[Toy Ground(토이 그라운드)]

15 주	2020. 4. 4 ~ 2021. 4. 10	작성자	김영준

이번 주 한 일

[0] 공동

04.05 주간 회의

- 일주일간 계획공유

[1] 김영준(서버)

주간목표:

- BattleServer - packet 재 설정 및 worker::Procpacket 뼈대 구현

*사진은 모두 notion을 통해 정리한 내용을 정리한 것을 캡처했습니다.

04.05 packet 정리

packet 설계한 문서를 매번 찾아보기 힘들어 notion에 다시 정리함

▶ Lobby → Client

▶ Client → Lobby

▼ Battle → Client

auto_accept_ok: id
→ Lobby에서 받은 정보를 통해 client와 accept

auto_accept_fail
→ Lobby에서 받은 정보를 통해 client와 accept 실패

room_entered: id, ready, player_no, mmr, isManager
→ 해당 게임 room에 접속 성공

room_leaved: id
→ 해당 게임 room에서 탈퇴

new_room_host
→ 해당 게임 room의 host으로 변경됨

ready: id, ready
→ 같은 게임 room의 id→ready의 상태가 변경됨

game_start:
→ 해당 게임 room의 게임 시작

game_start_available: bool
→ 해당 게임 시작 가능 여부 전달(host에게만)

game_over: char
→ 해당 게임 종료, 승리 팀 전달

left_time: unsigned char
→ 게임 시간 전달

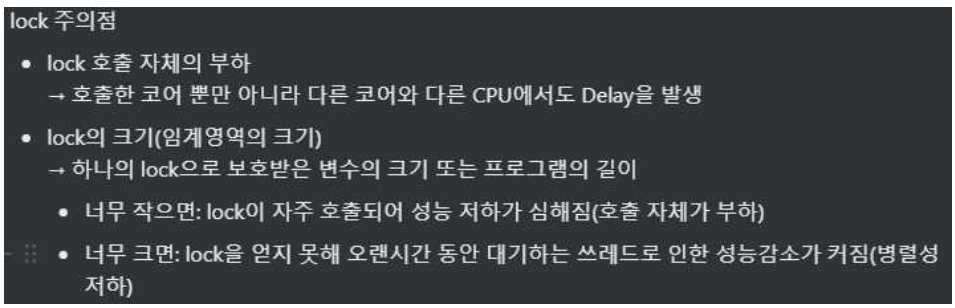
round_start:
→ 게임 countdown 시작

04.06~08 게임엔진 프로젝트 마무리

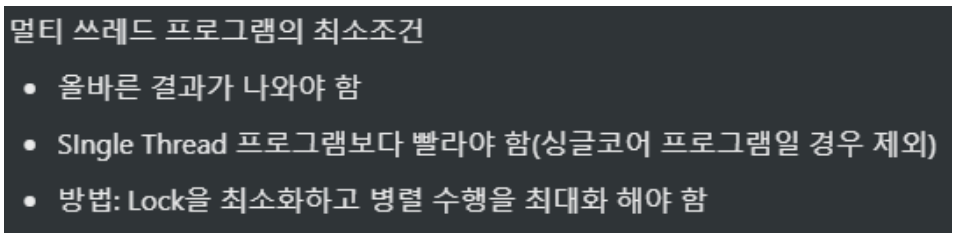
다음 주 까지 졸업작품 프로젝트에 집중하기 위해 랜덤 팀원과 분업하여 진행하는 프로젝트 진도를 중간고사 이후 내용까지 진행하였다. 다른 과목의 중간고사 일정과 졸업작품 중간발표 이전 일정이 겹치는 부분이 많아 해치울 수 있는 게임엔진 프로젝트를 마무리.

