

1. 주제

일반화된 슬라이딩 퍼즐 문제의 탐색과 시각화 프로그램

분반, 팀, 학번, 이름

1반, 10팀, 20251776, 이준빈

2. 요약

이 프로젝트는 수업에서 학습한 A* 탐색 알고리즘을 실제 문제에 적용하여 깊이 있게 이해하는 것을 목표로 한다. 또한 8-퍼즐과 같은 정형화된 문제를 넘어 블록의 개수, 모양, 그리고 퍼즐 판의 크기까지 사용자가 직접 설정할 수 있는 '일반화된 슬라이딩 퍼즐' 문제로 확장하여 알고리즘의 적용 범위를 넓히는 것이 목표이다.

프로젝트의 핵심 기능은 사용자가 정의한 퍼즐의 초기 상태가 주어졌을 때, A* 알고리즘을 통해 목표 상태에 도달하는 최단 경로를 탐색하는 것이다. 이 과정에서 각 단계별 블록의 움직임을 시각적으로 명확하게 제시하여, 사용자가 문제 해결의 전 과정을 직관적으로 파악할 수 있도록 구현한다.

이 프로젝트를 통해 A* 알고리즘을 직접 구현하며 휴리스틱 함수의 설계 중요성을 체감하는 실질적인 학습 효과를 기대한다. 또한, 완성된 프로그램은 게임 등에서 접하는 복잡한 슬라이딩 퍼즐에 대한 효율적인 해법을 제공하는 실용적인 보조 도구로서 기능할 수 있다. 나아가 사용자는 최적의 해결 과정을 관찰하며 문제 해결을 위한 효과적인 전략을 학습하고 논리적 사고력을 기를 수 있을 것이다.

3. 대표 그림



그림1. 풀고자 하는 문제(예시)

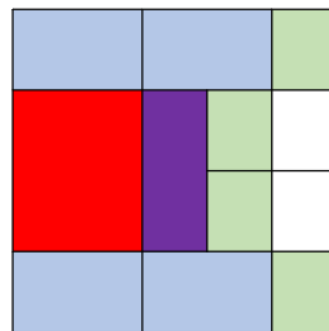


그림2. 퍼즐의 초기 상태 정의(예시)

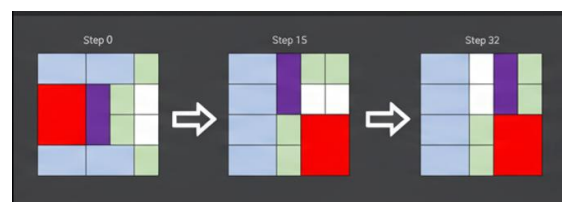


그림3. 단 단계별 블록의 움직임(예시)

4. 서론

컴퓨터 과학 분야에서 A* 탐색 알고리즘은 최적 경로 탐색을 위한 효율적인 방법론으로 널리 학습된다. 그러나 이론적인 이해만으로는 실제 문제에 적용하고 그 효용성을 깊이 있게 체감하는 데 어려움이 따른다. 일상생활이나 게임에서 접하는 슬라이딩 퍼즐은 다양한 형태를 가지며, 이는 알고리즘 학습의 실제 적용 사례가 될 수 있다.

일상생활이나 게임에서 접하는 슬라이딩 퍼즐은 대표 그림의 예시와 같이 매우 다양하다. 하지만 8-퍼즐, 15-퍼즐 풀이 알고리즘과 같은 기존 슬라이딩 퍼즐 해결 프로그램은 정형화된 문제에 한정되어 있어, 다양한 크기와 형태를 가진 비정형 슬라이딩 퍼즐을 해결하기 힘들다. 따라서 이러한 비정형 퍼즐을 풀다가 난관에 봉착했을 때, 효율적인 도움을 받기 곤란한 경우가 빈번하게 발생한다

위에서 정의된 문제들을 극복하기 위해, 이 프로젝트는 'A* 탐색 기반의 일반화된 슬라이딩 퍼즐 솔버(Solver)'를 만든다. 이 솔버는 사용자가 퍼즐 판의 크기, 블록의 개수와 모양, 그리고 초기 및 목표 상태를 자유롭게 설정할 수 있도록 설계하여 알고리즘의 적용 범위를 확장한다. 핵심 기능은 입력된 퍼즐에 대해 A* 알고리즘을 적용하여 최단 경로를 탐색하고, 블록의 움직임을 각 단계별로 시각적으로 명확하게 제시하는 것이다.

