

[Toy Ground(토이 그라운드)]

16 주

2020. 4. 11 ~ 2021. 4. 17

작성자

김동석

[0] 공동

04.12 주간 회의

- 일주일간 계획공유
- 맵 관련 패킷교환 회의

[1] 김영준(서버)

주간목표:

- BattleServer - packet 재 설정 및 worker::Procpacket 뼈대 구현
- Object Interface 생성

*사진은 모두 notion을 통해 정리한 내용을 정리한 것을 캡처했습니다.

04.12~14 procpacket 뼈대 구현

뼈대 구현 완료, 완벽한 구현은 아니지만, 작업하면서 추가할 예정

이번 주
한 일

```

85  message WorkerThread::ProcPacket(int id, void* buf) {
86      char* inputPacket = reinterpret_cast<char*>(buf);
87      message msg;
88      memset(&msg, 0, sizeof(message));
89      msg.id = id;
90      msg.type = NO_MSG;
91
92      switch (inputPacket[1]) {
93          case LB_REQUEST_ROOM: { ... }
115         case CB_LOGIN: { ... }
130         case CB_JOIN: { ... }
169         case CB_ROOM_LEAVE: { ... }
181         case CB_KEY_W_UP:
182         case CB_KEY_A_UP:
183         case CB_KEY_S_UP:
184         case CB_KEY_D_UP:
185         case CB_KEY_W_DOWN:
186         case CB_KEY_S_DOWN:
187         case CB_KEY_A_DOWN:
188         case CB_KEY_D_DOWN: { ... }
192         case CB_KEY_JUMP: { ... }
196         case CB_LOOK_VECTOR: { ... }
210         case CB_START: { ... }
217         case CB_READY: { ... }
224         case CB_TEST_TIME_PLUS:
225         case CB_TEST_TIME_MINUS: { ... }
229         default: { ... }
235     }
236     return msg;
237 }
238

```

04.15~16 게임 콘텐츠 개발

더미가 아닌 실제 클라이언트 내용을 전달받아 화면에 띄우기 위해 기존의 로그 형식이 아닌 실제 Object를 띄우기 위한 Interface 구현 게임수학, 다이렉트X 수업을 들으며 공부했던 내용을 다시 사용하면서 어려움을 느끼고 동석이에게 많은 도움을 받음

```
class Object {
public:
    Object();
    virtual ~Object() {};

    explicit Object(const Object& other) noexcept;
    Object& operator=(const Object& other);

public:
    virtual bool Update(float elapsedTime, bool is_player);
    bool BulletUpdate(float elapsedTime);
    void AddForce(XMFLOAT3 force, float elapsedTime, bool isBullet);

public:
    XMFLOAT4X4 GetMatrix() const;

    XMFLOAT3 GetPosition() const;
    void SetPosition(float x, float y, float z);
    void SetPosition(XMFLOAT3 xmfPosition);

    XMFLOAT3 GetPrePosition()const;
    void SetPrePosition(XMFLOAT3 xmfPosition);

    XMFLOAT3 GetLook() const;
    void SetLook(const float& x, const float& y, const float& z);
    XMFLOAT3 GetPreLook() const;
    void SetPreLook(XMFLOAT3 look);
    XMFLOAT3 GetUp() const;
    void SetUp(const float& x, const float& y, const float& z);
    XMFLOAT3 GetRight() const;
    void SetRight(const float& x, const float& y, const float& z);

    XMFLOAT3 GetRotation() const;
    XMFLOAT4 GetRotationQuaternion() const;
```

만든 Object Interface class을 통해 Player를 띄워 클라이언트와 연동하는 것이 목표

[2] 김동석(클라이언트)

주간 목표 :

- 맵 완성(100%)
- 카메라 설정

04.11 ~ 04.14 맵 완성

- 오브젝트 별 index값 설정(최종)

Map (0~127 사용가능)

Cube-01	1	나무대 - 가로 30 세로 31
Cube-02	2	
Cube-03	3	
Cube-04	4	

Cube-Bridge - 가로 20
세로 21

석물
돌1
돌2
나무1
나무2
BARREL
CHEST

}	R	5
	A	6
	N	7
	D	8
	M	9
		10
		11

Fence 1

}	50	□
	51	□
	52	□
	53	□

Fence 2

}	60	
	61	
	62	
	63	

Gravestone

}	40	
	41	

MathHelper const 적용!

mapArray → typeID

[오브젝트 별 index값 설정]

방향성이 의미있는 오브젝트와 없는 오브젝트를 분리했고 BYTE 자료형을 사용할 때 양수 인덱스 값으로는 0~127까지 사용할 수 있기 때문에 현재 맵을 생성하기 위해 필요한 오브젝트에 추가로 다른 오브젝트를 추가하더라도 인덱스 값으로 줄 숫자가 부족하지는 않을 것이다.

