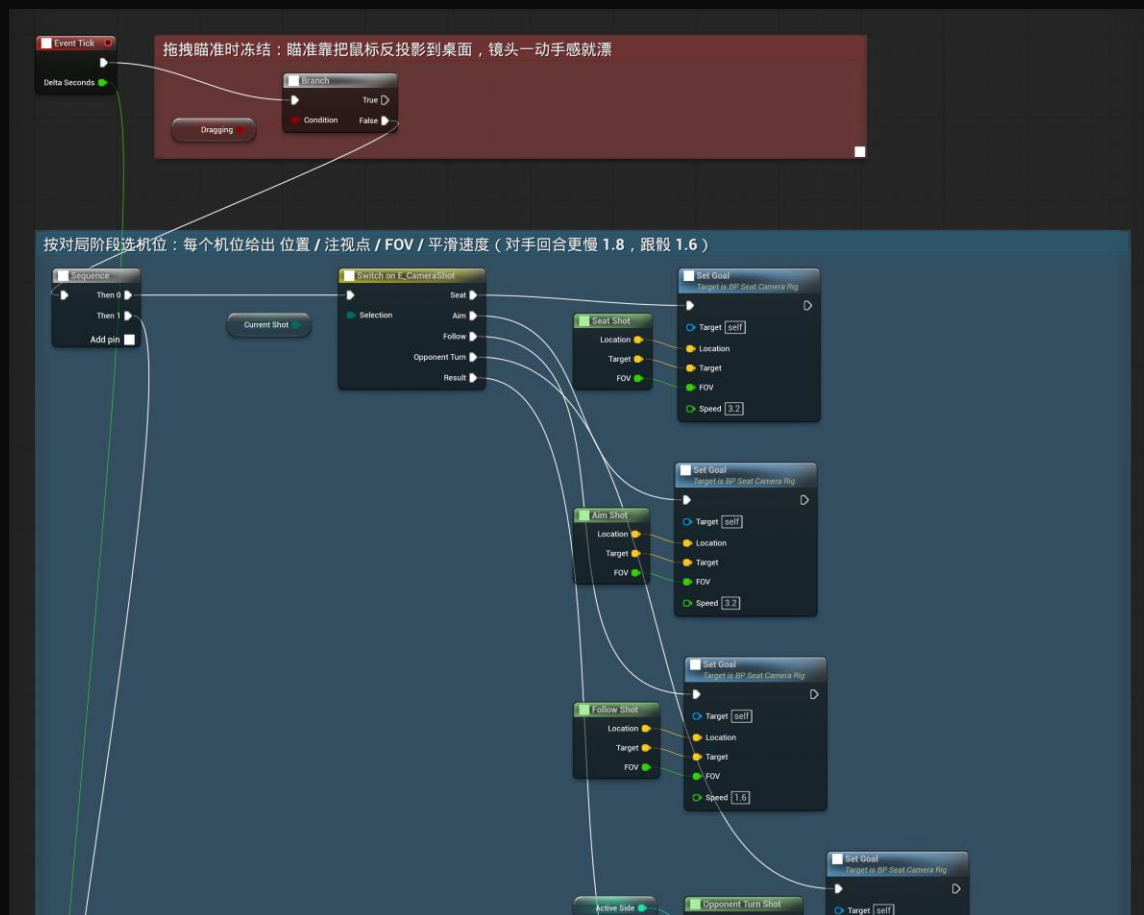
对应 C++: DicePrototype.cpp · PlanOpponentShot / EvaluateShot。每帧最多算 4 毫秒，剩下的候选留到下一帧；保留最高分。

