|  |  |  |  |
| --- | --- | --- | --- |
| 도둑들 | | | |
| **4월 3주** | **2023. 4. 16 ~ 2023. 4. 22** | **작성자** | **김혁동** |
| **이번 주**  **목표** | **김혁동: Collider 데이터 unity로부터 export**  **이도영: 팀원과 대면 후 move 관련 코드 완성**  **정극훈: Input렉 문제 해결**  **클라이언트-서버 연결 및 동기화** | | |
| **4/16**  **일** | **이도영:** Keytype에 따른 move 코드 추가 key 입력시 Vec값 time 저장하여 Player에 저장한 뒤 Send  텍스트이(가) 표시된 사진  자동 생성된 설명  오류 발생 클라이언트가 움직이지 않음.  Send와 recv의 값들을 저장하여 보았지만 클라 서버 통신간에는 문제 X  서버에서 변경된 값을 클라에서 받은 뒤에 값을 새로 저장할 때에 문제가 발생.  GetVec();를 통해 packet의 값을 저장하는 과정 이후 Vec값이 0으로 초기화 됨.  문제 해결을 위해 디버깅을 통해 지속적으로 값이 바뀌는 이유를 찾아보았지만 이유를 찾지 못함.  오류 예상  - 씬 update 과정에서 새로운 값을 호출하여 0으로 초기화 됨.  - 다른 곳을 호출하여 GetVec()을 함.  - 잘못된 곳에 저장하여 Vec 값이 계속 초기화 됨. | | |
| **4/17**  **월** | **이도영:** 서버에서 좌표를 직접 관리하기로 함. 굳이 클라에 좌표를 저장하지 않고 모든 move 좌표를 서버에서 관리하기로 함.  ForwardVec 삭제 프로토콜 다시 만듬.  텍스트이(가) 표시된 사진  자동 생성된 설명  Object의 초기값을 아이디를 받게 되면 서버에서 값을 다시 보내준 뒤 이동시 마다 계산하여 지속적으로 값을 전송해줌.  ‘O’ 누를시 gamestartpacket 전송 gamestartpacket recv시 서버에서 오브젝트의 정보를 넘겨주는 obj\_info packet을 전송하여 object의 값을 전송이 move값을 id에 맞추어 서버에 저장한 뒤 이동시 마다 packet의 값을 변경.  16일에 발생한 move 관련 오류 해결이 안됨.  일단 move 관련 오류 전 일단 id 값을 여러 개 받아 id값을 map에 저장하여 id 값 마다 mover의 값을 전송해 주는 서버코드 작성 후 실행화면에 cout하여 확인하며 코딩 함.  Move 관련 코드는 클라이언트와 시간을 맞추어 같이 고치기로 함. | | |
| **4/18**  **화** | **이도영:** Room의 id 값을 받아 같은 room 안의 플레이어 에게만 정보를 보내주는 코드를 작성  Room에는 round를 관리하는 time, room상태를 알려주는 state, obj값을 저장하는 list  일단 모든 object의 room 값은 1로 설정  Player startpacket이 올 때에 Id 값이 list에 저장된 뒤,  텍스트이(가) 표시된 사진  자동 생성된 설명  Match\_list의 id 값을 room에 넣어주어 모두 저장한다.  텍스트이(가) 표시된 사진  자동 생성된 설명  그 후 obj\_list에 room에 있는 obj를 전달해 준 뒤 obj\_list에를 이용해 플레이어를 관리한다.  이후 이 곳에서 enemy도 관리할 예정  **정극훈:** 프레임이 낮을 때에만 인풋 렉이 발생한다. 인터넷에 검색한 결과 directx12에서 인풋 렉과 관련된 자료를 찾지는 못했다… 그래서 몇 가지 추측을 해봤는데   1. 