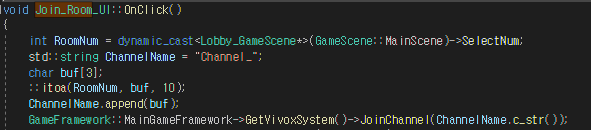
**2018180020 박재우(클라이언트) 30주차 기록**

**방 별로 Vivox 연결**

로비에서 방을 선택하여 입장하는 기능이 만들어졌으니 방에 입장할 때 Vivox를 연결할 수 있도록 만든다. 먼저 테스트를 위해 Vivox.cpp에서 자동으로 로그인부터 채널에 연결하는 것까지 만들었었는데 로그인까지는 게임이 시작될 때 자동으로 하도록 하고 채널입장을 버튼을 눌렀을 때 하도록 만들기 위해 채널 입장하는 부분을 삭제했다.

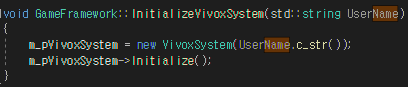


서연이가 만든 입장버튼의 클릭이벤트 함수이다. 입장버튼을 누르면 그 UI의 OnClick함수가 호출되는데 이곳에서 연결을 하도록 만들었다. 입장버튼을 누르기 위해서는 일단 방 목록을 먼저 선택을 할 것이다. 방 목록을 선택하면 방 목록 UI의 OnClick함수에서 그 방의 번호가 로비 Scene의 SelectNum에 저장하도록 되어있다. 방 번호는 유일한 값이기 때문에 이 값으로 채널을 구분하면 같은 방에 있는 사람끼리 통화가 가능하다. 그래서 Scene의 SelectoNum에 접근해서 이 값을 문자열로 바꾼 뒤에 Channel\_ 문자열 뒤에 붙여줬다. 그럼 Channel\_0, Channel\_1처럼 뒤에 방번호가 붙은 문자열이 완성된다. 이 문자열을 채널 이름으로 하여 VivoxSystem의 JoinChannel 함수를 호출한다.

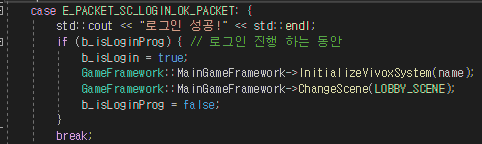


Vivox 채널에서 나갈 때는 Leave\_Room\_UI의 OnClinck함수에서 해준다. 원래 LeaveChannel은 채널의 이름을 받았었는데 그럴 필요가 없었다. 처음에는 SessionHandle에 채널 이름이 들어가는 줄 알았지만 아니었고 SessionHandle은 채널에 입장할 때 Listen 함수에서 반환되고 그것을 저장해서 가지고 있기 때문에 그냥 LeaveChannel함수를 부르기만 하면 음성채팅을 끊을 것이다.

방 별로 채널을 나누는 작업을 다 했다. 이제 유저의 이름을 받아서 그 이름으로 Vivox System을 초기화 해야 한다. 로그인 할 때 유저의 이름이 중복되면 안되기 때문이다. 유저의 이름은 게임을 시작하고 로그인을 할 때 확정된다. 플레이어의 이름은 로그인창에서 엔터키를 누를 때 저장된다고 한다. 그렇다고 엔터키를 누를 때 Vivox를 초기화하면 중복으로 계속 초기화 되는 상황이 올 수 있다. 그럼 로그인에 성공했을 때 초기화를 해준다면 이미 엔터키가 눌려서 유저의 이름은 저장되어 있고 게임에도 정상적으로 접속할 수 있으니 알맞은 타이밍에 초기화를 해줄 수 있을 것이다. NetWorkMGR에 로그인 성공 패킷을 받는 부분에서 초기화를 진행해 주면 된다.



먼저 프레임 워크의 생성자에서 해주던 Vivox초기화 부분을 따로 옮겨 함수로 만들었다. 그리고 생성할 때 유저 이름을 넘겨받도록 만들고 초기화를 하도록 했다. 생성자에서 받아온 유저의 이름을 이용해 URI 등 필요한 값을 만든다.



만든 함수를 로그인 성공 패킷을 받아왔을 때 호출해준다. Scene을 바꾸기 직전에 Vivox를 초기화 해준다.

**물**

맵에 물이 고여 있는 공간이 있다. 이를 표현하기 위해 Water 클래스를 만들고 이 물은 블랜딩을 하여 반투명 하게 그리도록 한다. 쉐이더 객체에서 Blend State를 수정한다. BlendEnable 옵션을 True로 해주고 SrcBlend와 DestBlend를 더하도록 만들어 반투명 하게 보이게 한다. Hlsl쉐이더에서는 단순하게 위치와 uv를 받아 텍스쳐를 그리도록 했고 물이 흐르는 걸 표현하기 위해 시간을 기반으로 uv 값을 지속적으로 더해줬다.

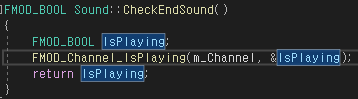




각 맵에 알맞은 위치에 물을 배치하였다.

**Sound**

이제 게임속에서 일어나는 행동들의 소리를 추가한다. 소리가 나는 경우는 거의 애니메이션이 재생될 때(걷기, 공격하기 등) 소리가 나기 때문에 저번에 만든 애니메이션 이벤트를 이용해 소리를 재생하도록 한다. 그 전에 Sound 리스트에서 재생이 끝난 객체를 자동으로 제거하는 코드를 만든다.



소리의 재생이 끝났을 때 false를 반환하고 계속 재생 중이면 true를 반환하도록 함수를 만들었다. 소리는 Game Scene에서 리스트로 관리를 하고 있기 때문에 update함수에서 이 함수가 false를 반환하는 곳을 찾아 지워주도록 한다.



컨테이너의 iterator를 이용한다. for문에서 iterator가 end가 아닐 동안 루프를 돌게 한 뒤 만든 Check End Sound 함수가 false를 반환할 때 erase 함수를 이용해 그 객체를 지워준다. Erase 함수는 제거한 객체의 다음 주소를 반환하기 때문에 iterator에 반환된 값을 넣어준다. 만약 true라고 한다면 iterator를 다음으로 옮겨준다.



플레이어 생성자에서 발자국 소리를 내기 위한 이벤트를 추가한다. 이벤트 함수에서는 발자국 소리를 재생하여 걷거나 뛸 때 발자국 소리가 나도록 했다. 시간도 잘 조절해서 발이 딱 땅에 닿을 타이밍에 재생하도록 만들어서 더 자연스럽게 했다.

다른 사운드들도 같은 방법으로 추가했다.