# 蓝图实现：AI 出牌还是投骰 (DecideCardOrShot)



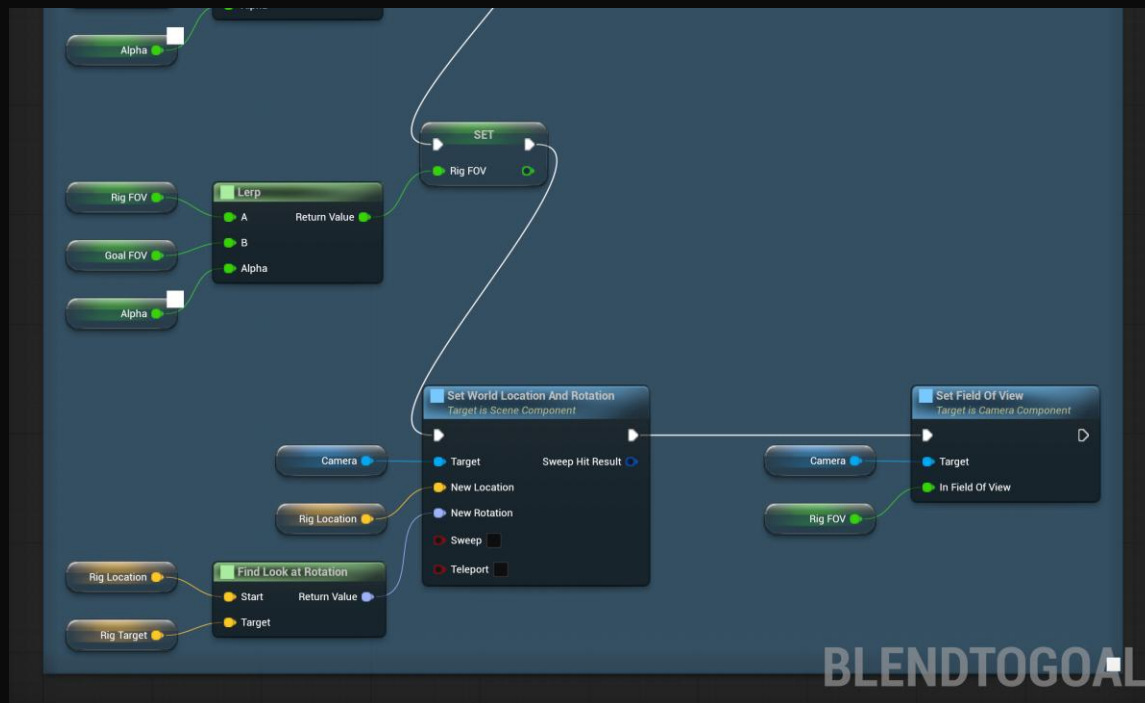
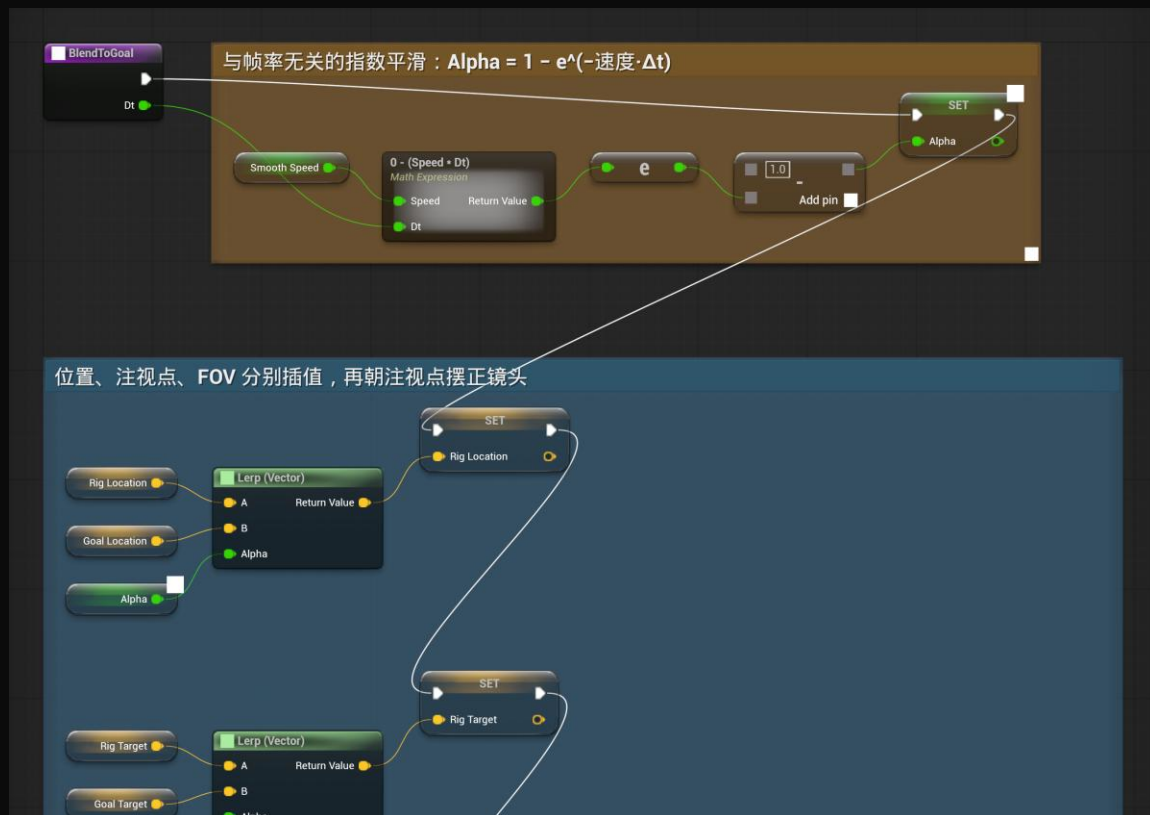
对应 C++: `DiceCards.cpp · AICardBar / TryAICard`。投骰预测收益按置信度 0.45 折算 (紫  $\times 1.25$ 、绿  $\times 0.85$ ) 再与卡牌比较。

