

Aellim ~The Dominion~

팀장 선엘림

부 팀장 이은영

부 팀장 최철욱

팀원 오소라

팀원 최경래

팀원 임경길

팀원 임이랑

팀원 김동훈

팀원 이진우

팀원 이동열

Pamatto 프로젝트 기술문서
목차

1. 프로젝트 소개
2. 일정
3. 사용 기술
4. 프로그램 기능
5. 코드 구조
6. User Process
7. 시연
8. 보안





풍요로운 강이 흐르고 사시사철 상록수가 가득한 작은 왕국. 집안 대대로 이 지역을 통치해 왔지만, 선조들과 달리 여러분에게는 큰 꿈이 있습니다. 그건 바로 더 넓은 영토를 차지해 손에 닿는 모든 지역을 통치하는 것입니다. 방방곡곡 굽이쳐 흐르는 강 끝없이 펼쳐진 숲, 풍요로운 마을과 백성에 대한 열망은 날이 갈수록 커져만 갑니다. 그리고 지금이야말로 주위로 눈을 돌릴 시간입니다. 왕국 근처에는 보잘것없는 영주들이 지배하는 토지들이 산재해 있으며, 무정부 상태나 마찬가지인 이곳은 문명의 혜택을 받지 못한 자들로 가득합니다. 이제 여러분은 이들에게 문명의 이기를 전하고, 모두를 자신의 깃발 아래 하나로 모아 진정한 군주로 거듭날 것입니다!

하지만 주변 군주들의 낯빛이 심상치 않습니다. 자신의 영토를 더 넓히고자 하는 것은 비단 여러분만의 생각이 아니었던 겁니다. 다른 군주들 보다 더 넓은 영토를 차지하려면 빠른 상황판단과 잘 짜인 계획이 필요합니다. 왕국을 발전시키고 재물을 모아 모든 것을 자신의 지배 아래에 두십시오. 때로는 하수인을 고용해 상대를 방해하고, 자신의 것을 지킬 줄도 알아야 합니다. 새로운 시대의 주인이 되는 것은 모두 여러분의 손에 달렸습니다.

-도미니언 게임 설명서 中 소개란

Pamatto 프로젝트 기술문서

프로젝트 소개



<Dominion>

도미니언은 2008년 도널드 X. 바카리노(Donald X. Vaccarino)가 제작한
덱 빌딩 보드게임입니다.

덱은 플레이어가 지배하는 왕국을 보여주며 가치를 지닌 재물 카드, 영토를
나타내는 승점 카드, 그 밖의 다양한 자원 카드로 이루어져 있습니다.

게임을 처음 시작하면 모두가 동일한 영토와 같은 재물을 가지고 게임을
시작합니다. 하지만 게임을 진행하며 넓은 속주와 그 위에 세워진 다양한 건물들,
그곳에 사는 거주민들이 플레이어의 왕국을 채워나갈 것입니다.

이후 게임이 종료되면 서로의 영토를 비교하여 승점이 가장 높은 플레이어가
우승합니다!

<Aellim the Dominion>

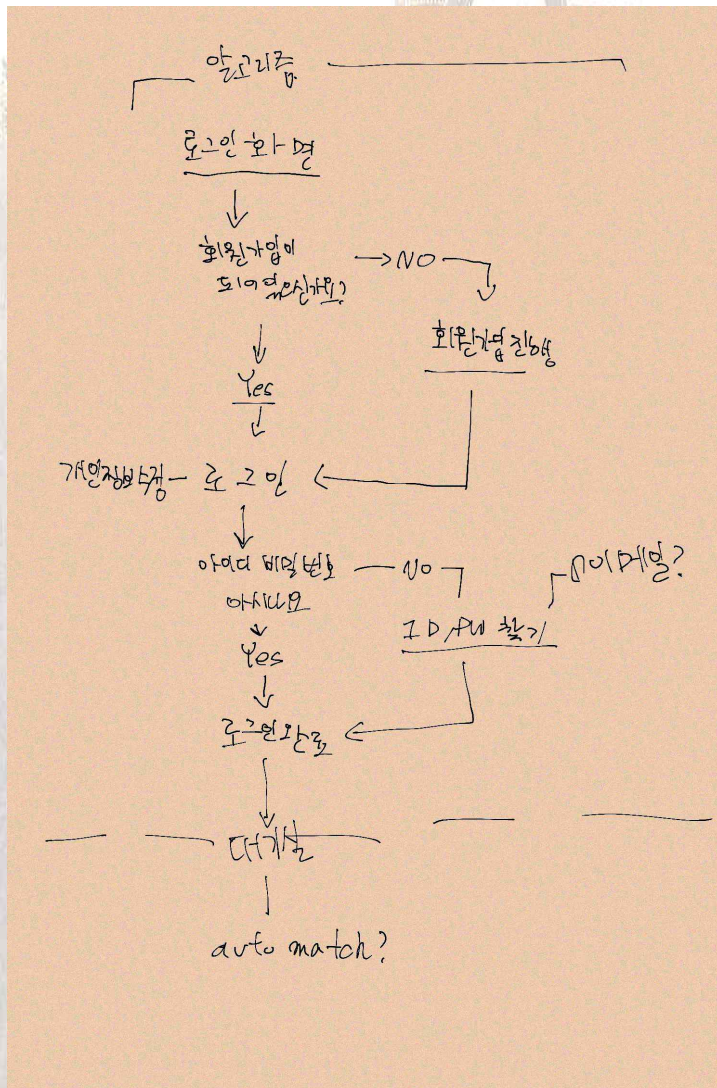
위의 보드게임을 기초로 삼아 제작한 네트워크 통신 게임입니다.

로컬 서버로 연결된 클라이언트가 해당 프로그램에 접속하여 게임을 즐길 수
있습니다. 다수의 플레이 보드를 생성할 수 있으며 각 보드 당 필요 인원수는
4명입니다. 각 보드는 비동기로 통신하며 서버 역시 다수의 클라이언트와 비동기
통신을 주고받습니다.

게임이라는 특성에 맞게 유지보수성과 확장성을 고려하여 설계하였으며 카드의
정보 확인 등 보드게임을 컴퓨터로 이식할 때 생기는 한계점들을 극복하여
설계하였습니다.

5/19

AM	<p>서기 : 임이랑</p> <p>주제 선정 및 전체 프로세스의 흐름도 (Flow Chart) 작성</p> <p>실물 게임 플레이로 게임의 패턴 이해</p> <p>카드 탐색 코드 및 클래스 정리</p>
PM	<p>각 프로세스의 구현 방식 및 난이도 논의</p> <p>서버에서 관리할 DB 테이블 및 내부 스키마 설정에 관한 논의</p> <p>승점, 액션 카드 등 게임에 사용할 카드 선정</p> <p>게임 진행 DFD 작성</p> <p>게임 카드 DB 테이블 논의</p> <p>카드 클래스 메서드와 객체 논의</p>



Pamatto 프로젝트 기술문서

일정

5/20

AM

강사님과 게임 기능에 대한 논의

공격 카드와 반응 카드 논의

게임 카드 선택의 메서드 논의

같은 종류의 카드는 한 객체 내 Count 사용 or 각 카드별 개별 객체 생성

게임 대기실 내 패널&버튼 정의

대기실 내 유저가 직접 사용하는 폼 정의

- 랭킹
- 설명서
- 매칭
- 유저 정보

PM

랭킹

- 메소드 1개
- 랭킹을 표시하는 프로시저
- 랭킹은 DB내 유저 정보와 연동하여 프로시저를 사용하여 승점 연산 후 표기

설명서

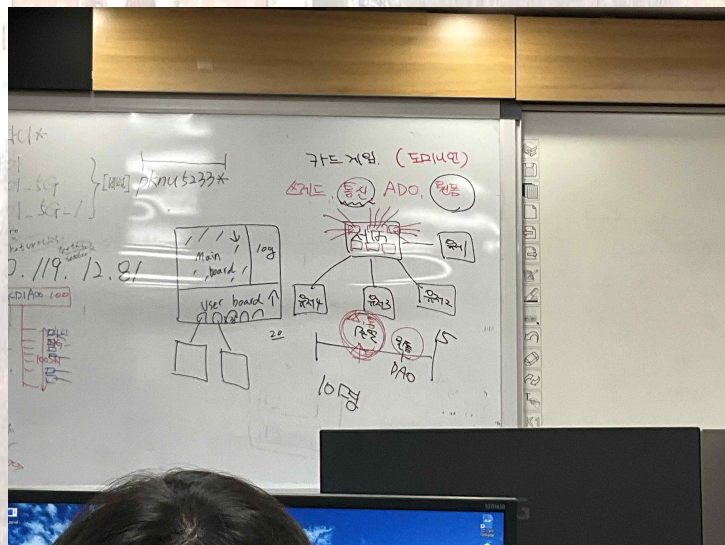
- 게임에 대한 설명을 사이트에 작성 후 하이퍼링크로 사이트와 연동
- 혹은 공식 홈페이지의 설명 페이지와 연동

