**이 력 서**

**1. 인적 사항**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C:\Documents and Settings\학생\바탕 화면\설은경.jpg | **성**  **명** | **한 글** | 설 은 경 | **지원부문 1** | Server |
| **한 자** | 薛 殷 璟 | **지원부문 2** |  |
| **주민등록번호** | | 890124 - | **생일(양)** | 1989. 01. 24. |
| **핸 드 폰** | | 010 6633 3848 | **전화번호** | 031 947 3848 |
| **주 소** | | 경기도 파주시 동패리 한울마을 603-902호 | | |
| **E- Mail** | | akeah@naver.com | | |

**2. 학력 사항(최근 학력이 위로 오도록 기재요망)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **기 간** | **출신학교명** | **학 과 (계열)** | **평점/만점** | **소재지** | **이수구분** |
| 07/03 ~ 12/02 | 한성대학교 | 멀티미디어공학 | 3.81 / 4.5 | 서울 | 졸업 |
| 04/03 ~ 07/02 | 신일정보산업고 | 컴퓨터그래픽과 |  | 경기 | 졸업 |
|  |  |  |  |  |  |

**3. 학원 프로젝트**

|  |  |  |  |
| --- | --- | --- | --- |
| **기 간** | **프로젝트** | **인원** | **담당** |
| 2012/05 ~ 진행중 | 3D | 5 | 서버/ DB |
| 2011/11 ~ 2011/12 | MFC | 3 | 맵툴/ 서버 |
| 2011/10 ~ 2011/11 | API | 3 | 메인 프로그래머 |
|  |  |  |  |

**4. 외국어 능력 / 자격사항 / 컴퓨터 활용 능력 (O.A)**

|  |  |  |  |
| --- | --- | --- | --- |
| **구 분** | **등급 및 수준** | **취득일** | **시행처** |
| 정보처리 | 기능사 | 2006/ 07/ 24 | 한국산업인력공단 |
| JLPT | 2급 | 2010/ 08/ 28 | 일본국제교육지원협회 |
|  |  |  |  |