# 蓝图实现：座位镜头的机位切换 (Event Tick)



对应 C++: DiceCameraRig.cpp · TickCameraRig。拖拽时冻结；按对局阶段选机位，各机位给出位置、注视点、FOV 与平滑速度。

# 蓝图实现：镜头平滑过渡 (BlendToGoal)



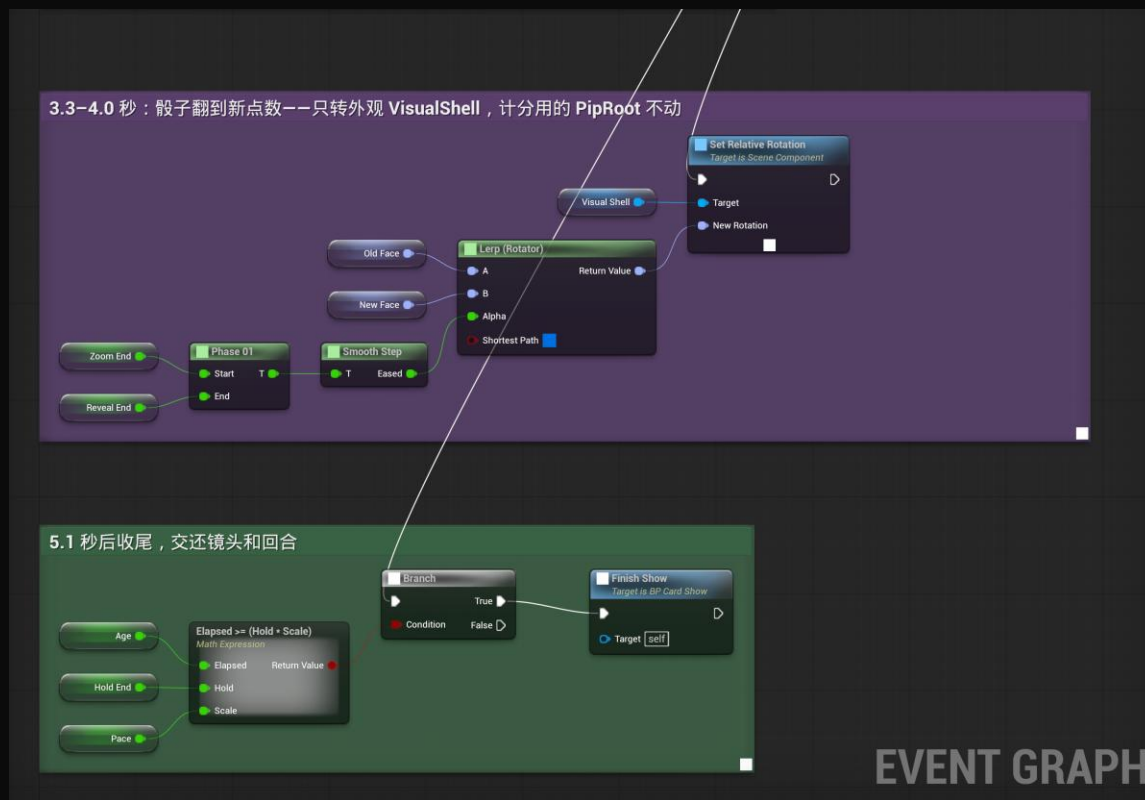
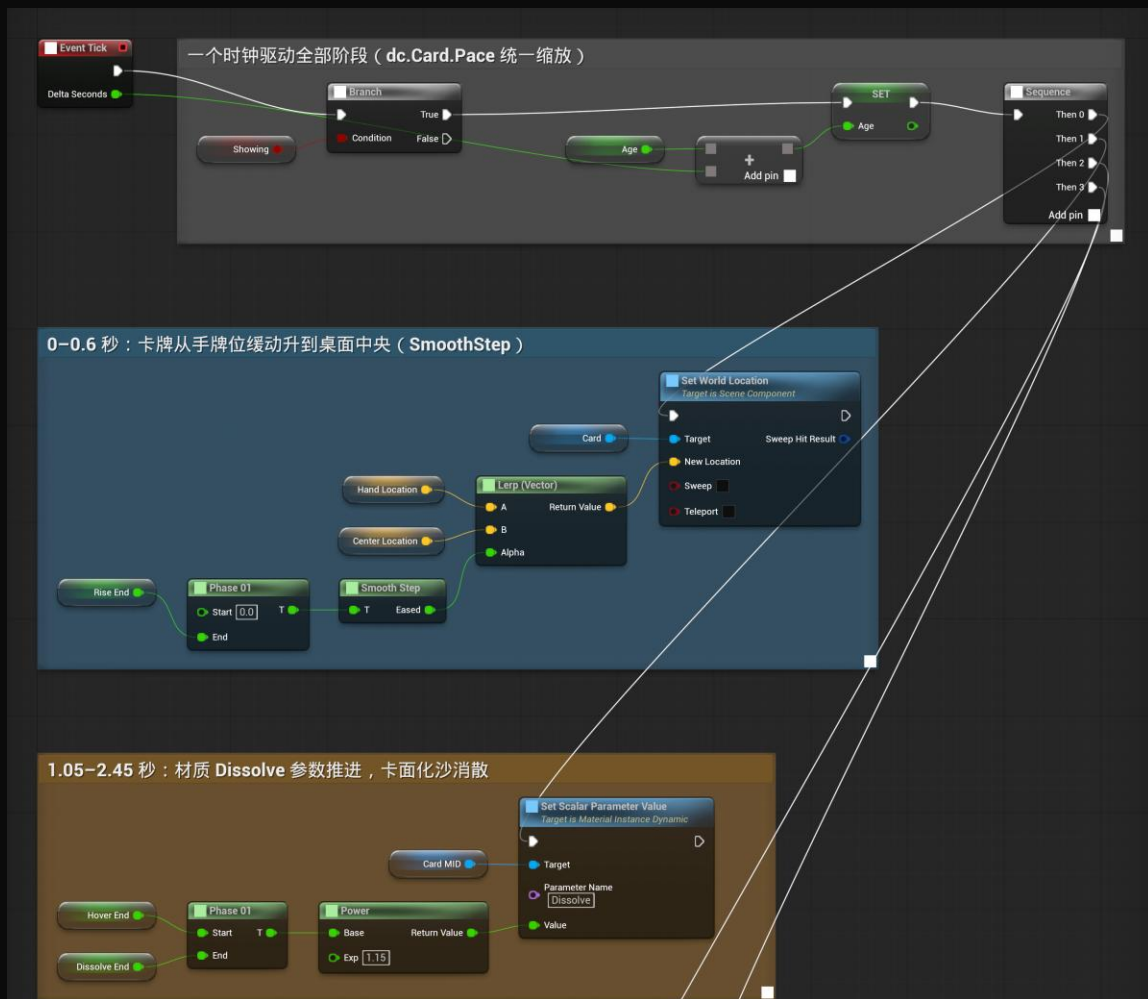
对应 C++：DiceCameraRig.cpp · TickCameraRig。Alpha =  $1 - e^{(-\text{速度} \cdot \Delta t)}$ ，与帧率无关；位置、注视点、FOV 分别插值。