유저 정보

- 닉네임 설정 가능한지 논의
- 아이디 설정 및 비밀번호 변경 메서드 논의

매칭

- 팝업 메시지 논의
- 매칭 중 취소할 수 있는 메서드 논의
- 매칭 완료 시 최종적으로 수락/거절 버튼 정의



5/21~22

All
Day

일정이 12일에서 6일로 감소

- > 이에 따라 실제로 구현할 수 있는 부분, 게임으로서 필수 기능만 추려서 선정
- > 로그인 화면 제외
- > 대기실의 간략화
- > 플레이어 인원 수 고정
- > 실행 > 매칭 > 게임 > 종료 순으로 간략화

GitHub 생성 : <https://github.com/AellimSun/pamatto.git>

서버 팀과 클라이언트 팀 분리

- 서버 팀 : 최철욱, 김동훈, 이동열, 임이랑, 이은영
- 클라이언트 팀 : 선엘림, 최경래, 임경길, 오소라, 이진우

추가 필요 기능 마스터 배분

- GitHub : 최철욱
- JSON : 오소라
- DFD : 김동훈

서버 팀

- 서버 시스템의 구체적인 기술 공유 (소켓 통신 및 스레드)
- 서버 마스터 : 이동열
- 서버 흐름도 작성
- 서버 사용 메서드와 클라이언트 사용 메서드 분리 논의

클라이언트 팀

- 필수 UI 논의 및 배치도 작성
- 통신이 필요하지 않은 클라이언트만의 필수 기능 모듈화 논의
- 게임 폼 배치 (시장, 재물, 승점, 플레이어 핸드, 덱과 무덤, 채팅 로그 위치) 논의
- 액션 카드 다중 사용 및 핸드 카드의 기술적인 배치 논의

5/23

PM

서버

- 유저 매칭 중 강제 종료 시 list에 그대로 남아있도록 코드 구현
- 4명의 유저가 모두 모이면 해당 유저들에게 수락/거절 메시지 전송
- 수락 시 : 새로운 방 생성하여 유저 정보를 넘겨줌
- 거절 시 : 매칭이 종료되고 새로 매칭 시작

클라이언트

- JSON 파일을 읽어 와서 카드 클래스 객체 생성
- 게임 관련 필요 메서드 구현 (30%)
- 게임 구동 테스트 (30%)

게임 로고 및 캐릭터 작성

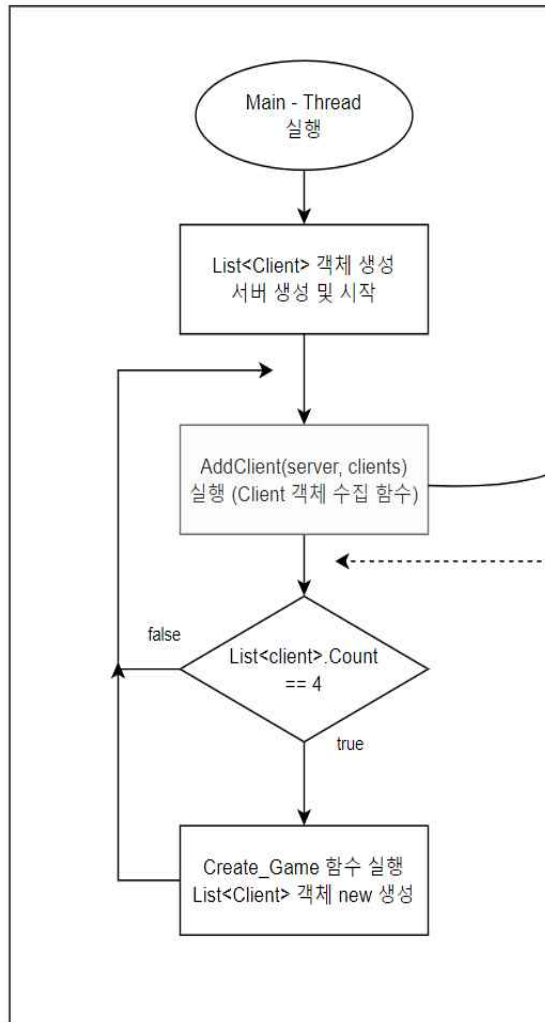


5/24

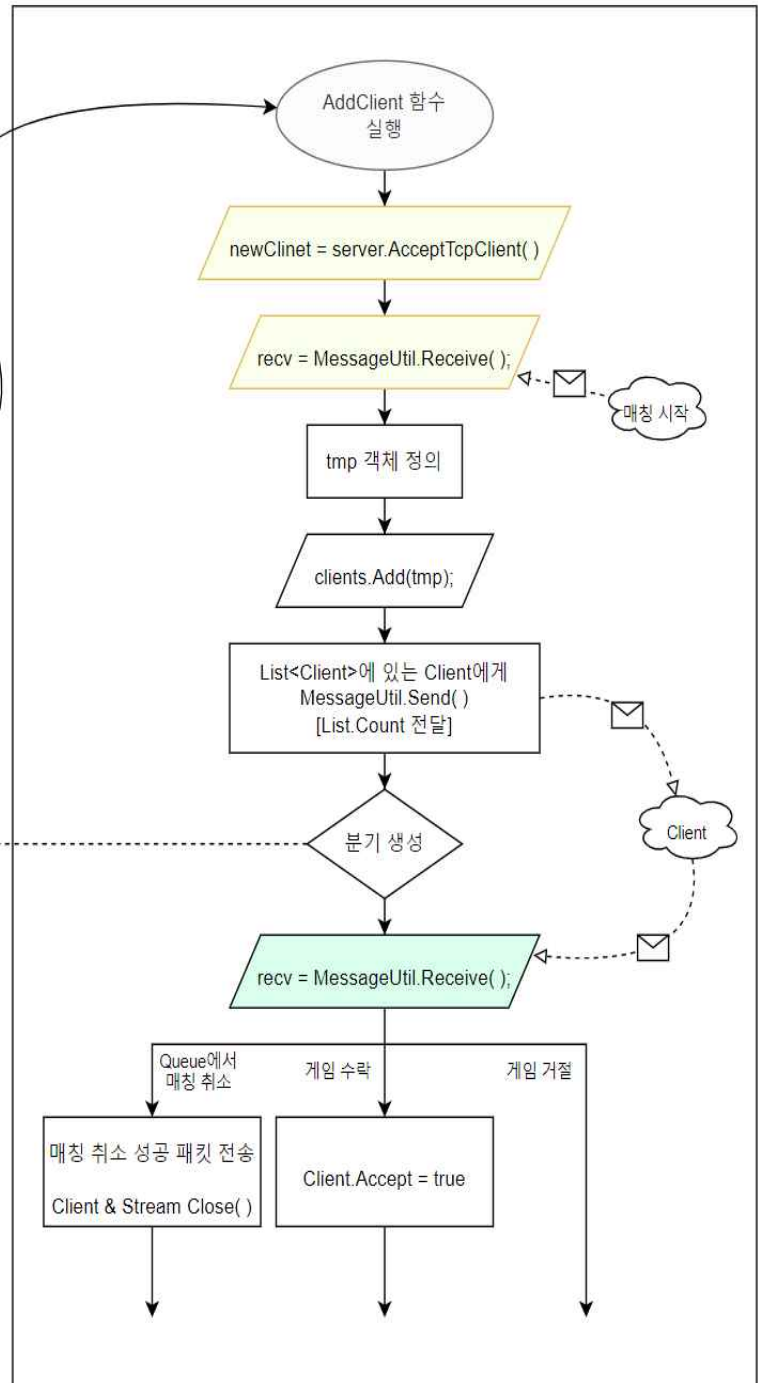
AM	<p>서버 전체 프로세스 1차 완료</p> <p>서버</p> <ul style="list-style-type: none"> - Play_Game(List<Clint>) 제작 -> 게임 내 참가하는 플레이어의 정보를 담는 메서드 작성 - API 1차 제작 - NICK 삭제 -> 아이디를 닉네임으로 사용하기로 통일 - Length 변경 <p>클라이언트</p> <ul style="list-style-type: none"> - 공격카드 확인 - 해자 유무 확인 - Switch Case 사용하여 다양한 공격에 부분적으로 코드 변경을 지원 -> 확장성 보장 - 액션카드 사용, 구매, 메서드 생성 - Game_Screen 클래스와 Game 클래스간 메소드 정리
PM	<p>서버 70% 완료</p> <p>클라이언트 30% 완료</p> <p>클라이언트</p> <ul style="list-style-type: none"> - 로그인창 -> 게임시작 전 매칭 폼 추가 - 내 액션, 구매, 재물 현황 UI 출력 메소드 생성 - 핸드 덱 영역 내 클릭이벤트 통합 메소드 생성 - 핸드 덱 잃어 가진 카드의 이미지 출력하는 메소드 생성 - AB 종료 버튼 클릭시 이벤트 메소드 생성 - 재물, 승점 카드 이미지 UI 작업 <p>UI -> 단조로운 프로그램적인 UI에서 게임적인 UI로 변경</p> <ul style="list-style-type: none"> - 담당 : 최경래 <p>서버 팀과 클라이언트 팀 회의를 통해 Log 메서드 작성</p> <ul style="list-style-type: none"> - 다른 플레이어 상황을 확인하기 위함 - 게임에 올라가는 DB와 id/Log/방번호를 string 문장 하나만으로 통신 되도록 코드 구현 - 현재 턴 플레이어는 서버와 통신 없이 자신의 행동이 Log로 남겨짐 - 자신의 턴이 아닌 플레이어는 서버와의 통신을 통해 Log를 전송받음 -> 메아리 통신 구현