**프로젝트 상세**

|  |  |
| --- | --- |
| 게임명 | Unknown (FPS) |
| 담 당  업 무 | 서버/ DB |
| 상 세  설 명 | 본 프로젝트는 1명의 기획자와 3명의 클라이언트, 1명의 서버로 구성되어 진행되고 있는 프로젝트로서 저는 서버/DB를 담당하고 있습니다.  서버는5개로 구성되어 있고 실제 클라이언트와 연결되는 서버는 login/ lobby/ game 서버 이렇게 세 개이며, 그 외로 DB서버와 Log서버가 있습니다. 서버는 모두 IOCP를 사용하고 있습니다.  서버간 연결 상태   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Client | |  | Login Server | |  | DB Server | |  | Lobby Server | | |  | 로그인 요청  로그인 결과 | | |  |  | 로그인 성공  \_캐릭터 정보  캐릭터 준비 완료 | | |  | | 로그인 확인 요청  로그인 결과 | | |   **로그인 처리 패킷 이동**  로그인 서버에서 로그인 요청을 받아 DB서버로 전송하게 되면 DB서버에서 로그인 체크를 하여 해당 캐릭터의 정보를 로비서버로 전송합니다. 정보를 받은 로비서버는 캐릭터의 정보를 임시적인 공간에 저장하고 유저를 받을 준비가 완료 되었다고 DB서버로 신호를 주게 됩니다. DB서버는 이 신호를 받아야만 다시 로그인 서버를 거쳐 클라에게 로그인 결과를 패킷으로 전송할 수 있고, 로그인 성공 패킷을 받은 클라는 로비로 이동할 수 있게 됩니다.   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Client | |  | Lobby Server | |  | Other Client | | |  | 로비로 접속  로비/게임에 있는 모든  캐릭터 정보  방 정보 | | | 캐릭터 정보 | | |  |   **로비로 접속**  예전 프로젝트에서 로그인 된 클라에게 모든 정보를 넘겨주고 로비서버에서 그 정보를 다시 받는 형식의 서버 구조를 만들었지만 보안상 완전히 클라의 정보만을 믿는 것은 위험하다는 생각에 이번 프로젝트에서는 모든 정보를 서버가 그 다음 서버로 넘기는 구조로 만들게 되었습니다. 클라는 로그인 후 자신의 정보로 고유 index ID만을 받아 로비로 전송하고 로비는 이를 확인하여 접속해 있는 캐릭터들의 정보와 방 정보를 모두 넘겨 줍니다. 물론 기존에 로비에 있는 유저들에게도 지금 접속한 유저의 정보를 전송합니다.   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Client | |  | Lobby Server | |  | Other Client | |  | Game Server | | |  | ready상태를 바꿈  ready상태가 바뀜  게임 시작  게임 시작 | | | ready상태가 바뀜  게임 시작 준비  게임시작 가능/ 불가능  게임 시작 | | |  | | |  |   **Ready와 게임시작**  방에 들어가 있는 유저들은 방장이 게임시작을 누르면 게임을 시작할 수 있는데, 이 버튼은 방장을 제외한 유저들이 모두 레디 상태가 되는 경우 서버에서 방장에게 신호를 보내주게 되어 있습니다. 게임 시작이 되면 로비서버는 우선 게임서버로 캐릭터들의 정보와 방의 정보들을 게임서버로 보내 준비를 하게 되어 있고 모든 준비가 끝났다는 신호를 받아야 모든 클라에게 게임을 시작하기 위해 게임서버로 옮기라는 패킷을 전송할 수 있습니다. 이때 클라는 각 방의 고유의 port번호를 받게 되는데 이는 캐릭터간에 주고받을 UDP설정을 위한 번호 입니다. 받은 port번호로 자신의 UDP서버를 열게 되고 게임서버에 접속하면서 받은 다른 유저들의 ip주소를 저장합니다. 게임이 시작되면 캐릭터들 간의 움직임은 서로 UDP전송을 이용하게 되고 상태와 공격 등 위치를 제외한 모든 패킷은 서버를 통해 받게 됩니다.  유저들은 게임서버에 접속해 로딩을 끝내고 모든 유저의 로딩이 끝났다는 신호를 받으면 게임서버에서 시작신호를 받아 무기 선택 화면으로 넘어가게 되는데 20초 동안 무기를 선택하여 게임을 시작할 수 있습니다. 게임 중 스캔과 은신 수치를 서버에서 관리하며 일정 시간에 따라 증가하는 수치를 클라에게 전송합니다.  또 한 유저는 1개의 지뢰를 사용할 수 있는데 신호를 받은 서버가 설치 가능 여부를 판단하여 해당 위치에 지뢰를 설치하고 지뢰에 따른 캐릭터와의 충돌체크를 합니다. 오브젝트에도 설치되는 지뢰로 방향성을 가지고 반구 형태의 체크를 하며 상대팀에 반응하게 되는데 터지는 지뢰의 체크에서는 아군까지도 피해를 입게 됩니다.  게임 종료 후 게임의 결과를 rank point(경험치 개념)으로 환산하여 DB서버로 전송하고 전송된 data는 lobby서버의 유저들에게 전송되며, 해당 data의 처리가 완료된 로비서버에서 게임서버로 신호를 보냅니다. 이 신호를 받아야만 게임서버에서는 유저들을 다시 로비로 이동하는 패킷을 보낼 수 있게 되는데 이를 받은 유저들은 다시 방으로 돌아가 기존의 다른 유저의 갱신된 정보를 받고 유저들과 다시 게임을 play할 수 있습니다.  DB서버는 위의 세 개의 서버와 연결상태에서 유저들의 data를 주고 받습니다. DB는 MS Sql Server 2008을 사용하여 만들었고, 간단한 저장프로시저를 만들어 서버 함수 내에서 사용하고 있습니다. 로그인 된 유저들의 정보를 DB에서 읽어와 저장하고 있으며, 로그인 서버에서 로그인 한 캐릭터가 로비서버로 넘어가기 전에 data를 먼저 로비서버로 전송해 주는 중개 역할도 겸하고 있습니다. 