.txt 파일에 Unity에 만들어둔 맵을 기준으로 값을 입력했다.

[illegible]

[.txt 파일에 생성한 맵]

- 맵 불러오기



[맵이 로드가 제대로 되지 않았다]

->문제점

맵을 불러왔는데 제대로 읽어오지 못하고 무덤과 펜스 오브젝트가 이상하게 복사가 되었다.

->원인

파일로부터 읽어올 때 BYTE 자료형을 사용했더니 52를 5와 2로 읽어와 밀려서 읽게 되었다.

->해결방안

int 자료형을 사용해 입력을 받고 형변환을 사용하여 배열에 넣었다.



[제대로 로드된 맵]

- 프레임드랍 현상 발생

맵을 불러오고나니 화면을 이동할 때 뚝뚝 끊기는 현상이 발생했다.

->원인

가장 아래 층(바닥)이 33 x 21개의 큐브로 이루어져 있기 때문에 오브젝트의 수가 매우 많아졌기 때문인 것 같다.

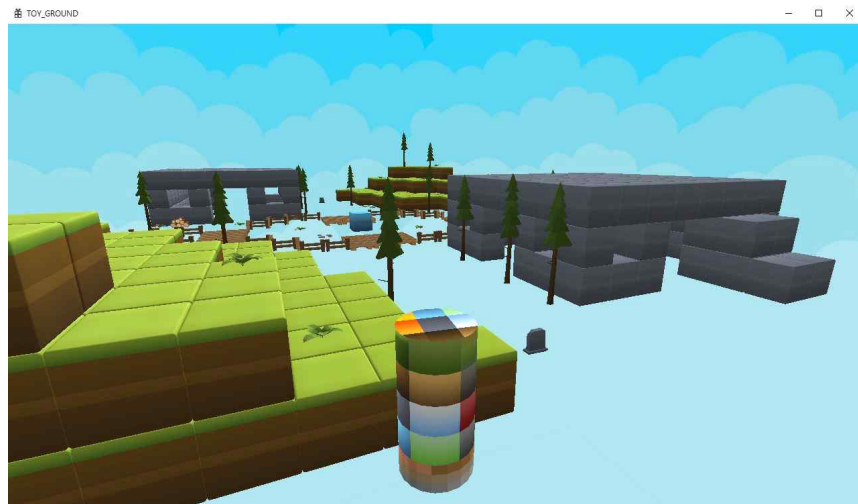
->해결방안

바닥을 이루는 큐브를 제거했더니 프레임 드랍이 현저히 줄었다.

게임을 플레이할 때 바닥의 큐브는 캐릭터의 스킨으로 파괴되지 않기 때문에 변할 일이 없다. 따라서 33 x 21개의 큐브 오브젝트가 아닌 1개의 평면에 큐브오브젝트를 불러온것과 동일한 텍스처를 씩워 처리하도록 수정할 계획이다.

->의문점

내가 생각하기에 33 x 21개의 큐브 오브젝트를 관리하는 것이 이렇게 버벅거릴 일인가에 의문이 들었다. 컬링 부분에 오류가 있거나 보이지 않는 오브젝트는 그리지 않는 기법 등을 사용해서 수정해 보는 것이 좋을 것 같다.



