

[Catch Bear (캐치 베어)]			
14 주	2022. 3. 27 ~ 2022. 4. 2	작성자	박소영
이번주 한 일	<p>[0] 공동</p> <p>3. 30) 임시 회의</p> <ul style="list-style-type: none"> - 서버) 플레이어 이동 동기화 체크 → 인스턴싱 문제 - 서버, 클라) 충돌체크 어떻게 할지 - 공동) 파일 정리 방식 정하기 - 클라) 맵에 객체 너무 많음. 종류 줄이기 - 클라) 전역 조명 제대로 되는지 확인 - 클라) 아이템 초기좌표 정하기 → 금요일 회의때 결정 <p>4. 1) 정기 회의 (14주차)</p> <p>* 14주차에 각자 한 일 공유</p> <p>1) 공동</p> <ul style="list-style-type: none"> - 아이템 생성 위치 결정 - 플레이어 카메라 오프셋 다시 설정 <p>2) 김우찬(서버):</p> <ul style="list-style-type: none"> - 클라이언트 2개 이상 접속했을 때 플레이어 사라지지 않는지 확인 → 사라지진 않지만 애니메이션 동기화 안됨 - 플레이어 이동 동기화 → 주말까지 완료하기로 함 - 아이템 생성 동기화 → 플레이어 애니메이션 동기화 후 하기로 함 <p>3) 박소영(클라이언트):</p> <ul style="list-style-type: none"> - 플레이어&오브젝트 충돌 완료 - 아이템 종류별로 다르게 렌더링 완료 <p>4) 고은비(클라이언트):</p> <ul style="list-style-type: none"> - 맵 렌더링 해오기 완료 - 플레이어 상태 확인 → 주말 안에 끝내기로 함 <p>* 15주차에 할 일 공유</p> <p>아래 다음주 할 일에 서술</p>		

*** 이번주 한 일**

[1] 김우찬 (서버)

주간목표 :

- 플레이어 이동 동기화 마무리 (100%)
- 맵에 모든 static오브젝트들을 불러와서 충돌 테스트 (0%)
- 아이템 생성 동기화 (0%)
- 플레이어-아이템 충돌 패킷 전송, 확인 (0%)

지난주에 설계한대로 하니 그냥 박스 오브젝트에선 이동 동기화 완료. 이동 동기화를 하면서 여러 가지 문제점을 발견하여 수정하였다. 우선 지금까지 내가 만든 방식은 ENTERGAME패킷이 계속 돌면서 플레이어의 이동이 감지되면 모든 플레이어에게 BroadCast해주는 방식으로 설계하였다. 하지만 이런 방법은 무늬만 IOCP 비동기 방식이지 동기 방식으로 돌아가고 있었다. 그래서 세션을 아예 전역변수로 선언하고 플레이어의 키 입력이 있을 때만 MOVE패킷을 보내도록 설계하니 기존의 스파게티 코드도 고치고 프레임도 약간 좋아졌다.

그리고 플레이어의 위치도 C_MOVE패킷에 같이 담아서 보내주고, S_MOVE패킷을 받아서 SetLocalPosition을 여기서 지정해주니 시간이 지남에 따라 위치 동기화가 안 되던 현상도 해결하였다.

이제 곰돌이 에셋을 씌우고 애니메이션 돌아가는 플레이어에도 똑같이 적용하려고 하니 플레이어 인스턴싱 문제로 멀티플레이 접속이 안되는 현상이 일어났다. 팀원들에게 물어가며 접속을 해결하니 이번에는 애니메이션 동기화 문제가 있었다.

혼자 공공대다가 이 부분도 팀원들에게 물어봐서 문제가 해결된 (플레이어 인스턴싱 과정에서 모든 플레이어가 같은 애니메이션이 재생되는 문제가 있었다) 코드를 받아서 작업하였다.

S_MOVE패킷으로 키 눌림 여부와 플레이어 State를 받아 각 플레이어마다 애니메이션을 재생시켜서 애니메이션 동기화 부분도 완료하였다

[2] 박소영 (클라이언트)

1. 아이템 랜덤 생성

아이템 랜덤 생성 작업을 완료했다. 팀원들과 회의때 우리 게임의 플레이 타임을 3분으로 잡고, 일반 아이템은 30초마다 모든 종류(5종)를 하나씩 생성, 유니크 아이템은 1분마다 모든 종류(2종)를 하나씩 생성, 보물은 1분마다 하나씩 생성하기로 정했다. 정한 아이템 생성 주기마다 아이템을 생성하고, 은비가 참고하라고 띄워준 아이템을 참고하여 아이템 종류마다 다른 모델이 띄워지도록 했다.

