0번ppt(-------------------------------------------------------------------------------------------------->)

안녕하세요 컴퓨터공학과 15학번 윤찬우입니다.

1번ppt(-------------------------------------------------------------------------------------------------->)

Skynet 에피소드2는, 네트워크상에 존재하는 에이전트가 출구를 빠져나가지 못하게 연결을 끊는 것이 게임의 핵심입니다.

저희의 알고리즘의 전체적인 흐름을 피피티에 있는 흐름도를 보면서 설명해 드리겠습니다.

메인 함수에서 그래프와 에이젼트의 위치를 FindEdge에 전달합니다.

FindEdge함수에서 BFS함수를 호출하여 에이젼트의 위치와 임의의 한 출구를 각각 시작점과 목적지로 전달합니다.

BFS함수에서 잘라줄 Node 및 거리 각 노드의 연결된 출구 수를 반환합니다.

그 후 FindEdge함수에서 후보군들 중 우선순위를 정하여 잘라야 할 최종 간선을 main함수로 반환해줍니다.

2번(----------------------------------------------------------------------------------------------------->)

저희의 전체적인 틀입니다.

구조체와 함수를 이용하여 구현하였습니다.

3번(----------------------------------------------------------------------------------------------------->)

그래프를 얻기 위해서 구조체로 Node를 구현하였습니다.

그리고 이 생성된 Graph에서 각 Node에 연결된 출구 수를 구해주는 NumOfGate함수를 구현하였습니다.

Main함수에서는 Node구조체를 이용하여 그래프를 그려주고, 각 Node들의 변수 값을 채워 주었습니다.

그리고 반복문을 통해 매번 에이젼트 위치와 FindEdge함수를 호출하여 잘라주어야 할 node 두 개를 받아 출력해줍니다.

4번(----------------------------------------------------------------------------------------------------->)

저희의 핵심 알고리즘인 BFS와 FindEdge함수가 있습니다.

우선 BFS를 보면, 너비 우선 탐색을 구현하면서 깊이랑 경로 그리고 경로상의 출구 수를 통한 위험도를 구할 수 있게 구현하였습니다.

다익스트라 알고리즘과 깊이 우선 탐색 알고리즘도 있는데 저희는 모든 거리를 구할 필요도 없고, 최단 경로를 구하는데 있어서 너비우선탐색을 선택하였습니다.

5번(----------------------------------------------------------------------------------------------------->)

다음은FindEdge함수입니다.

BFS에서 받아온 각 출구에 대한 정보들을 이용하여 우선순위를 두어 최종 간선을 선택하였습니다.

첫번째는 거리가 1인 후보가 있는지 확인합니다.

두번째는 2개의 출구와 연결된 Node에 대한 정보를 얻습니다.

세번째는 후보군을 각 종 sort함수를 통해 위험도로 1차 정렬을 하고 거리로 2차정렬을 한 후, 에이전트와 가까운 2개의 후보만 비교를 합니다.

네번째는 2개의 gateway룰 가진 Node가 하나일 경우 우선순위로 선정합니다.

다섯번째는 가까운 출구에 연결된 Node를 찾습니다.

마지막으로 제거해야할 두 Node를 main함수로 반환합니다.

이렇게 우선 순위를 두어 최종 선정을 하였습니다.

6번(----------------------------------------------------------------------------------------------------->)

지금 보시는 Sort함수들은 제가 각각의 우선순위를 정하는데 맞춰서 쓰기 위해 BubbleSort알고리즘으로 직접 구현하였습니다.

감사합니다.

끝(----------------------------------------------------------------------------------------------------->)