

[Catch Bear (캐치 베어)]			
13 주	2022. 3. 20 ~ 2022. 3. 26	작성자	박소영
이번주 한 일	<p>[0] 공동</p> <p>3. 21) 교수님 연구실 방문</p> <ul style="list-style-type: none"> - 4월 말까지 한 스테이지가 게임스럽게 돌아가는 것을 보여줘야 한다. - 4월 말까지 해야 할 일들을 리스트업 한 후, 우선순위를 정할 것 - 전체 게임을 봐주고 일정을 정리해줄 대표자 정하기 - 서버 패킷을 정해서 무엇을 전달하고 받을지 많이 얘기해보고 공유하기 - 4월 말까지 사용할 수 있는 시간을 계산하고, 그에 맞는 계획을 세우기 <p>3.21) 임시 회의</p> <p>안건 - 4월 말(중간발표)까지 해야 할 일 리스트업</p> <p>4월 말 중간발표까지 해야 할 일들을 함께 생각해본 다음, 각자 해야 할 일들의 우선순위를 정했다. 정한 우선순위를 바탕으로 4월 말까지의 주차별 계획을 세웠다.</p> <p>[김우찬(서버)]</p> <ol style="list-style-type: none"> 1. 서버에서 충돌체크 2. 플레이어 이동 동기화 3. 아이템 생성 동기화 4. 술래 정하기 5. 술래 넘겨주기 6. 대기실 7. 채팅 <p>[박소영(클라이언트)]</p> <ol style="list-style-type: none"> 1. 카메라 회전 2. 아이템 생성 3. 아이템 효과 → 플레이어 상태 or 능력치 변경 4. 보물 생성 5. 플레이어 점수 계산 6. 게임 종료 후 3인 플레이어들 순위 결정 7. 아이템 이펙트 		

[고은비(클라이언트)]

1. 플레이어 애니메이션 (IDLE, WALK, DASH, ATTACK, DAMAGE, VICTORY)
2. 키 입력에 따라 플레이어 상태 구현
3. 맵 불러오기
4. 게임 제한시간 타이머(3분)
5. 타이머(폰트 or 텍스처) 렌더링, 리소스 찾기
6. UI (3인 플레이어들 점수, 자신이 지닌 아이템)

[주차별 계획]

	김우찬	박소영	고은비
13주차) 3/20 ~ 3/26	충돌 처리, 플레이어 이동 동기화	카메라 회전	플레이어 애니메이션
14주차) 3/27 ~ 4/2	플레이어 이동 동기화, 아이템 생성 동기화	아이템 랜덤 생성 (위치 서버에서 정해줌)	맵 불러와서 렌더링, 플레이어 상태
15주차) 4/3 ~ 4/9	플레이어-아이템 충돌 클라이언트로 아이템 패킷 전송, 확인	아이템 효과 -> 플레이어 상태or능력치 변경	클라이언트 내부 개인 타이머 렌더링, 타이머
16주차) 4/10 ~ 4/16	서버에서 솔레 정하기, 대기실, 플레이어-플레이어 충돌 처리	보물 생성, 점수 계산, 순위 결정	플레이어 충돌로 인한 상태 변화
17주차) 4/17 ~ 4/23	못한 부분 구현, 서버와 통신하면서 플레이	아이템 이펙트	플레이어 UI (플레이어들 점수, 지닌 아이템)
18주차) 4/24 ~ 4/30	사운드(배경음악) 추가, 못한 부분 구현 및 마무리, 중간발표 준비		

3. 24) 정기 모임 (13주차)

- 13주차에 각자 한 일 공유

김우찬(서버): 대략적인 패킷 내용을 구상하여 팀원들과 공유했다. 애니메이션 동기화 방법에 대해서 고민해봄. 일단 서버에서는 플레이어마다 실행할 애니메이션을 enum값으로 보내서 각 플레이어마다 받은 enum값으로 분기문으로 애니메이션을 돌리는 방식으로 구현할 예정

박소영(클라이언트): 카메라 회전 구현 완료. 좌우 방향키를 누르면 카메라와 플레이어가 좌우로 회전하고 위아래 키를 누르면 플레이어와 카메라가 이동한다. 카메라 회전 다음 구현 예정이었던 아이템 생성 작업을 하는 중이다.

고은비(클라이언트): 애니메이션이 돌아가긴 하지만 제대로 돌아가지는 않음. 애니메이션 변환 행렬 때문인 것 같음. 상수버퍼로 바꿔서 전달해봐야겠음. GPU를 디버깅할 수 있는 PIX라는 프로그램이 있다길래 이 프로그램을 활용해서 디버깅 해볼것임.

