10주차 예비보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20191612 이름: 윤기웅

**1. (문제 및 제목은 작성하지 말 것. 답안만 작성하면 됩니다.)**

.................

테트리스 프로젝트 3주차에 구현하는 추천 기능은 플레이어가 블록을 놓을 수 있는 모든 위치와 회전 상태를 계산하여, 각 상태에서 얻을 수 있는 점수를 평가하는 원리로 작동합니다. 이를 위해 tree 구조를 사용하여 각 블록의 가능한 위치를 노드로 하고, 노드 간의 연결(간선)을 블록의 이동 및 회전으로 표현합니다. 각 노드에서의 점수는 해당 위치에서 블록을 놓았을 때 얻을 수 있는 점수로 계산됩니다. 이러한 방식으로 최적의 위치를 추천하는 기능을 구현합니다.

Tree 구조의 장점은 다음과 같습니다:

1. **효율성**: 모든 가능한 블록의 위치와 상태를 체계적으로 탐색할 수 있어, 최적의 블록 배치를 찾는 데 효율적입니다. 특히, 블록의 배치 후 얻을 수 있는 점수를 계산하여 최대 점수를 얻을 수 있는 위치를 추천할 수 있습니다.
2. **가지치기(Pruning) 가능**: 점수가 낮은 브랜치는 조기에 제거(가지치기)하여 탐색 공간을 줄일 수 있습니다. 이를 통해 탐색 시간을 크게 단축할 수 있습니다.

단점은 다음과 같습니다:

1. **공간 복잡도**: 모든 가능한 위치와 상태를 나타내기 위해 많은 노드를 저장해야 하므로, 공간 복잡도가 높습니다. 특히, 테트리스 보드의 크기가 크거나 블록의 종류가 많을수록 더 많은 공간을 차지합니다.
2. **비효율적인 탐색**: 가지치기를 하더라도 여전히 많은 수의 노드를 탐색해야 할 수 있으며, 최적의 결과를 보장하기 위해 많은 계산이 필요할 수 있습니다.

**2.**

Tree 구조의 비효율성을 해결할 방법은 다음과 같습니다:

1. **휴리스틱 함수 사용**: 탐색 과정에서 휴리스틱(Heuristic) 함수를 사용하여 우선순위를 두고 탐색을 진행합니다. 예를 들어, 블록이 가장 낮게 쌓이는 위치나 줄 삭제 가능성이 높은 위치에 더 높은 우선순위를 부여하여 탐색할 수 있습니다. 이 방법은 탐색해야 할 노드의 수를 줄여 시간 복잡도를 감소시킬 수 있습니다.
2. **메모이제이션(Memoization)**: 이미 계산한 노드의 점수를 저장하고 재사용하는 기법을 사용합니다. 이는 동일한 상태에 대한 중복 계산을 방지하여 전체적인 계산 시간을 줄일 수 있습니다. 메모이제이션을 통해 효율적으로 공간을 사용하면서도 탐색 시간을 단축할 수 있습니다.

이와 같이, Tree 구조를 사용한 테트리스의 추천 기능 구현은 효율적인 탐색과 최적의 블록 배치 추천이 가능하지만, 공간 복잡도와 탐색 과정의 비효율성이 단점으로 작용할 수 있습니다. 이를 해결하기 위한 방안으로 휴리스틱 함수 사용과 메모이제이션 등을 적용할 수 있습니다.