

컴퓨터정보통신공학부  
2011253020 이화중

A\* Search를 이용한   
퍼즐 풀이 시스템 개발

인공지능 A\* 프로젝트

# 프로그램 실행 방법

1. 프로젝트의 “SlidingPuzzleDoc.cpp”에 있는 CSlidingPuzzleDoc::initializePuzzle()의 정의에서 startNode와 goalNode 변수의 초기값을 변경하여 시작 노드와 목표 노드를 변경한다.
2. 프로젝트를 빌드하고, 프로그램을 실행시킨다.[[1]](#footnote-1)
3. 화면 좌측 상단의 인공지능 버튼을 클릭한다.
4. 컴퓨터가 A\* 알고리즘을 사용하여 퍼즐 풀이를 시작하며, 풀이가 완료되면 퍼즐의 움직임을 화면에 1초 간격으로 보여준다. 각 동작의 체크 포인트에서 로그를 띄운다.
5. 결과 확인 후 인공지능 버튼을 다시 클릭하면 풀이와 결과를 다시 확인할 수 있다.

참고로, 보고서 작성을 위한 프로젝트 수행 시에 사용한 IDE는 Visual Studio 2015이며, 컴파일러 버전은 v140이다. 또한, 소스코드는 C++로 작성하였다.

퍼즐 기본적인 동작과 A\* 알고리즘에서 에서 사용할 클래스와 함수는 “SlidingPuzzleEngine” 프로젝트에서 정적 라이브러리로 구현하였다. 퍼즐 블록의 움직임을 화면에 보여주고 실제로 A\* 알고리즘을 수행하는 동작은 “SlidingPuzzleMFC” 프로젝트에서 Win32 MFC 프로젝트로 구현하였다. 특히, 실제로 A\* 알고리즘을 수행하는 동작은 “SlidingPuzzleView.cpp”에 있는 CSlidingPuzzleView::OnGameAI()의 정의와 expandChild()의 정의에서 구현하였다.

# 프로그램 결과 화면

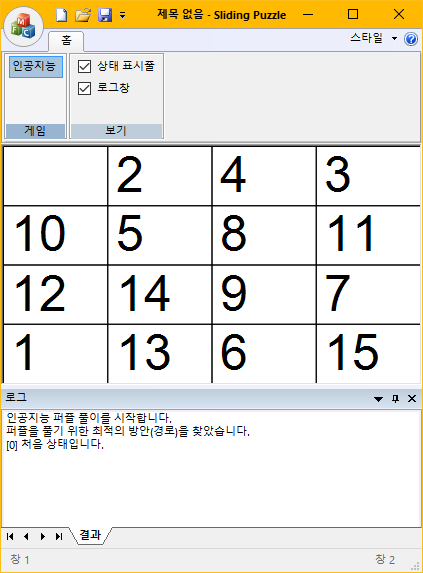


Figure 1 A\* 경로 탐색 완료 (인공지능 버튼을 누른 직후 화면)

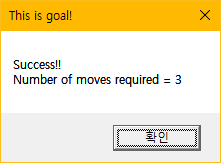


Figure 2 A\* 경로 탐색 결과창

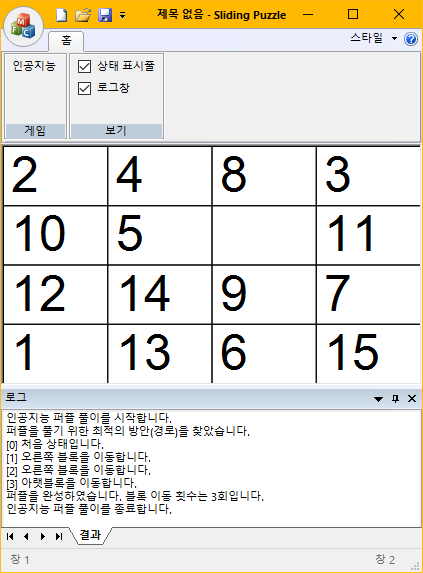


Figure 3 퍼즐 풀이가 완료된 화면

1. 만약 SlidingPuzzleEngine이 빌드가 되지 않는다면 프로젝트 속성에서 대상 플랫폼 버전을 낮추면 된다. 그래도 되지 않는다면, 함께 동봉한 “SlidingPuzzleEngine.lib” 파일을 SlidingPuzzleMFC 프로젝트 폴더로 옮기고 솔루션에서 SlidingPuzzleEngine 프로젝트를 제거시킨 후, SlidingPuzzleMFC 프로젝트 속성에서 구성 속성-링커-입력-추가 종속성에 “SlidingPuzzleEngine.lib”를 추가하면 된다. Visual Studio 2015가 아니면 SlidingPuzzleEngine 프로젝트의 빌드에 제한이 있을 수 있다. 빌드가 불가능하다면, 함께 동봉한 “SlidingPuzzleMFC.exe”를 실행하여 참고할 수 있다. [↑](#footnote-ref-1)