10주차 예비보고서

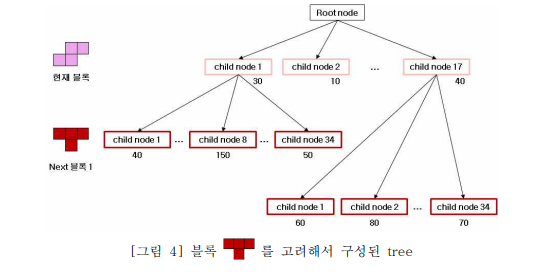
전공: 아트엔테크놀로지 학년: 3학년 학번: 20191172 이름: 함승우

1. 교재를 참조하여 추천 기능은 어떤 원리로 작동되는지 설명하고, 추천 기능을 구현하는 tree 구조의 장점과 단점을 기술하시오.

추천 기능은 현재에 떨어지고 있는 블록과 그 이후의 블록들의 정보와, 이 블록들의 회전수, 그리고 필드에 위치할 수 있는 경우의 수를 통해 블록이 어디에, 어떤 회전수로 놓이면 좋을 지에 대한 추천 시스템이다. 이 모든 경우의 수 중에서 가장 높은 점수를 얻을 수 있는 위치와, 회전수를 추천 시스템을 통해서 제공한다. 이 모든 경우의 수는, tree를 통해서 구현할 수 있는데, 이 tree 구조의 장점과 단점을 기술한다면

추천 기능을 tree로써 구현을 한다면 추천 기능을 구현하는 tree 구조의 장점은 많은 정보를 저장할 수 있다는 점이다.

강의 자료의 이미지를 첨부하자면,



이 그림 4의 경우 현재의 블록에 대한 경우의 수 17개에 대한 nextblock의 34가지의 경우의 수를 저장하고 있다. 이는 사용자가 직관적으로 자료를 이해할 수 있게 하고, 체계적으로 정보를 저장해 효율적이라고 볼 수 있다.

추천 기능을 tree 구조로 구현하는 단점은 시간적 비효율성과 공간적 비효율성이 있다.

각 블록에 대한 회전수, 위치를 tree로 설정을 하며, 높은 점수의 정확도를 올리기 위해서는 tree의 depth가 깊어야 한다. 즉 고려하는 block이 현재 떨어지는 block뿐 아니라 그 이후로 고려하는 block이 많아질수록 정확도는 증가한다. 하지만, 이 많은 데이터들을 비교하고 저장하기 위해서는 매우 시간 복잡도가 증가하고, 공간 복잡도도 증가하게 된다. 이는 게임의 진행을 막기도 하며, 데이터를 많이 잡아먹어 다른 프로그램의 진행을 막기도 한다.

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

위에서 언급한 tree 자료구조의 단점들을 해결할 방법으로는 강의자료에서 언급한 pruning tree가 있다. 우선 pruning tree는 tree의 가지를 자르는 방식인데, 작은 누적 점수를 가지는 가지들을 자르는 방식이다. 이는 데이터를 저장할 필요도, 읽을 필요도 없기에 시간 복잡도면에서 이득이 된다고 볼 수 있다. 하지만, 만약 level 1에서 pruning tree를 했을 경우에 level 2에서 가지 치기 당한 부분에서 큰 누적 점수를 갖는 경우가 있을 수 있다. 물론 완벽한 전술을 짜지는 못하지만, 이 level 1에서 가지치기를 하는 것이 아닌 더 낮은 level에서 가지치기를 통해 최대한 높은 점수를 가질 수 있는 프로그램을 만들 수 있다.

추가로 레벨을 구성하는 트리를 다르게 하는 것이다. 우선, 강의자료에서 제시한 트리는 하나의 블록에 대해서 회전수를 고려하고 위치를 고려하는 것이지만, 미리 회전수에 따라 현재 블록과 다음 블록이 결합된 상태를 분석해서 자료구조를 만드는 것이다. 테트리스 게임에서 높은 점수를 얻는 방법은 여러가지가 있고, 현재 블록이나 다음 블록에서 판단이 불가능하지만, 불변의 법칙은 최대한 빈칸 없이 구성하는 것이다. 빈칸 없이 구성한 현재 블록과 다음 블록의 결합된 모양으로 미리 트리의 구성요소를 구성한다. 이는 블록 하나씩 구성하여 데이터를 읽는 것보다는 미리 구성 요소를 만들어야 하기 때문에 효율적이라고 볼 수 있다.

마지막으로 강의자료에서 제시한, 공간 복잡도를 줄이기 위한 방법은 data simplification이다. 예를들어, 테트리스의 필드의 y축, 즉 높이만 보는 방법이다. 필드의 높이만 생각하는 이유는 gameover가 되는 조건은 블록의 현재의 위치인 (i+blockY)의 좌표가 -1이 될 때인데, 이 gameover를 피하며 계속 게임을 진행하는 것이다. x축, 즉 가로의 길이를 생각하지 않으니 데이터적인 측면으로 장점이 있다.