Magic World 종합설계 작업일지

팀원 : 최규화, 이관구, 유창민

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 작성자  (학번,이름) | 최규화  2013182042 |
|  |  | 기간 | 2018-03-07 ~ 2018-03-14 |
| 이번주  작업일지 | 1.비동기 입출력이 가능한 boost asio를 사용하여 서버작업을 하기위한 공부.  2.서버에서 비동기 입출력을 통한 클라이언트로의 플레이어 초기화 정보 전달.  3.통신용 클라이언트로부터 로그인정보 동기 입출력을 통해 서버로 데이터정보 전달 수정.  4.서버 및 통신용 클라이언트 프레임워크 수정  (패킷 구조체 수정 -> 플레이어 데이터의 pos값 수정, Changed\_Pos 및 Changed\_Dir 구조체 내부 변경 ) | | |
| 다음주 할일 |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 작성자  (학번,이름) | 최규화  2013182042 |
|  |  | 기간 | 2018-03-15 ~ 2018-03-21 |
| 이번주  작업일지 | 1.패킷 송수신을 위한 비동기 IO를 기존에는 async\_write\_some과 async\_read\_s  Ome을 사용하였으나 이를 async\_write 와 async\_read로 바꿔 보내거나 받을 버퍼에 한번에 데이터를 쓰도록 함  2.서버에서 비동기 입출력을 통해 클라이언트로 초기화 정보를 전달하여 클라이언트에서 플레이어1이 해당 위치에 생성되는 것을 확인함  3.클라이언트에서 키조작을 통해 이동된 위치를 서버로 보내는 과정 확인. 서버에서 해당 데이터를 다시 클라이언트로 보내줄 때 (async\_read 부분) 버그 발생, 이를 수정 중에 있음  4.프로토콜 헤더파일의 패킷 구조체 수정 | | |
| 다음주 할일 |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 작성자  (학번,이름) | 최규화  2013182042 |
|  |  | 기간 | 2018-03-22 ~ 2018-03-28 |
| 이번주  작업일지 | 1. Async\_read를 사용하여 패킷을 수신하는데 문제가 있어서 다시 async\_read\_some으로 패킷 수신용 입출력함수를 바꿈. 이에 추가적으로 수정해야 되는 서버 부분을 수정 2. 클라이언트에서 서버와 연결 후 바로 RecvPacket 함수에 진입하여 ProcessPacket을 처리하도록 함 (이 부분을 수정하기 전에 RecvPacket과 ProcessPacket 함수를 따로 클라이언트의 FrameAdvance 부분에 뒀었는데, 서버로부터 받은 패킷을 RecvPacket함수에서 처리하지 못했었음) 3. 클라이언트에서 키 조작 (W,A,S,D)을 통해서 서버로 해당 패킷을 보내고 서버가 다시 이 패킷을 클라이언트로 보내는 것을 확인했음   즉, 캐릭터를 W,A,S,D 키를 눌러서 이동시킬 수 있음   1. 클라이언트에서 키 조작 패킷을 처리하는 부분(ProcessPacket 에서 ChangedPos 위치)을 수정하였음 | | |
| 다음주 할일 |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 작성자  (학번,이름) | 최규화  2013182042 |
|  |  | 기간 | 2018-03-29 ~ 2018-04-04 |
| 이번주  작업일지 | 1. 클라이언트에서 키조작을 통해서 서버에 보내주는 패킷을 처리하는 부분 (ProcessPacket의 STC\_ChangedPos 패킷 처리부분 수정) 2. 다수의 클라이언트 접속에 대한 처리를 하기위해서 이에 따른 패킷구조체 추가 및 변경 (SetMyClient, SetOhterClient , 서버와 클라이언트에 OtherClient Packet을 처리하기 위한 추가코드 작성 ) 3. 서버와 클라이언트가 주고받는 클라이언트 정보가 담겨있는 Player\_Data 구조체 (pos, id, status, connect\_state 등을 담고있음)의 패킷 사이즈를 줄이기 위해서 기존의 int, bool 형 등의 데이터타입을 모두 short 혹은 char형으로 바꿈 4. 2명의 클라이언트가 서버에 접속하여 원하는 위치에 놓여지고 키조작을 통해 이동하는 모습을 서로의 클라이언트에서 볼 수 있도록 작업을 하였음 5. 클라이언트에서 연결된 서버에 대한 접속을 끊었을 때 상대방의 클라이언트에서 내 모습이 사라지도록 하기위해 Disconnect 패킷 처리를 위해 서버에서 이에 따른 코드 추가. 클라이언트에서도 접속이 끊긴 플레이어의 정보를 삭제해주는 작업을 해야함(아직 못했음) | | |