- 14주차에 할 일 공유

아래 다음주 할 일에 서술

- 월요일 임시 회의 후 목요일 회의에서 정해야 할 일들 상의

1. 데이터 충돌 – 서버에서 할지 클라이언트에서 할지

서버에서 하기로 결정

2. 전체 게임에 대한 그림(맵 크기 등)

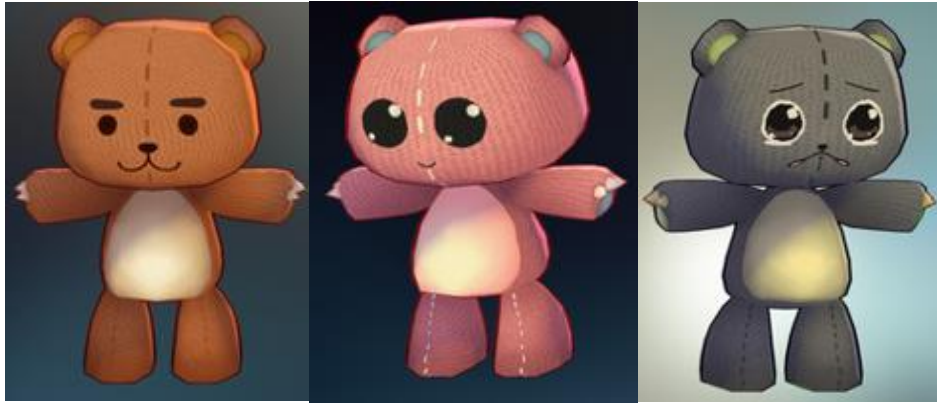
static 오브젝트들을 다 불러온 후 다다음주(15주차)에 다시 상의

3. 서버 패킷 – 무엇을 주고받을지

S_ : 서버에서 전송하는 패킷	C_ : 클라에서 전송하는 패킷
S_LOGIN	C_LOGIN
bool success	string 닉네임
uint64 플레이어ID	
S_LOBBY	C_LOBBY
bool 모든 플레이어 준비?	uint64 플레이어ID
	bool isPlayerReady
S_ENTER_GAME	C_ENTER_GAME
uint64 플레이어ID	uint64 플레이어ID
S_CHAT	C_CHAT
uint64 플레이어ID	uint64 플레이어ID
string 메시지	string 메시지
S_MOVE	C_MOVE
uint64 플레이어ID	uint64 플레이어ID
float 플레이어 위치	float 플레이어 위치
uint64 플레이어 이동 방향	uint64 플레이어 이동 방향
bool success (벽과 충돌하면 실패)	
S_ITEMCREATE (아이템 생성)	C_ITEMCREATE (아이템 생성)
float 아이템 위치	bool success
uint 아이템 index	
-> 아이템 인덱스에 보물까지	
S_HANDOVER_PLAYER (다른 플레이어와 충돌하여 술래 넘겨주는 패킷)	C_HANDOVER_PLAYER (다른 플레이어와 충돌하여 술래 넘겨주는 패킷)
uint64 술래의 플레이어ID	uint64 술래의 플레이어ID
uint64 술래 넘겨받을 플레이어ID	uint64 술래 넘겨받을 플레이어ID

서버 패킷은 게임 만들면서 수정될 수 있음.

4. 플레이어 3인 에셋 정하기



5. 일반 아이템, 유니크 아이템, 보물 생성 주기

총 플레이타임: 3분

일반 아이템 - 30초마다 모든 일반 아이템(5종) 하나씩 생성

유니크 아이템 - 1분마다 모든 유니크 아이템(2종) 하나씩 생성

보물 - 1분에 하나씩 생성

- 게임 일정 관리 관련

교수님께서 전체 게임을 관리하고, 각자 일정을 관리해줄 수 있는 사람이 필요하다고 하셔서 은비가 그 일을 담당하기로 했다.

임시 회의때 정한 주차별 계획을 바탕으로 매일 어떤 작업을 했고, 어떤 문제가 있었으며 어떻게 해결했는지와 같은 내용을 trello에 정리해놓기로 했다. 월요일 회의 후부터 오늘(3/26) 까지의 일별 일지를 작성한 상태이다.

