

# 컴퓨터 그래픽스 과제3

18011683 조현우

## A-Star 알고리즘

- A-star 알고리즘은 다익스트라 알고리즘과 비슷한데 가장 최소의 코스트로 도착한 곳부터 탐색하는 것으로 현재 상태의 비용을  $g(x)$ , 현재에서 다음 상태로 이동할 때의 휴리스틱 함수를  $h(x)$ 라고 하면  $f(x) = g(x) + h(x)$ 가 최소가 되는 지점을 우선으로 탐색합니다. 그렇기에 우선순위 큐 방식으로 구현됩니다. 다익스트라 같은 경우는 굉장히 큰 공간에서 사용하면 비효율적이지만 a-star 알고리즘을 사용하면 더 효율적으로 길찾기가 가능합니다.
- $H(x)$ 를 저는 Manhattan distance를 이용하여 목적지 좌표를  $p$ , 비교할 좌표를  $q$ 라고 할 때  $|p.x - q.x| + |p.y - q.y|$ 를 통해 코스트를 구하였습니다.
- 알고리즘의 실행 방식은 다음과 같습니다.
  1. P = 시작점 및  $f, g, h$  값을 갖고 있는 노드
  2. P에  $f, g, h$  값 넣어주기
  3. OpenList에 P노드 넣기
  4. B = OpenList에서 가장 낮은 코스트인  $f$  값을 가진 노드
    - a. B가 목표점이면 경로 완성
    - b. OpenList의 사이즈가 0이면 경로가 존재하지 않습니다..
  5. C 는 B에 연결된 노드들 순회
    - a. C에  $f, g, h$  값 할당
    - b. OpenList/CloseList에서 C가 있는지 체크
      1. 있으면 더 낮은  $f$  값으로 경로 업데이트
      2. 없으면 C를 OpenList에 넣기
    - c. 5번으로 돌아갑니다.
  6. 4번으로 돌아갑니다.

## 수행과정 및 해결 아이디어

- **카메라의 움직임 조작:** A, D 키를 눌렀을때 회전 구현은 viewDirection 벡터를 y축 기준 zx 평면 회전을 사용하였고 얼마만큼 회전했는지 각도를 구하기 위해 추가적으로 viewDirection과  $\text{vec3}(0,0,-1)$  사이의 각도를 구하여 그 각도 만큼 카메라 물체에 회전을

주었습니다. W, S키를 눌렀을 때 이동은 viewDirection벡터에 이동거리를 곱한 벡터를 카메라 좌표에 더해줌으로써 구현하였습니다.

- **카메라가 벽과 충돌:** 카메라가 벽과 충돌하는 것은 카메라의 중앙 물체의 위치에서 벽들과의 중심좌표까지의 거리가 (카메라의 사이즈/2 + 벽돌 사이즈/2) 보다 작다면 충돌이 일어났다고 판단하여 이동을 제한하였습니다.
- **Q클릭시 경로 표시:** a-star알고리즘을 통해 최적의 경로를 탐색하고 카메라좌표에서 목표점까지의 경로좌표가 담긴 연결된 노드들을 하나씩 순회하면서 빨간 막대로 경로를 표시하였습니다. 현재 노드와 다음 노드간의 좌표를 비교하여 방향에 따라 막대를 조금씩 이동시켜(한칸 크기의 반) 연결되게 표현하였습니다.
- **스페이스바 클릭시 경로따라 이동:** 스페이스바를 클릭시에 경로의 시작점으로 카메라를 위치시키고 viewDirection도 알맞게 설정을 해줍니다. 그상태에서 현재노드와 다음 노드간의 방향벡터를 구하여 해당방향으로 조금씩 이동을 하고 현재노드와 다음노드의 방향벡터가 viewDirection벡터와 방향이 맞지 않으면 현재노드와 다음노드의 방향벡터로 viewDirection을 천천히 움직여 방향을 맞추어주었습니다.

## 시행착오

- 스페이스바 클릭시에 경로따라 이동을 구현할 때 viewDirection벡터와 다음 노드좌표로 이동할 방향벡터가 맞지 않으면 회전을 하게 하였습니다. 이때 조건을 float과 float끼리 일치하는지 비교하였는데 이때는 조금의 오차가 발생할 수 있어 생각대로 움직이지 않는 경우가 발생하였습니다. 그래서 이동할 방향벡터와 viewDirection벡터 간의 사이의 각도가 1도이상 벌어지면 방향이 같지 않다고 보았습니다.
- A-star알고리즘을 처음에는 옳지 않은 경로까지 포함된 경우로 경로가 출력되어서 이를 노드 연결로 변환하여 노드가 추가될때마다 이전노드를 연결시키는 방식으로 하였습니다. 이렇게 하면 옳지 않은 경로(목적지에 도달하지 못하고 끝난 경로)가 포함되지 않았습다. 또한 목적지에서 출발지까지의 방향으로 경로가 나오게 되고 이를 반대로 뒤집어 사용하였습니다.
- 목적지에서 q를 눌렀을 때 경로가 생성되지 않는데 이럴 경우에 잘못된 접근 오류가 발생하였습니다. 그래서 이에 대한 예외처리를 해주었습니다.

## 배운점, 느낀점

- A-star알고리즘을 처음 접해보았는데 다익스트라와 비슷하면서 조금 더 효율적이라는 부분에서 앞으로 많이 사용할 알고리즘일 것 같다는 생각이 들었습니다.
- 이번 과제를 하면서 가장 유의할 점은 float끼리 비교할 때 정확하게 맞아떨어지는 경우는 드물다는 점과 오차가 발생하는 경우를 잘 고려해야겠다는 생각이 들었습니다.