| 다음주 할일 | 1. 이동 애니메이션 추가 2. 플레이어 공격 및 공격 애니메이션 추가 3. 클라이언트에서 접속을 끊었을 때 상대방의 클라이언트에서도 자신의 모습이 보이지않게 추가 작업필요 4. 구버전 클라이언트에서 신버전 클라이언트와 연동되게끔 변환작업필요 | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 작성자  (학번,이름) | 최규화  2013182042 |
|  |  | 기간 | 2018-04-04 ~ 2018-04-11 |
| 이번주  작업일지 | 1. 플레이어 2명이 이동하는 것 구현 2. 플레이어의 회전 정보를 서로 주고받아 서로 다른 클라이언트에서 서로 다른 캐릭터가 회전한 모습을 볼 수 있음 3. 신 버전 클라이언트(2.5version)와 연동 4. StaticObject인 상자에 대한 정보를 서버에서 보냄 5. 서버에서 충돌처리작업중 | | |
| 다음주 할일 | 1. 공격 애니메이션 및 기능 추가 2. 클라이언트에서 접속을 끊었을 때 상대방의 클라이언트에서도 자신의 모습이 보이지않게 추가 작업 필요 3. 플레이어 공격에 대한 충돌처리, StaticObject(상자) 충돌처리 | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 작성자  (학번,이름) | 최규화  2013182042 |
|  |  | 기간 | 2018-04-12 ~ 2018-04-18 |
| 이번주  작업일지 | 1. 서버로의 패킷 송신에 문제가 생김, 이 현상을 해결 중  * 이동과 회전을 동시에 했을 때 서버의 RecvPacket 함수에서 클라이언트 패킷을 처리할 때, 이전에 처리해야 될 패킷 정보를 다음 패킷 정보가 덮어씀. * 서버에서 SendPacket 함수에서 RecvPacket 함수를 호출해서 위 현상 발생. RecvPacket은 반드시 Accept가 끝난 뒤 한번 호출하고 RecvPacket 함수에서 ProcessPacket을 통해 패킷 처리를 하고 다시 호출하는 방식으로 선언해야 한다  1. 기존에는 캐릭터이동 및 회전을 할 때마다 클라이언트에서 서버로 패킷을 보냈는데, 이렇게 하니 1초에 수백개의 패킷이 서버로 전송됨. 서버에서 패킷 데이터 처리하는데 버그발생(처음에는 서버에서 넘어오는 패킷 대로 처리를 하다가 어느 순간 제대로 패킷 처리를 못 하게됨) 이를 해결하기위해서 클라이언트에 SendRegularPacket함수를 만들어 이동 및 회전이 일어났을 때 변화된 값을 저장하고 이 저장한 값을 위 함수에서 0.5초마다 서버로 보내도록 바꿈 (추후 1초에 서버로 몇 개의 패킷을 보내야 되는가를 생각해보고 초당 서버로 보내는 패킷의 개수를 달리해줘야 될 것 같음)  * 현재 1초에 20개의 패킷을 보내도록 수정 (20 ~ 30이적당?)  1. 클라이언트에서 마우스 클릭을 했을 때 큐브가 발생되고 해당 큐브 정보를 서버로 보내는 bullet data에 대한 sendpacket 코드 추가 및 서버의 ProcessPacket함수에서 위 패킷 정보를 받아 처리할 수 있도록 코드 구현. 2. StaticObject인 상자에 대한 패킷 정보를 서버에서 클라이언트로 보내줄 때, 첫 클라이언트는 잘 송신이 되는데, 두 번째 클라이언트에서 패킷 정보에 들은 위치 값이 전송되지 않는 현상이 발생했었음. 이를 해결함. -> 서버에서 SendPacket을 해줄 때 사용한 async\_write의 handler 호출 함수 부분에서 람다를 사용했었는데, 이를 레퍼런스 타입인 &로 설정해둬서 bytes\_transferred 값과 실제 받은 packet\_size 값이 서로 뒤죽박죽이 되어버렸었음. 이를 읽기타입인 = 로 바꿔서 해결 3. 서버에서 받은 플레이어 정보, StaticObject인 상자 정보, 공격 스킬에 대한 정보를 합쳐서 충돌처리해주는 Collision 함수를 따로 생성하여 Update 주기 마다 충돌처리를 해줘야 됨. | | |