[바닥의 큐브들을 제거하니 정상 프레임으로 돌아옴]

04.16 ~ 04.17 카메라 설정

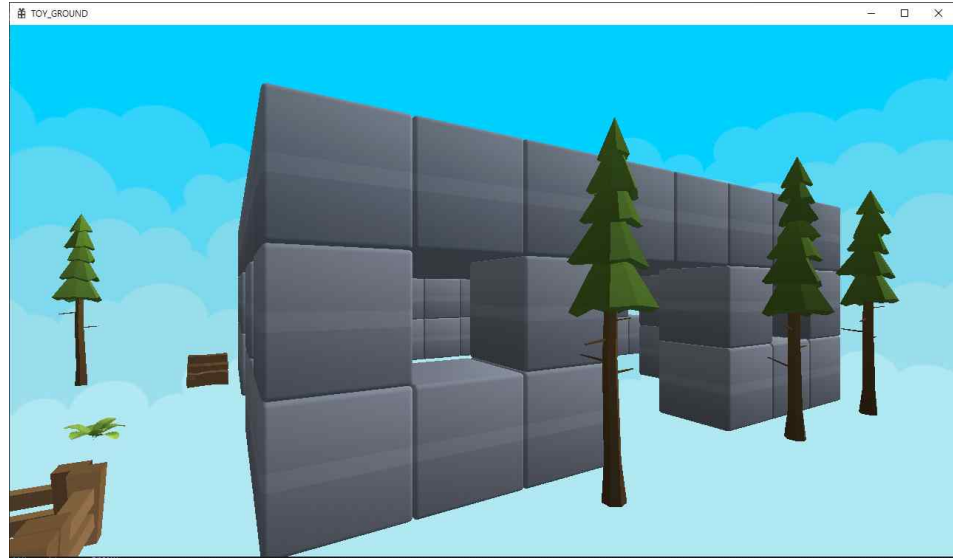
- 카메라 타입 설정

기획할 때 3인칭 슈팅게임으로 기획했지만 TPS 특성상 캐릭터가 화면을 가리는 단점이 있다(장점은 자신의 캐릭터가 화면에 보이기 때문에 상대방의 투사체를 피하기가 더 쉽다).

이러한 단점을 수정하기 위해 조준(우클릭) 시 1인칭으로 전환하도록 수정하기로 했다.

- 1인칭 카메라 설정

1인칭 카메라 관련해서 Camera 클래스에 추가하고 마우스 우클릭(다운) 시 3인칭->1인칭, 마우스 우클릭(업) 시 1인칭->3인칭으로 전환하도록 수정했다.



[1인칭 시점]

- 문제 발생

카메라 전환 시 자연스럽게 전환이 안된다.

->문제점

1. 3인칭->1인칭 변환 시

카메라가 보고 있는 방향이 아닌 캐릭터가 보고 있는 방향으로 시점이 변경된다.

2. 3인칭에서 이동 시

카메라가 보고 있는 방향 기준으로 좌우앞뒤 이동이 되지 않는다.

3. 1인칭에서 화면 이동 시

캐릭터가 카메라가 바라보고 있는 방향으로 회전하지 않는다.