# 蓝图实现：瞄准机位 (AimShot)



对应 C++：DiceCameraRig.cpp · AimShot。骰子后方 650、高 480，看向桌内，保证 260 的拖拽范围在画面里。

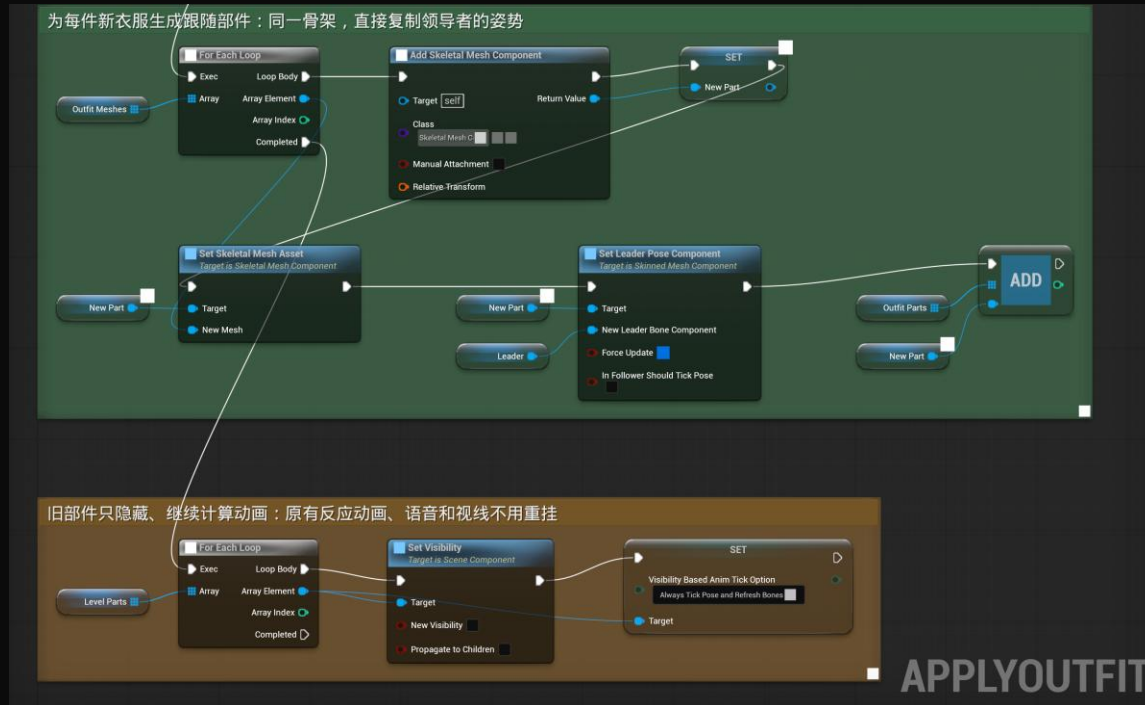
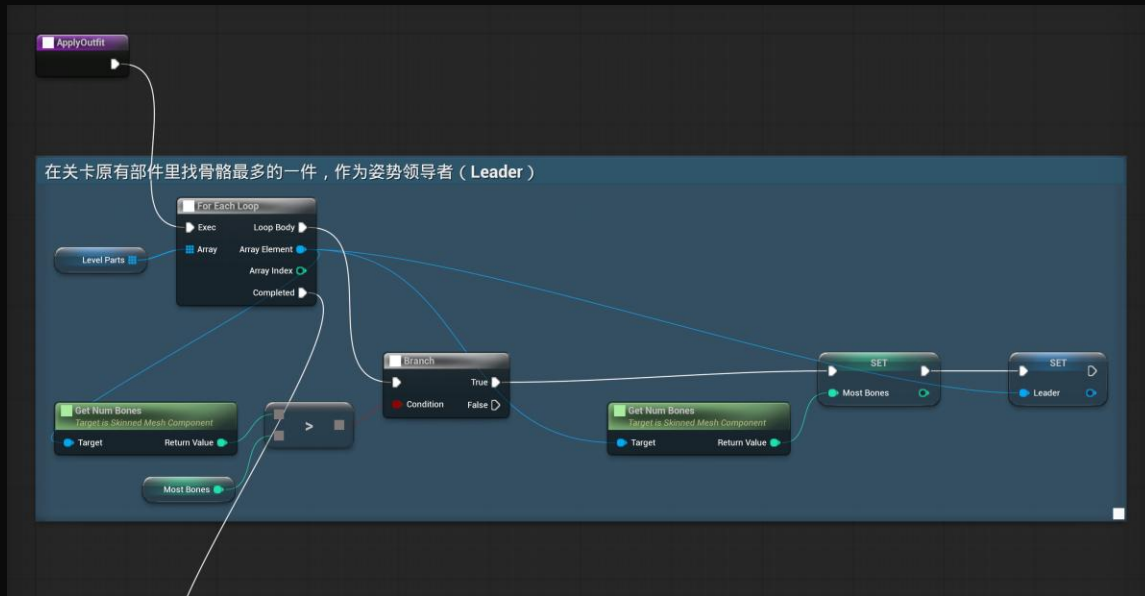
# 蓝图实现：出牌演出时钟（Event Tick）



