		작성자	2022180007 노훈철 2021182032 임동건 2021182037 진성준	**팀명**	Pulse
주차	**1,2**	**기간**	**2025.09.01 ~ 2025.09.11**	**지도교수**	**정 내 훈**(서명)
이번주 한일	**노훈철: 텍스트 출력 기능 제작.** **임동건: UI (Text 출력 포함) => HP Bar, NPC HP Bar, 총알 장탄수, Kill/Death 카운터, 플레이어 이름, hp bar 출력** **진성준: 다른 플레이어가 클라이언트에서 보이지 않는 오류 수정, 사망시 NPC 제거되지 않는 오류 수정, 플레이어가 죽으면 리스폰, UI에 필요한 정보 보내주기**				

<상세 수행내용>

노훈철: 텍스트 출력 기능 제작.

D3D12에서 텍스트 출력을 하기 위한 방법은 여러가지가 있지만, DirectX 공식 샘플에 나와있던 방법은 D3D11과 Direct2D를 사용하여 텍스트 렌더링을 하는 방법이었다.

하지만, DirectX11의 CPU와 GPU 동기화 방법이 D3D12와 매우 다르기 때문에, 렌더링을 하면 GPU가 다 끝나고 내부의 동기화 객체가 있을 것이라 생각되었다. 그것으로 텍스트 출력을 하게 된다면, 텍스트의 활용도는 스크린에서만 출력이 가능하고, D3D12의 게임자체의 GPU동기화와 D3D11의 동기화가 같이 돌아가 결국 렌더 한번에 두번의 Fence객체가 동기화를 진행하는 요상한 모양이 될 것 같아서, 결국 TTF 데이터를 해석하여 글자의 Texture를 만들기로 했다.

오픈소스 : (https://github.com/kv01/ttf-parser)

이렇게 결정한 이유는 다음과 같다.

1. ttf 정보를 사용하게 되면, 텍스트를 2d 텍스쳐 뿐 아니라, Mesh로도 표현이 가능해 활용성이 높아진다. (Mesh로 텍스트를 출력하는 작업은 이미 3D 프로그래밍 1 과제에서 해놔서 그걸 가져오면 된다.)

2. 글자의 렌더링 알고리즘 전체를 모두 만들기 때문에, 게임 안에서 다양한 텍스트의 움직임이라던가 모습이 나올 수 있다. ex> 글자의 간격에 점차 넓어지는 연출, 글자가 출렁거리는 연출, 발로란트 폰트 연출 - 부분적으로 길어졌다 짧아지는 연출(Mesh를 사용)

3. 글자 렌더링에 HLSL 셰이더를 쉽게 적용시킬 수 있다. 글자가 지지직 거리는 연출이나, 깜빡거리거나, 위치가 흔들리거나, 글자를 회전하고, 외곡하는 것도 쉽게 할 수 있다고 생각되었다.

>> 이는 모두 표현력의 향상으로 이어지며, 후에 게임을 만들때 특별한 연출로써 쓸 수 있다고 생각된다.

SDF라는 외곽선과의 거리를 픽셀의 자리에 놓아 부드럽게 출력이 가능한 기법이 있다고 한다. 나중에 적용해보자 생각했다.

글자 텍스쳐 구현 방식

글자의 텍스쳐는 다음과 같이 진행된다.

1. 글자의 크기를 받아 raw image가 저장될 버퍼를 할당하고, 그곳에 외곽선을 그린다.

글자의 외곽선을 그릴때는, glyph 정보를 활용하고, y변화량과 x변화량 중 더 많은 축 쪽으로 반복문을 돌아가면서, 해당 선의 픽셀을 체운다.

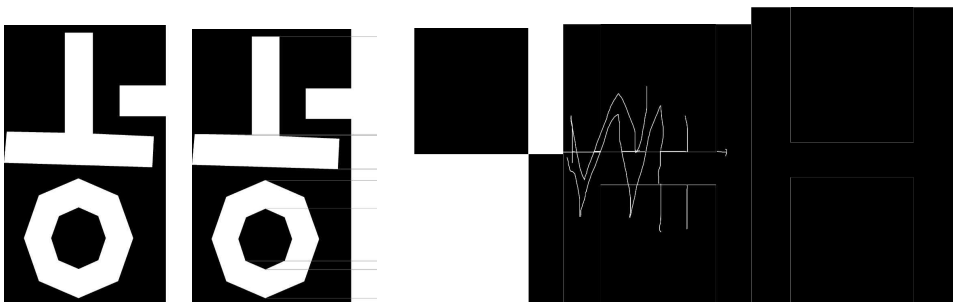
또한 베지어 곡선의 식을 이용하는 것도 중요했다.