입력에 너무 많은 데이터가 포함되어 있어서 한 프레임에 다 해결하지 못해서 발생 2. 입력 버퍼에 키보드입력이 들어가는 속도가 프로그램이 입력 버퍼에 있는 데이터를 처리하는 속도보다 빨라서 발생 3. 프레임관련 코드에 문제가 있어 입력이 비정상적으로 많이 들어가거나 처리를 못해서 발생 4. 코드의 최적화 문제   네가지 중 답이 있지 않을까 생각했다.  일단 첫번째 추측을 해결하기 위해 입력 버퍼를 제어하는 방법들을 검색해보았다. 인터넷에 검색해보니 MS에서 입력 버퍼 이벤트와 관련된 내용을 찾을 수 있었다. 그 중에 콘솔 입력 버퍼에서 데이터를 읽고 제거하는 ReadConsoleInput 함수를 사용해보았다. Input 클래스를 실행하고 함수를 호출해보았으나 바뀌는 게 없었다. 그래서 콘솔 입력 버퍼를 플러시하는 FlushConsoleInputBuffer 함수를 사용했음. 이 함수는 MS에서 사용하지 않는 것을 추천하지만 이 함수를 사용해 버퍼를 플러시하는 것으로 문제가 해결된다면 버퍼를 플러시하는 다른 방법을 찾을 생각이였다. 하지만 이 경우에도 아무것도 바뀌지 않았다. 두 경우를 시도해보니 입력 버퍼 자체는 원인이 아닌 것 같다는 생각이 들어서 첫 번째 경우는 원인에서 제외했다. | | |
| **4/19**  **수** | **이도영:**  클라이언트와 함께 코드를 변경  발생하는 오류의 근본적인 원인을 분석  오류를 발생하는 지점이 클라이언트 프로젝트에서 엔진 프로젝트로 전달될 때에 문제가 발생한다.  모든 값이 전달할 때에 0으로 초기화가 된다.  전달하지 않고 서버에서 클라로 packet이 올 시 이 패킷을 다른 저장소에 저장하지 않고 직접적으로 엔진에 전달하기로 함.  직접적으로 엔진에 전달 시 오류가 해결되나. 근본적으로 저장소를 통해 전달 시 이유가 생기는 원인이 규명되지 않아 추후 더 이야기 해보기로 함.  클라이언트 업데이트 함수안에 send recv update가 함수 하나안에 다 있었는데 이럴시 recv가 일어나지 않아도 update가 지속적으로 일어나는 문제가 발생하여 send recv update를 나누어 각 멤버 함수로 작성  클라이언트 서버 해야 할 일들 재지정 중간 발표 전 어디까지 목표를 잡고 제작을 할 것 인가.  서버는 맵이 만들어진 다면 충돌체크까지 진행할 예정  일단 동기화 코드 작성 후 충돌체크, 충돌체크가 빠른 시일안에 해결되면 게임 컨텐츠 손대기로 함. 목요일 교수님을 뵌 뒤 좀 더 자세히 계획하기로 함  **정극훈:** 아직까지 명확한 원인을 찾지 못했다. 일단은 프레임 관련 기능에 문제가 있는지 확인해보았음. 도둑들의 프레임은 Timer class에서 CPU클럭을 확인해 프레임을 측정함. 그리고 모든 코드들은 Engine class에 포함되어 있고 Engine class의 다른 코드들은 Timer class가 업데이트되면서 프레임이 바뀔때마다 같이 바뀌므로 프레임 관련 코드에 문제는 없다는 것을 확인. Timer class의 프레임관련 코드는 원인이 아닌 것으로 파악했다. 그래서 입력버퍼에 인풋정보가 들어가는 속도가 프로그램이 동일한 시간동안 처리하는 속도보다 빨라서 발생하는 문제인지 추측했다. 입력버퍼에 입력 레코드가 들어가는 속도를 조절하는 것은 불가능했음. 키보드 입력이 버퍼에 들어가는 것은 프로그램의 업데이트과 무관하므로 프레임 관련 코드로 조절하는 것도 불가능함. 그리고 입력버퍼에 들어가는 입력 레코드를 조절하는 것도 문제를 해결하는데 도움이 되지 않았음. 결국 2번째, 3번째 추측도 원인이 아니였고 4번째 코드 최적화 문제만 남았는데 인풋 관련된 코드 전체를 다시 확인할 예정. | | |
