综合实验

创意前端游戏

2011013259 邱泓钧 / 2012013309 王程鹏

children1930928@gmail.com / mhnfwln@126.com

目录

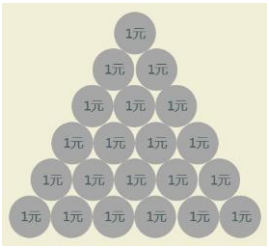
摘要

从一个古老的取子游戏中得到启发，原创出一款有趣的取星星游戏，英文名为 Last Is Worst。同时设计出 AI 实现了人机对战，并从界面优化、音效配置多个方面提升游戏的用户体验。本文档将从游戏构思、AI 设计、UI 设计等三个方面介绍 Last Is Worst 这段游戏，结尾处附上游戏效果图及开发日程。

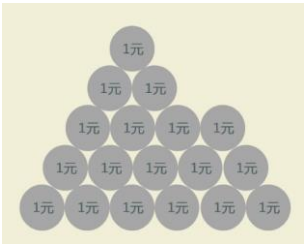
关键词

原创 博弈 益智 AI 游戏 状态约简

游戏构思



源于一个取硬币游戏，见上图。游戏规则为，双方轮流取出相邻的 1—3 枚硬币，取到最后一枚硬币者便败北例如可取上图右上角三个相邻的硬币，操作后得到下图。



在网上大致搜寻了一下，并没有出现过类似的游戏，因此我们决定设计出这款网页游戏，在 7*7 的方格中随机产生 36 个星星。游戏的规则如下：

轮流在每行或每列选取连续 1~3 个分值不同的星星，胜者最终能获得输者一半的积分。

经过讨论，我们确定优先实现游戏的单人模式，玩家可与 AI 对战，并将游戏得分存入积分榜。

总的来说，这个游戏是近乎原创的，其设计思路也是自行实现的。

AI 设计

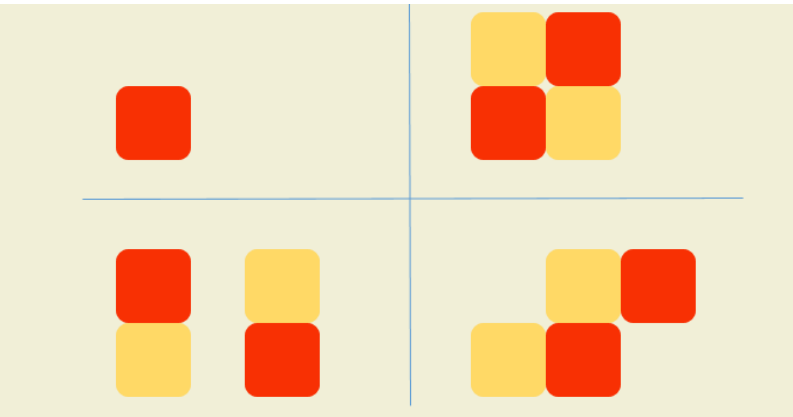
此部分源代码参见 AI.js 文件。

设计思路

首先想到的是上学期厚厚的那本算法导论，但是回顾了一下所学的动态规划，发现不能很好地实现本游戏。除了本游戏的方块是二维显示的，规则中还规定了必须取相连成直线的方格才算获胜。所以转换思路，由最初获得的规则

称上下左右无星星的星星为孤立星星。注意到两个孤立星星的存在不影响结果，因此可以对状态空间做进一步得约减，即去除所有孤立星星对。只需关注约减后的状态空间中的情形即可得到原状态空间中对应的情形的胜负结果。

在约减后的状态空间中考虑 AI 操作后的星星数 RestRectNum 为 1、2、3、4 四种取值分类讨论，每个 RestRectNum 对应不同的星星排布。在假定玩家每步操作均为最优成立的前提下，得出以下四种必胜排布，即 AI 操作结束后若出现以下四种排布（经平移，旋转，颜色变换后得到的排布与原排布等价）



剩余必胜状态图

基于上面的观察，设计出的 AI 算法如下：

步骤 1、玩家操作结束后，AI 对状态为“active”的星星（亮星星）做约减，消除孤立星星对，得到约减后的星星数 VIRectNum。若 VIRectNum 大于 7，进入步骤 2；否则进入步骤 3。