2. 텍스쳐 데이터의 외곽선 내부를 체우기 위해 픽셀을 탐색한다. 왼쪽 위부터 오른쪽 아래까지 순서대로 탐색한다. 외곽선을 체우는 원리는 한 행에서 [체워진 픽셀의 열]A들을 만나면, 그 다음 [체워진 픽셀의 열]B를 탐색하고, A~B 까지 픽셀을 체운다. 이를 모든 열에 대하여 반복한다.

만약 마지막에 체워진 픽셀의 열A가 있을 경우, 해당 열을 마지막으로 체우고 난 Xindex 부터 같은 작업을 반복한다. 한 열에 두번 이상 작업을 진행하지는 않는다.

3. constexpr int textureMipLevel = 16; 만큼 텍스쳐 데이터의 크기를 줄여 GPU Resource에 올린다. 대강 한 글자당 50x90 픽셀을 먹는다.

시행착오들.. 알고리즘이 오류를 많이 냈고 또 고쳤다.



셰이더 이용 방식

셰이더는 새로운 셰이더 클래스 ScreenCharactorShader를 사용한다. 알파블랜딩을 처음으로 해보았다. DirectX 공식 샘플의 D3D12PredicationQueries 프로젝트에서 알파블랜딩이 있길래 따라써보았다.

글자를 렌더링하는데 쓰이는 Mesh는 오직 하나로, 그 하나의 메쉬가 모든 글자를 표현한다.

Root Constant로 RECT의 영역과 화면의 Width, Height를 받고,

그것을 활용해 버택스 셰이더는 텍스트를 렌더링할 Mesh를 구성한다. 그 후 해당 글자의 텍스쳐를 매핑하여 렌더링한다.

표현방식

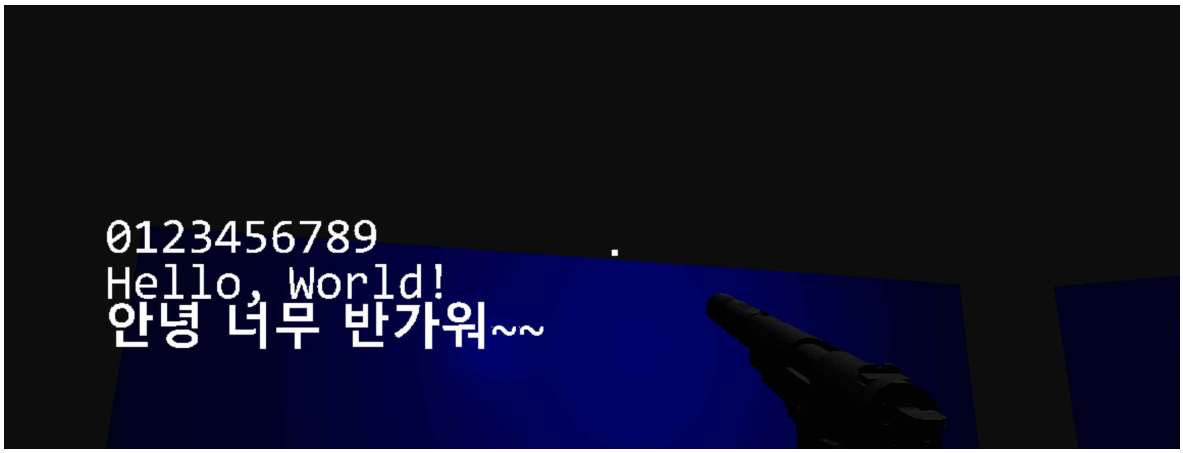
void Game::RenderText(const wchar_t* wstr, int length, vec4 Rect, float fontsiz)

wstr : 문장 - ₩n : 줄바꿈 작동한다. 한글 또한 렌더링이 된다.

length : 문장의 글자 길이이다.

Rect : 글자가 쓰여질 영역이다. 해당 영역을 벗어나면 줄바꿈된다.

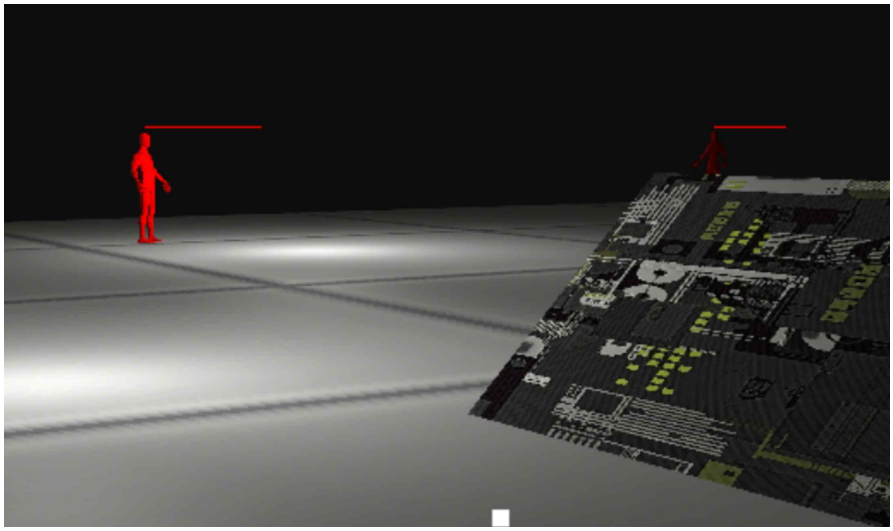
fontsiz : 0123456789 숫자를 적었을때, 그들의 숫자 픽셀이 fontsize 높이를 가지도록 의도했다.