| **4/20**  **목** | **이도영:** 클라이언트와 서버 동기화를 위한 코드 작성  Server: packet\_manager의 room에 따른 object를 저장하는 자료구조를 제작  Room과 로비 관련 ui 코드가 없기에 모든 room 번호를 1로 설정함.  client에서 ‘O’를 누를시 게임시작 CS\_PACKET\_GAME\_START 패킷이 서버로 오게 됨.  서버에서 packet을 process\_packet에서 패킷 판별 후 PacketManager::ProcessGameStart 함수로 이동  Player id를 객체에 저장 후 id 값의 player ready 상태를 true로 변경  Player의 room ID 값을 받은 후 room의 id 값에 맞는 objList를 얻어 안에 있는 player의 정보를 얻는다.  만약 플레이어가 Ready 상태가 아니면 거기서 리턴 하도록 함.  텍스트이(가) 표시된 사진  자동 생성된 설명  그 후 룸 아이디에 따른 void PacketManager::StartGame(int room\_id) 메서드 호출  패킷을 받은 후 Room\_id에 따른 room 호출 후 저장된 room\_id의 obj\_list[i]에 따른 player를 맵에 스폰 시킨다.  동기화 테스트를 위해 test\_start 함수를 사용하여 player의 pos 값을 초기해 준 후 Start의 ready 상태에 따라 게임을 시작하는 것이 아닌 일단 맵 안에 접속하게 한 뒤 지속적으로 패킷 값을 받아오게 설정하여 동기화 테스트 예정.  테스트 결과 서버에서 모든 아이디 값으로 데이터를 보내는 것을 확인함.  **정극훈:** 드디어 인풋렉의 원인을 찾았다. 객체를 움직이는 스크립트들을 확인하는 도중에 객체의 회전을 담당하는 Rotation함수를 제외하니 인풋 렉이 발생하지 않았다.  Rotation 함수에는 마우스의 현재 좌표를 받고 마우스를 화면 가운데에 고정시킨다음 중앙값과 마우스의 현재좌표의 차이를 계산해서 객체를 회전시키는 함수인데 마우스를 움직일 때 항상 커서를 가운데로 고정시키는데 사용했던 SetCursorPos함수가 호출될 경우 인풋 렉이 발생하였다.  SetCursorPos는 호출될 때 다른 이벤트를 무시하고 강제로 커서를 이동시키는데 이러한 특성 때문에 키보드 입력이 제 때 처리되지 못했던 것이 문제였다.  SetCursorPos를 제외한 다른 함수를 사용해 마우스 입력을 제어할려고 했으나 사용할 수 있는 함수가 SendInput이고 사용하기 난해한 함수여서 일단 SetCursorPos 함수를 잘 활용해서 문제를 해결하기로 했다. 일단 SetCursorPos의 위치와는 상관없이 인풋 렉이 발생하는 것을 확인했다. 그렇다면 SetCursorPos가 여러 번 호출되는 것이 문제다.  지금은 매 프레임마다 SetCursorPos를 호출하는데 대신 커서가 화면 밖을 나갈 때마다 함수를 호출하는 것으로 바꿨다. 바꿀 때 코드가 살짝 바뀌는 부분이 있었지만 동작은 동일하게 하는 것으로 확인했다. 이 경우에는 인풋 렉이 항상 발생하진 않지만 움직이는 도중에 SetCursorPos가 호출되면 인풋렉이 발생했다.  **김혁동:** 교수님을 만나 뵙고 난 이후, 팀 전체가 졸업작품 보고서에 대한 이해를 잘못한 것으로 이야기가 진행됨. 교수님께서 카톡으로 교수님께 보여드리기 위한 것이 아닌 자신들을 위해 적으라는 말을, 팀원이 이해할 수 있는 정보를 적음과 동시에 교수님께는 간략한 정보만 보내도 된다는 이야기로 이해하였음. 