Server Diagram 1

Main Thread



AddClient Thread



5/25

AM

클라이언트 명칭 어플리케이션(Application)으로 통합
어플리케이션

- 초기 세팅 (로그인, 텍 세팅, 핸드 카드 랜덤 5장) 완료
- 오류 수정
- 로그인 정보 서버 DB와 연동하여 ID 정보 전송
- 핸드 텍 클릭시 UI 이벤트 처리
- 마켓 영역 내 픽처박스 클릭이벤트 정의 및 메소드 통합

서버

- 전반적인 서버 통신 프로그램 완성
- 클라이언트 측 넘겨주는 메서드 작성
- 메서드를 짜면 그대로 API처럼 사용할 수 있도록 작성

PM

어플리케이션

- 마켓 내 액션카드 랜덤 10장 선정 후 드로우텍 셔플 작업
- 선정된 마켓- 카드 UI 출력 메서드 생성 작업
- 액션카드 구매 메소드 수정
- 재물, 승점카드 구매 메소드 생성
- 재물, 승점카드 영역 내 클릭 이벤트 정의 및 메소드 통합
- 핸드 텍에 n장을 드로우하는 메소드 생성
- 액션 예외카드(merchat, cellar) 사용 작업

Pamatto 프로젝트 기술문서

프로젝트 소개

5/26

AM

어플리케이션

- 카드 버린 후 드로우 이어질 때 핸드 덱 이미지 재정렬 작업
- 액션카드 사용즉시 무덤 덱으로 이동되도록 수정
- 핸드 덱 이미지 새로 불러오는 메소드 수정
- workshop, remodel, mine 예외 액션카드 사용 작업
- 게임 종료 여부 판단 메소드 생성
- 드로우 덱, 무덤 덱 카드 개수로 덱 이미지 및 카드 장수 출력여부 판단 작업

PM

어플리케이션

- 핸드 덱의 모든 카드를 무덤 덱으로 보내는 메소드 생성
- 내 점수 승점 계산 메소드 생성
- 게임종료시 승점 계산하여 로그 전달 작업
- 서버로 로그정보 전달 작업
- 마켓, 승점-재물, 핸드에서 카드 오른쪽 클릭시 카드 확대 출력 작업
- 플레이어 리스트 서버에서 받아와 출력
- 매칭 후 게임 시작버튼 클릭 작업

Pamatto 프로젝트 기술문서
프로젝트 소개

5/27

AM

PM

어플리케이션

- 게임 UI 폰트 및 텍스트 수정
- 턴 끝난 후 핸드 덱 제거 후 재드로우 작업
- 셔플 메소드 오류 수정
- 게임테이블 초기화 메소드 추가
- 턴 끝난 후 액션, 구매, 재물 초기화 작업
- 폐기장 기존 클래스 -> string 배열로 변경
- 카드 구매시 서버로 구매 카드 정보 전송
- 서버에서 받은 메시지 분석해주는 메서드 생성
- 마켓, 승점·재물 그룹박스 배경화면 추가

Pamatto 프로젝트 기술문서
프로젝트 소개

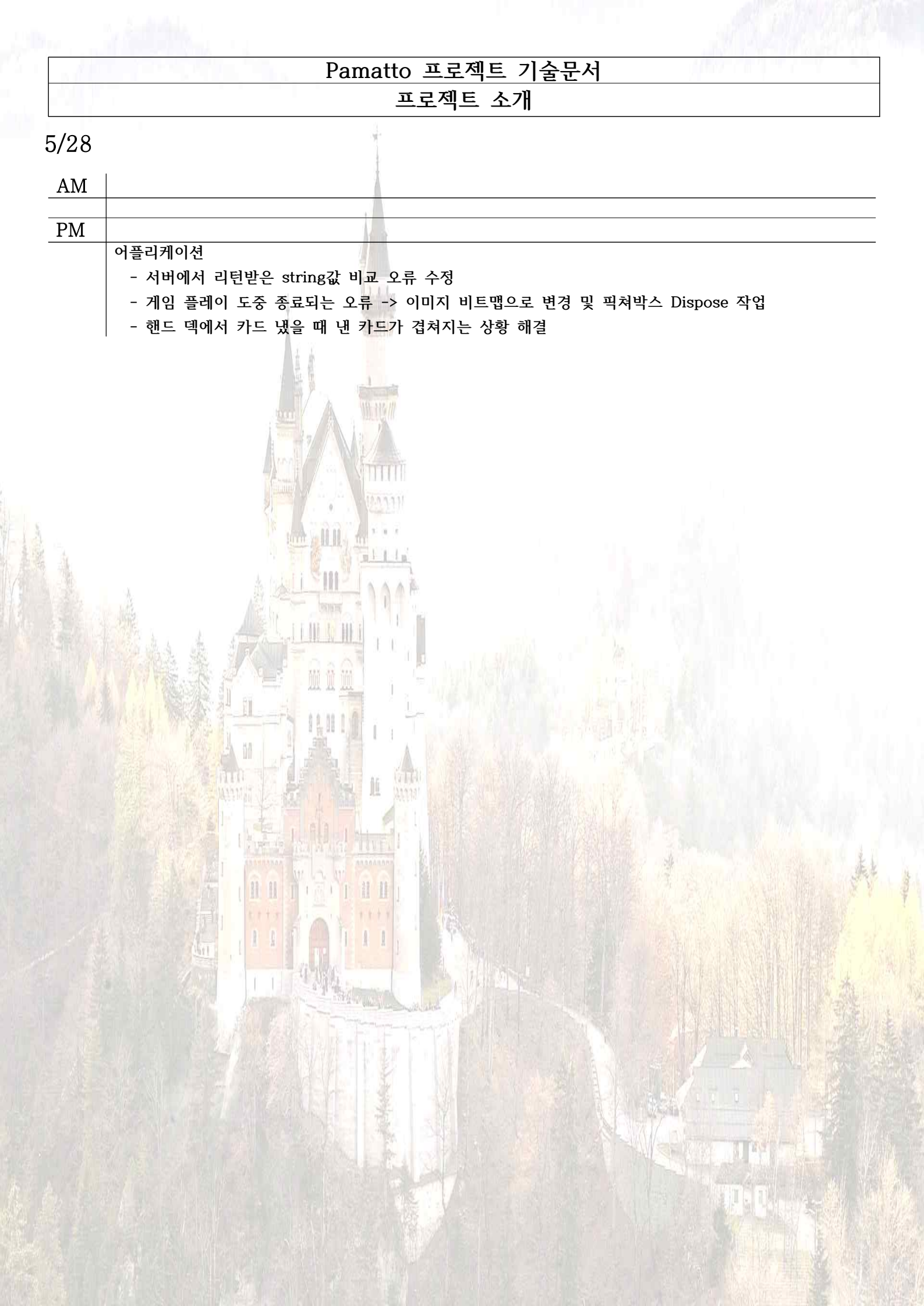
5/28

AM

PM

어플리케이션

- 서버에서 리턴받은 string값 비교 오류 수정
- 게임 플레이 도중 종료되는 오류 -> 이미지 비트맵으로 변경 및 픽처박스 Dispose 작업
- 핸드 덱에서 카드 냈을 때 낸 카드가 겹쳐지는 상황 해결



