**[현명한 선택] – STL**

1. vector가 배열에 비해 가지는 장점과 단점

vector의 장점은 자동으로 메모리를 할당해준다는 점이다. 값을 삽입하거나 삭제할 때 메모리를 할당하거나 해제하기 때문에 벡터를 정의할 때 크기를 지정할 필요가 없기 때문에 메모리를 효율적으로 관리할 수 있다. 하지만 벡터는 배열에 비해 삽입과 삭제의 속도가 느리기 때문에 많은 양의 데이터를 다뤄야 하는 경우에는 적합하지 않다.

2. Vector와 List의 차이점

연속적인 메모리로 이루어진 벡터와 다르게 리스트는 double linked list로 이루어져 있기 때문에 원소를 인덱스로 접근 할 수 없고 특정 원소에 접근하려면 처음이나 끝부터 선형탐색을 해야 한다. 그리고 벡터는 컨테이너 끝이 아닌 곳에 삽입/제거 시 비용이 많이 들지만 리스트는 어느 위치에서도 삽입/제거가 빠르고 메모리를 재 할당할 필요가 없다.

3. vector나 list를 써도 되는데 굳이 stack이나 queue를 쓰는 이유

스택과 큐는 데이터를 추가하는 방향과 빠져나오는 방향이 정해져 있기 때문에 용도에 맞게 사용하면 더 효율적이고 빠르게 데이터를 처리할 수 있다.

4. set과 map은 구현이 비슷한데 둘이 나눠진 이유

set와 map는 중복된 값을 허용하지 않는 다는 점은 같다. 하지만 map는 key와 value를 가지는데 set는 key만 갖는다. map을 사용하면 키를 기준으로 정렬되고 키를 이용하여 데이터를 찾을 수 있다.

5. 다음 상황에 대해 어떤 자료구조가 좋을 지 단답형으로 적은 후 간단한 이유를 적어주세요. (복수정답 존재, 그러나 1개라도 이유가 합당하면 OK)

- 캐릭터의 대사를 저장하고 출력하는 경우

배열: 게임 내에서 수정될 일이 없기 때문에 메모리 크기가 정해진 배열을 사용하는 것이 좋다.

- 추가, 삭제, 수정이 빈번한 플레이어가 가지고 있는 종류가 다양한 아이템을 관리하는 경우

리스트: 수정이 빈번하고 많은 아이템을 다루어야 하기 때문에

- 위치와 칸이 정해져 있는 아이템창의 정보를 관리하는 경우

벡터: 위치가 정해져 있으면 인덱스로 접근해야 하기 때문에 벡터가 적절하다.

- string 또는 enum 값을 통해 효과음을 재생해야 하는 경우

배열: 효과음은 메모리에 빠르게 접근하는 것이 중요하게 때문에

- 플레이어 주변 특정 범위에서 다수의 적 종류를 판단해야 하는 경우

맵: 위치기준으로 정렬하고 해당 적의 종류까지 저장해야 하기 때문에

- 플레이어 주변 특정 범위에서 다수의 적 위치를 감지해야 하는 경우

세트: 위치 기준으로 정렬되면 다루기 편하기 때문에

- 행동 횟수 제한이 있는 게임에서 바로 이전 상태로 되돌리기 기능이 필요할 경우 (다수호출 가능)

스택: 마지막에 들어온 상태가 가장 먼저 나가기 때문에