5. 본론

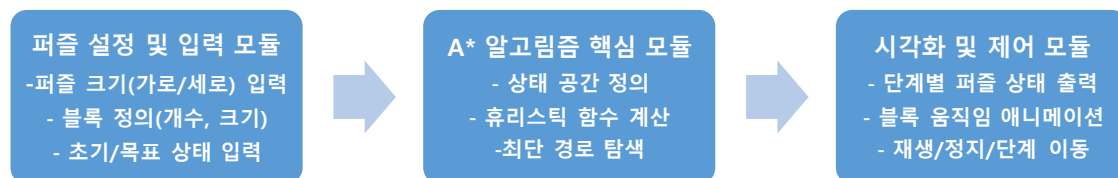


그림4. 시스템 개요 그림

이 프로젝트에서 개발할 '일반화된 슬라이딩 퍼즐 솔버'는 크게 세 가지 모듈로 구성된다. 사용자는 '퍼즐 설정 및 입력 모듈'을 통해 원하는 퍼즐의 조건을 정의한다. 이 정보는 'A* 알고리즘 핵심 모듈'로 전달되어 최단 경로 탐색이 이루어진다. 탐색 결과로 얻은 해결 경로는 '시각화 및 제어 모듈'에서 사용자에게 단계별로 보여진다.

프로젝트의 모든 로직 구현에는 파이썬 언어가 기본적으로 사용된다. UI는 tkinter 또는 Pygame를 활용하여 기본적인 설정 및 시각화 화면을 구성할 수 있다. 퍼즐 보드의 상태나 블록 정보를 저장하고 관리하기 위해 리스트와 딕셔너리를 활용하고, A* 알고리즘의 효율적인 탐색 노드 관리를 위해 큐와 우선순위 큐가 사용될 것이다. 이는 파이썬의 heapq 모듈을 통해 구현할 수 있다.

이 프로젝트의 구현은 크게 세 단계로 진행한다. 첫째, 2차원 리스트와 클래스 등을 활용하여 퍼즐 판과 블록 정보를 표현할 데이터 구조를 정의한다. 둘째, 휴리스틱 함수와 상태 전이 로직을 포함한 A* 알고리즘의 핵심부를 구현한다. 이때 우선순위 큐를 이용하여 탐색 효율을 높인다. 마지막으로, 사용자가 퍼즐을 직접 설정하고 해결 과정을 단계별로 확인할 수 있는 사용자 인터페이스(UI)를 개발한다.

개발 방향은 초기 단계에서 고정된 퍼즐에 대한 알고리즘의 작동을 검증하는 데 집중한다. 이후 사용자가 퍼즐의 크기나 블록 구성을 자유롭게 설정할 수 있도록 기능을 고도화하고 시각적 피드백을 강화하는 방향으로 진행한다.

6. 결론

이 보고서는 A* 알고리즘의 깊이 있는 학습을 위해, 사용자가 직접 퍼즐을 설정하는 '일반화된 슬라이딩 퍼즐 솔버' 개발 계획을 제안했다. 기존 솔버의 정형화된 한계를 지적하고, 파이썬과 tkinter를 기반으로 한 A* 알고리즘의 구현 및 시각화 방안을 구체적으로 설명했다.

향후 개발은 핵심 알고리즘 구현부터 시작할 것이다. 그 후, 기본 UI를 개발하여 알고리즘과 연동하고, 마지막으로 블록의 움직임을 보여주는 시각화 기능을 구현하여 프로젝트를 완성할 계획이다..

7. 출처

[1] 러셀, 스튜어트 J., and 피터 노빅. *인공지능: 현대적 접근방식*. 류광 옮김, 4판, 제이펍, 2022