| 다음주 할일 | 1. 클라이언트를 신버전(2.7version)으로 업데이트  * Hpbar 구현, 조준점 구현, 상자 오브젝트 추가 및 리지드바디 오브젝트 추가 및 위치 조정, 맵의 테두리에 벽 추가, 스킬 1개 추가, 스킬 쿨타임추가  1. 공격기능 추가 구현 2. 플레이어 접속 종료 시 후 처리 | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 작성자  (학번,이름) | 최규화  2013182042 |
|  |  | 기간 | 2018-04-19 ~ 2018-04-26 |
| 이번주  작업일지 | 1.물리효과 작업들을 서버에서 하게 끔 바꿈  - 가속도, 중력, 물리효과 클래스 업데이트  2. Bullet, Player, StaticObjects 에 전용 충돌 함수 제작  \* 추후 충돌 처리할 때 호출하여 사용  3. TimerWorker 클래스 제작  \* 서버에서 주기적으로 처리해줘야 되는 작업들을 실행  \* 별도의 스레드에서 작업처리  - WorkerThread와의 동기화 필요. 동기화는 lock을 통해서 구현  \* Bullet 생성 후 서버에서 주기적으로 위치를 업데이트 시켜서 모든 클라이언트 에서 Bullet이 움직이는 것이 보이게 해야 됨  4. protocol 업데이트  - Bullet 정보를 담고 있는 패킷의 구조 변경  \* 굳이 패킷으로 주고 받을 필요 없는 데이터 삭제  \* Bullet의 velocity 데이터 추가 | | |
| 다음주 할일 |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 작성자  (학번,이름) | 최규화  2013182042 |
|  |  | 기간 | 2018-04-27 ~ 2018-05-02 |
| 이번주  작업일지 | 1. TimerWorker 클래스 제작 중  \* Bullet 위치 변경 이벤트를 0.025초마다 주기적으로 실행  -> 클라이언트에서 Bullet과 서버에서의 Bullet 이동 및 생명주기 동기화가 안되는 문제 발생 ( 서버가 2초느리게 Bullet의 생명주기가 끝남)  2. 충돌 함수 제작 중  - 충돌 시 사용할 Bullet, Player, StaticObjects 클래스가 가지고 있는 충돌함수 부분 수정   * 시험 기간 및 연구과제가 겹쳐서 저번주와 이번주 모두 서버작업을 별로 못했음 | | |
| 다음주 할일 |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 작성자  (학번,이름) | 최규화  2013182042 |
|  |  | 기간 | 2018-05-02 ~ 2018-05-09 |
| 이번주  작업일지 | 2018. 05. 03  1. bullet 정보를 클라이언트로 주기적으로 보내는 작업 중  \* 패킷을 전송하는 부분과 클라이언트를 관리하는 부분을 담당하는 Player\_Session 클래스와 주기적인 패킷 혹은 이벤트를 처리해야 되는 클래스인 TimerWorker 클래스 간의 연동을 시도  -> 원래는 bullet 정보를 클라이언트로 주기적으로 보내는 작업을 Player\_Session 내부에서 하려 했으나 예상했던 방식으로 작동되지 않음. 그래서 위에서 언급한 방식으로, 클래스간 연동을 통해서 TimerWorker 클래스에서 작업처리를 하려함.  -> 또 문제발생. 1. TimerWorker 스레드는 Player\_Session처럼 많은 스레드를 할당받은게 아니라 callback함수들은 되도록 여기서 처리하면 안됨. but, sendpacket은 callback함수  2. 1초에 20번씩 sendpacket을함. 불렛리스트와 플레이어리스트를 모두 돌아야함 시간 복잡도 0(n2). 성능상에 큰 문제가 생김.  3. 위 단점을 극복하고 처리를 하려면 주기적 패킷을 처리하는 작업을 다른 스레드를 또 할당해서 처리해야함. 