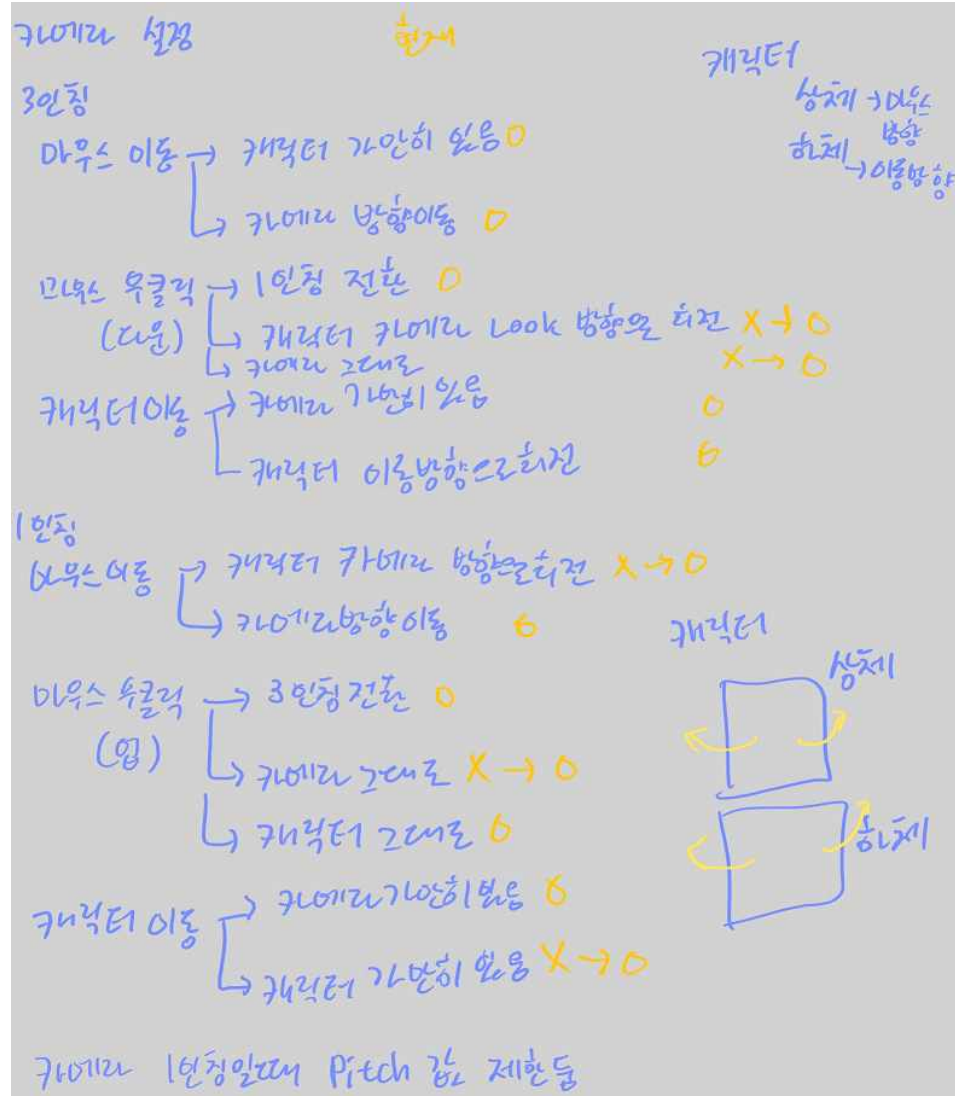
->원인

이러한 문제점이 발생한 이유는 설계할 때 캐릭터를 상체와 하체로 분리하여 하체의 방향은 이동 방향을 향하고 상체의 방향은 카메라의 방향(조준 방향)을 하도록 설정했기 때문이다.

따라서 지금까지 3인칭으로 이동 시 캐릭터의 방향과 카메라가 바라보는 방향이 달랐음

->해결방안

태블릿에 현재 상황과 수정해야하는 부분을 정리하고 하나하나 수정했다.



[태블릿에 정리한 내용]

현재는 캐릭터를 위아래가 분리되지 않은 하나의 원기둥으로 사용하고 있기 때문에 카메라 전환 관련해서만 자연스럽게 전환되도록 수정하였고 애니메이션을 적용하면서 원래 계획한대로 자연스럽게 캐릭터가 움직이도록 수정할 것이다.

이번 주에는 애니메이션 작업을 하기로 했었는데 카메라와 맵 설정으로 인해 시간을 투자하지 못했다. 다음 주에는 무조건 애니메이션을 적용할 것이다.

<p>다음 주 할 일</p>	<p>[0] 공동 04.19 주간 회의 (17주차) 안건 : 일일계획 공유</p> <p>[1] 김영준 (서버) 주간 목표 - Player 완성? - Network 연결이 간헐적으로 실패함(디버깅 하지 못함), 원인 확인 및 해결 - 시험 준비 및 시험 치르기</p> <p>[2] 김동석 (클라이언트) 주간 목표 - 애니메이션 프레임워크에 추가</p>
<p>문제점</p>	<p>[1] 김영준 (서버) - 물리, 백터, 행렬 등 Object을 만들면서 다시 공부하고 찾아보고 동석이에게 물어보면서 코딩을 했는데, 어려움을 느꼈다. 클라이언트와 합쳐야 한다는 생각에 일정에 없는 작업을 먼저 했는데 당장 server 연결에 간헐적으로 실패하는 일을 우선으로 해결해야 한다.</p> <p>[2] 김동석 (클라이언트) - 시험 기간이 다가오고 있다. 다음 주 안에 애니메이션을 프레임워크에 완벽히 적용하는 것을 목표로 시험 공부와 병행해 열심히 작업해야겠다.</p>

[추가]