**Pathfinder Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Conditions** | **Expected Result** | **Actual Result** |
| Base Test | pMap[] = {1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1};  FindPath(0, 0, 1, 2, pMap, 4, 3, pOutBuffer, 12); | FindPath Return = 3  pOutBuffer = 1,5,9 |  |
| Base Fail Test | pMap[] = {0, 0, 1, 0, 1, 1, 1, 0, 1};  FindPath(2, 0, 0, 2, pMap, 3, 3, pOutBuffer2, 7); | FindPath Return = -1  pOutBuffer = N/A |  |
| Follow Edge Test | pMap[] = {1, 0, 0, 1, 0, 0, 1, 1, 1};  FindPath(0, 0, 2, 2, pMap, 3, 3, pOutBuffer, 7); | FindPath Return = 4  pOutBuffer = 3,6,7,8 |  |
| No Wrap Around Test | pMap[] = {1, 0, 1, 1, 0, 1, 1, 0, 1};  FindPath(0, 0, 2, 2, pMap, 3, 3, pOutBuffer, 7); | FindPath Return = -1  pOutBuffer = N/A |  |
| Line L to R Test | pMap[] = {1, 1, 1, 1, 1, 1, 1, 1};  FindPath(0, 0, 7, 0, pMap, 8, 1, pOutBuffer, 8); | FindPath Return = 7  pOutBuffer = 1,2,3,4,5,6,7 |  |
| Line R to L Test | pMap[] = {1, 1, 1, 1, 1, 1, 1, 1};  FindPath(7, 0, 0, 0, pMap, 8, 1, pOutBuffer, 8); | FindPath Return = 7  pOutBuffer = 6,5,4,3,2,1,0 |  |
| Line top to bottom Test | pMap[] = {1, 1, 1, 1, 1, 1, 1, 1};  FindPath(0, 0, 0, 7, pMap, 1, 8, pOutBuffer, 8); | FindPath Return = 7  pOutBuffer = 1,2,3,4,5,6,7 |  |
| Line bottom to top Test | pMap[] = {1, 1, 1, 1, 1, 1, 1, 1};  FindPath(0, 7, 0, 0, pMap, 1, 8, pOutBuffer, 8); | FindPath Return = 7  pOutBuffer = 6,5,4,3,2,1,0 |  |
| Short Buffer Test | pMap[] = {1, 1, 1, 1, 1, 1, 1, 1};  FindPath(0, 7, 0, 0, pMap, 1, 8, pOutBuffer, 8); | FindPath Return = 7  pOutBuffer = 6,5,4,3 |  |
| Invalid Input Test | pMap[] = {1, 1, 1, 1, 1, 1, 1, 1};  FindPath(0, 7, 0, 0, pMap, 1, 0, pOutBuffer, 8); | FindPath Return = -2  pOutBuffer = N/A |  |

Notes: To condense the file, I formatted the conditions as if they were the function call. The function call has the following format:

int FindPath(const int nStartX, const int nStartY, const int nTargetX, const int nTargetY, const unsigned char\* pMap, const int nMapWidth, const int nMapHeight, int\* pOutBuffer, const int nOutBufferSize);