그로 인해 보고서에 내용을 자세히 적지 않아, 교수님이 팀의 진행상황을 잘 알지 못함과 동시에 남아 있는 데이터가 부족할 것이라고 지적하심. 이후부터는 간략한 진행상황만이 아닌, 직접적인 시행착오 및 진행상황에 대해 자세하게 서술하여 보고서에 적기로 팀원들과 이야기를 나눔.  교수님이 현재 상황으로서는 통과하기 어렵다고 하셨으나, 팀이 할 수 있는 곳 까지는 해보기 위하여 발표 날인 5/8일 까지 최대한 진행하기로 회의에서 결정됨. 나의 경우는 플레이어인 도둑뿐만이 아닌 경찰의 **애니메이션도 최대한 빠르게 만든 뒤,** **FBX를 읽을 때 UV가 부서지는 문제를 해결**하고 나서, **맵을 최대한 빠르게 제작하여 충돌 처리**까지 할 수 있도록 하기로 함.  맵 제작까지는 이미 기획서에 맵의 구조가 그려져 있으므로 에셋만 어느 정도 갖추어 지면, 시간 안에는 만들 수 있을 것으로 보임. 그러나, 충돌 처리는 클라에서만 처리할 수 있는 문제가 아니기 때문에 서버의 진행도에 따라 가능성이 달라질 것임.  또한, 충돌 처리를 위해서는 맵을 제작할 때에 콜라이더도 추가하여 모든 콜라이더들에 대한 데이터를 서버에 전달해 줄 수 있어야 함. 이를 위해 유니티 엔진에서 파일 입출력을 할 수 있는 방법을 찾아야 함. | | |
| **4/21**  **금** | **정극훈:**  **김혁동:** **블렌더를 사용하여 애니메이션 작업을 모두 완료**함.    플레이어인 도둑의 경우 뛰기, Idle, 공격, 점프 4가지의 모션을 구현하고, NPC인 경찰의 경우 뛰기, 점프 2가지의 모션만을 사용하기로 이야기함.  이후, 맵 에셋을 구매하여 맵을 구현하기로 함. 기본적으로 유니티를 사용하여 모두 배치하고, 콜라이더도 유니티에서 배치하여 파일 입출력으로 콜라이더 및 FBX데이터를 빼내와서 적용하기로 결정함.  **FBX를 로딩할 때에 시간이 조금 걸리는 문제**가 있음. FBX 메쉬를 불러오는 데에 걸리는 시간이 아닌 애니메이션을 받아올 때에 시간이 걸리는 것으로 보임. 이 Mesh를 하나씩만 로드하고 복사를 하는 형식으로 하지 않으면 게임 실행 시 시간이 매우 오래 걸릴 것으로 보이므로 극훈이와 상의하여 방법이 있는지 확인할 예정. | | |
| **4/22**  **토** | **이도영:** 스타트 버튼을 누를 시 로그인하여 서버에서 object 정보를 받는 코드 작성 중 오류 발생  스타트 버튼을 누를 시 로그인이 여러 번 되어 서버에서 여러 객체를 받게 되어 유령 id 값이 생기는 오류가 발생  유령 아이디에 모두 데이터를 전송하여 심한 바틀넥이 생기게 됨.  스타트를 할 때에 예외처리를 하여 한번 만 로그인 되게끔 코드를 고침  텍스트이(가) 표시된 사진  자동 생성된 설명  하지만 이럼에도 불구하고 여러 유령 id가 생기게 되어 \_start 값을 디버깅 하니 계속해서 \_start값이 false로 바뀌게 됨.  이는 스타트할 시 씬이 여러 개 생기며 그 과정에서 \_start값이 false로 변경되는 오류였음. GetButton()이 아닌 GetButtionDown으로 버튼이 다운될 때에만 한 번 스타트를 하도록 씬이 여러 개 생기지 않도록 변경 후 오류 수정  텍스트이(가) 표시된 사진  자동 생성된 설명  **정극훈:** 현재 코드는 GetKeyboardState 함수로 키보드 입력을 받고 동일한 Update에서 마우스 입력을 받는다. 입력 자체는 동시에 받기 때문에 SendInput을 사용할 필요가 없다.  일단 챗gpt는 엉뚱한 답을 한 것 이였고 해결책이 되지는 못했다. 