

게임 데이터베이스 기말평가

Made By. 고성준

INDEX

01. 데이터베이스 설계 구조

02. 인게임 작동 방식

03. 플레이 영상

01

데이터베이스 설계 구조

01 데이터베이스 설계 구조

- NoSQL인 Redis를 사용하여 제작
- Key 값을 보내면 Value를 받는 1대1 형식
- 패킷을 나누어 클라이언트가 서버에게 데이터베이스에게서 데이터를 받아오도록 요청

```
// Block  
BLOCK_POSITION,  
  
ADDITEM_REQUEST = 400,  
ADDITEM_RESPONSE,  
  
USEITEM_REQUEST = 402,  
USEITEM_RESPONSE,  
  
GETALLITEM_REQUEST = 404,  
GETALLITEM_RESPONSE  
;
```

```
REQUEST_ADDITEM,  
RESPONSE_ADDITEM,  
  
REQUEST_USEITEM,  
RESPONSE_USEITEM,  
  
REQUEST_GETALLITEM,  
RESPONSE_GETALLITEM
```

← 패킷 관련 Enum

02

인게임 작동방식

02 인게임 작동 방식

- 플레이어의 상호작용, 행동에 따라서 데이터베이스에게서 값을 받아온다.
 1. 각 플레이어가 움직일 때 마다 모든 클라이언트에게 해당 플레이어의 좌표를 보낸다.
 2. 플레이어가 왼쪽 클릭을 하면 블록을 파괴, 해당하는 블록을 데이터베이스에 저장한다.
 3. 플레이어가 오른쪽 클릭을 하면 블록을 설치, 해당하는 블록을 데이터베이스에서 값을 갱신한다.

```
if (Input.GetMouseButtonDown(0))
{
    if (Physics.Raycast(ray, out hit, 6f))
    {
        P_UpdateBlockSpawn updateBlockInfo = default;
        updateBlockInfo.pos = hit.transform.position;
        updateBlockInfo.rotate = hit.transform.rotation;
        updateBlockInfo.type = (int)hit.transform.GetComponent<Item>().GetItemType();
        updateBlockInfo.isBroken = 1;

        Destroy(hit.transform.gameObject);

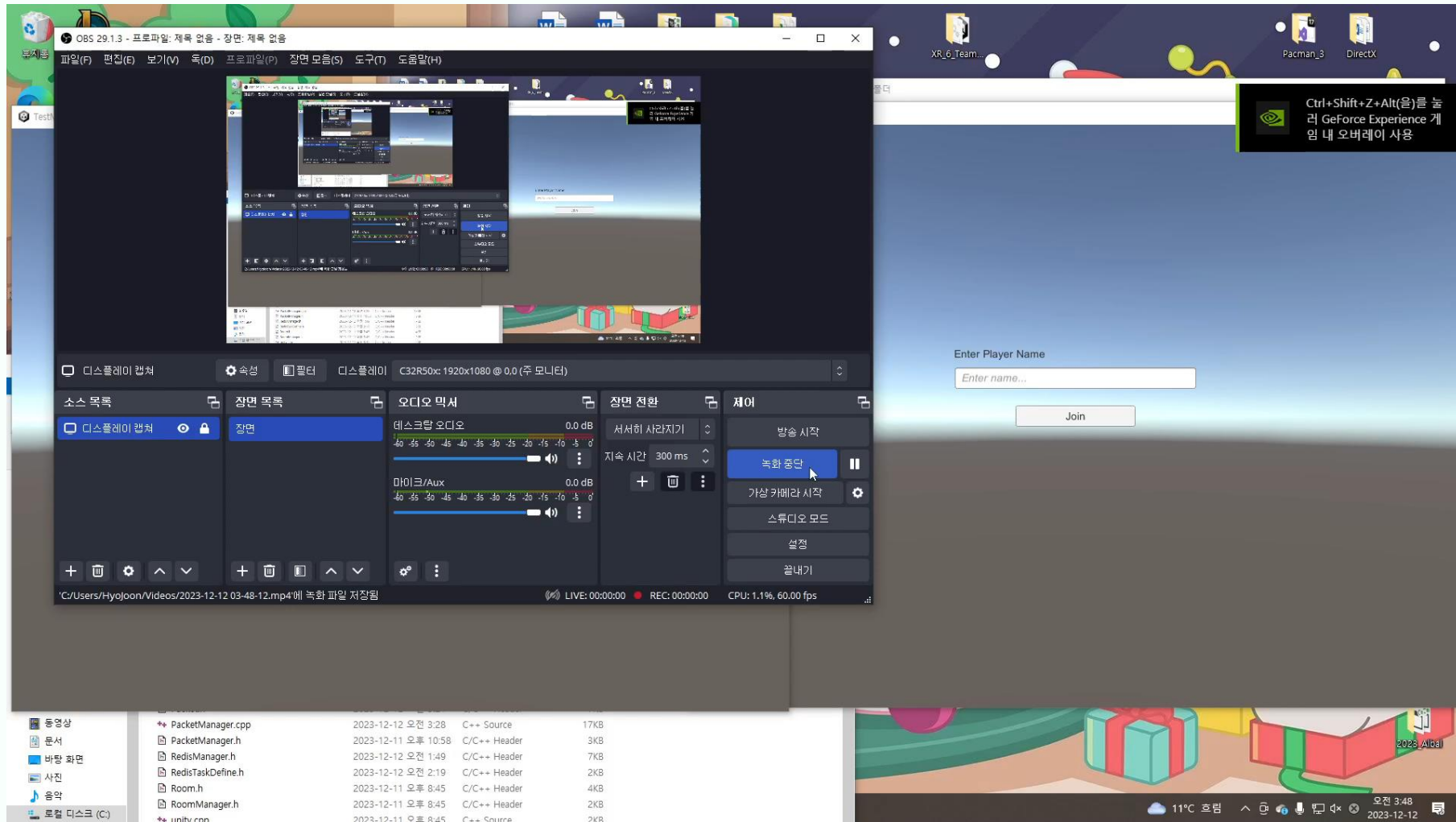
        Client.TCP.SendPacket2(E_PACKET.BLOCK_POSITION, updateBlockInfo);
    }
}
```

← 블록 설치 클라이언트 코드

03

플레이 영상

03 플레이 영상



Thank you
202013144 고성준