컴퓨터그래픽스 과제3 레포트

19011493 김준우

1. 접근 방법/아이디어

1) 카메라 회전/이동

- A, D키를 입력받아 rotate변수 값 변형
  - 음수이면 += 360, 양수이면 %= 360 (0~360 유지)
- display()의 ViewMat를 rotate만큼 RotateY
- W, S로 전진/후진 여부 입력
  - cos(rotate), sin(rotate)로 x축 속도, z축 속도 벡터 분해
  - 충돌이 일어나기 전까지 이동

2) 충돌

- 현위치의 왼쪽에 벽블록('*')이 있다면 leftLock = true
- 현위치의 오른쪽에 벽블록('*')이 있다면 rightLock = true
- 현위치의 위쪽에 벽블록('*')이 있다면 upLock = true
- 현위치의 아래쪽에 벽블록('*')이 있다면 downLock = true
- 카메라가 이동할 때 lock이 잠겨있는 방향으로는 속도 0으로 초기화

3) A* 알고리즘

- A*을 이용한 최단경로 찾기 알고리즘은 pathFinding() 함수에 구현되어있다.
  - 시작지점에서부터 모든 방향으로 한 칸씩 이동하고, 도착 지점에 도착하면 최단경로를 얻어내는 방식
- 미로 내 모든 위치의 이동 거리를 최댓값(INT_MAX)으로 초기화
- 우선순위 큐 pq에 시작지점의 (이동 거리(비용), 위치 좌표) 노드 추가
- 벽을 피해 주변 맵 상하좌우로 한 칸 이동
  - (현재 이동 거리 + 1)이 (기존의 이동 거리)보다 작을 경우, 이동 거리를 업데이트하고 우선순위 큐에 추가

- 도착지점에 도달할 때까지 반복

- 도착지점 노드에서 부모 노드를 타고 출발지점까지 올라가고, 그 역순을 경로로 저장하면 출발지점-도착지점 사이의 최단경로 탐색 가능

4) 경로 표시

- 경로 내 n번째 지점과 n+1번째 지점을 양 끝으로 하는 빨간 선을 미로 위에 표시

  - n번째 지점과 n+1번째 지점의 x좌표가 같을 경우 세로로 길게, 다를 경우 가로로 길게

- 빨간 선의 색상은 sin함수 내에 g_time, 시작 지점으로부터 떨어진 거리 i를 넣어 빨간 그라데이션이 파도처럼 지나가는 효과 부여

5) 자동 이동

- 카메라가 다음으로 이동할 좌표를 획득

- 해당 좌표의 방향에 맞춰서 이동은 정지한 채카메라를 회전

- 반복

2. 마주친 문제/해결방안

1) 카메라 이동을 구현하는 과정에서 분명 x축 y축 속도 분해를 잘한 것 같았는데 실행하고 ADWS를 누르면 카메라가 난데없이 엉뚱한 방향으로 이동했다.

   A. 카메라의 회전 각도 rotate를 sin, cos 함수 안에 degree(0~360) 단위 그대로 넣었었다. rad으로 변환해준 후 넣으니 정상적으로 실행됐다.

2) 스페이스바를 한 번만 눌렀음에도 카메라가 이동했다 멈췄다를 반복했다.

   A. 스페이스바를 GetAsyncKeyState로 입력받았기 때문이다.
   체감상 GetAsyncKeyState는 키가 눌려있는 한 매 프레임 input이 들어가는 것 같아서, 한 번 입력받고 약간의 딜레이가 생겨 단타 클릭에 더 적합한 glutKeyboardFunc(myKeyboard);에서 입력받는 것으로 변경했다.

3) 자동이동을 켜면 카메라가 한 블록만 이동한 채 프로그램이 멈춰버렸다.

   A. 최단경로를 매 프레임 계산하게끔 코드를 짜두었기 때문에 카메라의 좌표가 바뀐 순간 최단 경로가 바뀌었고, 현재 이동하고 있던 자동이동 알고리즘에 문제가 생겼다. 최단경로를 매 프레임 계산하는 것이 아닌 Q를 눌렀을 때에만 계산/갱신하는 것으로 수정해 해결하였다.