

AUTORI

BONALI LUCA

LAMPARELLI ANDREA



DOCUMENTAZIONE PROVA FINALE

INDICE:

STRUTTURA DEL PROGETTO	3
DIAGRAMMI UML	4
DIAGRAMMI DI SEQUENZA	17
MANUALE UTENTE	0
RINGRAZIAMENTI	0

STRUTTURA DEL PROGETTO:

PACKAGE SERVER :

MAINSERVER :

API : PAG 4

GAME_SERVER : PAG 5

MODEL:

ACTION_SPACES : PAG 6

BOARD : PAG 7

EFFECTS : PAG 9

FIELDS : PAG 10

PACKAGE CLIENT:

MAINCLIENT;

API;

CLI;

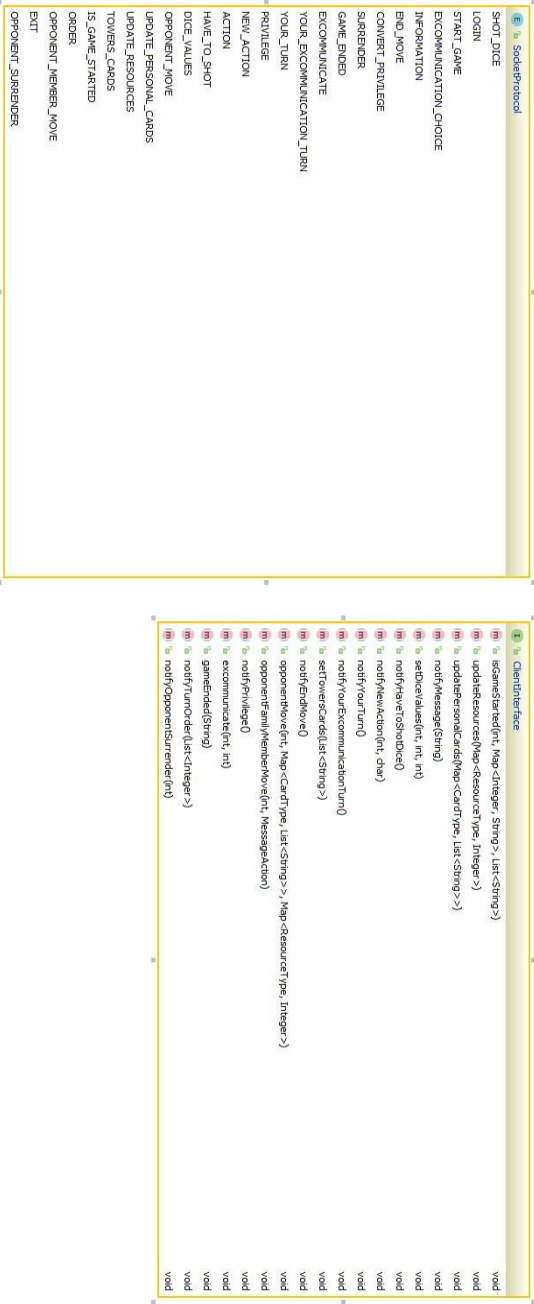
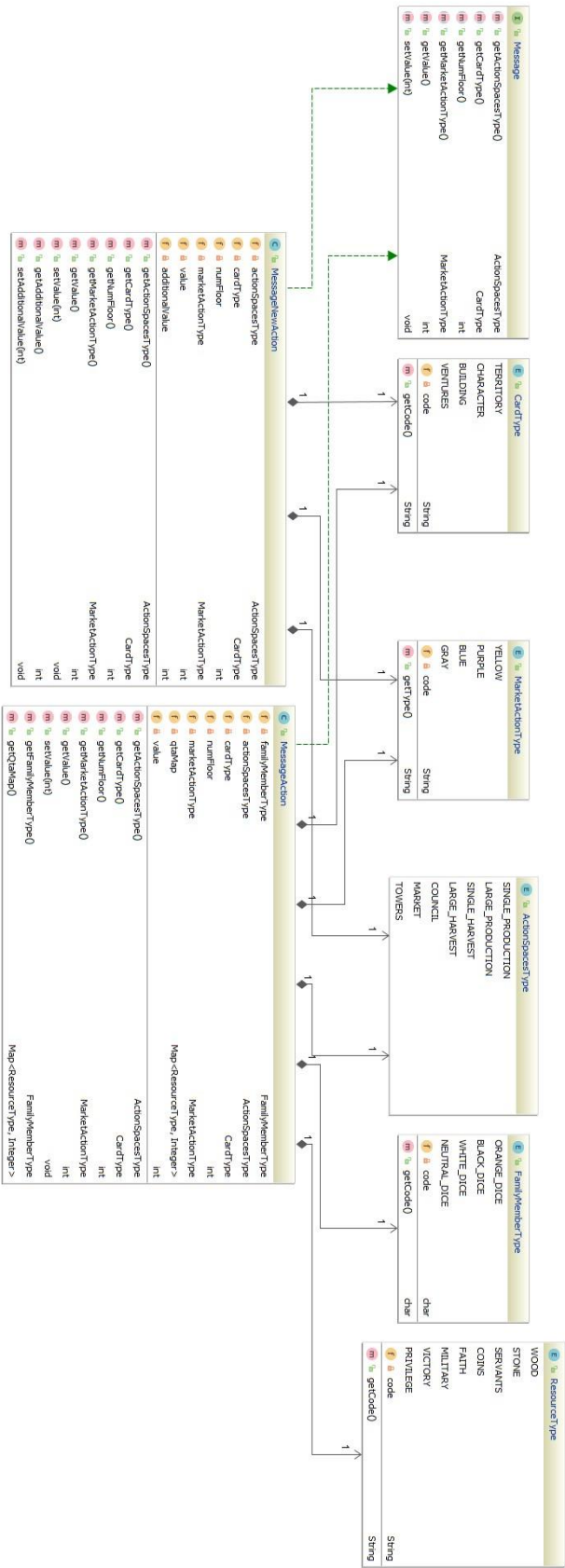
GUI;

