10주차 결과보고서

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

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

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

**알고리즘 요약:** 테트리스 추천 알고리즘을 구현하기 위해 BFS(너비 우선 탐색)를 사용했다. 이 알고리즘은 현재 게임 보드 상태에서 가능한 모든 블록 배치를 탐색하여 최적의 배치를 추천한다. BFS를 통해 가능한 모든 배치를 탐색하여 가장 적합한 배치를 찾는다.

**자료구조:**

* **큐(Queue)**: BFS를 구현하기 위해 사용되며, 탐색할 보드 상태와 해당 상태까지의 경로를 저장한다.
* **게임 보드 배열**: 현재 게임 보드 상태를 저장한다.
* **상태 노드**: 각 상태는 게임 보드 배열과 현재 블록의 위치 및 회전 상태를 포함한다.

**시간 복잡도:**

BFS의 시간 복잡도는 노드(상태)의 수에 비례한다. 테트리스에서는 각 블록의 가능한 모든 위치와 회전 상태를 고려해야 하므로, 시간 복잡도는 아래처럼 나타난다. 𝑂(𝑏⋅𝑝⋅(𝑚⋅𝑛))*O*(*b*⋅*p*⋅(*m*⋅*n*)) 여기서 𝑏는 블록의 수, 𝑝는 각 블록의 가능한 회전 수, 𝑚과 𝑛은 게임 보드의 행과 열의 수이다.

**공간 복잡도:**

BFS의 공간 복잡도는 큐에 저장되는 노드의 수에 비례한다. 최악의 경우, 큐에는 모든 가능한 상태가 저장될 수 있으므로 공간 복잡도는 아래와 같다. 𝑂(𝑏⋅𝑝⋅(𝑚⋅𝑛))*O*(*b*⋅*p*⋅(*m*⋅*n*))

**2.**

**향상된 점:**

* **효율성**: BFS를 사용하면 최적의 배치를 찾기 위해 가능한 모든 상태를 체계적으로 탐색할 수 있다. 이는 무작위 탐색이나 깊이 우선 탐색(DFS)보다 효율적이다.
* **최적화**: BFS는 최단 경로를 보장하므로, 가장 적은 움직임으로 블록을 배치할 수 있다.

**향상되지 않은 점:**

* **공간 사용**: BFS는 많은 메모리를 사용한다. 트리 구조에서 모든 가능한 상태를 저장하므로, 메모리 사용이 비효율적일 수 있다.
* **시간 소요**: 모든 가능한 상태를 탐색해야 하므로, 실행 시간이 오래 걸릴 수 있다.

**3.**

**습득한 내용:**

* **알고리즘 설계 및 구현**: BFS를 사용하여 테트리스 게임의 최적 배치를 찾는 알고리즘을 설계하고 구현하는 방법을 배웠다.
* **자료구조의 활용**: 큐와 배열을 사용하여 게임 보드 상태와 블록 배치를 효율적으로 관리하는 방법을 익혔다.
* **문제 해결 능력**: 복잡한 문제를 작은 단위로 나누어 해결하는 방법을 학습했다.

**느낀 점:**

* **체계적인 접근의 중요성**: BFS를 통해 체계적으로 모든 상태를 탐색함으로써 최적의 결과를 얻을 수 있음을 깨달았다.
* **효율성 고려**: 알고리즘의 시간 및 공간 복잡도를 고려하여 구현하는 것이 중요하다는 것을 느꼈다.
* **실습의 가치**: 직접 알고리즘을 구현하고 테스트해보는 과정에서 많은 것을 배울 수 있었고, 이론을 실제로 적용하는 경험이 매우 유익했다.

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