아이템을 생성하고 관리하기 위한 아이템 매니저를 만들고, 리스트를 사용해 아이템을 관리하도록 하였다. 내가 임시로 정한 좌표대로 아이템을 띄우다가, 금요일 회의 때 팀원들과 아이템 생성 위치를 정하고, 그곳에서 아이템이 생성되도록 구현했다.

2. 클라에서 아이템-플레이어 충돌

수요일 회의때 아이템과 플레이어가 충돌하는 부분을 만들어달라고 해서 3DGP때 한 과제를 참고하여 BoundingOrientBox로 오브젝트끼리의 충돌을 구현했다. 일단 디버깅을 하여 충돌 조건문 안에 들어가는지만 확인했었다.

금요일 회의때 아이템과 플레이어가 충돌하면 아이템을 삭제하는걸로 하자고 해서 아이템과 플레이어가 충돌하면 아이템을 삭제하는 부분까지 구현했다.

이제 다음주에 충돌했을 때의 후처리(플레이어 상태 변경, 아이템 획득) 부분을 구현할 예정이다.

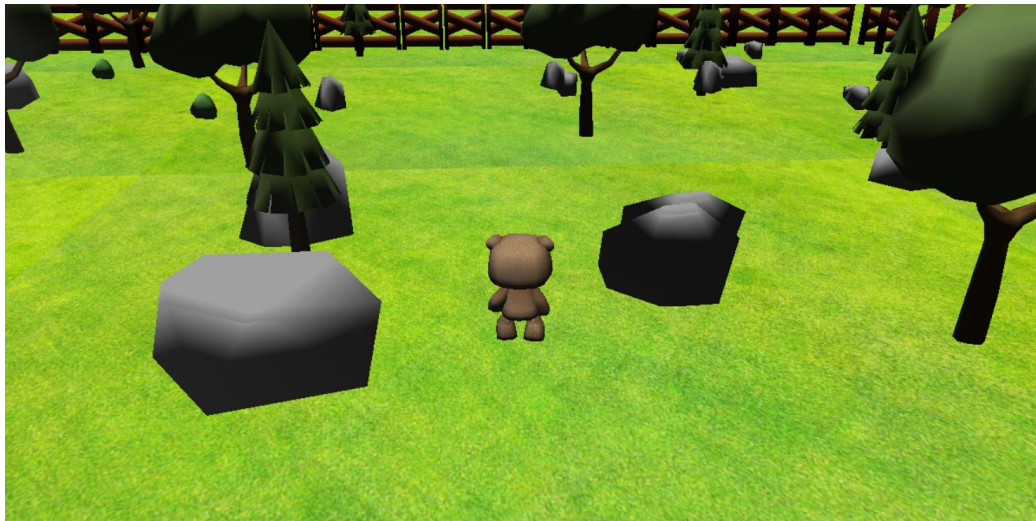
[3] 고은비 (클라이언트)

주간목표 및 진척도

- 바이너리 파일로부터 데이터를 읽어들이어 인스턴싱을 활용하여 맵 렌더링 (100%)
- FSM 패턴을 활용해 플레이어의 상태별로 클래스를 구현 (100%)

유니티에서 맵 오브젝트들을 배치한 후 전체적인 맵을 확인한 후, 오브젝트의 이름, Transform, Scale, Rotation 값을 파일 입출력으로 바이너리 파일에 적어 생성한 후, 이 바이너리 파일을 클라이언트 프로젝트에서 순서대로 읽어 들였다. 읽어들이는 오브젝트의 이름을 key값으로 해서 만약 Resources에서 관리하는 오브젝트들에 현재 읽어들이는 오브젝트의 이름이 없으면 그 때 모델 정보를 읽어들이고, Resources에서 관리하도록 추가하였다. 그리고 만약 Resources에 해당 key값이 존재한다면 그 값을 반환하도록 하였다.