EVENT GRAPH

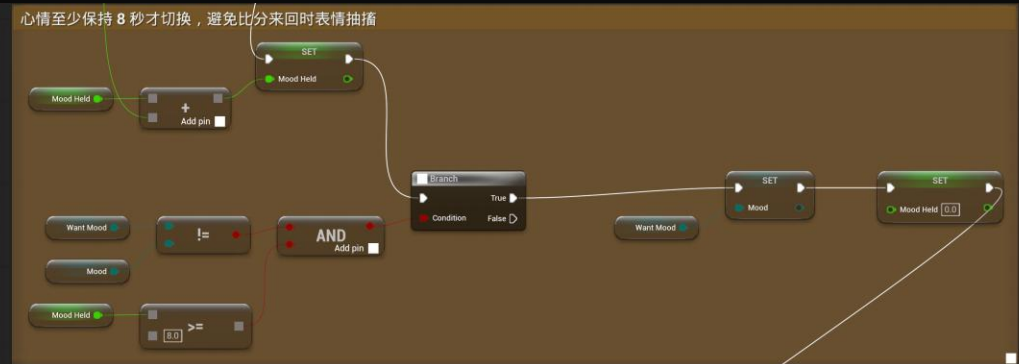
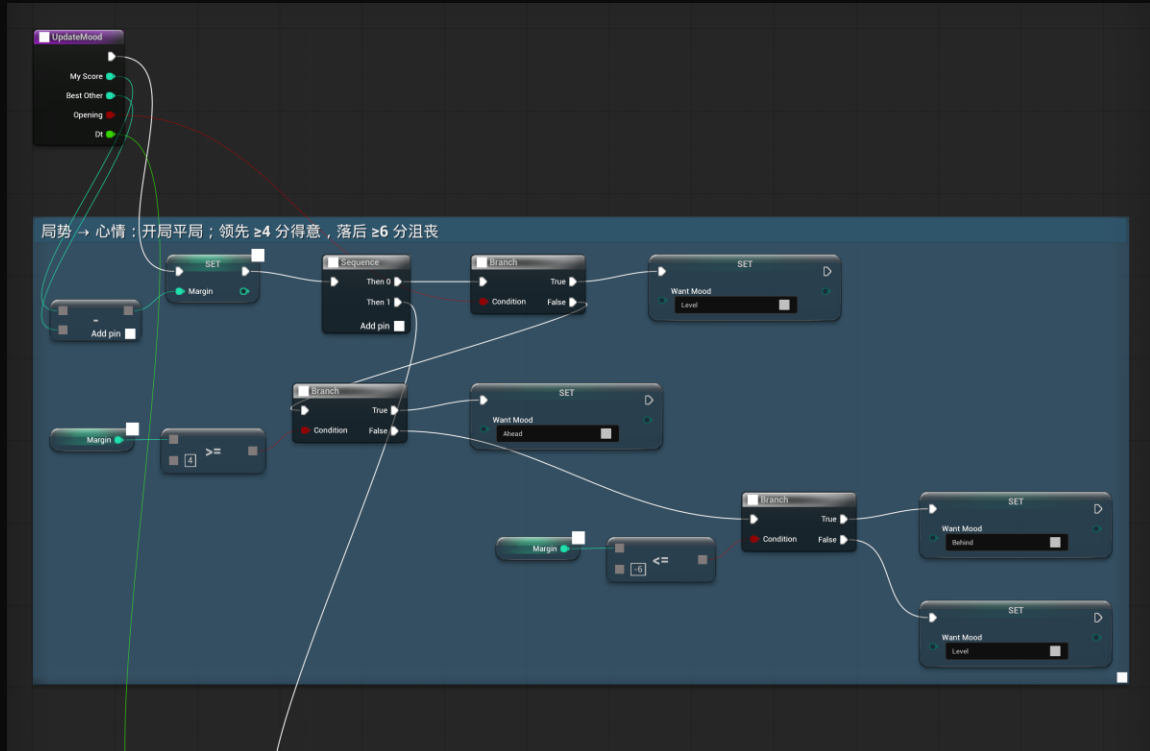
对应 C++：DiceCardShow.cpp · TickCardShow。一个时钟驱动升起、沙化、翻面与收尾；只转外观，计分的 PipRoot 不动。

