10주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20192135 이름: 윤영인

**1.**

1) RecNode 구조체

텍스트, 스크린샷, 폰트, 디스플레이이(가) 표시된 사진

자동 생성된 설명

트리를 구성하는 노드 구조체이다. 트리의 depth를 나타내는 level, 누적된 점수를 기록하는 accumulatedScore, 블록이 놓인 테트리스 필드를 나타내는 recField, 트리의 각 root에서 자식을 가리키는 포인터 child, 블록이 놓일 추천 위치의 회전, x , y좌표를 저장하는 r, x, y로 구성되어있다.

2) recommend 함수

recommend 함수는 BFS로 구현하였다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

인자로 받아온 root의 자식 노드를 CHILDREN\_MAX만큼 동적할당 하고, 전체 필드를 돌면서 블록이 회전하면서 최대로 내려갈 수 있는 y좌표를 찾는다. y 좌표를 찾았으면 RecNode 타입의 새로운 노드를 할당해서 root의 정보를 담고, level을 1 증가한다. 사용자에게 추천할 nextBlock을 AddBlockToField으로 필드에 추가하고 맞닿은 면적을 계산하여 accumulatedScore에 저장한다. 갱신된 recField를 DeleteLine으로 꽉 찬 줄을 찾아 삭제하고 accumulatedScore에 저장한다. 현재 nextBlock이 놓일 위치 및 회전 수를 newNode의 x, y, r에 저장한다. root의 정보를 자식 노드에 저장하는 makeNewNode 함수는 아래의 사진과 같다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

새로운 노드인 newNode를 할당하여 root의 level보다 1만큼 큰 level을 저장하고 root의 accumulateScore 및 recField를 newNode에 저장하여 리턴한다. 이 함수의 시간 복잡도 및 공간 복잡도는 O(1)이다.

CHILDREN\_MAX만큼 자식 노드를 할당하고 root의 정보를 담아 점수를 갱신했다면 selectionSort 함수로 accumulatedScore을 기준으로 하여 자식 노드를 내림차순으로 정렬한다. selectionSort 함수는 아래와 같다. selectionSort 함수의 시간 복잡도는 O(size \* size) 즉 O(N^2)이고, 공간 복잡도는 O(1)이다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

자식 노드들이 담긴 배열을 받아와서, 자식 노드의 개수만큼 배열을 돌면서 각 노드의 accumulatedScore을 비교하면서 accumulatedScore가 큰 순서대로 가진 노드를 배열한다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

선택 정렬로 자식 노드를 정렬했다면, MAX\_BFS만큼, 즉 10개의 자식 노드를 뽑아 각 자식 노드를 루트로 하여 recommend 함수를 재귀를 돈다. 이때 현재 자식 노드의 accumulatedScore이 max보다 크다면 max를 갱신한다. 만약 root의 level이 1이라면 재귀가 모두 끝나고 올라오면서 트리의 부모 노드인 root이므로 전역 변수인 recommendR, recommend, recommendY를 root의 r, x, y로 갱신한다. 10개의 자식 노드 외의 나머지 노드들은 free를 하여 pruning branch를 수행한다.최종적으로 가장 큰 accumulatedScore이 담긴 max 값을 반환한다. 따라서 최종적으로 recommend 함수는 최대 CHILDREN\_MAX 만큼 재귀 호출이 발생하고, 각 호출은 NUM\_OF\_ROTATE \* (WIDTH + 2 \* BLOCK\_WIDTH) \* HEIGHT 만큼 반복문을 돈다. 따라서 시간 복잡도는 O(CHILDREN\_MAX \* NUM\_OF\_ROTATE \* (WIDTH + 2 \* BLOCK\_WIDTH) \* HEIGHT)이다. 또한 공간 복잡도는 RecNode가 CHILDREN\_MAX 만큼 동적할당 되므로 O(CHILDREN\_MAX)이다.

3) InitTetris, BlockDown

recommend 함수를 수행하기 위해선 InitTetris 함수와 BlockDown 함수에서 트리를 초기화 해야한다.

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

InitRecTree 함수는 트리의 root를 할당하여 트리를 초기화를 하는 함수이다. 점수 및 recommendX/Y/R을 0으로 초기화하고 level을 1로 설정, 현재 필드 값을 할당한 root의 recField에 복사하여 할당한 root 노드를 리턴한다.

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

InitTetris 함수에서 트리를 초기화하여 recommend 함수를 수행하였다. 또한 nextBlock을 BLOCK\_NUM만큼 할당하여 트리의 깊이에 따라 다음 블록의 위치를 추천하도록 하였다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

BlockDown 역시 트리를 초기화하여 recommend 함수를 초기화하였다. 또한 nextBlock 함수를 BLOCK\_NUM만큼 갱신하여 트리의 깊이에 따라 다음 블록의 위치를 추천하도록 하였다.

4) DrawRecommend

텍스트, 스크린샷, 폰트, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

인자로 받아온 좌표 및 회전 수로 블록을 ‘R’로 표시하여 필드에 그린다.

5) DrawBlockWithFeatures

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

블록이 그려질 때 DrawBlockWithFeatures에서 블록의 추천 위치를 표시하도록 DrawRecommend 함수를 추가하였다.

**2.**

구현한 recommend 함수에선 현재 level에서 트리의 자식 노드를 accumulated가 큰 순서대로 정렬한 후 점수가 가장 큰 자식 노드 10개를 뽑는다. 뽑힌 자식 노드를 root로 하여 recommend 함수를 재귀를 돌면서 점수를 누적하고 최종적으로 점수가 가장 큰 경로를 찾아 사용자에게 위치를 표시한다. 따라서 모든 경우를 고려한 트리 구조보다 가장 큰 경우를 미리 찾아 추천할 경로를 탐색하므로 시간 및 공간 복잡도가 많이 들지 않는다는 장점이 있다. 하지만 아무리 최대의 점수를 가질지라도 완벽하게 최적의 경로를 추천한다고 보장할 수 없다는 단점이 있다.

**3.**

테트리스 프로젝트 3주 과정을 진행하면서 ncurses library를 사용하여 테트리스 게임을 직접 만들어 보았다. 이 과정동안 연결리스트와 트리 자료구조에 대해 자세하게 파악할 수 있었다. 뿐만 아니라 BFS를 사용한 트리 탐색 구조를 사용하여 최대 점수를 얻을 수 있는 최적의 경로를 찾는 pruning branch 알고리즘을 공부할 수 있었고 알고리즘 및 자료구조에 대해 더 자세하게 공부할 수 있는 시간이었다.