

## 20011742 권혁재 컴퓨터 그래픽스 과제3\_Maze Navigator 보고서

이번 과제의 핵심은 A-star 알고리즘이었다. A-star 알고리즘은 최단 경로를 찾는 데에 사용되는 탐색 알고리즘으로, 현재 상태의 비용을  $g(x)$ , 현재 상태부터 다음 상태로 이동할 때의 함수를  $h(x)$ 라고 할 때, 둘을 더한 값이 최소가 되는 지점을 우선적으로 탐색하는 방법이다. 둘을 합한 함수를  $f(x)$ 라고 할 때, 다음과 같은 방식으로 수행 가능하다,  $f(x)$ 를 오름차순 우선순위 큐에 삽입한후, 큐에서 최우선 노드를 pop 함수를 사용해 pop하고, 그 노드에서 이동 가능한 노드를 찾은 후 노드들의  $f(x)$ 를 구한 뒤, 다시 우선순위 큐에 삽입. 이방식을 반복해서 수행한다.  $h(x)$ 가 정확할수록 더 효율적으로 동작하며, 더욱더 좋아진다. 이는 코드 내의

```
void astar() {  
  
    Open.clear();  
  
    Close.clear();  
  
    Goal.clear();  
  
  
    start = TakeIndex();  
  
    pair<int, int> currentIndex = start;  
  
  
    node first;  
  
  
    first.index = currentIndex;  
  
    Close.push_back(first);  
  
  
    int f = 0;  
  
    int s = 0;  
  
    while (1) {  
  
        FindOpen(currentIndex);
```

```

currentIndex = SetClose();

if (currentIndex == goal) {
    break;
}

if ((GetAsyncKeyState('O') & 0x8000) == 0x8000) {
    break;
}

}

pair<int, int> parentIndex;

parentIndex = Close[Close.size() - 1].parent;
Goal.push_back(Close[Close.size() - 1]);

for (int i = Close.size() - 1; i >= 0; i--) {

    if (parentIndex == start) {
        Close[i].child = Goal[Goal.size() - 1].index;
    }

    if (parentIndex == Close[i].index) {
        Goal.push_back(Close[i]);
        parentIndex = Close[i].parent;
    }
}

```

```
}
```

```
for (int i = 1; i < Goal.size() - 1; i++) {  
    Goal[i].child = Goal[i - 1].index;  
}
```

} 함수에서 정확하게 확인할 수 있다.

시행착오:

Astar알고리즘을 구현할 때, 리스트를 관리를 제대로 하지 못하여 동일한 노드를 반복확장하여 무한루프가 발생하였다

해결 아이디어: List를 Closed와 Open으로 나누어 불필요한 확장을 방지하였다.

시행착오:

Path를 보여주는 것을 설정할 때, exe 프로그램처럼 반짝이는 빨간 길을 사용하고 싶어  
서 g\_time을 섞어서 코드를 짰으나, 색상이 이상하게 나오고 오류가 발생하였다.

해결 아이디어: 반짝이는 것까지는 구현하기 힘들다고 생각해 단순히

glUniform4f(uColor, 1.0f, 0.0f, 0.0f, 1.0f); 를 이용하여 루트가 빨강게 나오도록 설정하였다.

시행 착오:

카메라를 회전하는 것을 구현하였을 때, 오른쪽 미로와 카메라 사진이 있는 부분의 카메라는 회전하지 않았다.

해결 아이디어: drawCamera 함수에서 turn부분 관련 변수를 사용하여 처리하였다.