<실행화면>

- 임동건: HP, Kill/Death, Player name, Bullet UI Text 출력 작성

-npc : hp bar 출력

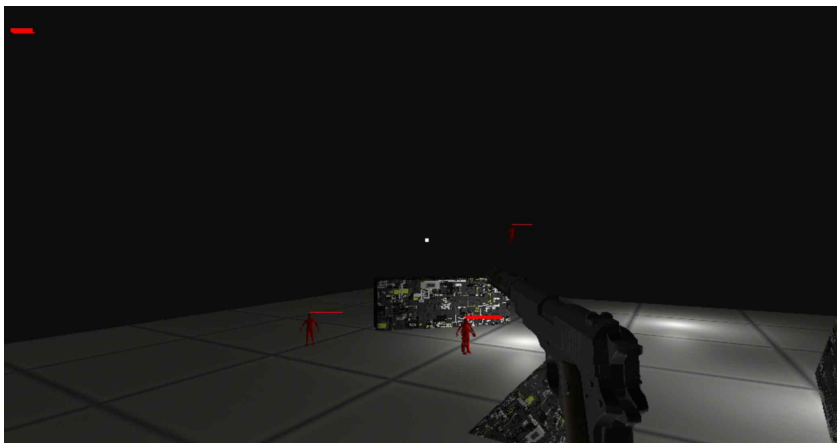


진성준: 다른 플레이어가 클라이언트에서 보이지 않는 오류 수정, 사망시 NPC 제거되지 않는 오류 수정, 플레이어가 죽으면 리스폰

다른 플레이어가 클라이언트에서 보이지 않은 오류 수정

기존의 코드에서 새로운 클라이언트의 매쉬정보를 보내고 있지 않다는 것을 파악하고, 매쉬정보를 보냄. 하지만 아직 보이지 않았음. 3인칭과 1인칭 시점을 전환함에 따라 자신과 다른 클라이언트를 랜더링하지 않는자는 것을 파악하고, 조건을 나눠 해결.

사망시 NPC 제거되지 않는 오류 수정, 리스폰, 플레이어 사망시 리스폰



bool isExist 같은 존재하는지 아닌지에 대한 변수를 통해 isExist가 false라면 랜더링하지 않는 방식으로 해결하려 했으나, 리스폰을 관리하기 위해 isExist가 false가 되면 더 이상 업데이트가 되지않아 isDead라는 새로운 변수로 관리. isDead를 올바르게 서버에서 보내니 몬스터의 hp가 0이하가 되었을때 정상적으로 사라지게 되었음. 그리고 몬스터가 isDead일때 몬스터의 리스폰 타이머가 작동하여 deltatime을 사용해 update에서 리스폰 타이머의 시간이 일정이상 지났을때 몬스터가 랜덤 위치에 생성되도록 하여 해결하였음. 여기서 리스폰이 보이기 쉽게 하도록 스폰 지점의 y값을 높였음. 또한 플레이어가 공격을 맞아 hp가 0이하가 되면 체력이 초기화되고,

플레이어가 생성된 처음위치로 전송하는 것으로 해결.

UI에 필요한 정보 보내주기

처음에는 게임 오브젝트가 생성될때 모든 클라이언트에 필요한 정보를 보내는 것으로 해결하려 했으나, 새로운 오브젝트를 생성하는 시점에 보냈을때 클라이언트가 접속하기 전에 모든 정보를 어딘가로 보내고 새로 접속한 클라이언트에 정보가 전송되지 않는 문제를 발견.

게임 오브젝트가 생성되는 시점과 클라이언트가 서버에 접속하는 시점이 다르다는 것을 파악하고, 새로운 클라이언트가 서버에 접속했을때, 다른 모든 정보들을 새로 입장한 클라이언트에 전송하는 것으로 해결. 노훈철 팀원이 만든 `void SendingAllObjectForNewClient`가 큰 도움이 되었음.

문제점 정리		**해결 방안**	
다음 주차	**3**	**다음 기간**	**2025.09.11 ~ 2025.09.**
다음주 할 일			
지도교수 Comment			