# 蓝图实现：对手换装 (ApplyOutfit)



对应 C++：DiceOpponentLooks.cpp · TickOpponentLooks。骨骼最多的原部件作领导者，新衣服以跟随部件复制姿势，旧部件只隐藏。

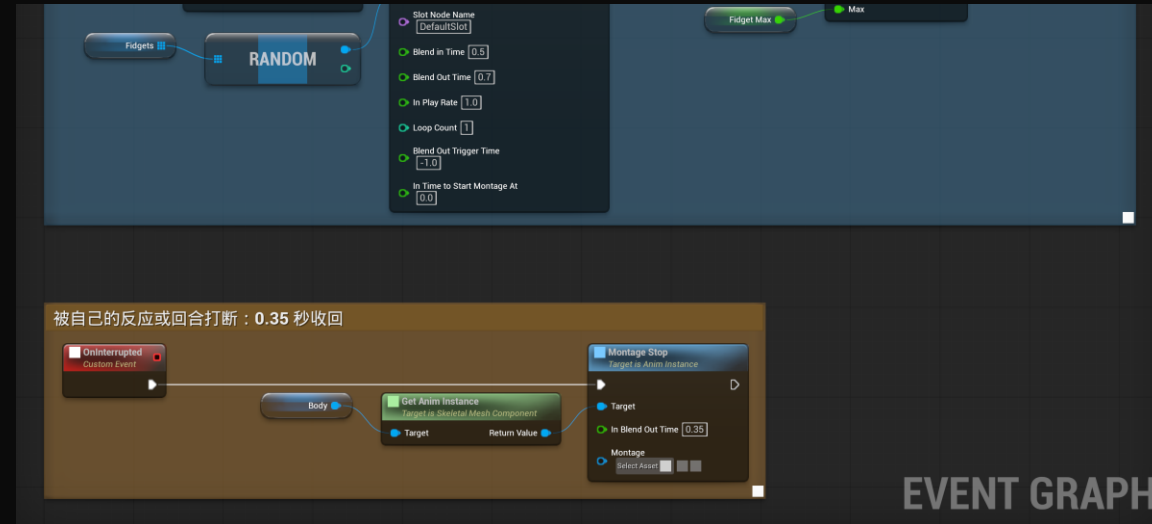
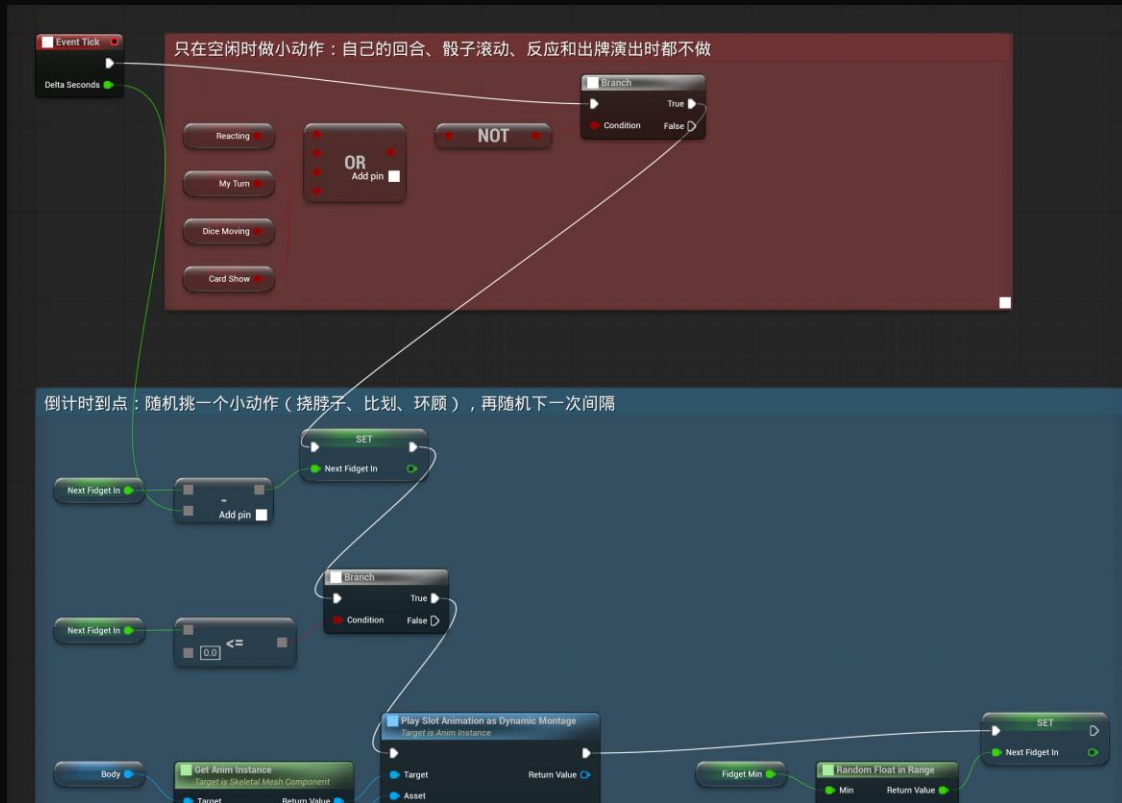
## 蓝图实现：待机心情 (UpdateMood)



