

DemoVI

设计文档

执行摘要

旨在制作基于 Love2D 引擎，在 DEOT（GS）上面是可以运行的 Lo-Fi 游戏。你现在有权限管理一篇场地，上面可以你可以通过重新布局地块儿、购买新鲜地块；同时也要防止各种各样的灾害。来保证你的资产顺利运行。在一周内，能管理好你的场地吗？

游戏设计——创美部分

摘要

本游戏主要是在一个成矩阵的场地上，具有若干功能不同的“地块(Tile)”，经过一定的操作；在场上组成一定的地形后就可以进行收获金钱。自然，这些地块儿都有一定的保养费，如果安排不当，则反而有可能赔钱。

游戏的时间推进的时间颗粒是对地盘操作的一个步骤，没进行一次操作会进行一次场地金钱的结算、允许购买一块地块、或者地图上有一定地块会被随机消除等等。并且一个步骤可以是设定为一天、一小时、一个月之类的。整个游戏时长就是一周、一天、一年等，这个就是包装成一局游戏的不同长度，比如展会现场就是推荐玩一个月的长度，然后网上公开游玩时就可以推荐一年或者更多的时间。

自然，玩家的游戏任务就是要保证在这段时间中尽量不要赔款（破产），反而利用这块儿资产赚钱。

设计目标

核心功能

玩家目标

- 主要目标：保证游戏过程中不破产且尽可能在结束时资产更多
- 次要目标：资产的收益效率尽可能的高
- 挑战类型：排列好资产地图上的地块，并且权衡投入和收入的关系
- 冲突类型：玩家在限定步数中排列和布局的能力，和权衡收入与支出的能力
- 胜利条件：在游戏结束时未破产且排行榜最高

主要规则和游戏流程

标题界面：简单的选择内容，“开始游戏”/“游戏说明”/“排行榜”/“制作名单”等

游戏界面：具体进行游戏的界面。画面的大部分由场地占据，上面的游标令玩家可以方便移动已布置地块。在画面剩下的地方由现有金钱显示，通过目前场地计算出将会获得（损失）的金钱，以及剩余时间颗粒的标记的显示。其中还包含一个可以伸缩/显示的供玩家购买地块的 overlay 商店页面。

游戏说明界面：独立界面，上面可以对游戏的操作、规则、地块信息相关的进行文字、图形化的说明。

排行榜界面：独立界面，里面分为不同的游戏时长排列成的不同种类的分数排行榜

制作名单界面：简单的可能是 Overlay 的窗口可以展示制作组名单的窗口

玩家资源

金钱：玩家计算分数和基本购买单位，降至 0 后游戏结束，也使用这个去购买新的地块。获取方式就是每一次时间颗粒将场地上的收获和成本结算一次，进行结算。

地块：在每个时间颗粒中，玩家可以花费金钱购买一块地块，并且会自动植入场地中。

场地：玩家的基本控制的基本场地，上面由大量的地块构成，消耗时间对场地进行操作以试图获取金钱。

边界条件和限制

场地大小限制：原则上玩家可以拥有的场地大小是有极限的，目前设定为 6x6，是一开始给满 6x6 还是慢慢放开还有讨论空间。

板块移动限制：目前按照板块只能在场地上滑动的逻辑进行移动。

板块旋转限制：

时间限制：一句游戏中时间颗粒也是有限制的，通过这个数量不同，来控制游戏一句不通的长度。大约设计为，25、75、100。

防止停止限制：板块的布局理论上可能存在一个短期的最优解，但是为了防止玩家此时积极怠工；游戏机制中必须存在以某几个时间颗粒为周期场地中有一定量的地块会被敲掉。防止玩家怠工。

视效

待定

世界观、角色背景故事

背景故事

世界观

角色

交互模型

用户操作

菜单操作：十字键导航，A 键确认，B 键返回。

游戏内操作：十字键移动光标，按住 A 键和方向键移动地块，B 键旋转地块（顺时针），按住 Y 键和方向键从商店中选择购买。

分数和玩家表现统计

游戏设计——技术实现部分

设计假设

硬件

硬件：基于 Clockwork 公司的 Gameshell 的 DEOT

系统：Gcores Stable 1.0

软件

引擎：Love2D

在 PC 上开发，但是之后需要在 DEOT 上面进行调试和验证。

算法与细节

大块算法摘要：

地块在场地上的滑动：通过操作直接对地块进行移动旋转的基本逻辑，自然地块不能重叠、滑出场地等（已经实现）

地块通过拼接的逻辑进行显示：在设计原型时计划了 tile 的 Core 和 Side 的拼接逻辑，这样可以创建出大量的地块，美术生产时也只生产最小逻辑的 Core 和 Side 零件。需要程序上进行相关的拼接和显示。自然这个要和底层的数据逻辑相联。