이렇게 해야하는가? 아닐듯 . . . 좀 더 고민해 봐야함  2018. 05. 04  1. bullet 에 대한 이동 관련  - 서버: 일단 데모에 시연할 캐릭터 수가 많지않으므로 시간 복잡도가 큰 편이여도TimerWorker 클래스를 통해서 주기적으로 클라이언트로 SendPacket해주는 방식을 선택  - 클라이언트: A 클라이언트가 공격을 눌렀을 때 A 클라이언트는 bullet 객체가 생성되지만, 다른 클라이언트는 클라이언트에서 서버로 보낸 bullet 객체에 대한 정보를 다시 받아서 클라이언트에서 따로 생성을 해야 한다.  [1]. A 클라이언트에서 공격 -> 서버로 bullet 데이터 전송 -> 서버에서 B클라이언트로 해당 bullet 데이터 송신  -> B 클라이언트에서 해당 정보를 가지고 bullet 객체 생성  = 이 때, 기존의 CreateBullet 함수로는 bullet 생성불가(GameObject 클래스자체를 서버와 클라이언트가 송수신할 수 없으므로)  별도의 CreateOtherClientBullet 함수 생성. 이를 이용. 여기에 문제가 있을 듯. 일단은 작성함  \* 또한, 클라이언트에서 bullet 아이디 관리. bullet 삭제 시 해당 아이디 삭제. 새로운 bullet 생성 시 삭제 된, 즉 가장 앞 순서에 있는 아이디를 넣어야함. -> 리스트를 이용해서 아이디 관리  [2]. [1]과정을 거친 bullet은 A와 B가 모두 bullet 데이터를 갖고 있기 때문에 주기적으로 서버에서 받은 데이터를 이용해 위치를 업데이트 시켜 줌    2. 충돌 처리 관련  - 서버: 충돌 처리를 해주는 클래스 생성, 이를 호출하는 스레드 할당.  일단, static 오브젝트와 플레이어. 플레이어와 플레이어간 충돌처리 작업중  - 클라이언트: 서버에서 충돌처리는 계속해주고 이 정보는 1초에 20번의 패킷을 통해 클라이언트로 보내주므로 크게 따로 처리해야 될 부분이 있다고는 생각이 들지 않음, 고민해 봐야 됨.  3. 새로운 클라이언트와 서버 연동  - 아직 진행 안 한 상태임.  2018. 05. 05 ~ 2018. 05. 07  1. Bullet 관련 수정  - 초기생성을 제외한 모든 bullet에 대한 작업을 서버에서 처리하도록 수정  - 서버에서 보낸 bullet 위치와 플레이어에게 보여지는 bullet의 위치가 달랐었고  클라이언트에서 사라지는 bullet의 위치가 서버와 달랐었는데, 이 부분을 수정.  즉, 클라이언트와 서버 간의 bullet delay를 없앰  - 클라이언트에서 GetFocus 함수 추가에 따른 bullet 생성 버그가 있었고 이를 수정  (기존에 GeteFocus함수를 사용하지 않았을 때에는 동시 키입력 때문에 하나의 클라이언트가 키를  눌러도 다른 클라이언트에서 bullet이 생성됐었음(따로 서버에서 정보를 받지 않아도)  그렇기 때문에 서버에서 주기적으로 bullet이 이동한 값만 받아도 정상적으로 2개의 클라이언트에  이동하는 bullet이 표시됐었음)  - Release 모드에서 Bullet이 이동할 때 프로그램이 터지는 현상은 아직 해결하지 못함.  Debug 모드에서는 Bullet이 잘 이동됨.  2018. 05. 07 ~ 2018. 05. 08  1. Bullet 관련 수정  - Release 모드에서 Bullet이 이동할 때 프로그램이 터지는 현상  -> 1. Release 모드에서는 Debug 모드와는 다르게 구조체나 new 로 할당된 객체가 생성될 때  초기화가 자동으로 되지 않음. 즉, 데이터의 초기화가 제대로 안돼서 발생하는 현상일 가능성이 높음  2. 멀티 스레드를 사용해서 그런 현상이 발생하는 것일수도 있음  2. Collision 작업중  - Bullet과 Player간의 충돌 처리 중  -> Player가 Bullet을 맞았을 때 hp가 한번만 감소하는게 아니라 연속적으로 감소되는 현상 발생  -> Bullet이 생명주기를 다했을 때와 캐릭터에 충돌했을 때의 순간이 겹쳐  이미 사라진 bullet 데이터를 접근하는 경우 발생  위 모두 동기화를 제대로 안 해줘서 발생한 것 같음. 현재 작업 중 | | |
| 다음주 할일 |  | | |