## UPDATEMOOD

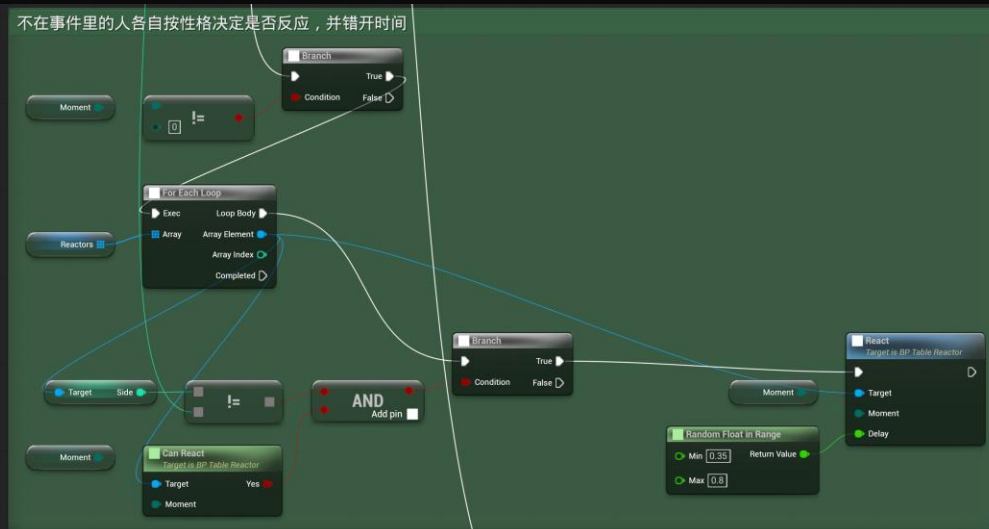
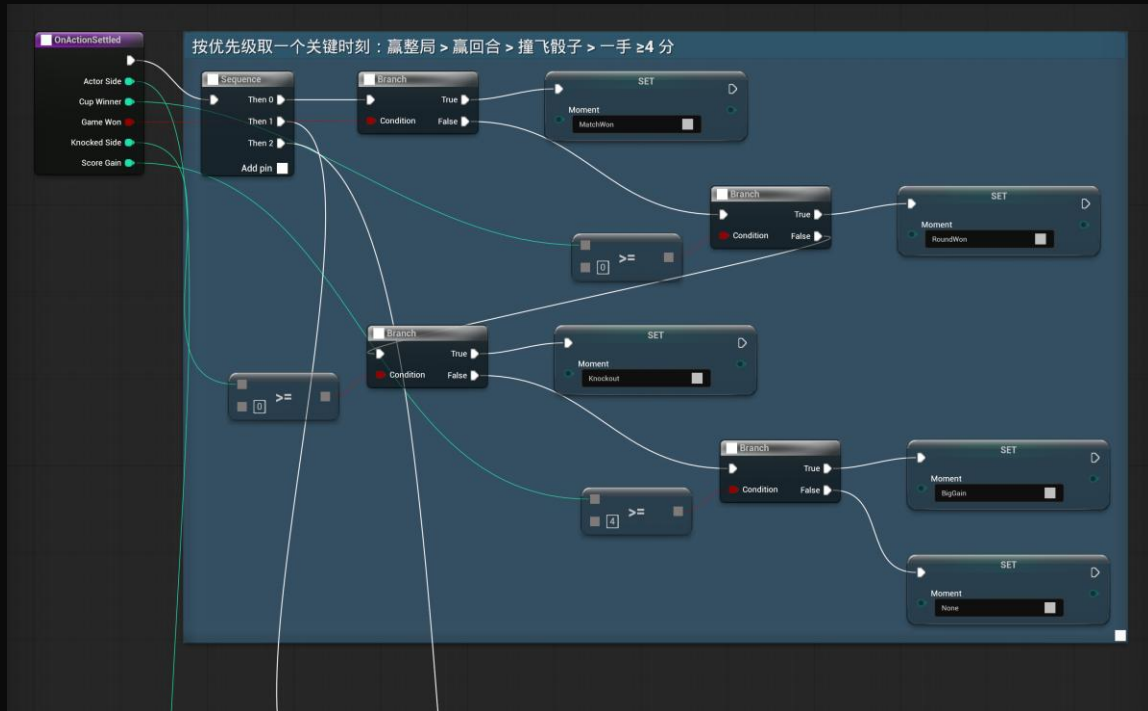
对应 C++: DiceNPCIdles.cpp · TickNPCIdles。领先  $\geq 4$  得意、落后  $\geq 6$  沮丧, 至少保持 8 秒; 从动作池随机取段交叉淡入。