1. Tool



<Visual Studio 2019>

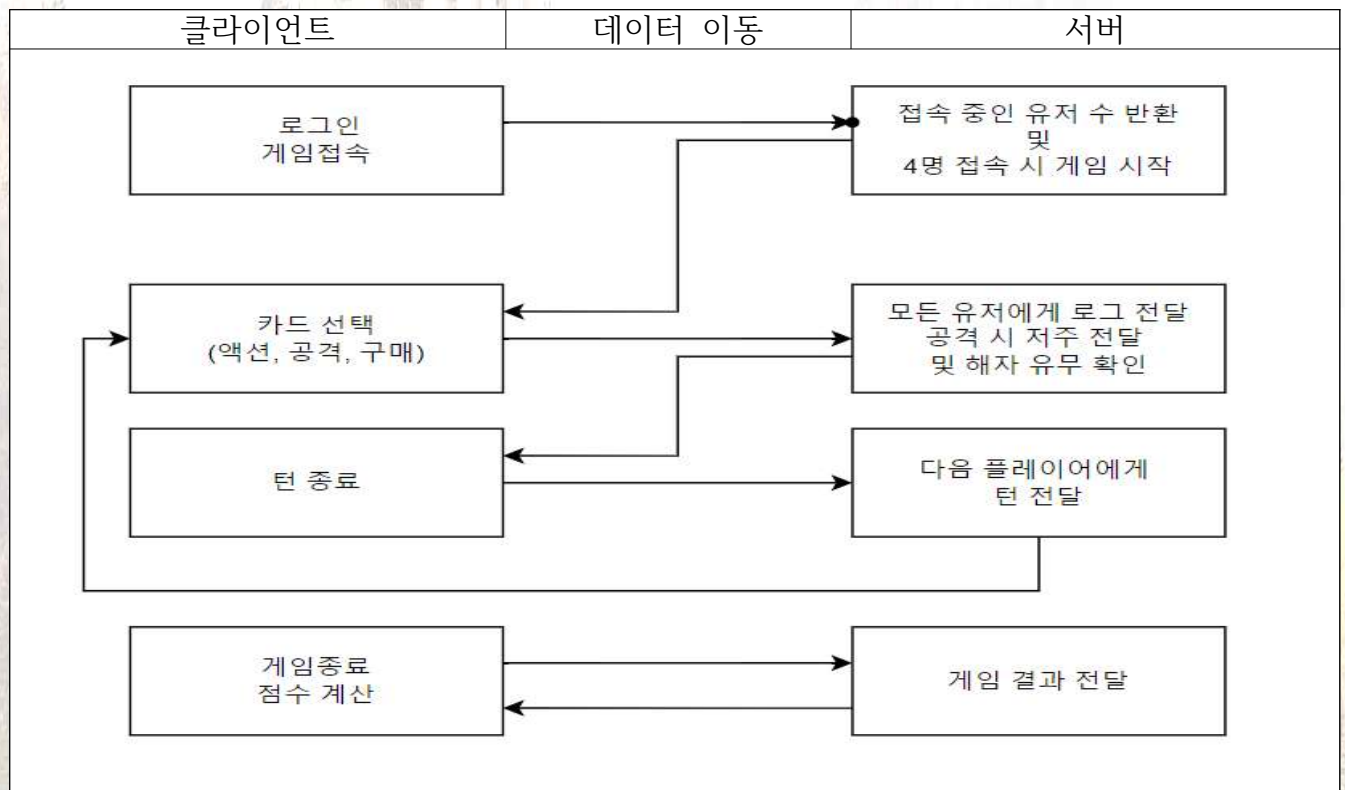


<GitHub>

2. 사용 기술

가) TCP/IP Socket 통신

클라이언트 프로그램과 서버 프로그램이 데이터를 서로 전송하기 위해 이동 경로 하나를 설정하여 실시간으로 데이터를 공유합니다. 클라이언트가 플레이어의 액션에 대한 로그 데이터를 소켓 통신을 통해 보내면 서버는 다른 클라이언트에게 전달하거나 액션에 대한 반응을 보냅니다.



*user process 이미지로 변경

나) JSON

데이터를 저장하거나 전송할 때 많이 사용되는 경량의 DATA 교환 방식으로 프로젝트의 내부에서 카드의 종류, 이름, 가격 등등의 속성들을 저장하고 읽어와서 사용하였습니다.

//JSON 코드 캡처

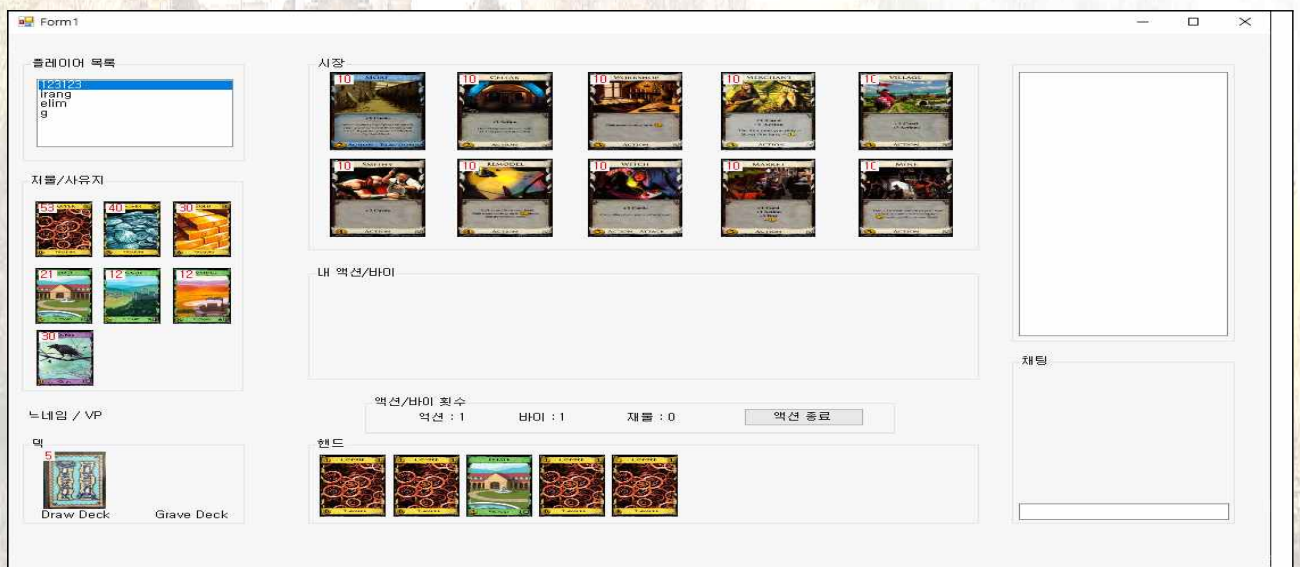
다) Thread

프로세스 내에서 실제로 작업을 수행하는 주체를 의미하며, 모든 프로세스에는 한 개 이상의 Thread가 존재하여 작업을 수행합니다. 또한, 2개 이상의 Thread를 가지는 프로세스를 multi-threaded process라고 합니다. 서버 내에서 각 플레이어 접속에 대한 대기Thread, 플레이 중인 방에 대한 Thread를 사용하고 있습니다. 클라이언트 내에서는 서버에서 오는 데이터를 대기하기 위한 Thread를 사용하고 있습니다.



라) WINFORM

Visual Studio 2019에서 지원하는 C# WINDOW FORM을 사용하여 제작하였습니다. 사용된 도구로는 카드 이미지를 나타내기 위한 PictureBox, 플레이어와 로그를 나타내기 위한 listbox 등 그 외에도 label, button, groupbox를 사용하였습니다.



Pamatto 프로젝트 기술문서

프로그램 기능

DTL

HEADER	9 Byte
MSGTYPE	4
BODYLEN	4
HASBODY	1

BODY				
Type	변수명	HEADER_MSGTYPE	BODY_LEN	BODY 내용
서버 접속	ENTER_SERVER	0x01	0	x
접속 성공	SUCCESS_SERVER	0x02	0	x
매칭 시작	START_MATCHING	0x03	20	ID(string)
큐 입장	INSERT_QUEUE	0x04	4	user_count(int)
큐 취소	CANCLE_MATCHING	0x05	20	ID(string)
매칭 취소	SUCCESS_CANCLE_MATCHING	0x06	0	x
큐 인원 추가	ADD_PLAYER	0x07	0	x
큐 인원 감소	SUB_PLAYER	0x08	0	x
큐 인원 충족	FULL_QUEUE	0x09	0	x
매칭 수락	ACCEPT	0x0A	0	x
매칭 거절	DECLINE	0x0B	0	x
게임 시작	GAME_START	0x0C	80	ID1(string), ID2(string), ID3(string), ID4(string), Host_Number(int)
게임 취소	GAME_CANCLE	0x0D	0	x
액션	ALERT_ACTION	0x0E	25	TYPE(int), CARD(int)
카드 얻기	GET_CARD	0x01	25	ACTION(int), CARD(int)
카드 폐기	SCRAP_CARD	0x02	25	ACTION(int), CARD(int)
공격	ATTACK	0x03	25	ACTION(int), CARD(int)
로그 전송	LOG_SEND	0x0F	51	LOG(string)
턴 종료	TURN_END	0x10	0	x
턴 넘김	TURN_SEND	0x11	0	x
게임 끝	GAME_FIN	0x12	0	x
점수 요청	SCORE_REQUEST	0x13	0	x
점수 전송	SCORE_SEND	0x14	4	SCORE(int)
점수 총 점	TOTAL_SCORE_SEND	0x15	16	SCORE1(int), SCORE2(int), SCORE3(int), SCORE4(int)

