【完美国际】集成自动寻路

**文档撰写时间：**2013年10月17日

**关键词：Angelica 2.0，VC6，AutoPFImp**

**集成背景：**自从游戏上线后，一直采用与碰撞物接触面切向移动作为无自动寻路情况下的临时方案。对老玩家影响不大，但要拉新玩家时已经不合时宜。公司范围内，以Angelica为引擎的游戏、目前也就最早推出的《完美世界国际版》和《完美世界经典版》没自动寻路，其它如《诛仙》、《赤壁》等都已经集成了AutoPF（不支持分层寻路）或AutoPFImp（支持分层寻路）。

**策划需求：**主要提供新手村、大地图各主城及附近的自动寻路。

**集成难点：**32位游戏客户端面临的内存环境越来越差，32位win7操作系统下尤其恶劣。已经有很多内存不足导致崩溃的问题发生，已经针对周围玩家的模型加载做了优化处理，但除了“不加载周围玩家”的策略外，其它未见明显改善（见客户端optimize\_mem.xml界面中提供的功能）。自动寻路在《完美国际》的另一个特殊情况是，完美国际的大地图world有11\*8个地图块，每个地图块大小是1024\*1024，这种情况下，如何减少内存占用是必须考虑的问题。

**支持大地图的集成策略：**允许只在world大地图部分重要地图块寻径，要求地图块相连，支持大地图内同时布置多个地图块。如configs.pck\instance.txt里对大地图的如下配置"r1\_5-c4\_5-l0.cfg","r3\_3-c6\_6-l0.cfg","r4\_7-c2\_3-l0.cfg","r8\_8-c1\_3-l0.cfg","r8\_9-c6\_6-l0.cfg"表明在第1-5行4-5列的10个相邻矩形地图块、第3行第6列的1个地图块、第4-7行2-3列的8个相邻地图块、第8行1-3列的3个地图块、第8-9行第6列的2个地图块，总计24块地图内寻径。各配置内独立寻径、允许重合，相邻的合并到一个配置内才能实现这些相邻块内任意两点寻径（即相邻的配置之间无法互通、只有起点和终点在同一配置内才行）。

**AutoPFImp内存优化：** 直接使用与其它项目组相同的AutoPFImp模块，world大地图全图范围内寻径的情况下，将占用内存250~300MB，寻径时间约3秒钟。因此，首先对AutoPFImp库进行内存优化，优化条目如下：

1. COptimizePath::m\_LookUp 类型由 short[m\_MapWidth\*m\_MapHeight] 修改为 abase::hash\_map<int, short>，只在必要点进行记录，单块地图1024\*1024节省内存2MB，大地图全图节省176MB。
2. 加载过程优化，除去vector因resize策略带来的所有多余的空间。由于在游戏中只读，这些内容除了占用内存无意义。能提前知道大小的使用vector::reserve函数，不能知道的先统计大小。加载优化部分在CluCommon.cpp中LoadCluGraph函数。
3. 数据结构修改。修改规则有两条：(a).double 修改为float（如Graph.h中LabelValue定义）；(b).vector<Labels>直接修改为数组（如CGNode::m\_Labels）。(a)中double改为float主要用于搜索中权重weight，没必要使用double；(b)中不需要动态扩充，所需内容都是固定的，修改后可以将vector本身的内容节省下来。double改为float除了影响内存外、也暴露出在使用double时未出现的BUG（见下）、更影响文件存储格式（CLU\_FILE\_VERSION、MLU\_FILE\_VERSION）。

**AutoPFImp库BUG修改：**

1. CMAStar::Search有重大BUG，表现为提取 closelist 内容时无法形成合法路径，原因是搜索过程中对 closelist 已有节点的删除和重新放入 openlist 有2 处逻辑漏洞，第一处是浮点误差处理不当，导致同一节点从 closelist 反复删除再放入 openlist，第二处是从closelist中删除的节点，其后续路径结点应同样删除，再放入 openlist。此BUG不光会导致崩溃、更影响效率，因为1个已经处理完的节点会反复从closelist中提取出再放入openlist，并有可能反复循环。此BUG在double改为float时重现。
2. bool CMoveAgent::Search 实现有漏洞，未考虑 CMoveAgent::SetStartEnd 调用后状态。
3. COptimizePath::\_AddPathPortion 修改2处BUG，SetFootPrint 清除[m\_CurIndex+1, m\_curIndex+oldCount] 时，最后一个没清，添加新节点后，没对新节点调用 [m\_CurIndex+1, m\_CurIndex+newCount] SetFootPrint(1)。
4. CPfClu::SetupSearch 当 pStartClu == pGoalClu 时，if (cost > 0) 修改为 if (cost >= 0)，以正确处理待寻径两点在同一搜索位置的情况。

