游戏设计文档

一、基础结构

1. 玩家
   * 控制方式：使用 WASD 键进行移动， Shift 键实现翻滚。
   * 属性：拥有基础生命值和法力值，法力值根据时间回复。
2. 敌人
   * 小怪：初步设计包括三种（其中两种为近战类型，一种为远程类型）。
   * A型近战小怪：接触攻击
   * B型近战小怪：扇形范围攻击
   * C型远程小怪：保持一定距离，发射弹幕
   * Boss：设定为单个 Boss，分为 2 至 3 个阶段设计，使用特定卡牌攻击可迫使boss更换状态。
3. 关卡
   * 章节设定：单个章节，包含一小关。
   * 房间布局：包括1个初始房间， 2-4 个敌人房间、1 个宝箱房间
4. 卡牌
   * 初步设计五种：
     + 火球：从主角身上发射，锁定并自动追踪指定敌人，发射一连串火球。
     + 毒雾：对范围内敌人造成伤害。
     + 雷电：直接对单体敌人进行一次性高伤害攻击。
     + 狂暴：增加玩家30%的攻击力。
     + 护盾：生成带有持续时间的护盾，相当于额外生命值。

二、核心机制

1. UI 界面
   * 左上角：实时显示当前生命值和法力值。
   * 右上角：展示当前关卡地图。
   * 中央区域：为游戏主画面。
   * 底部：呈现半圆形转盘卡牌队列，卡牌进入时的动态效果。卡牌分布在半圆形圆盘上，系统生成的卡牌从右侧滑入，若有卡牌被使用，则剩余卡牌会向左侧滑动填补空缺。
2. 战斗机制
   * 玩家通过移动与翻滚闪避敌人攻击。
   * 利用鼠标拖拽卡牌释放，可用于攻击敌人或强化自身。
   * 拖拽卡牌时，卡牌变小，游戏也进入子弹时间，便于精准选中目标；右键点击则可取消当前卡牌选择。拖拽至敌人（红色高亮）触发攻击，拖拽至地面触发区域效果。
   * 卡牌每间隔一段时间生成，使用需消耗法力值，最多同时存在4张，若队列中已有4张则停止生成。
   * 需达到法力值回复与系统生成卡牌，卡牌蓝条消耗量的动态平衡，尽量营造“蓝条不够而卡牌充足”的情形，防止蓝条机制成摆设。
3. 关卡机制
   * 怪物生成：2-3波次（可选），随机生成3-6名敌人，敌人将在房间内的合法位置随机生成；进入敌人房间后，房间将自动封锁。
   * 战斗胜利：当玩家击败房间内所有敌人后，系统提示战斗胜利，并自动回复玩家一定生命值。
   * 宝箱奖励：战斗胜利后，系统会在随机合法位置生成宝箱，宝箱奖励包括（60%属性增益/40%卡牌强化）：
     + 玩家属性增益（如提升生命值上限、加快法力回复速度或提高法力上限）。
     + 卡牌属性增益（如提升火球的伤害或数量、增强毒雾效果等）。
4. 敌人机制
   * 近战敌人：主动追踪玩家，并采用近身攻击。
   * 远程敌人：寻找合适的射程位置进行远程攻击。
   * Boss 设计：尚在探索阶段，建议设计多阶段战斗，并可加入特定卡牌打破某些阶段的防御或限制。
5. 游戏流程
   * 开始流程：启动游戏 → 进入主界面 → 点击“开始游戏” → 选择难度（可选） → 生成关卡/地图 → 玩家进入初始房间。
   * 通关流程：击败 Boss 后显示通关动画或提示 → 游戏结束时给予特定增益奖励，同时提供给下一次挑战增加难度，给出重新挑战的选项。
   * 失败流程：若玩家在战斗中被击败，则显示失败界面，并同时提供重新挑战的选项。

三、美术概念

1. 整体风格
   * 采用像素风格，画面需涵盖以下元素：
     + 玩家：移动、翻滚、站立等动态表现。
     + 敌人：移动、弹幕攻击等动作效果。
     + 道具：宝箱、增益物品、金币等图标设计。
     + UI：按钮、状态栏、卡牌样式等界面元素。
     + 地图：地板，墙壁，障碍物。
2. 音效与 BGM
   * 音效：选择免费素材，涵盖卡牌释放、攻击命中，增益获得、宝箱开启、角色受击、按钮点击等常用音效。
   * 背景音乐：（可选）免费素材，设计未战斗、进入战斗以及 Boss 战三个不同氛围的 BGM。

分工

桑：主角移动/属性相关，ui界面（ui布局/卡牌队列实现）

林：敌人移动/ai行为树，房间/地图的生成和布局。

崔：美术/bgm/音效 图标，敌人/主角的动画，地图

，卡牌设计，技能的特效/粒子

计划

1. 主角移动/属性的实现，基础ui布局，敌人的移动/攻击方式，
2. 抽卡队列/卡牌攻击方式，房间/地图，敌人ai/行为树
3. 完善ui,功能整合（玩家与敌人，地图的交互），boss机制设计/实现
4. 游戏完整流程的实现和游戏调试。

四、git的使用与开发环境搭建

1. unity版本：2022.3.55f1c1

2. 仓库地址：https://github.com/CUC-HAINAN1/Black-Dungeon-Draw-A-Card

3. 确保看完大话unity git版本控制教程，特别是第三章与第四章多人协作和unity git实践部分。注意git LFS的配置和unity中版本控制和资产序列化的设置。

4. 远程库的使用与克隆：.gitignore和. gitattributes配置文件都已存在于main的根目录。克隆时创建一个新的项目文件夹，然后从远程克隆即可。使用unity打开该项目文件夹时，会自动创建缓存和配置文件，再把代码/美术资源导入该项目即可。

5. 最后检查：

- Version Control Mode: Visible Meta Files

- Asset Serialization: Force Text

6.注意：

-所有提交都需提交至dev/分支下，main/分支仅当开发工作进展到一定程度之后通过合并更新。

-新功能需从dev创建子分支（如feature/技能系统），开发完成后合并到dev。

-美术资源通过git LFS提交。提交前检查文件是否显示为LFS标识（避免误传文本指针）。

-提交前先拉取最新代码， 遇到冲突时，不要强制推送代码或删除他人修改。先把冲突发群里看看，同时利用教程中提到的解决冲突方式

-遇到冲突时，推荐使用unity官方提供的合并工具unityYAMLMerge, 安装在unity的tools文件夹中，具体配置方式和使用方式可询d指导或gpt。