GAME_MODE_SELECTION;

GAME;

COMPONENT;

PACKAGE API:



PACKAGE API: PACKAGE CONDIVISO LATO CLIENT E SERVER, CHE CONTIENE:

- LE INTERFACCE DI COMUNICAZIONE RMI (CLIENTINTERFACE , PLAYERINTERFACE , SERVERINTERFACE), CONTENENTI I METODI NECESSARI PER LO SCAMBIO DI INFORMAZIONI, CHE VERRANNO IMPLEMENTATE IN MANIERA DIFFERENTE A SECONDA DEL METODO DI CONNESSIONE.
- IL PACKAGE MESSAGE, CHE INCAPSULA I VARI TIPI DI MESSAGGI SCAMBIATI ATTRAVERSO SOCKET E GENERALIZZATI ATTRAVERSO L' INTERFACCIA MESSAGE . OGNI MESSAGGIO CONTIENE GENERALMENTE LA DESCRIZIONE DI UNA MOSSA, (SPAZIO AZIONE, GIOCATORE , FAMILIARE SELEZIONATO, VALORE DEL FAMILIARE ECC). ALL' INTERNO DI QUESTO PACKAGE È PRESENTE ANCHE LA ENUM SocketProtocol, CHE RACCHIUDE TUTTI I TIPI DI MESSAGGI POSSIBILI DI GESTIONE DELLA PARTITA.
- IL PACKAGE TYPES, CHE CONTIENE ENUMERAZIONI E TIPOLOGIE RIGUARDANTI SPAZI AZIONE, CARTE, FAMILIARI , MERCATO , RISORSE E FASI DEL GIOCO.