그리고 같은 모델을 공유하는 오브젝트들은 인스턴싱을 활용하여 렌더링하였다. Scene에서 Render()를 호출하면 InstancingManager::Render()가 호출되어 현재 scene의 객체들을 받아와서 객체의 InstanceID를 읽어와 같은 InstanceID끼리 분류해서 하나만 그려야 하는 객체, 여러 개를 한 번에 그려야 하는 객체들로 나눠서 Render()를 호출했다. 하나만 그려야 하는 객체는 일반 Render(), 인스턴싱이 적용돼야 한다면 InstancingBuffer를 사용해 객체들의 월드행렬, 뷰행렬, 투영행렬을 모두 한 번에 모아서 넘겨줬다.



	<p>저번주까지는 키 입력에 따라 애니메이션만 바뀌서 재생하도록 코드를 구현해 상태도 enum값으로 갖도록 하였다. 그래서 이전 상태enum값과 현재 상태enum값만 갖고 있어도 된다. 하지만 이렇게 하면 Player 클래스 내 KeyCheck(), Update(), LateUpdate() 안에서 if문이 많아지고 잘못하면 복잡해질 수 있다. 그래서 상태별로 클래스를 만들어 각각의 상태에서 체크해야 할 키입력, Update()를 나눠서 Player 클래스 내 코드도 간단해졌다.</p> <p>아직까지는 IDLE, MOVE, JUMP, ATTACK, DASH만 구현하였다. 개발을 진행하면서 추가해야 되는 상태가 있다면 추가할 예정이다.(DAMAGE, VICTORY 등)</p>
다음주 할 일	<p>[0] 공동 4. 7) 정기 모임(15주차)</p> <p>[1] 김우찬 (서버)</p> <ul style="list-style-type: none"> - 아이템 생성 동기화 - BoundingOrientedBox를 사용한 충돌 방식으로 변경 - 플레이어-아이템 충돌 패킷 전송, 확인 - 플레이어-static오브젝트 충돌 처리 <p>[2] 박소영 (클라이언트)</p> <ul style="list-style-type: none"> - 아이템 효과에 따른 플레이어 상태 or 능력치 변경 - 그림자 오류 해결 - 조명부분 다시 보기 <p>[3] 고은비 (클라이언트)</p> <ul style="list-style-type: none"> - 클라이언트 내부 개인 타이머 UI 렌더링 (시간이 지날 때 마다 타이머 텍스처도 바뀌도록 하기) - 아이템 효과에서 애니메이션 관련 수정 사항이 생긴다면 수정하기
문제점	<p>[0] 김우찬 (서버)</p> <p>기존의 코드는 패킷이 계속 돌아서 비동기방식인 iocp서버의 장점을 전혀 활용할 수 없었다. 게임 서버 수업을 듣고 뭔가 잘못됐다고 느껴서 늦게라도 코드를 수정하여 문제를 해결하였다.</p> <p>동기화 과정에서 Engine안의 코드를 수정할 일이 많았는데, 아무리 서버를 맡았다고 해도 그래픽에 굉장히 무지함을 느껴 시간도 많이 쓰고 팀원들에게 의지하는 일이 많아 내가 얼마나 부족한지 깨달았다. 졸업작품이 끝나면 그래픽 부분을 다시 공부하고 싶다.</p> <p>[1] 박소영 (클라이언트)</p> <p>프레임워크를 만들 때 임시로 플레이어를 띄우고 그림자를 적용했을 때는 잘 되었는데 플레이어랑 아이템을 띄우니까 그림자가 잘 되지 않는다.. 그림자 부분을 다시 보</p>

	<p>고 수정해야 겠다. 또 조명 부분도 약간 이상하게 되는 것 같아 그림자를 고치고 다시 한번 수정해 봐야겠다.</p> <p>[2] 고은비 (클라이언트)</p> <p>인스턴싱을 활용하는 부분에서 플레이어도 인스턴싱 처리를 했다가 서버에서 애니메이션 동기화가 잘 안되는 문제점이 생겼다. 각각의 플레이어마다 재생되는 애니메이션이 다르면 애니메이션 변환행렬도 달라진다. 이 부분을 생각 못하고 인스턴싱을 구현할 때 일반 메쉬를 렌더링할 때와 똑같이 했다가 여러 플레이어들이 모두 같은 동작이 렌더링되었다.</p> <p>이후에 문제점을 확인하고 수정하여 알려주었지만 안그래도 할 일이 많은 서버의 시간을 뺐은 것 같아 미안했다. 앞으로 서버 동기화도 생각해가면서 코드를 구현해야 겠다.</p>
--	--