Application												
Class	Member Variable		Method	Parameter		return type	function	Detail	etc			
	type	name		type	name							
Card	int	price										
	int	amount										
	string	Name										
	string	kind										
ActionCard (Card 상속)	int	add_Draw	ActionCard	string	key n		생성자	card.json 파일에서 받아온 키값을 변수명으로 하는 변수에 제이슨의값내용대입하여액션카드객체생성				
	int	goto_Grave										
	int	goto_Trash										
	int	add_Action										
	int	add_Buy		Jtoken	jtoken							
	int	add_Money										
	int	gain_to_Deck										
	int	gain_to_Hand										
	bool	attach										
	bool	protect										
MoneyCard (Card 상속)	int	money	MoneyCard	string	key		생성자	card.json 파일에서 받아온 키값을 변수명으로 하는 변수에 제이슨의값내용대입하여재물카드객체생성				
				Jtoken	jtoken							
EstateCard (Card 상속)	int	score	EstateCard	string	key		생성자	card.json 파일에서 받아온 키값을 변수명으로 하는 변수에 제이슨의값내용대입하여점수카드객체생성				
				Jtoken	jtoken							

Deck	List<Card>	HandDeck	Deck	List<Card>	estimated	생성자	HandDeck, DrawDeck, GraveDeck 리스트를 생성하며 기본덱(동7,사유지3)을배부하여객체를생성하고, DrawDeck을섞어(Shuffle())다섯장드로우(DrawToHand())	
				List<Card>	monelist			
				Game_Screen	g			
	List<Card>	>DrawDeck	Hand_To_Grave	Hand_To_Grave		HandDeck의 카드를 GraveDeck으로 모두 이동	HandDeck을 돌며 카드 한장한장을 GraveDeck에 추가시킨 후 HandDeck을비움	
			Shuffle	List<Card>	obj	GraveDeck의 카드를 HandDeck으로 랜덤순서로 섞어 이동	GraveDeck을 돌며 0 ~ GraveDeck내의 객체수 사이의 난수를 발생시켜 HandDeck에 추가하고, 추가한 객체는 GraveDeck 내에서 제거	
	List<Card>	GraveDeck		Show DrawDeck		DrawDeck이 미지 출력해야할지 여부 리턴	DrawDeck의 카운트가 0이면 false, 0이 아닐 경우 true 리턴	
				Show GraveDeck		GraveDeck이 미지 출력해야할지 여부 리턴	GraveDeck의 카운트가 0이면 false, 0이 아닐 경우 true 리턴	
			DrawToHand	int	i	HandDeck에 카드 추가	매개변수 i의 수만큼 DrawDeck에서 HandDeck으로 카드객체 이동시킴 DrawDeck이비었을경우Suffle()메소드호출	
				Game_Screen	g			
	List<string>	trash	GoToGrave	int	number	HandDeck의 카드를 GraveDeck으로 이동	HandDeck 내 매개변수 num번째 객체를 GraveDeck으로 보내고 HandDeck에서 제거, 서버로 로그를 전송하고 ui에도 출력하는 MakeString()메소드 호출 매개변수mode가"u"일경우공격카드인지를체크하여서서버로전송	
				string	mode			
	Rand	rand		Ga	g			

				me_Scr een					
	om	om	BuyC ard	Car d	car d		Grave Deck에 카드를 추가	매개변수로 받은 카드를 GraveDeck에 추가	
			gainC ardT oHan d	Car d	car d		HandD eck에 카드를 추가	매개변수로 받은 카드를 HandDeck에 추가	
			goto Tras h	stri ng	car d		trash 에 카드명 을 추가	매개변수로 받은 문자열을 trash(폐기장 string 리스트)에 추가	
Game	int	score	Game	Game_Scr een	form		생성자	매개변수로 받은 form을 멤버변수 form에 할당 market,deck,gameTable초기화 마켓ui에이미지를입히는메소드호출(mar ketImgInit()) 핸드ui에이미지를입히는메소드호출(setH andDeckImg) abc영역ui의수치를수정해주는메소드호 출(changeABC())	
				stri ng	now			1. 점수카드의 경우 : 에러메세지 출력 2.재물카드의경우 1) 클릭한카드가 상인카드이고, silverUsed의 값이 0일경우 silverUsed 값을 ++시키고, gameTable의 Coin값을 ++ 2) 매개변수 now의 값이 Action End인경우 에러메세지 출력 3) 매개변수 now의 값이 Action End가 아닌 경우 클릭한 카드를 낸 카드 그룹박스에 올리는 메소드 호출(pictureBox_SetImg()) pictureBox_SetImg()의 리턴값이 true일 경우 1] gameTable의 Coin값을 기존값과 클릭한 재물카드의 money값을 더한 값으로 세팅 2] 클릭한 재물카드를 grave로 보내는 메서드 호출(GoToGrave()) 3] abc영역 ui의 수치를 수정해주는 메서드 호출(changeABC()) 4] 드로우덱과 무덤덱의 이미지 노출여부, 남은카드수 출력하는 메서드 호출(pictureBoxTF()) 5] 핸드 ui에 이미지를 입히는 메소드 호출(setHandDeckImg)	
			click Hand				핸드덱 의 카드를 클릭했 을때의 처리	3.액션카드의경우 1) 매개변수 now의 값이 Action End일 경우 클릭한 카드를 낸 카드 그룹박스에 올리는 메소드 호출(pictureBox_SetImg()) pictureBox_SetImg()의 리턴값이 true일 경우 1] gameTable의 ActionNumber값을 1 줄임	
				int	idx				

						2) 클릭한 재물카드를 grave로 보내는 메서드 호출(GoToGrave()) 3) 액션 카드의 효과를 사용하는 메서드 호출(useCard()) 4) abc영역 ui의 수치를 수정해주는 메서드 호출(changeABC())		
						5) 드로우덱과 무덤덱의 이미지 노출여부, 남은카드수 출력하는 메서드 호출(pictureBoxTF()) 6) 핸드 ui에 이미지를 입히는 메소드 호출(setHandDeckImg) game테이블의 ActionNumber의 값이 0보다 작고, 사용한 카드의 이름이 workshop, remodel, mine이 아닐 경우 -> button1의 텍스트를 변경하는 메서드 호출(turn_button1()) 2) 매개변수 now의 값이 Action End가 아닌 경우 : 에러메시지 출력		
			useCard	ActionCard	card	매개변수 card의 add_Action, add_Buy, add_Money의 값이 0이 아닐 경우 gameTable의ActionNumber,BuyNumber,Coin값을증가시키기 매개변수card의add_Draw값이0이아닐 경우DrawToHand()메서드호출 예외사항이있는카드들일경우 1)merchant카드:merchantUsed값을true로변경 2)cellar카드:form의clickMode를grave로변경,button1의텍스트를변경하는메서드호출(turn_button1()) 3)worckshop카드: form의clickMode를 actionEffectMode로 변경, button1의 텍스트를 변경하는 메서드 호출(turn_button1()), gameTable의 coin값을 4로 설정 4)remode카드:form의clickMode를trash로변경,button1의텍스트를변경하는메서드호출(turn_button1()) 5)mine카드:form의clickMode를moneyTrash로변경,button1의텍스트를변경하는메서드호출(turn_button1()) abc영역ui의수치를수정해주는메소드호출(changeABC())	클릭한 액션카드의 효과 실행시키기	
			buyCard	int	i	Card	gameTable의 Coin값이 cardList의 매개변수 i번째 카드의 값보다 같거나 높고, gameTable의BuyNumber값이1보다같거나클 경우 1) gameTable의 Coin값을 cardList의 매개변수 i번째 카드의 값만큼 줄이기 2) gameTable의 BuyNumber을 1 줄이기 3) 마켓리스트에서 카드를 판매하는 메서드 호출(SellCard()) 4) grave에 카드를 추가시키는 메서드 호출(BuyCard()) 5) 드로우덱과 무덤덱의 이미지 노출여부, 남은카드수 출력하는 메서드 호출(pictureBoxTF()) 6) abc영역 ui의 수치를 수정해주는 메소드 호출(changeABC())	Grave Deck에 액션카드를 추가