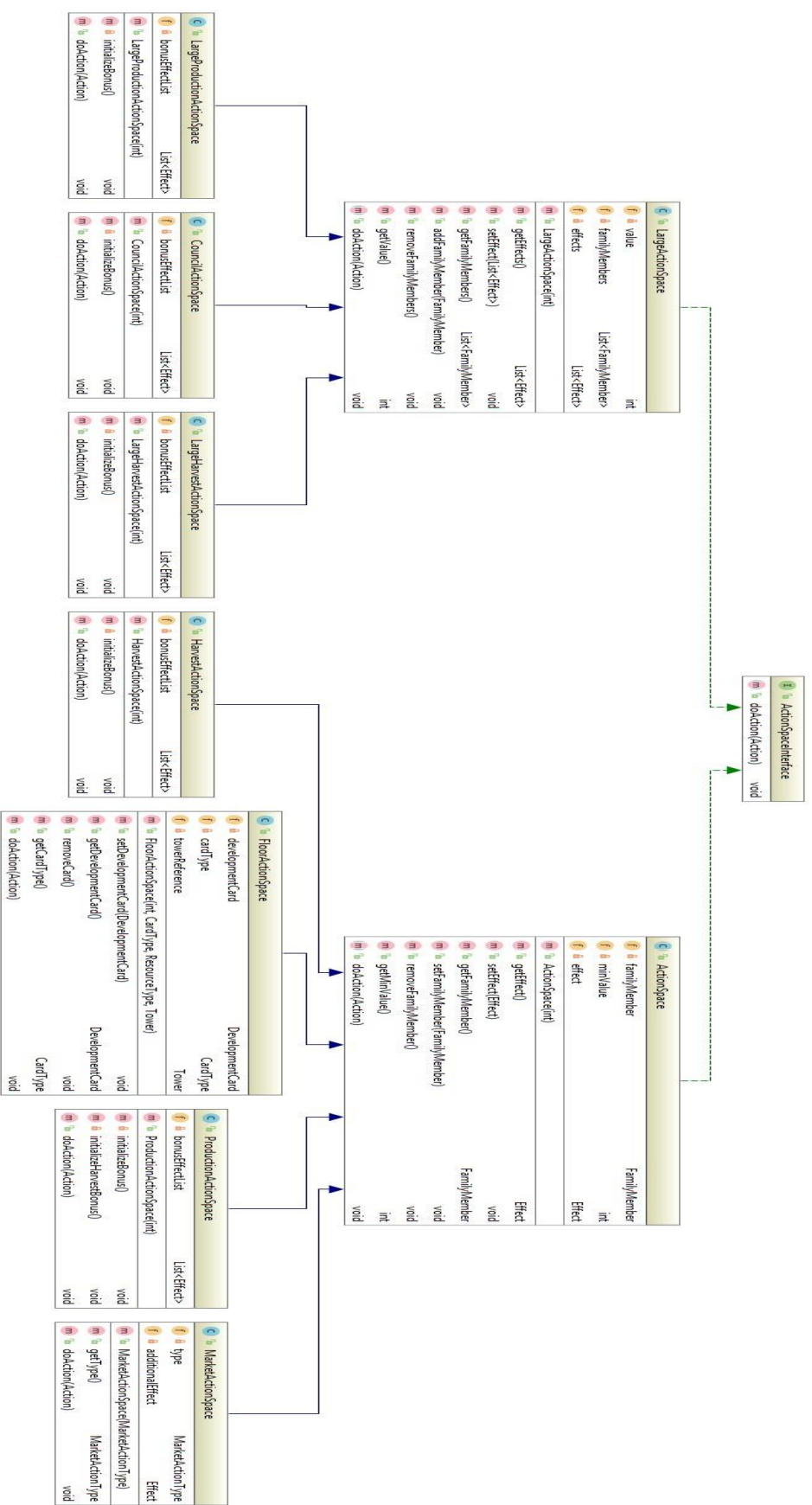
TUTTI QUESTI SONO ELEMENTI UTILI A ENTRAMBI I LATI DEL PROGETTO, ED ESSENDO SOLO GENERALIZZAZIONI E ASTRAZIONI, SI PRESTANO BENE A ESSERE UTILIZZATE IN ENTRAMBI GLI AMBITI, IN MODO DA NASCONDERE IMPLEMENTAZIONE E FUNZIONAMENTO AL CLIENT, E ALLO STESSO TEMPO DI FACILITARE LA COMUNICAZIONE AL SERVER BASANDOSI SU ELEMENTI COMUNI.