步骤 2、枚举所有可行操作，并计算每种可行操作的得分，取得分最大的操作作为此轮 AI 的操作。

步骤 3、枚举所有可行操作，判断操作后的星星排布是否为剩余必胜状态图中四种情形之一。若是，则取此操作为此轮 AI 操作；若所有可行操作枚举后仍未进入必胜状态图中的情形，则进入步骤 2。

实现难点

在上面的 AI 算法流程中，最大的实现难点在于步骤 3 中如何描述图四中的四种必胜状态。VIRestRectNum 为 AI 操作后剩余的状态为“active”星星的个数。

首先根据 VIRestRectNum 做初步判断，若 VIRestRectNum 不为 1 且不为 4，则不属于图四中的四种情况；若为 1，则一定为图四中的第一种情况。

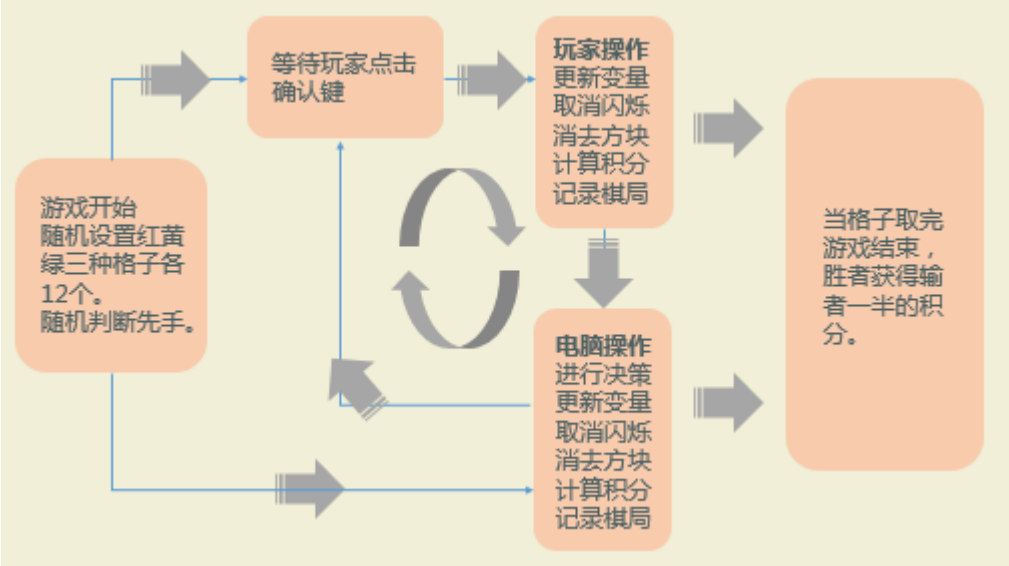
VIRestRectNum 为 4 时的判定过程稍复杂。若两个星星相邻，则称其为一个相邻对。计算剩余的状态为“active”星星中的相邻对数。若为 4，则为状态图中的情形二；若为 2 且孤立星星数为 0，则为状态图中的情形三。状态图中的情形四可通过对剩余的状态为“active”星星按 x 坐标和 y 坐标排序后简单判定即可确定。