04.09 workerThread::ProcPacket 구현(계속)

클라 -> 로비 -> 인게임으로 이어지는 패킷 연결을 위해 로비에서 클라이언트의 정보를 인게임으로 가져가는 패킷 처리 진행 중
lock의 주의점은 인지하고 프로그래밍 중이지만



멀티스레드의 최소조건을 만족하지 못하고 있다



(클라이언트 정보를 받지 못함)

[2] 김동석(클라이언트)

주간 목표 :

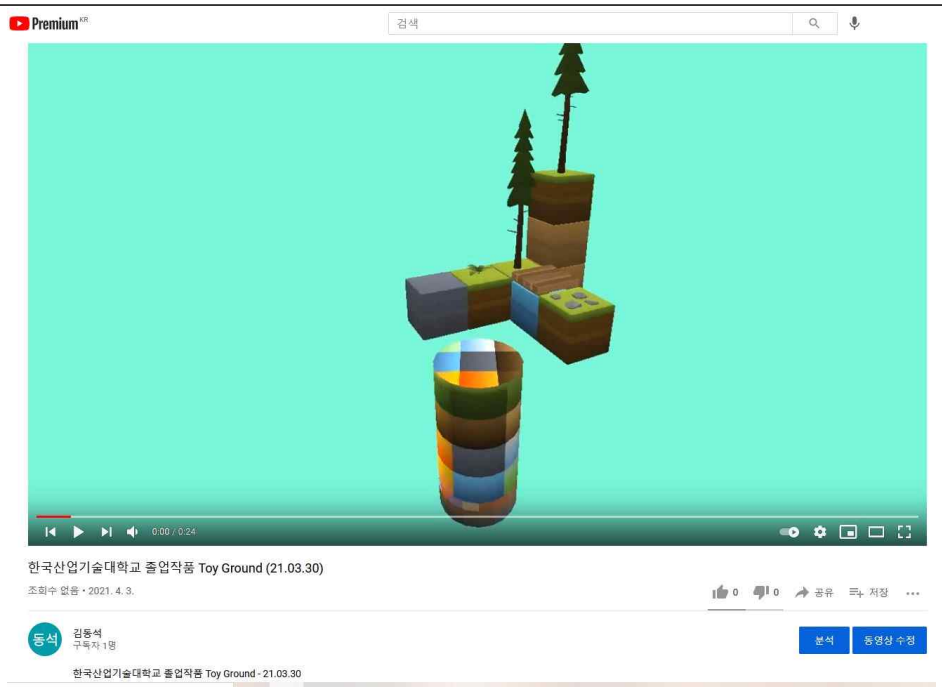
- 바운딩 박스 생성

- 맵 완성(80%)

04.04 유튜브 재생목록 생성

- 유튜브에 졸업 작품 진행 상황을 기록할 재생목록을 만들었다.

앞으로 꾸준히 영상을 찍어 올릴 생각이다.



[04.03에 게시한 동영상]

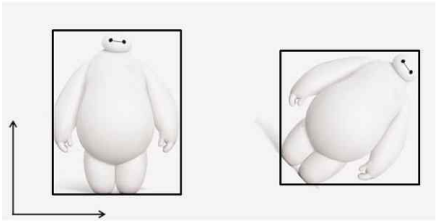
04.05 ~ 04.06 바운딩 박스 생성

- 오브젝트별 바운딩 박스 지정

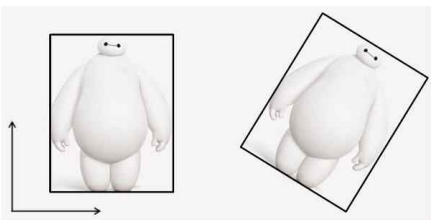
저번주에 생성해 놓은 오브젝트들에게 각각 바운딩 박스를 생성하였다. 오브젝트들은 미리 만들어 놓은 데모 맵에 사용되는 오브젝트를 모두 추가해 두었으며 새로운 오브젝트를 추가할 때도 쉽게 바로 추가할 수 있도록 만들었다.

Toy Ground의 맵은 큐브 맵이기 때문에 지형이 큐브 형식이고 큐브는 회전할 일이 없다.