PACKAGE GAME_SERVER:



PACKAGE GAME_SERVER: PACKAGE CHE CONTIENE I MODULI DI COMUNICAZIONE ATTRAVERSO LE DUE IMPLEMENTAZIONI DELL' INTERFACCE PLAYERINTERFACE (ABSTRACTPLAYER) E SERVERINTERFACE (ABSTRACTSERVER), E IL PACKAGE EXCEPTION.

- **ABSTRACTSERVER:** CLASSE CHE IMPLEMENTA L' INTERFACCIA SERVERINTERFACE E CHE CONTIENE I METODI INIZIALI DI CONNESSIONE ALLA PARTITA E CHE SARÀ ESTESA DALLE DUE IMPLEMENTAZIONI DIFFERENTI SOCKETSERVER E SERVERRMI (LE CUI ISTANZE VENGONO CREATE DIRETTAMENTE NELLA CLASSE MAINSERVER, CHE CONTIENE IL METODO MAIN).
- **ABSTRACTPLAYER:** CLASSE CHE ESTENDE PLAYERINTERFACE E CHE GENERALIZZA PLAYERSOCKET E PLAYERRMI, CHE CONTENGONO I METODI DI COMUNICAZIONE SERVER-CLIENT.
- **EXCEPTION:** CONTIENE I DUE TIPI DI ECCEZIONI CHE VENGONO LANCIATE QUANDO SI HA UN MALFUNZIONAMENTO O UN COMPORTAMENTO IMPREVISTO LEGATO SOLAMENTE ALL' IMPLEMENTAZIONE DEL GIOCO, AD ESEMPIO QUANDO L' UTENTE PROVA A FARE UNA MOSSA NON VALIDA (FORZA INSUFFICIENTE, SPAZIO AZIONE GIÀ OCCUPATO, NON È IL TUO TURNO...).
- **PLAYERRMI:** QUESTA CLASSE CONTIENE I METODI DI NOTIFICA DELLA PARTITA, DA SERVER VERSO CLIENT, E OGNI METODO VERRÀ CHIAMATO SULL' IMPLEMENTAZIONE SU SERVER DELL' INTERFACCIA CLIENTINTERFACE , COME RAPPRESENTAZIONE DEL CLIENT SUL SERVER.
- **PLAYERSOCKET:** QUESTA CLASSE CONTIENE I METODI CHE CREANO MESSAGGI DA INVIARE COME OGGETTI NELLO STREAM, SPECIFICANDO QUINDI LE MOSSE CHE IL CLIENT POTRÀ ESEGUIRE.



PACKAGE ACTION SPACES:

PACKAGE ACTION SPACES: PACKAGE CHE CONTIENE LA DEFINIZIONE E IL COMPORTAMENTO ASSOCIATO DI UNA MECCANICA DI GIOCO FONDAMENTALE, GLI SPAZI AZIONE:

OGNI SPAZIO AZIONE IMPLEMENTA L' INTERFACCIA ACTIONSPACEINTERFACE, CHE CONTIENE IL METODO DOACTION(ACTION ACTION) COMUNE A TUTTI GLI SPAZI E CHE PRENDE COME PARAMETRO UN' ISTANZA DELLA CLASSE ACTION ALL' INTERNO DI QUESTO PACKAGE.