또한 유저들의 정보를 따로 가지고 있으니 중복으로 로그인 되는 것을 막아 주고 게임 종료 후 넘겨 받은 게임 data로 유저의 data를 계속 갱신하여 저장해 두었다가 유저가 게임을 완전히 종료하였다는 신호를 받으면 최신의 data를 DB에 갱신하게 됩니다.  Log서버는 개인적인 공부용 서버로 mfc를 이용해 제작되었지만 디버그에는 유용하지 않아 제작 중에는 실시간으로 콘솔창에 출력하는 로그클래스를 사용합니다. 실행하면 사용할 로그테이블과 서버를 생성하여 사용할 수 있으며 서버 설정을 할 때 원하는 로그테이블을 선택할 수 있습니다. 저장되는 로그는 서버에 따라 볼 수 있고, 메시지의 종류(info, error, 등)에 따라 확인할 수 있습니다.    **라이브러리 클래스 다이어그램( 부분 )**  **라이브러리 상세**   * **SObject** : SSession클래스에서 각 서버에 맞는 Session을 사용하기 위해 기본적으로 RTTI를 사용하기 때문에 SSessionMgr에서 세션을 생성하기 위해서는 반드시 이 class를 상속받아야 합니다. * **SServerObj** : 거의 모든 서버 객체들이 상속받는 클래스로 크리티컬 섹션 객체를 가지고 있는 클래스 입니다. SSyncronize class의 생성자와 소멸자를 사용하여 간단하게 동기화 객체를 사용할 수 있습니다. * **SSession** : 세션의 기본기능을 담고 있는 class로 사용자는 SSession class를 상속받고 있는 가상class인 SSessionObj를 상속받아 세션class를 만들 수 있습니다. * **SSocket** : 소켓객체를 간단한 함수를 사용하여 서버용 소켓을 만들거나 클라용 소켓으로 생성하여 서버에 연결할 수 있는class로 초기화, 소켓생성, 넌블럭킹 설정 등의 소켓기능을 함수로 제공합니다. * **SPacket** : 패킷 class. 데이터 형에 따라 Put/ Get 기능을 지원하며, 패킷 아이디, 크기등을 쉽게 가져 올 수 있습니다. * **SLogger** : 로그 클래스, 디버그 모드에선 콘솔창을 띄워 로그를 남길 수 있는데 이 로그는 설정되는 로그 레벨에 따라 디버그출력창/ 콘솔창/ text파일로 로그를 남길 수 있습니다. * **SIndexQueue** : 공간을 관리하기 위한 indexQueue class로 배열형식으로 만들어져 있으며 시작 인덱스와 인덱스의 크기를 설정하여 만들 수 있습니다. 더 이상 인덱스를 할당 받을 수 없을 경우 -1을 return함으로 공간의 제한을 확인할 수 있습니다. |
| 상 세  이미지 | 로그서버( 선택된 table의 로비서버관련 로그 )  **<< 아웃 게임 >>**  C:\Documents and Settings\학생\바탕 화면\모음\메인.png  로그인 화면 로비 화면  C:\Documents and Settings\학생\My Documents\GomPlayer\Capture\Sequence 01.avi_000033016.jpg  방 생성 방  **<< 인 게임 >>**  C:\Documents and Settings\학생\바탕 화면\모음\무기선택.png C:\Documents and Settings\학생\바탕 화면\모음\mainImg1.PNG  총 선택 게임 화면(기본)  C:\Documents and Settings\학생\My Documents\GomPlayer\Capture\Sequence 01.avi_000083000.jpg C:\Documents and Settings\학생\바탕 화면\stealth.png  게임 스킬(좌\_스캔, 우\_은신)사용 중 스킬 포인트 감소  C:\Documents and Settings\학생\바탕 화면\boom.pngC:\Documents and Settings\학생\My Documents\GomPlayer\Capture\UnknownR 2012-07-04 16-13-01-73.avi_000061816.jpg  설치된 지뢰 폭발 게임 종료 |
| 게임명 | Dungeon of Hell (쿼터뷰 액션게임) |
| 담 당  업 무 | 맵툴, 서버 |
| 상 세  설 명 | 본 프로젝트는1명의 기획자와 2명의 프로그래머로 진행된 프로젝트이며 이 프로젝트에서 맵툴과 서버를 담당하였습니다.  맵툴은 mfc를 이용하여 만들었고, .bmp 이미지 형식의 저장을 지원하며 기본적으로 게임 바닥으로 깔릴 맵 이미지를 만들어 저장한 후 다시 그 이미지 위에 오브젝트를 배치하며 맵 data파일을 뽑을 수 있게 만들었습니다. 마름모 방식의 타일은 마우스 클릭 위치와 맵 타일 배열을 계산하여 선택된 타일의 index를 얻는 방법을 사용하였고, 툴 내에서 각 타일의 기본적인 속성을 부여 할 수 있습니다. 타일 역시 타일 이미지의 index로 그 위치의 이미지를 직접 편집 할 수 있게 만들었습니다.  서버는 select모델을 사용하여 만들었습니다. 내부 네트워크를 사용함과 적은 인원수라는 점이 속도에는 큰 무리를 주지 않았습니다. 클라에서는 쓰레드를 사용하여 서버에서 보내는 패킷을 받아 처리 하였는데 매 프레임마다 캐릭터의 정보를 보내는 방법으로 캐릭터 간의 움직임, 공격 등의 동기화가 정상적으로 이루어 졌습니다.  서버는 총 3개의 로그인을 처리하는 로그인 서버와 캐릭터를 선택하여 사람들이 모이는 로비서버, 그리고 게임 play가 진행되는 게임서버로 나누어져 있습니다. |
| 상 세  이미지 | 맵툴  C:\Documents and Settings\학생\바탕 화면\Login.bmp C:\Documents and Settings\학생\바탕 화면\CharSel.bmp  로그인 화면 캐릭터 선택 창  C:\Documents and Settings\학생\바탕 화면\ReadyRoom.bmpC:\Documents and Settings\학생\바탕 화면\InGame.bmp  로비 게임화면 |