**其它修改：**

1. **多CMoveAgent支持：** 为支持同时存在多个CMoveAgnet，删除全局变量g\_AutoMoveAgent；修改CMoveMap::m\_vOrigin含义为CMoveMap左下角对应世界中的三维位置，并修改TransMap2Wld 和 TransMap2Wld 方法；CMoveMap 增加 SetOrigin 对已有的\_SetMapCenterAsOrigin 做出补充；
2. CPfClu、COptimizePath 对 CMoveAgent 的依赖，修改为都依赖 CMoveAgent 中的 CMoveMap ，避免由于 CMoveAgent 包含 CPfClu 和 COptimizePath 带来的相互引用情况，实现寻路模块的层次化依赖。

**最终内存测试：**针对大地图88块子图的寻路图加载和不同项目配置做一完整的测试，结果显示如下：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 未加载KB | 加载KB | 差距MB |
| DebugCRTAllocator | 953,580 | 1,165,476 | 212 |
| DebugLAA | 985,960 | 1,234,240 | 249 |
| DbgReleaseLAA | 818,784 | 919,336 | 101 |

其中DebugCRTAllocator是采用CRT内存分配器的调试版本，DebugLAA是采用引擎内存分配器的调试版本，DbgReleaseLAA是发布给玩家的发行版本（带调试信息，用于dump查错）。结果显示，在发布给玩家的版本DbgReleaseLAA版本中，仅增加了101MB内存。实际中策划只制作了88块中的24块，DbgReleaseLAA版本最终占用内存36MB。

**预处理辅助工具：**策划从地图编辑器编辑出玩家寻路可达图.prmap和高度差图.pdhmap，然后交由预处理编辑器进行处理生成游戏中使用的寻路图，此中使用到2个参数g\_iClusterSize和g\_iEntranceWidth（默认值分别是100和30），在寻路编辑器项目源码AutoPFImp中能找到。此两个系数影响游戏客户端内存占用和寻路效率，因此，国际撰写了这两个参数的辅助求解工具链（由3个EXE构成）。工具链输入是地图编辑器导出的prmap和pdhmap，并向入口程序AutoPFImpFindPara.exe（由AutoPFImpFindPara项目编成）指定此两个参数的求解范围，由AutoPFImpFindPara.exe遍历每种参数组合调用AutoPFImpGenerator.exe（由AutoPFImpGenerator项目编成）生成处理结果，再交由AutoPFImpOptimize.exe（由AutoPFImpOptimize项目编成）进行内存使用测试和随机寻路测试效率（有中间文件生成、确保不同参数情况下使用的测试样例相同、从而生成公平评价结果）。评价结果放在csv文件中，以下是样例：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 64 | 20 | 35.254 | 863 | 0.004 | 137 | 0.009 | 0.005 |
| 64 | 21 | 35.254 | 863 | 0.004 | 137 | 0.009 | 0.005 |
| 64 | 22 | 34.75 | 863 | 0.004 | 137 | 0.009 | 0.005 |
| 64 | 23 | 34.75 | 863 | 0.004 | 137 | 0.009 | 0.005 |
| 64 | 24 | 34.75 | 863 | 0.004 | 137 | 0.009 | 0.005 |
| 64 | 25 | 34.75 | 863 | 0.004 | 137 | 0.009 | 0.005 |
| 64 | 26 | 34.75 | 863 | 0.004 | 137 | 0.009 | 0.005 |
| 64 | 27 | 34.75 | 863 | 0.004 | 137 | 0.008 | 0.005 |
| 64 | 28 | 34.75 | 863 | 0.004 | 137 | 0.008 | 0.005 |
| 64 | 29 | 34.75 | 863 | 0.004 | 137 | 0.008 | 0.005 |
| 64 | 30 | 34.75 | 863 | 0.004 | 137 | 0.008 | 0.004 |
| 65 | 20 | 34.223 | 863 | 0.004 | 137 | 0.009 | 0.005 |
| 65 | 21 | 34.223 | 863 | 0.004 | 137 | 0.009 | 0.005 |
| 65 | 22 | 34.223 | 863 | 0.004 | 137 | 0.009 | 0.005 |
| 65 | 23 | 34.223 | 863 | 0.004 | 137 | 0.009 | 0.005 |

前两列为g\_iClusterSize和g\_iEntranceWidth参数，第3列为内存消耗（单位为MB），第4-7列为寻径成功次数及平均时间（单位：秒）、失败寻径次数及平均时间（单位：秒），最后一列为寻径平均时间（单位：秒）。实际大地图中，策划使用的m\_iClusterSize有115、66、64等。