下面分享一下设计思路，主要还是《开发者文档》中的内容：

1、基本功能：  
1.1、在4\*4的方格内随机生成2和4：  
扩展为任意大小方格，为界面显示，方格大小限定在3\*3至6\*6之间；  
随机生成2和4，设计为可加权的生成2和4以及其他值：  
如设置为“1,1”表示2和4出现概率各占50%，即题目要求设置，  
如设置为“2,1,1”表示2出现概率(2/(2+1+1))=50%，4出现概率25%，8出现概率25%；  
随机生成数的位置为当前所有空余位置随机取一个。  
1.2、能够使用键盘方向键控制移动：  
采用事件结构检测键盘输入。  
1.3、显示当前得分，得分规则请参考http://2048game.com/：  
原网页无效，根据其他游戏参考所得分数即为合成的新值。  
1.4、显示历史最高分，通过读写配置文件记录该数据；  
写明为配置文件，采用ini配置文件，同时利用配置文件队列特性，保存程序运行中的数据；该数据只有退出时才保存到文件中。  
1.5、实时显示当前已运行时间；  
多开一个循环实时显示，采用对齐的延迟等待，0.1s刷新一次。  
1.6、至少两个按键“新游戏”和“停止”，点击“新游戏”运行时间重新开始计时；  
至少的功能。

2、基础功能：  
2.1、方向移动：按指定的方向，利用数组的旋转特性统一成一种形式，去除0，合并相同的值，补充0；  
2.2、随机加点：在空位置中随机选取一个位置，按设定的权值生成新值；  
2.3、主体控件：可选取图片下拉列表组成的数组，考虑移动特效，可通过多个控件组成引用数组实现；  
又扩展为任意大小方格，多个控件的引用数组比较困难，选取二维图片控件更为方便有效，同时为在子VI中实现图像输出，使用控件的引用。

3、扩展功能  
3.1、数字移动时，能够看到移动的特效：（难度较高）  
对移动进行分析，确定每个点移动的原值、新值、移动的距离（及该点起始坐标）；  
根据分析结果将所有的点分两类：不动点、移动点；  
将不动点组成新的数组（其中的移动点的位置用背景0代替）画图输出为背景；  
将移动点的坐标循环微量变动实现图像的变动，以一个方格为单位，每移动一个方格距离-1，将距离为0的点从移动点中删除作为不动点；  
循环以上过程至移动点为空，完成移动特效。  
程序中分析偷懒了，而是直接画全0背景，之后直接进行循环，效果一样，理论上效率降低了，实际未测试。  
3.2、AI功能，用LabVIEW写个自动运行的程序，不建议使用动态链接库；能自动运行到512就满足要求了：（难度中高）  
基于512的设计需要，采用贪心策略（就是找目前看最好的移动策略，不去搜索多少多少步……）。  
关键是要有一个好的评估函数，之前对该游戏不熟，评估函数也不太会写，随便写了个，仅考虑了局势的杂乱度（即每个点与其上下左右四个点的差值的绝对值的和），杂乱度越低，局势越有利；  
模拟局势按各个方向移动一步或两步，计算新局势的评估函数值，选择当前最有利的移动方向作为移动策略。  
该策略仅能80%概率保证达到512，测试概率大致为：512以下20%，512占60%；1024占15%；2048在5%以内；4096及以上不可能，最高得分33000左右。  
该思想应该没有问题，主要是评估函数还不太完善，如果评估函数写好了，应该能有95%以上保证512，20%以上到2048。  
本AI限制了左上角为最大值点，对于设定开局、人间游戏提示效果不理想，可通过改善评估函数解决；  
本AI可适用于任何加点设置、任何大小方格；

4、其他扩展：  
4.1、扩展图片样式，实现加强版、朝代版扩展等不同的图片版本；  
4.2、实现存档读档；

5、未做扩展：  
5.1、增加存档读档的选择功能，实现更多的存档；  
5.2、利用修改ini文件读档，实现设定模式、开局模式等；  
5.3、利用计时器实现限时功能；

6、优化总结  
6.1、界面优化；  
6.2、程序模块化，建立子VI；  
6.3、程序注释；  
6.4、图标设计：直接复制的别的图标，有的形象，有的不形象，最好不要是LabVIEW的默认图标；  
6.5、文档完善。