클릭한
액션카
드의
효과
실행시
키기

Grave
Deck에
액션카
드를
추가

					<p>7) 서버로 로그를 전송하고 ui에도 출력하는 MakeString()메소드 호출</p> <p>8) 얻은 카드의 정보를 서버로 전송하는 메서드 호출(Get_Card())</p> <p>cardList의매개변수i번째카드리턴</p>	
	notBuyCard	int	i	Card	<p>Grave Deck에 액션 효과로 액션카드를 추가</p> <p>gameTable의 Coin값이 cardList의 매개변수 i번째 카드의 값과 같거나 높은 경우</p> <p>1) gameTable의 Coin값을 0으로 변경</p> <p>2) 마켓리스트에서 카드를 판매하는 메서드 호출(SellCard())</p> <p>3) grave에 카드를 추가시키는 메서드 호출(BuyCard())</p> <p>4) 드로우덱과 무덤덱의 이미지 노출여부, 남은카드수 출력하는 메서드 호출 (pictureBoxTF())</p> <p>5) abc영역 ui의 수치를 수정해주는 메소드 호출(changeABC())</p> <p>6) form의 clickMode를 market으로 변경</p> <p>7) gameTable의 ActionAmount값이 0일때와 아닐때를 판별해 button1의 텍스트를 변경하는 메서드 호출(turn_button1())</p> <p>cardList의매개변수i번째카드리턴</p>	
	buyCard	int	i	Card	<p>매개변수 i가 3보다 작을 경우 재물카드리스트, 아닐 경우 점수카드 리스트 사용</p> <p>gameTable의Coin값이cardList의매개변수i번째카드의값보다같거나높고, gameTable의BuyNumber값이1보다같거나클 경우</p> <p>1) gameTable의 Coin값을 cardList의 매개변수 i번째 카드의 값만큼 줄이기</p> <p>2) gameTable의 BuyNumber을 1 줄이기</p> <p>3) 마켓리스트에서 카드를 판매하는 메서드 호출(SellCard())</p> <p>4) grave에 카드를 추가시키는 메서드 호출(BuyCard())</p> <p>5) 드로우덱과 무덤덱의 이미지 노출여부, 남은카드수 출력하는 메서드 호출 (pictureBoxTF())</p> <p>6) abc영역 ui의 수치를 수정해주는 메소드 호출(changeABC())</p> <p>7) 서버로 로그를 전송하고 ui에도 출력하는 MakeString()메소드 호출</p> <p>선택된카드리스트의i번째카드리턴</p>	
	notBuyCard	int	i	Card	<p>Grave Deck에 액션 효과로 재물or 점수카드를 추가</p> <p>매개변수 i가 3보다 작을 경우 재물카드리스트, 아닐 경우 점수카드 리스트 사용</p> <p>gameTable의Coin값이cardList의매개변수i번째카드의값보다같거나높을 경우</p> <p>1) gameTable의 Coin값을 0으로 변경</p> <p>2) 마켓리스트에서 카드를 판매하는 메서드 호출(SellCard())</p> <p>3) grave에 카드를 추가시키는 메서드 호출(BuyCard())</p> <p>4) 드로우덱과 무덤덱의 이미지</p>	

Game_Screen	Game	game					노출여부, 남은카드수 출력하는 메서드 호출 (pictureBoxTF()) 5) abc영역 ui의 수치를 수정해주는 메서드 호출(changeABC()) 6) form의 clickMode를 market으로 변경 7) gameTable의 ActionAmount값이 0일때와 아닐때를 판별해 button1의 텍스트를 변경하는 메서드 호출(turn_button1()) 선택된카드리스트의i번째카드리턴	
			gainCurse			Card	Grave Deck에 curse카드를 추가 마켓리스트에서 curse카드를 판매하는 메서드 호출(SellCard()) grave에curse카드를추가시키는메서드호출(BuyCard()) 드로우덱과무덤덱의이미지노출여부,남은카드수출력하는메서드호출 (pictureBoxTF()) abc영역ui의수치를수정해주는메소드호출(changeABC()) curse카드리턴	
			gainCardToHand	int	i	Card	HandDeck에 재물or점수카드를 추가 매개변수 i가 3보다 작을 경우 재물카드리스트, 아닐 경우 점수카드리스트 사용 마켓리스트에서카드를판매하는메서드호출(SellCard()) handDeck에카드를추가시키는메서드호출(gainCardToHand()) 선택된카드리스트의i번째카드리턴	
	Image_Class	images	Game_Screen				생성자 Form의 컴포넌트들을 초기화하는 메서드 호출(InitializeComponents())	
			getLower			PictureBox[]	lower 리스트 getter lower 픽처박스 리스트 리턴	
			Form1_Load	object	sender		form의 컴포넌트들로드 서버와의 통신상태를 시작하는 Listen_Method() 호출 form내에TypographerGotischB-Bold.ttf폰트를적용 카드재고label의위치를설정 upper,lower,marketAmt,marketPics,CSPics,CSAmt배열초기화 game객체생성 market,deck객체초기화 listBox1에플레이어리스트출력 재물,점수카드이미지,재고출력 드로우덱과무덤덱의이미지노출여부,남은카드수출력하는메서드호출 (pictureBoxTF()) button1의텍스트를ActionEnd로설정	
				EventArgs	e			
	DB_ACCESS	dB	changeABC				드로우덱과 무덤덱의 이미지 출력 DrawDeck의 카드 유무 판단하는 메서드 호출하여 드로우덱 이미지 출력여부 판단 GraveDeck의카드유무판단하는메서드호출하여그레이브덱이미지출력여부판단 GraveDeck의카드유무판단하는메서드호출하여그레이브덱재고출력여부판단 드로우덱과그레이브덱의재고변경	
				GameTable	gameTable		남은액션, 구매, 재물출력 매개변수 gameTable의 ActionNumber, BuyNumber, Coin값을 출력	

			setHandDeckImg	Deck	deck		핸드덱의 이미지 재정렬 or 초기 세팅	핸드덱의 모든 이미지 visible, enabled값 false HandDeck리스트에 존재하는 카드 수만큼 돌려 이미지 출력, visible, enabled값 true	
			printMessageBox	string	content		MessageBox 출력	매개변수 string의 내용을 MessageBox로 출력	
			marketClick	object	sender		마켓카드 영역 이미지 클릭이벤트 처리	1. 왼쪽 클릭시 1) clickmode의 값이 market일 경우 선택한 카드 구매 메서드 호출(buyCard()) 후 재고 줄여 출력 2) clickmode의 값이 grave, trash, moneyTrash일 경우 : 에러메시지 출력 3) clickmode의 값이 actionEffectMode일 경우 선택한 카드 얻는 메서드 호출(notBuyCard()) 후 재고 줄여 출력 2. 오른쪽 클릭시: 클릭한 카드 확대하여 보여주는 메서드 호출(rightclick())	
	Market	market	CSClick	object	sender		재물, 점수카드 영역 이미지 클릭이벤트 처리	1. 왼쪽 클릭시 1) clickmode의 값이 market일 경우 선택한 카드 구매 메서드 호출(buyCSCard()) 후 재고 줄여 출력 2) clickmode의 값이 grave, trash, moneyTrash일 경우 : 에러메시지 출력 3) clickmode의 값이 actionEffectMode일 경우 선택한 카드 얻는 메서드 호출(notBuyCSCard()) 후 재고 줄여 출력 2. 오른쪽 클릭시: 클릭한 카드 확대하여 보여주는 메서드 호출(rightclick())	
			pictureBox_SetImg	int	idx	bool	핸드에서 클릭한 카드 제출	제출된 카드의 픽처박스 리스트를 돌려 제출할 자리를 정하기 제출할 자리의 칸의 visible, enabled값 true 선택된 핸드칸의 이미지가 비어있을 경우 1. 선택한 카드의 이미지를 제출할 자리에 보이게 하고 선택된 카드의 이미지를 해제 2. true 리턴 선택된 핸드칸의 이미지가 비어있을 경우: false 리턴	
			Sum_Score	CardItems		int	점수 더하여 리턴	1. 매개변수 item의 카드가 duchy일 경우 : 매개변수 s + 3 2. 매개변수 item의 카드가 estate일 경우 : 매개변수 s+1 3. 매개변수 item의 카드가 province일 경우 : 매개변수 s+6 2. 매개변수 item의 카드가 curse일 경우 : 매개변수 s-1 s리턴	
	Deck	deck	button1_Click	object	sender		button1 클릭이벤트 처리	1. button1의 텍스트값이 Action End일 경우 button1의 텍스트값을 Buy End로 변경 gameTable의 ActionNuber값을 0으로 변경 abc영역 ui의 수치를 수정해주는 메소드 호출(changeABC()) 2. button1의 텍스트값이 ThrowEnd일 경우	
		uppe							