在场地上地块计分逻辑：不同的地块和相关逻辑，在场地上需要不同的计分逻辑，这个需要进行相关的设计而且不同种类的地块会有不同的算法。并且还要考虑不同玩法之间的交叉。

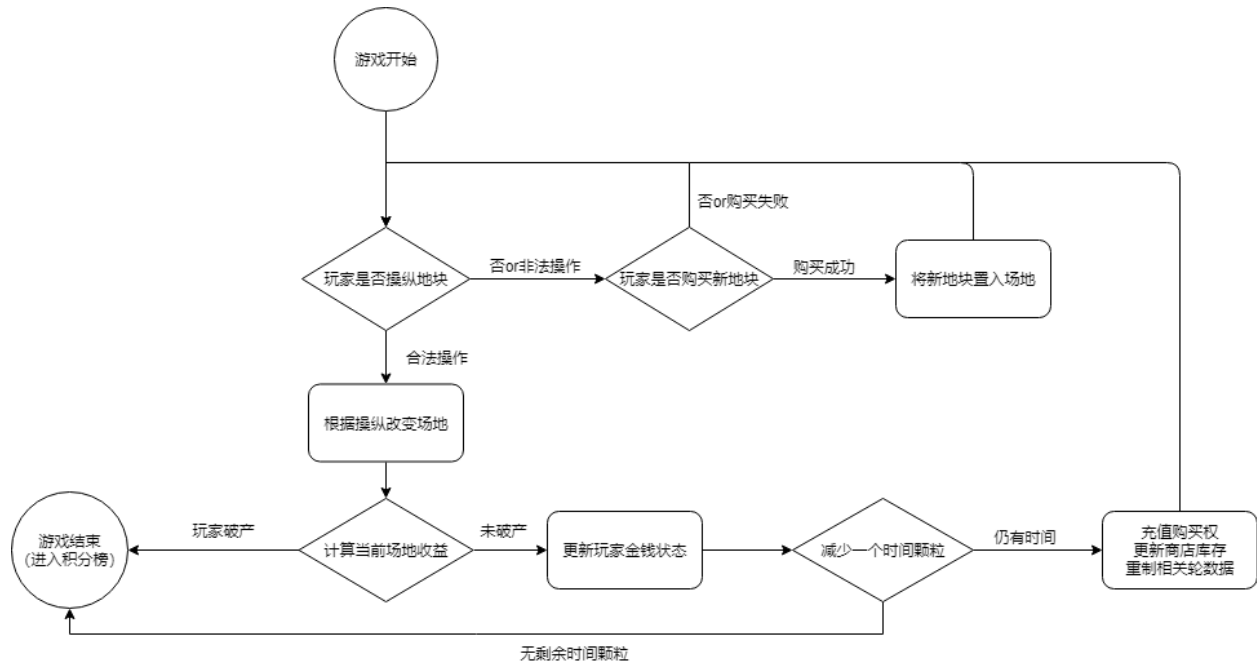
直接系统级对场地地块的直接操作：这个实现难度不会很高，但是还是写出来；主要是考虑到要敲格子和购买格子的功能。需要对场地直接注入地块信息，还是需要实现的。

BOOOM Jam Gcores

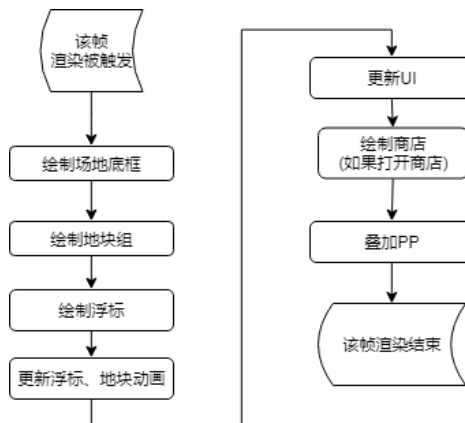
故事板：

主干逻辑：

状态机：



渲染管线：



地块说明：

目前有 5 种 Core：

PCB/无分数

Network/网线/服务器（至少有一个 Serial-side，如果 side 上的 Serial 有且只有两个则为总线）、任何其他情况都是服务器、另两侧如果互相 side 材质相同，则视为拓扑通路，如果不同则会打断。而且总线不计算联通材质的分数。且非对侧的 side 不能联通

Bridge/网桥（至少有两个 cable-side 而且对侧、另对侧的连续性参考总线

BOOOM Jam Gcores

Driver/硬盘（本身不计分，需要 Processor，至少要有有一个 Driver-side（草绿）

Processor/处理器（硬盘集群的分数计算核心、至少有一个 Driver-side。（橙色）

目前有 X 种 Side：

PCB/链接 PCB/打断链接

SerialConnector/串行链接/链接服务器、网线

ParllelConnector/并行链接 /链接硬盘。链接处理器

Firewall/防火墙、用来圈硬盘集群一圈。在链接上完全替换 ParllelConnector（纯粹上位替换）

Core:

Network：接续的流道越长越好，只有最长的计算一次分数。

Bridge：只计算链接到网桥上最大完整服务器的大小。目前 Bridge 上下游无所谓。

Processor：需要连接在 Driver 上面，只计算单纯的面积。如果边上有防火墙一整圈被圈住的硬盘翻倍。如果有网线/服务器穿过，周边硬盘分数翻倍。