김우찬	박소영	고은비
3/21 월 ≡	3/21 월 ≡	3/21 월 ≡ 1/1
3/22 화 ≡ 1/2	3/22 화 ≡ 2/2	3/22 화 ≡ 0/1
3/23 수 ≡	3/23 수 ≡	3/23 수 ≡ 2/3
3/24 목 ≡	3/24 목 ≡ 3/5	3/24 목 ≡ 3/4
3/25 금 ≡	3/25 금 ≡	3/25 금 ≡
3/26 토 ≡	3/26 토 ≡ 0/1	3/26 토 ≡
+ Add a card	+ Add a card	3/27 일 + Add a card

[1] 김우찬 (서버)

주간목표 :

- main브랜치에 있는 실제 게임에도 서버 붙이기 (100%)
- 충돌 버그 해결하기 (100%)
- 플레이어 접속 동기화 (100%)
- 플레이어 이동 동기화 (50%)

Main에 있는 프로젝트에도 서버 붙이고 버그 해결함.

다시 테스트해보니 내가 만든 충돌 로직에는 문제가 없다. 충돌 시 밀리는 부분에서 이동 방향의 반대 방향으로 밀어주다 보니 어색하게 이동하는 문제를 충돌한 순간 방향 정보를 받아서 충돌 상태일 때 그 방향으로만 이동 패킷을 보내지 않는 방식으로 수정하니 깔끔하게 해결되었다.

하지만 문제점이 한 방향으로 이동하다가 부딪히면 깔끔하게 막히는데 앞으로 가면 서 좌,우 키를 막 누르면서 부딪히면 방향 정보를 받아서 그 방향으로만 이동 못하게 막는 로직으로는 뚫는 버그가 생긴다. 이 문제를 해결하려고 생각해보니 우리 게임은 좌,우로 이동하지 않고 카메라를 요리조리 돌려서 앞으로만 이동하기 때문에 지금 방식으로든 문제가 없다고 판단하여 우선 플레이어 동기화부터 시작하였다.

이 부분은 이동 동기화까지 완료하고, main에 구현되어있는 카메라 이동까지 완성된 프로젝트에서 병합 후 테스트해 볼 예정이다.

클라이언트가 접속하면 바로 enterGame패킷을 보내는 것이 아닌 login패킷을 보내 어 서버에서 map컨테이너로 클라이언트ID와, 플레이어 객체를 저장하는 방식으로 플레이어 정보를 관리하였다. 서버에서 임의의 InGame씬을 만들어서 서버로 접속한 플레이어들은 InGame씬 안의 컨테이너로 관리하고, 앞으로 인게임 씬 안에서의 모든 행동은 플레이어ID로 구분하여 모든 플레이어들에게 BroadCast하는 방식으로 동기화를 맞출 생각이다.

서버에서 새롭게 접속한 플레이어가 있을 때 각 클라이언트로 플레이어 정보들을 넘겨준다. 클라이언트에서는 이 정보를 받고, 플레이어ID 따라서 동적으로 새롭게 다른 클라이언트의 플레이어를 생성한다.

플레이어 접속 동기화까지는 테스트 완료했고, 남은 주말동안 이동 동기화까지 완료 할 생각이다.

[2] 박소영 (클라이언트)

1. 카메라 회전 (100%)

좌우 키를 누르면 카메라와 플레이어가 회전하도록 구현했다. 우리 게임은 3인칭 시점이라 플레이어가 회전하면 카메라가 플레이어를 공전해야 하는데, 그 부분에서 계속 오류가 발생해 행렬 계산을 하나씩 다시 생각해보면서 코드를 쳐봤더니 오류가 수정되었다. 이렇게 카메라 작업이 완료되었다.

원래 기획발표때는 마우스로 에임을 조절한다고 했는데, 우리게 게임을 기획할 때 참고한 게임인 Freeze Bomb 영상을 보니 키보드로 플레이어 방향을 바꾸는듯 했고,

그렇게 하는게 구현하기도 편할 듯 하여서 팀원들에게 의견을 물어보고 키보드 입력으로 수정하였다.

플레이어는 좌우 키로 방향을 바꾸고, 위아래 키로 이동할 수 있다.

2. 아이템 작업 시작 (30%)

아이템(일반, 유니크, 보물) 클래스를 어떤 식으로 만들지 정리해보고, 회의때 정한 아이템 생성 주기에 따라 아이템을 생성하는 함수를 만드는 작업부터 하고 있다. 아이템은 일단 큐브로 띄워보고 잘 생성되는 것이 확인되면 은비에게 모델 어떤 식으로 띄웠는지 물어보고 그대로 적용해 봐야겠다.