	Pictu reBo x[]	r					<p>1)핸드에서 선택한 카드 리스트(selected)가 비어있지 않을 경우 selected를 돌려 선택된 카드들을 무덤덱으로 이동(GoToGrave()) 핸드덱 이미지 재정렬하는 메서드 호출(setHandDeckImg()) selected 내의 카드 수 만큼 드로우(DrawToHand()) selected 비우기 clickMode값 market으로 변경 button1 텍스트값 Action End로 변경 3.button1의텍스트값이EffectEnd혹은S crapEnd일경우 clickMode의 값 market으로 변경 gameTable의 Coin 값을 0으로 변경 abc영역 ui의 수치를 수정해주는 메소드 호출(changeABC()) gameTable의 ActionNumber값이 0일경우 button1의 텍스트값을 Buy End로, 아닐 경우 Action End로 변경 4.button1의텍스트값이BuyEnd일경우 1) 게임종료가 아닐 경우 턴 엔드 서버로 전송</p>	
	Pictu reBo x[]	lowe r					<p>button1 텍스트값 Action End로 변경, enabled 값 false 핸드덱 비우고 5장 새로 드로우(Clear(), DrawToHand()) 낸 카드들의 픽처박스 지우고 visible, enabled 값 false 서버와의 통신상태를 시작하는 Listen_Method() 호출 gameTable 초기화시키는 메서드 호출(initGameTable()) abc영역 ui의 수치를 수정해주는 메소드 호출(changeABC()) 2) 게임종료일 경우 : 서버로 내 점수 전달</p>	
			My_S core		int	점수 집계	<p>핸드덱의 모든 이미지 지우기 핸드덱모두무덤덱으로보내기(Hand_To_ Grave()) 무덤덱,드로우덱에서승점계산(Sum_Sco re()) 집계된점수값리턴</p>	
	Pictu reBo x[]	mar ketPi cs	Go_t o_Ma in_Fo rm			모든 유저 점수 집계하 여 결과창 이동	<p>서버에서 모든 유저의 점수 받아오기 모든유저를얻은점수가많은순으로정렬 결과창(Form7)객체생성하여창띄우기</p>	
			Finis h_Ga me			내 점수 집계하 여 서버로 보내기	<p>내 점수 집계 메서드 호출(My_Score()) 서버로내점수전송(Game_End()) 모든유저점수집계하는함수호출(Go_to_M ain_Form())</p>	
			turn_ butto n1	stri ng	cont ent	button 1의 텍스트 값 변경	<p>매개변수 content값으로 button1의 텍스트값 변경</p>	
			hand Click	obje ct	sen der	핸드 덱 내 이미지	<p>1. 왼쪽 클릭일 경우 1) clickMode가 market일 경우 카드의 종류에 따라 처리하는</p>	

	Pictu reBo x[]	CSPi cs						클릭이 벤트 처리	메서드 호출(clickHand() 2) clickMode가 market일 경우 핸드에서 선택된 카드 리스트(selected)에 추가 선택된 카드 visible, enabled 값 false 3) clickMode가 actionEffectMode일 경우 : 에러메세지 출력 4) clickMode가 trash일 경우 gameTable의 Coin값을 선택한카드의 price + 2로 설정 abc영역 ui의 수치를 수정해주는 메소드 호출(changeABC()) 선택한 카드 폐기하는 메서드 호출(gotoTrash()) HandDeck에서 선택한 카드 삭제 핸드 ui에 이미지를 입히는 메소드 호출(setHandDeckImg) clickMode를 actionEffectMode로 변경 button1의 텍스트값을 Effect End로 변경 5) clickMode가 moneyTrash일 경우 선택한 카드가 copper, silver가 아닐 경우 에러메세지 출력 선택한 카드 폐기하는 메서드 호출(gotoTrash())				
	Labe l[]	mar ketA mt	Eve ntA rgs	e					HandDeck에서 선택한 카드 삭제 선택한 카드 얻는 메서드 호출(notBuyCSCSCard()) 후 재고 줄여 출력 핸드 ui에 이미지를 입히는 메소드 호출(setHandDeckImg) clickMode를 actionEffectMode로 변경 gameTable의 ActionNumber값이 0일경우 button1의 텍스트값을 Buy End로, 아닐 경우 Action End로 설정 2.오른쪽클릭시:클릭한카드확대하여보여 주는메서드호출(rightclick())				
								mark etImg Init	List <Ca rd>	mar ketli st	마켓 이미지 초기화	매개변수 marketlist를 돌며 저장된 카드의 이미지를 출력	
								right click	Pict ure Box	sen der	카드 오른쪽 클릭 이벤트	선택한 이미지를 확대하여 보여줄 폼(Form6) 객체 생성 후 띄우기	
	Labe l[]	CSA mt	setLo gBox	stri ng	mes sag e	로그메 세지 출력	매개변수 message가 null이 아닐 경우 list_log에 로그 한줄을 추가하고 마지막줄로포인터를이동						
			Log_ Hand le	stri ng	mak e	로그 서버로 전송 및 출력	매개변수 make를 서버로 전송(Log_Send()) 서버에서받은값을make에저장 로그메세지출력메서드호출(setLogBox())						
			Make Strin g	stri ng	car dna me	액션카 드 로그 메시지 생성	매개변수 cardaction의 값에 따라 로그메세지를 조합 로그를서버로전송및출력하는메서드호출(Log_Handle())						
	stri ng	car dac tion											

			Make String	int	dCount		카드 드로우 로그 메시지 생성	매개변수 dCount를 사용해 로그메세지 생성 로그를서버로전송및출력하는메서드호출(Log_Handle())	
			Make String				카드 버림 로그 메시지 생성	핸드 덱에서 선택된 카드 리스트(selected) 내의 카드 수를 사용해 로그메세지 생성 로그를서버로전송및출력하는메서드호출(Log_Handle())	
	string	click Mod e						1. 서버에서 받아온 값이 1일 경우(내 턴) button1의 enabled값 true 내 턴을 표시하는 로그메세지 출력(setLogBox()) 2.서버에서받아온값이2일경우(공격받음) 1) 핸드덱 내에 moat카드가 없을 경우 서버로 curse카드 얻었음을 전송(MakeString()) GraveDeck에 curse카드 추가(gainCurse()) curse카드의 재고 ui 수정 2) 핸드덱 내에 moat카드가 있을 경우 서버로 moat 카드가 있음을 전송(MakeString()) 3.서버에서받아온값이3일경우(상대가가 카드를먹음) 상대가 먹은 카드의 재고를 줄이고 ui 반영 4.서버에서받아온값이4일경우(상대가가 카드를폐기함) trash에 받아온 카드의 이름을 추가(gotoTrash()) 5.서버에서받아온값이5일경우(상대방에게서로그받음) 로그메세지 출력 메서드 호출(setLogBox()) 6.서버에서받아온값이6일경우(상대방이 게임종료시킴) 내 점수 집계 메서드 호출(My_Score()) 서버로 내 점수 전송(Score_send()) 모든 유저의 점수 집계하여 결과창 띄우기(Go_to_Main_Form())	
			Listen_Me thod				서버와 통신하여 받아온 값을 처리		
	List<int>	selected							
			Attack_Receive				공격을 전한 유저가 사용하는 메서드	턴을 진행 중인 유저를 제외한 나머지 인원 3인은 Listen_Method를 통해 턴 진행자의 상황을 계속해서 수신 받는다. 하지만, 턴을 진행 중인 유저가 타인에게 공격을 할 경우 자신도 타인에 대한 응답을 수신 받아야 한다. 그때 공격 유저는 해당 메서드를 실행하게 되며, Game_Listener 메서드의 리턴 결과에 따라 다음과 같은 행위를 진행한다. case 1 : Attack_Recive 종료 case 3 : 상대가 먹은 카드의 ui 반영 case 5 : 로그메세지 출력 메서드 호출(setLogBox())	

			Game_Scren_FormClosed	object EventArgs	sender		게임창 닫힘	부모창(Form2)에 결과값 OK 전송	
Game Table	int	ActionNumber	GameTable				생성자	ActionNumber와 BuyNumber의 값을 1로, Coin의 값을 0으로 설정	
	int	BuyNumber	initGameTable				게임테이블 초기화	ActionNumber와 BuyNumber의 값을 1로, Coin의 값을 0으로 설정	
	int	Coin							
Market	List<Coin>	MarketPile	Market				생성자	card.json 파일 읽어들이는 메소드 호출(getJson()) action카드객체생성하여MarketPile에 추가x10 money카드객체생성하여MoneyPile에 추가x3 estate카드객체생성하여estatePile에추가x4	
			getMarketList			List<Card>	MarketPile getter	MarketPile 리스트 리턴	
			sellCard	Card	card		카드객체의 amount 줄임	매개변수 card의 amount값을 1 줄임	
	List<Card>	MoneyPile	ShuffleDraw				MarketPile에서 카드 10장 뽑아정렬	MarketPile을 돌며 0 ~ MarketPile내의 객체수 사이의 난수를 발생시켜 새 리스트에 추가하고, 추가한 객체는 MarketPile 내에서 제거 새리스트를MarketPile로덮어씌우기	
	List<Card>	estatePile	GameOver			bool	게임종료여부 판단	province카드가 모두 팔렸다면 게임 종료 true 리턴 MarketPile,MoneyPile,estatePile내에서 3종류이상의카드가모두팔렸다면게임종료 true리턴	
Image_Classes	Bitmap	게임상에서 필요한 이미지	Image_Classes				생성자	게임상에서 필요한 모든 이미지들을 Directory에서 불러온 뒤 Bitmap으로 정의한다.	
			getBitmap	string	name	Bitmap	필요한 이미지 리턴	이미지 정보를 이름으로 매개변수로 받아와 그에 맞는 Bitmap 이미지를 리턴한다.	