따라서 바운딩 박스의 종류는 오브젝트가 회전할 때 같이 회전하는 OBB 방식 대신 오브젝트가 회전해도 바운딩 박스를 이루는 면의 노말 벡터들이 x,y,z 축과 일치하는 AABB 방식을 택하기로 했다.



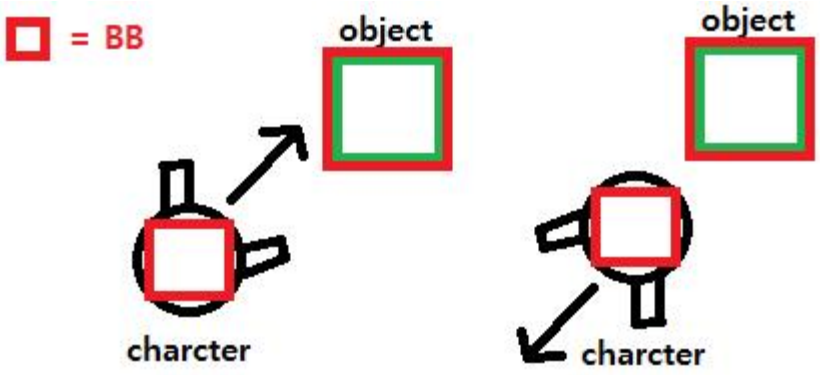
[AABB(Axis-Aligned Bounding Box)]



[OBB(Oriented Bounding Box)]

단, 기존의 AABB는 모델을 이루는 다각형의 x,y,z 좌표의 최소 최대를 각 박스의 버텍스로 해서 생성하기 때문에 모델이 회전할때마다 바운딩

박스의 크기가 달라진다는 특징이 있는데, 우리 게임에서 처리해야 할 충돌처리는 (캐릭터-보석, 캐릭터-총알, 캐릭터-지형) 이렇게 3가지인데 캐릭터를 기준으로 탑뷰에서 바라봤을 때 캐릭터는 y축으로 회전할 일 밖에 없기 때문에 회전에 따라 크기가 변하는 것이 아닌 크키는 캐릭터 기준으로 정사각형 크기로 고정하기로 결정했다.



[캐릭터가 회전하더라도 바운딩 박스의 크기와 y축의 각도는 변하지 않는다.
캐릭터의 바운딩 박스는 캐릭터보다 살짝 작게 설정]

위의 규칙을 적용하여 DirectXCollsion.h에서 제공하는 Bounding Box 구조체를 사용해 바운딩 박스를 생성하였다.

- 절두체 컬링

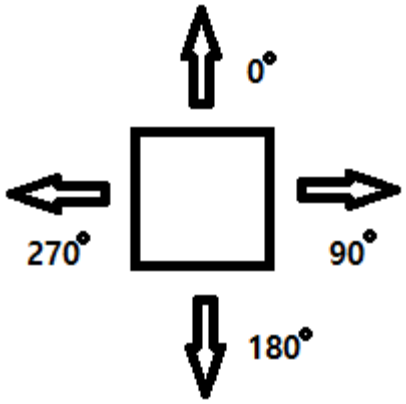
위에서 생성한 바운딩 박스를 기준으로 절두체 컬링을 적용하여 시야 범위에 포함되지 않는 객체들은 그리지 않도록 하였다.

04.07 맵 설계

- 맵 설계

큐브 맵의 이점을 살리기 위해 배열을 사용하여 맵의 정보를 주고 받을 수 있도록 패킷을 설정했다.

2차원 배열을 사용하였고 배열에서 가지고 있어야 하는 값은 오브젝트의 종류와 y축의 회전량인데 기본 베이스가 큐브 형식이다 보니 y축 회전량은 4방향(0˚, 90˚, 180˚, 270˚) 밖에 없다.