[3] 고은비 (클라이언트)


주간목표 및 진척도

- 플레이어 애니메이션 렌더링 (100%)
- 키 입력에 따라 다른 애니메이션만 재생 (100%) IDLE, MOVE, JUMP, DASH, ATK

드디어 플레이어 애니메이션을 구현했다. 저번주에 애니메이션 정보를 다 받아와서 애니메이션 정보를 포함한 구조체를 기반으로 애니메이션 컨트롤러까지 구현했었다. 하지만 이 방식대로 하는 것보다 애니메이션 동작별로 정보를 담고 있는 클래스를 구현하는 것이 더 편할 것 같아서 애니메이션 컨트롤러를 수정하고 관련 클래스들도 추가로 구현하였다. (AnimationSet, AnimationTrack)

애니메이션이 돌아갈 때 마다 플레이어의 계층구조도 변경되어야 한다. 따라서 계층구조에 대한 정보를 Transform 컴포넌트로 갖고 있도록 하였다. (계층구조에 대한 정보는 world행렬, toParent행렬, 뼈 이름, parent, child, sibling에 대한 포인터 밖에 없기 때문에 하나의 GameObject 객체로 만들 필요는 없다고 생각함.) 그리고 루트 Transform을 플레이어의 Transform 컴포넌트의 child로 설정해서 플레이어가 계층구조를 갖고 있도록 하였다. 그리고 애니메이션 컨트롤러에서 동작을 재생시키고 플레이어의 로컬 변환까지 끝난 후에 전체 계층구조를 업데이트하는 방식으로 구현했다.

애니메이션 컨트롤러에서 애니메이션 변환행렬을 업데이트한 후, 플레이어의 계층구조를 이루는 뼈들 중 애니메이션에 활용되는 뼈들의 toParent행렬에 방금 애니메이션 컨트롤러에서 업데이트한 변환행렬을 넣어주고 마지막에 플레이어의 계층구조를 업데이트해서 월드행렬을 다시 설정해주었다. 그리고 애니메이션에 사용되는 뼈들의 월드행렬을 셰이더로 넘겨주었다. 오프셋 행렬은 애니메이션 변환 행렬과 다르게 한번 설정되면 바뀌지 않으므로 처음에 한 번만 넘겨주었다.

	
다음주 할 일	<p>[0] 공동 3. 31) 정기 모임(14주차)</p> <p>[1] 김우찬 (서버)</p> <ul style="list-style-type: none"> - 플레이어 이동 동기화 마무리 - 맵에 모든 static오브젝트들을 불러와서 충돌 테스트 - 아이템 생성 동기화 - 플레이어-아이템 충돌 패킷 전송, 확인 <p>[2] 박소영 (클라이언트)</p> <ul style="list-style-type: none"> - 아이템 랜덤 생성 - 아이템 효과에 의한 플레이어 상태 or 능력치 변경 - 보물 생성 <p>[3] 고은비 (클라이언트)</p> <ul style="list-style-type: none"> - 데이터 파일로부터 오브젝트들의 정보 불러와서 맵 렌더링 - 플레이어 상태 구현
문제점	<p>[0] 김우찬 (서버)</p> <p>게임 서버 시간에 분기문/제어문이 많으면 성능이 떨어진다고 하셨는데 현재 서버에서 for, if, switch문이 너무 많다.</p> <p>현재 오브젝트 하나로만 충돌테스트 해서 프레임 문제가 없지만 모든 맵의 오브젝트들을 불러왔을 때는 문제가 생길 수 있다. 다음 주에 맵의 모든 오브젝트들을 불러와서 테스트해 보고 문제가 있다면 최적화 방법을 고민해 봐야 할 수도 있다.</p> <p>[1] 박소영 (클라이언트)</p> <p>여태까지 중간발표 전까지 대부분의 작업을 완료해야 겠다는 생각은 하고 있었지만, 해야 할 일들을 리스트업 해서 계획을 세운 적은 없었다. 월요일에 교수님께서 해주신 조언을 듣고 팀원들과 중간발표 전까지 우선적으로 해야 할 일들을 정리하고, 계</p>

	<p>획을 세우니 앞으로 어떤 작업들을 해야 할지가 머리에 잡혔다.</p> <p>[2] 고은비 (클라이언트)</p> <p>제일 걱정됐던 애니메이션을 구현해서 마음이 조금 가벼워졌다. 남은 작업들도 빨리 해서 4월까지 게임이 진행될 수 있도록 해야겠다. 소영이가 지금 아이템이 랜덤으로 생성되는 작업을 하고 있고, 우찬이 오빠가 지금 충돌을 구현하고 있어서 빨리 맵을 불러와서 테스트를 할 수 있도록 맵 렌더링을 우선적으로 구현해야겠다.</p> <p>내가 맡은 파트만 신경쓰지 않고 전체적으로 개발이 어느정도까지 진행됐는지 항상 체크해야겠다. 그동안 너무 내가 맡은 파트만 신경썼었다. 전체적인 개발 일정을 항상 생각하고 팀원들과 소통하며 구현해야겠다.</p>
--	---