|  |  |
| --- | --- |
| 게임명 | Mine Island (2D 퍼즐) |
| 담 당  업 무 | 메인 프로그래머, 게임 로직 |
| 상 세  설 명 | KGCA 교육기관에 입학해 첫 프로젝트로 만들었던 간단한 지뢰 찾기 게임입니다. 기획 1명과 프로그래머 2명의 인원으로 진행된 이 프로젝트에서 메인을 맡아 게임의 로직을 담당하였습니다. 게임을 씬(scene)단위의 클래스로 나누어 state패턴을 이용하였습니다. 게임 내의 캐릭터 이동과 충돌은 타일 맵을 이용하였고, 맵타일의 index를 이용하여 맵을 만들었습니다. |
| 상 세  이미지 | 메인화면 게임 스테이지 선택    게임화면 true엔딩 |

**자기 소개서**

|  |  |
| --- | --- |
| 성 장  과 정 | 무언가 만드는 것을 좋아하던 아이가 게임을 접하고 게임을 만들고 싶다고 생각하는 것은 자연스러운 일이었습니다. 그렇게 게임 프로그래머를 목표로 막연히 게임을 만들고 싶다는 생각을 하며 대학을 들어와 공부했지만 더 전문적인 지식을 얻고 싶다는 생각과 혼자서는 많은 정보를 얻을 수 없다는 생각에 전문기관을 알아 보게 되었습니다. 그렇게 졸업 후 남들보다 1년을 더 공부하게 되면서 사람들과 함께 프로젝트를 진행하며 게임을 만들어 보았고 현재 3D게임을 만들고 있습니다. |
| 장 단 점  및  특기사항 | 저의 장점은 하나에 집중하여 끝까지 해내는 점입니다. 또한 그 과정에서 화를 내기보다는 긍정적으로 생각하려고 노력하기 때문에 짜증을 내거나 좌절하는 경우가 적습니다. 안 된다고 짜증을 내거나 화를 내면 주위사람들에게 피해만 줄 뿐 문제가 해결되지 않는다고 생각하며 그렇게 불평을 하기 보다는 잠시 기분 전환을 하거나 사람들과 대화, 혹은 인터넷 등을 이용하는 것이 문제해결에 도움이 된다고 믿기 때문입니다. 실제로 학원에서 처음으로 네트워크가 적용된 프로젝트(위의 MFC프로젝트)에서 짧은 시간과 적은 인원에도 client 담당과 계속적으로 대화를 나누면서 게임을 만들었습니다. 처음배우는 네트워크를 게임에 붙이고 생각도 못한 문제들이 나왔지만 짜증을 내기 보다는 팀원들과 문제를 하나하나 해결해 나갔고 결국 그 프로젝트는 학원 내부 공모전에서 1등을 하였습니다.  단점은 한가지에 몰입하면 다른 것을 보지 못하는 점입니다. 한번 한가지에 흥미를 가지면 어느 정도 그 일에 대해 만족할 만큼 알게 될 때까지 반복적으로 하게 되는데 이는 하나에 빠져 다른 것을 보지 않는 점에서 단점이 되기도 합니다. 하지만 다르게 말하면 한가지 일을 시작하면 만족할 결과가 나올 때까지 더 나은 방법을 찾고 더 좋은 결과를 보게 된다는 점에서 자신에게 긍정적인 의미로도 작용됩니다. |
| 지원동기  및  포 부 | 제 인생의 목표는 제 이름이 있다는 이유만으로 사람들이 게임을 play하는데 있어 서버 안정성에 대한 걱정을 하지 않는 것입니다.  게임 서버는 언제나 많은 게임 유저들의 타킷이 되지만 그것은 그만큼 게임 서버가 중요하다는 말이 됩니다. 제가 만든 게임 서버가 사람들에게 인정을 받기 위해서는 그 만큼의 경험과 노력이 필요할 것입니다. 함께 노력하고 많은 것을 공유하며 같이 성장 할 수 있는 팀을 만나고 싶다는 생각에 지원하게 되었습니다. 더 많은 것을 보고 노력하며, 항상 그 자리에 만족하지 않고 계속 발전해 나가는 모습을 보여드리겠습니다. |
| 상기 이력사항에 대하여 허위가 없음을 증명합니다.  2012 년 07 월 20일  작성자 **설 은 경** | |