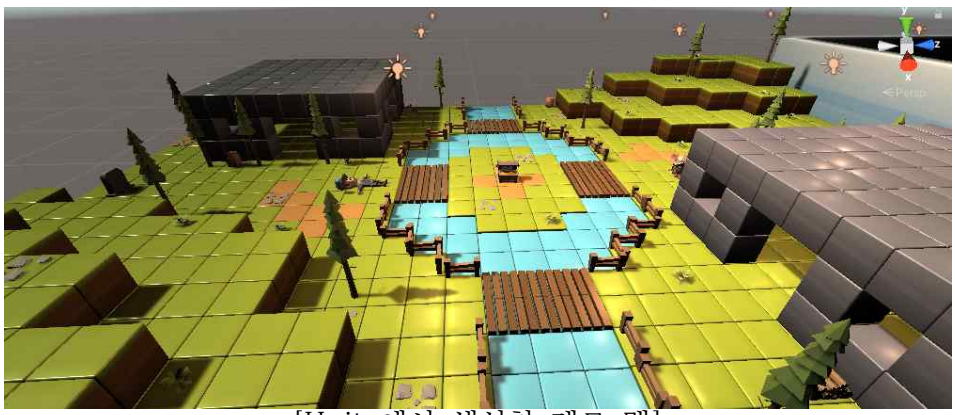


큐브 4개
나무 2개
울타리 2개
돌 2개
풀 2개
발판 1개
나무통 1개
비석 1개
보물함 1개
나무더미 1개

총 10 종류

[회전각도 4종류] [맵에 사용되는 오브젝트 종류]

그리고 unity에서 생성한 데모맵을 기준으로 Toy Ground에서 사용되는 오브젝트의 종류는 10가지이기 때문에 0~39까지의 index값을 사용해 맵 패킷을 생성하기로 결정했다.(이 방법이 가장 효율적인 방법일지는 고민을 더 해봐야겠다.)



[Unity에서 생성한 데모 맵]

04.09 ~ 04.10 맵 생성, 애니메이션 복습

- 오브젝트 위치 설정

위에서 설계한대로 맵을 생성해주는 프로그램을 만들고 있다, 일단 맵에서 오브젝트의 종류와 회전량을 읽어오면 그에 따라 오브젝트가 생성되도록 설정을 했다.(발판 같은 경우는 에셋 자체가 공중에 떠 있는 상태이기 때문에 y축 값을 바닥에 붙도록 수정하는 작업을 했다.)

- 맵 에디터(80%)

일단 기획서에서는 맵의 종류를 1개로 만들기로 했지만 직접 플레이해보면서 오브젝트의 위치등의 수정이 필요할 수도 있기 때문에 바로바로

	수정이 가능하도록 맵 에디터를 제작하고 있다. 현재 유니티에서 설정한 맵을 읽어와 Toy Ground에서 사용할 2차원 배열로 설정하는 방법을 시도 중이다. 맵 에디터가 주가 아니기 때문에 최대한 빨리 마무리를 해야겠다. **- 애니메이션 복습** 애니메이션을 적용하기 앞서 애니메이션 관련해서 강의 복습을 했다.
다음 주 할 일	**[0] 공동** **04.12 주간 회의 (16주차)** **안건 : 일일계획 공유** **[1] 김영준 (서버)** 주간 목표 - Battle Server - workerthread->procpacket 완성 **[2] 김동석 (클라이언트)** 주간 목표 - 맵 완성 - 애니메이션 프레임워크에 추가
문제점	**[1] 김영준 (서버)** - 다른 과목 중간고사를 더 이전에 생각해서 일정을 수립했어야 했는데 중간고사 시즌이 오면서 다른 일정들이 생기며 기존 일정보다 더 빠르게 일정을 소화해야 한다. **[2] 김동석 (클라이언트)** - 게임에서 가장 중요한 애니메이션을 프레임워크에 적용할 때가 되었다. 2월달에 만들어 놓은 FBX Converter와 Sample Animation을 적용한 프로젝트를 Toy-Ground 프로젝트에 바로 붙이는 것을 다음 주 목표로 삼았다. - 중간 발표날짜가 5월 17일로 정해졌기 때문에 이제 날짜별로 세부계획을 세워야겠다. 나의 목표는 그림자까지 적용하여 중간발표를 하는 것이다. 더 열심히 해야겠다.

[추가]