그래서 결국에는 SetCursorPos를 사용하긴 해야 한다.  가장 확실한 방법은 멀티 스레딩을 사용해 키보드와 마우스를 다른 스레드에서 처리해야하나 현재 코드는 멀티 스레드를 지원하지 않는다.  그래서 어제 생각했던 SetCursorPos 함수 호출을 최대한줄이는 방법으로 코드를 구현했다. 지금은 임시방편으로 구현해놨지만 추후에 멀티 스레딩으로 바꾸거나 더 좋은 방법을 찾으면 그 방식대로 코드를 구현해야한다.  **김혁동:**  텍스트이(가) 표시된 사진  자동 생성된 설명**간단한 큐브에 대해서 UV 문제가 해결**함. 그러나 애니메이션이 있는 데이터의 경우 깨져서 움직이게 되는 것을 확인함.    텍스트이(가) 표시된 사진  자동 생성된 설명  위 코드와 같이, UV를 받아오는 데이터에 대해 이미 데이터가 추출된 정점은 새로 정점을 만들고, 그 이외에는 정점의 데이터를 변경하는 방식으로 변경하였음.  텍스트, 밤하늘이(가) 표시된 사진  자동 생성된 설명  애니메이션이 있는 FBX의 경우는 문제가 생기는 것을 확인. 정점 수를 늘렸으나, 해당하는 정점 들에는 애니메이션 데이터가 들어 있지 않기 때문인 것으로 추정됨. 새로 생성된 정점들에 대해서도 weight치를 설정해 주어 제대로 표현하도록 하여야 함.  텍스트, 밤하늘이(가) 표시된 사진  자동 생성된 설명  **UV가 깨지지 않는 FBX를 읽어 들이는 데 성공**함.  텍스트, 시계, 벽, 실내이(가) 표시된 사진  자동 생성된 설명    텍스트이(가) 표시된 사진  자동 생성된 설명  overlappedIndex라는 UV가 겹쳐진 정점에 대해 저장하는 데이터를 추가함. int 두개를 가지는 변수로 만들어, 하나는 몇 번째 정점에 대해 UV가 중첩되었는지, 하나는 정점이 새로 몇 번째에 생겼는지에 대해 저장해 둠. 이후, 애니메이션 데이터를 읽어 들일 때에 FBX에 기록된 정점의 weight 데이터를 읽을 때에, 같은 정점의 데이터를 같은 weight로 설정하여 움직일 수 있도록 함. 이 부분을 추가함으로서 UV가 깨져서 나타나는 문제는 해결되었음.  그러나, **FBX를 읽어오는 과정에서 속도가 너무 느린 문제**가 있음. FBX를 그렇게 많이 읽지 않음에도 로딩에 굉장히 많은 시간이 걸림. 예상으로는 애니메이션 데이터를 읽어오는 데에 오랜 시간을 쓰게 되는 것으로 보임. 이 부분은 테스트를 해 보아야 함. 또한, vector에 데이터를 넣을 때에 push\_back으로 데이터를 넣고 있어 복사 생성이 많이 되고 있을 것으로 보이므로, 해당 부분을 emplace\_back으로 변환할 수만 있다면 시간을 좀 더 줄일 수 있을 것이라 생각됨. | | |
| **이번 주 문제점 및 해결방안** | **클라이언트에서 패킷을 받아 저장할 때에 값이 초기화 되는 문제가 있음.**  **코드 분석 및 디버깅을 진행하여 해결하는 수밖에 없음.** | | |
| **다음 주 계획** | **김혁동: 에셋 맵 제작, 맵 데이터 서버에 전달, 서버 도와주기**  **정극훈: 플레이어의 행동에 따른 애니메이션 출력, 로딩한 애니메이션을 여러 객체에 적용, 공격버튼을 추가하고 공격시 파티클 출력**  **이도영: MOVE 패킷 동기화 완료, 맵 제작 종료 시 충돌체크**  **서버 오류코드 변경** | | |
| **비고** | Github 도둑들 주소: <https://github.com/rmrgns/Thieves.git>  Github 졸업작품 회의록 주소: <https://github.com/rmrgns/gameproject_proceedings.git> | | |