**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 |  |
| Big Test | pMap[] = {1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,0,0,1,0,0,0,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,0,0,1,0,0,0,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,1,1,1,0,0,0,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,0,0,1,0,0,0,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,0,0,1,0,0,0,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,0,0,1,0,0,0,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,0,0,1,0,0,0,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,0,0,1,0,0,0,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,0,0,1,1,1,1,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,0,0,1,0,0,0,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,1,1,1,0,0,0,1,0,0,1,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,1,1,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};  FindPath(1,1,35,12,pMap10,39,22,pOutBuffer10,256); | FindPath Return = 95  pOutBuffer = 41 80 119 158 197 236 275 314 353 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 446 485 484 523 522 521 482 481 480 519 558 597 636 637 638 639 640 641 680 719 720 721 722 723 724 725 686 647 608 609 610 611 612 613 574 535 496 495 494 455 416 377 338 299 260 221 182 143 104 105 106 107 146 185 224 263 302 341 342 343 344 345 346 347 348 387 426 465 504 503 |  |

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);