# 蓝图实现：待机小动作与打断 (Event Tick)



对应 C++：DiceNPCIdles.cpp · TickNPCIdles。只在空闲时做小动作，到点随机挑一个；被反应或回合打断时 0.35 秒收回。

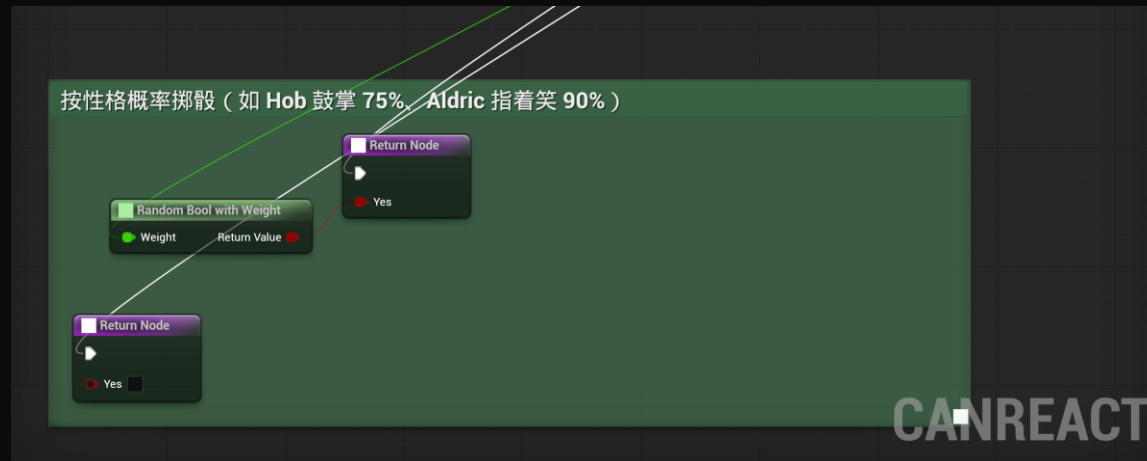
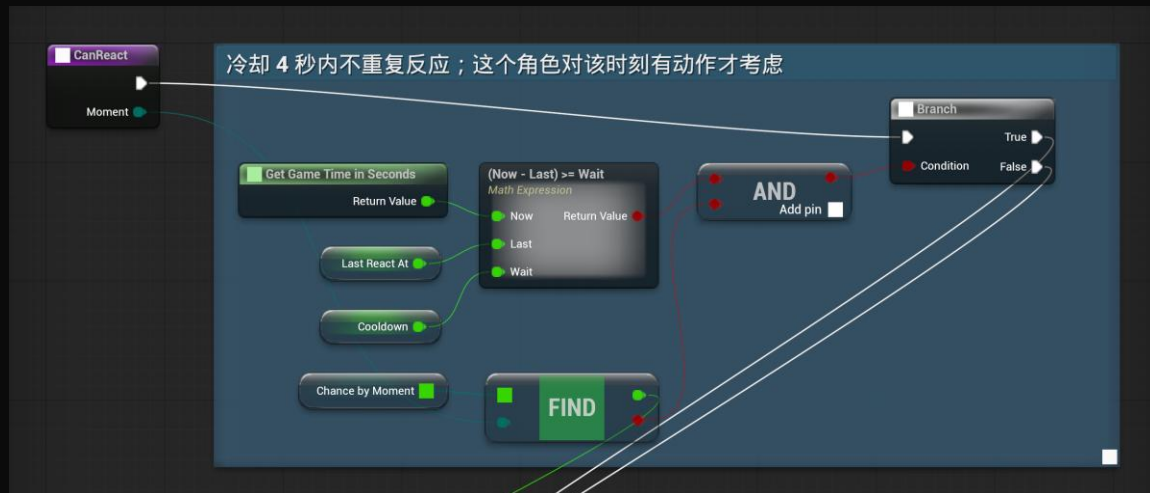
# 蓝图实现：桌边关键时刻 (OnActionSettled)



ONACTIONSETTLED

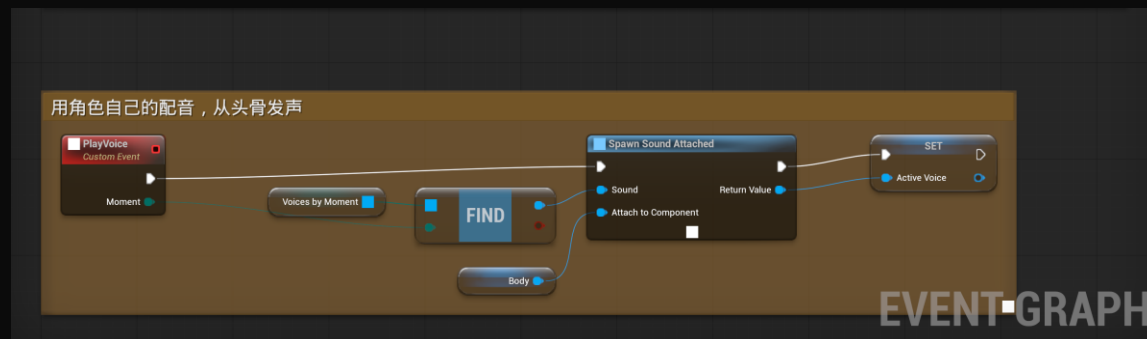
对应 C++: DiceAudience.cpp · DispatchAudienceReactions。按优先级取一个时刻，不在事件里的人各自决定是否反应。

# 蓝图实现：是否反应 (CanReact)



对应 C++: DiceAudience.cpp · FireAudienceEvent。冷却 4 秒；按性格概率掷骰 (Hob 鼓掌 75%、Aldric 指着笑 90%)。

# 蓝图实现：反应动作与语音 (React / PlayVoice)



对应 C++：DiceAudience.cpp · TickAudience。错开 0.35-0.8 秒后在反应层播放动作；用角色自己的配音从头骨发声。