10주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231515 이름: 김다은

1. 교재를 참조하여 테트리스 프로젝트 3주차에 구현하는 추천 기능은 어떤 원리로 작동되는지 설명하시오. 그리고 추천 기능을 구현하는 tree 구조의 장점(효율성)과 단 점(비효율성)을 기술하시오.

3주차에는 블록 추천 기능을 구현한다. VISIBLE\_BLOCK만큼의 nextblock를 고려하여, 현재 블록을 놓을 최적의 위치를 추천해주는 기능이다. 이 기능을 구현하기 위해 tree 자료구조를 활용한다. 먼저 level 0인 노드에서 시작해서 블록의 모든 회전의 경우와 놓이는 위치의 경우를 찾는다. 이후 다음 블록의 모든 회전의 경우, 놓이는 위치의 경우를 찾는 과정을 반복한다. VISIBLE\_BLOCK 만큼 nextblock를 고려하였다면, 그 중 제일 높은 점수를 받을 수 있는 현재 블록의 위치를 추천해주는 원리이다.

tree 구조를 이용할 경우의 장점은 루트 노드에서 시작해서 depth1, depth2 … …의 child node까지 구조적으로 순차적으로 고려하게 되므로, 경우의 수를 효율적으로 계산 가능하다.

tree 구조를 이용할 경우의 단점은 VISIBLE\_BLOCK의 값이 증가함에 따라 고려해야 하는 경우의 수가 기하급수적으로 늘어난다는 것이다. 따라서 블록의 추천 위치가 산출되기까지 걸리는 시간또한 기하급수적으로 증가한다.

2. Tree 구조의 비효율성을 해결할 방법에 대해서 2가지 이상 생각하고, 그 idea를 기술하시오.

**2-1. Pruning Tree 기법 사용**

게임에서 블록의 모든 경우의 수를 계산하게 될 경우, 너무 많은 메모리가 사용되고 연산시간이 오래걸린다는 치명적인 단점이 존재한다. Pruning Tree 기법이란 최적의 해가 될 확률이 적은 경우의 수는 제외하고 연산을 진행하는 방법을 말한다.

모든 경우의 수를 고려하는 것이 아니므로, 전체 경우의 수를 고려했을 때보다 정확도가 떨어진다. 하지만 게임에서는 지연 현상이 발생하지 않도록 하는 것이 중요하다. 따라서 즉각즉각 추천 블록의 위치가 나타나도록 Pruning Tree 기법을 사용하여 연산 시간을 단축하는 것이 효율적이다.

**2-2. 게임 필드를 1차원 배열로 관리**

현재 게임 필드를 2차원 형태로 저장하여 관리하고 있다. 하지만 2차원 배열의 경우 메모리도 많이 차지하고, 블록을 놓을 때마다 2차원 배열 전체를 탐색하게되므로 시간복잡도 측면에서도 비효율적이다. 이 문제를 해결하기 위해서 2차원 배열 대신 1차원 배열을 만들고, 배열의 원소로 해당 index를 column 값으로 하는 field의 맨 위의 블록 위치를 저장하도록 하여 구현할 수 있다.

하지만 이렇게 1차원 배열을 사용할 경우, 맨 위 블록 아래로 공간이 존재하더라도 이를 고려하여 추천 블록 위치를 계산하지 못하는 문제가 있다. 하지만 마찬가지로 게임에서는 지연 현상이 발생하지 않도록 하는 것이 중요하다. 따라서 즉각즉각 추천 블록의 위치가 나타나도록 1차원 배열을 사용하여 연산 시간을 단축하는 것이 효율적이다.