判定是否属于状态图中的情形在 AI.js 文件中的 function SureToWin()中实现。

UI

此部分源代码参见 UI.js 文件。上半部分主要实现游戏逻辑，下半部分主要为绑定函数。

游戏逻辑

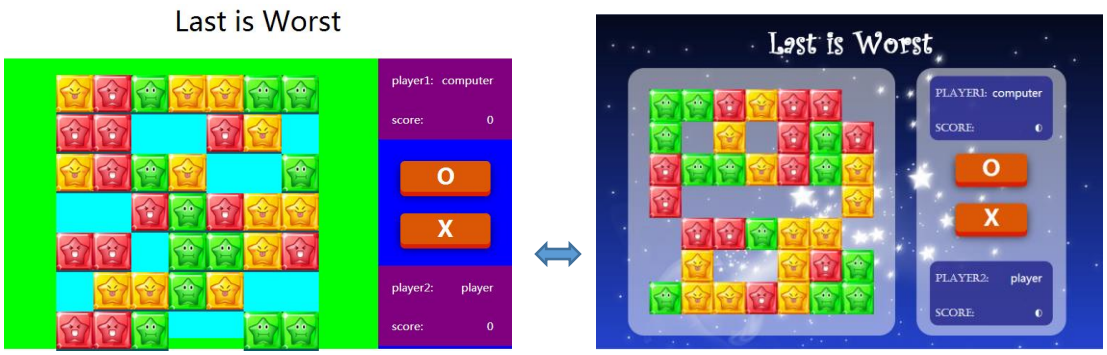


界面部件

立体按钮插件：从网上找到相关插件只修改了颜色和大小。作为游戏点击次数最多的按钮，采用了立体效果，点击时体验较好。

左侧菜单插件：从网上找到相关插件并做了较多的修改。更改了圆圈的大小，使不同圆圈边缘叠加，更具层次感。

边框背景透明：在参考了一系列相关设计素材后，发现将不同去区块另用一个盒子包围做一定的分离，并采用透明化的背景，美感顿时提升。



界面图层

这次的游戏设置了很多 hidden 的图层，如剧情层次，用户名输入，top 排行榜，about 游戏介绍，rule 规则介绍等。

游戏开始

结束剧情对话框后，最后一个点击会出现输入框填入用户名。

基础公告

主要有 top、about 和 rule，在左下角菜单可以执行弹出。在切换两个图层（如 top 和 about），会先将原本的淡出，再出现新的图层公告。

游戏结束

音乐音效

主要用到的音乐和音效如下：

bgmusic.mp3:

背景音乐是很重要的一环。选取了和游戏背景较为相近较为轻盈的音乐。

correctSelect.wav:

当用户正确点击方格出现提示音。和错误点击提示音一样都采用了较为简洁的音效。

incorrectSelect.wav:

当用户错误点击翻个出现提示音。之前对于用户错误点击处理一直比较头疼，通过代码可以实现智能判断，但是使用浮窗等提示仍然容易给用户觉得厌烦，而不进行提示玩家可能会以为是程序设计问题不能很快意识到是点击错误（尤其现在很多玩家懒得看规则）。通过该提示音则很好解决这个问题。

OBtn.wav:

当用户点击正确并按 O 按钮时获得积分时会调用此提示音，是经典获得积分的提示音。

XBtn.wav:

当用户点击错误按了 O 按钮，或者选择了几个方格想取消时调用，类似于弹簧的声音，与取消和错误等场景也较为相符。

SetEnd.wav:

游戏结束提示音，表示本游戏结束。

用户体验

对于电脑的 AI 实现取方块操作，经历了一下四个阶段。

1. 考虑电脑取方块时速度很快会让玩家来不及反应，采取缓慢消去方块的动画效果。
2. 消去速度再慢效果仍然有限，很多用户不能明显感受到变化。因此使用了 setTimeout 函数，让电脑闪烁 1s 后调用 AIisWorking 函数，执行消去方块动画。
3. 实际进行游戏体验时，我们发现，当玩家做出抉择后，AI 几乎没有反应时间就立刻执行闪烁了，容易造成用户的挫败心理，因此为提升用户体验，再度使用用 setTimeout 函数，让电脑发呆 1s 后调用 downShiftAI 函数，执行方块闪烁动画。
4. 在电脑消去方格时间也限定玩家点击。如果电脑还没完全消去方块就选取，会没有反应。

附录

游戏连接

http://children19930928.github.io/LastIsWorst

在 chrome、IE、firefox 等主流浏览器皆能使用，在 chrome 上体验风味更佳。

任务分工

邱泓钧：游戏构思、设计 AI、实现 UI、游戏逻辑、接口拼接

王程鹏：实现 AI、界面优化、添加音效、游戏测试、剧情设计

开发日程

7 月 12 日	下午	查阅游戏开发相关书籍；体验游戏
	晚上	初步构思；思考 AI 算法；查找星星素材
	深夜	实现游戏界面框架
7 月 13 日	下午	构建游戏逻辑；查找按钮素材
	晚上	实现 AI 除与 VictoryPriorDecision 的所有函数
	深夜	没干正事，背着电脑看球去
7 月 14 日	下午	完整地实现 AI；构建界面
	晚上	AI 白盒测试，调 AI 已发现的 bug；界面初步确定
	深夜	AI 与界面拼接，出现一堆 bug
7 月 15 日	下午	调接口拼接的 bug
	晚上	完善游戏逻辑；修补 AI

	深夜	参考资料，进行界面优化（用户输入、游戏结束）
7 月 16 日	下午	搜集音频素材；添加游戏音效
	晚上	利用 localStorage 实现积分榜；制作 PPT
	深夜	调积分榜的 bug；完善 PPT（做的最认真的一次）