Client Method

Class	Membervalue		Method	Function	Parameter		return	Detail	ETC
	Variable type	Variable name			Variable type	Variable name			
public class TransHandler	IPEndPoint	ServerAddress	TransHandler	생성자	string	Server_IP		서버IP, 서버Port, 유저 ID를 입력하여 객체를 생성	
	IPEndPoint	ClientAddress			int	serverPort			
	TcpClient	Client			string	ID			

	Network Stream	Stream	disconn	Stream, Client 종료				Stream 및 Client를 Close()	
	string	ID	Start_Matching	Server 연결, Queue 인원 리턴				ID(11byte) 정보를 Body로 하여 Message 전송 후 현재 큐에 존재하는 인원 수를 리턴 받는다	
			Wait_Full_Queue	Message 분석	Form3	f	int	1. ADD_PLAYER : Form3에 인원 수 증가 2. SUB_PLAYER : Form3에 인원 수 감소 3. 매칭 완료 후 게임 시작 거부 : -1 리턴 4. 매칭 완료 후 게임 시작 수락 : 1 리턴	
			Cancle_Matching	매칭 취소			int	현재 유저의 ID(11byte) 정보를 Body로 하여 취소 Message 전송	async
			Respond	Res값에 따라 Message 전송	int	Res	int	Res = 1 수락 Message 전송 Res = -1 거절 Message 전송	
					string[]	ID_List		모든 유저가 전송을 완료하면 서버로부터 Message를 받아서 분석	
					int	Host_Number		GAME_START ID_LIST[]를 정의 후 1 리턴 GAME_CANCEL -1 리턴	
			Game_Listener	자신의 차례가 아닐 때 실행	ref string	Card_Name	int	요청받은 분석에 맞는값을 return하여 Application으로 전송 TURN_SEND : 내 차례일 경우 1 반환 ALERT_ACTION : ATTACK : 액션을 사용한 경우, 2 반환 GET_CARD: 카드를 얻은 경우, 3 반환 SCRAP_CARD: 카드를 버린 경우, 4 반환 LOG_SEND : LOG기록을 전달한 경우 5 반환 SCORE_REQUEST : 점수 요청이 왔을 경우 6 반환	
					ref string	Log			
			Attack	공격 Message 전송	string	Card_Name		byte로 인코딩(21byte)한 Card_Name을 Body로 하여 서버 측에 공격 알림 Message 전송	
			Get_Card	획득 Message 전송	string	Card_Name		byte로 인코딩(21byte)한 Card_Name을 Body로 하여 서버 측에 습득 알림 Message 전송	
			Scrap_Card	폐기 Message 전송	string	Card_Name		byte로 인코딩(21byte)한 Card_Name을 Body로 하여 서버 측에 폐기 알림 Message 전송	
			Turn_end	턴 종료				서버에게 턴종료 Message 전송	

		d	Message 전송					
		Log_Sen d	Log Message 전송	string	Log		byte로 인코딩(51byte) 한 Log를 Body로 하여 서버 측에 Message 전송	
		Log_Rec eive	Log Messag e를 받아 디코딩			string	byte로 인코딩(51byte)된 Log를 string으로 디코딩하여 리턴	
		Game_En d	게임종료 Message 전송					
		Score_se nd	자신의 Score 전송	int	Score			
		Recv_Tot al_Score		int[]	all_scor e		수신 받은 유저들의 점수 정보를 배열로 가공	

Server Method

Class	Property		Function	Detail	ETC
	type	name			
Client	TcpClient	client	client 정의	TCP 네트워크 서비스에 대한 클라이언트 연결 정의	
	NetworkStream	stream		네트워크 액세스에 대한 데이터의 내부 스트림 정의	
	string	ID		유저 ID 정의	
	bool	Accept	유저 상태	유저 상태 정의. true : 게임 시작 동의 default : false	
	bool	respond	유저 상태	유저 상태 정의. true : 큐 내부에 존재 default : false	

Server Method

Class	Method	Function	Parameter		return	Detail	ETC
			Variable type	Variable name			
Program	Main	메인	string[]	args	x	게임 접속 시 필요한 기본 정보. 서버가 시작되고 종료될 때 까지 유지되는 Main_Thread 초기 생성되는 방 번호 1번으로 설정 Client 객체를 생성하여 List로 가공 후 AddClient로 전송, Client List의 객체 길이가 4가 되었을 때, 객체 리스트는 Create_Game으로 client 정보 및 방 번호와 함께 넘어감	
	Add Client	큐에 Client 추가	TcpListener	server		TcpClient를 통해 요청된 정보를 토대로 Client 객체에 저장. 큐 리스트에 Client 객체 전송. 비동기 통신을 통해 다수의 Client 객체들을 큐 List에 전송 가능하도록 구현. Main으로 돌아가 다시 유저의 접속을 기다림	async, static, private
			List<Client>	clients	x	CANCEL_MATCHING: ID값을 이용하여 index 결정, 해당 List의 client & stream 종료 및 List배열에서 제거. 매칭 취소 인원이 존재함을 나머지 Client에게 전송	
			int	host_number		ACCEPT: respond 및 Accept의 값을 true로 변경. 큐 내부에 존재함을 알리며 게임을 수락함을 알림 DECLINE: respond 값만 true로 변경. 인원 충족 상태에서 게임 거부 했음을 나타냄	
	Create_Game	게임 방 생성	List<Client>	clients	x	게임 생성. Client 정보를 토대로 랜덤 정렬. 정해진 순서를 full_list라는 Client 객체 배열에 할당. AddClient 메서드에서 넘긴 클라이언트에게 큐가 다 차다는 메시지를 전송. Client 중 response 값이 false인 경우, 해당 큐 나감 accept 값이 false인 경우, 게임 거절로 간주, 각 Client에게 GAME_CANCEL 메시지를 전송, 통신중인 client 및 stream 모두 close response 및 accept 값이 true인 경우 유저들의 ID값과	

					GAME_START 메시지를 각 Client에게 전송 List형으로 Play_Game으로 full_list를 매개변수로 전송	
Play_Game			Client[]	full_list x	<p>1. 시작 턴을 넘겨줌 TURN_SEND 메시지를 현재 턴인 유저의 Stream으로 전송.</p> <p>2. 주고받기 LOG_SEND: 다른 클라이언트들에게 받은 메시지를 전송해준다. ALERT_ACTION: 카드 사용 시 액션을 분석 후 그에 맞는 값을 전송 ATTACK : 마녀카드를 사용 한 경우. 다른 유저에게 ALERT_ACTION 상태를 알려주고, 그것이 공격카드일 경우 다른 유저들로부터 반응을 리턴 받는다. (방어카드가 존재하는 경우 방어를 했다는 Log만 전송받으며, 그렇지 않은 경우 저주카드를 습득했다는 Log 및 해당 유저로부터 새로운 ALERT_ACTION(Get_Card) 메시지를 추가로 받아 다른 유저에게 전달한다. SCRAP_CARD : GET_CARD : ALERT_ACTION 중 위 두 케이스의 경우 그대로 다른 유저에게 전달하여, 해당 클라이언트 측에서 해석을 하여 UI를 수정한다. TURN_END: 현재 플레이어가 자신의 턴을 종료했다는 의미의 메시지로, 서버는 해당 메시지를 통해 기존의 유저와의 통신을 중단하고 다음 플레이어와 통신을 시도한다. GAME_FIN: 현 플레이어가 게임을 끝냈다는 의미의 메시지로, 서버는 해당 메시지를 통해 각 클라이언트 측에게 현재 점수를 집계하여 서버측에 전송을 요구하며, 서버는 집계된 점수들의 목록을 클라이언트 측으로 전송하여, 클라이언트 측은 이를 이용하여 최종 스코어를 출력한다.</p>	
game_Fin			Client[]	full_list t	<p>먼저, int 형 배열로 점수를 저장 할 공간을 생성</p> <p>1. client에게 점수 요청 2. client 요청 응답 받기 각 client에게 응답받은 정보를 int형 배열에 저장 3. 점수 다시 보내기 및 끊기 각 client에게 모든 유저의 점수를 보낸 후 stream 및 client 접속 close</p>	