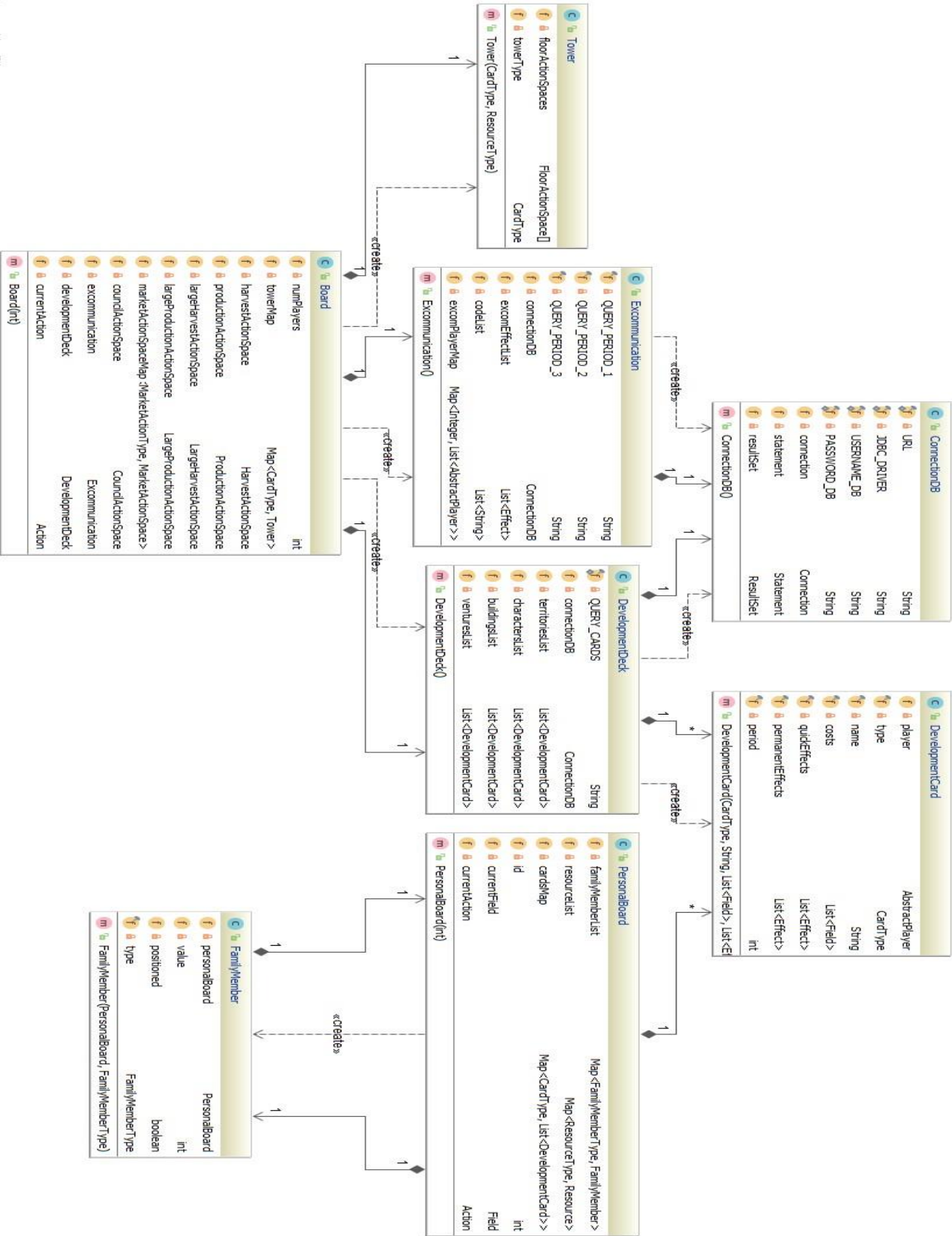
- ACTION: CLASSE CHE DESCRIVE UN' AZIONE ATTRAVERSO ATTRIBUTI COME FAMILIARE, GIOCATORE, VALORE DELL' AZIONE.

IL PACKAGE POI SI DIVIDE NEI DUE PACKAGE DEI TIPI DEGLI SPAZI AZIONE SINGLEACTIONSPACE E LARGEACTIONSPACE OGNUNO DEI QUALI HA UNA CLASSE ASRATTA CHE IMPLEMENTA L' INTERFACCIA E CHE NE GENERALIZZA IL COMPORTAMENTO , ACTIONSPACE E LARGEACTIONSPACE.

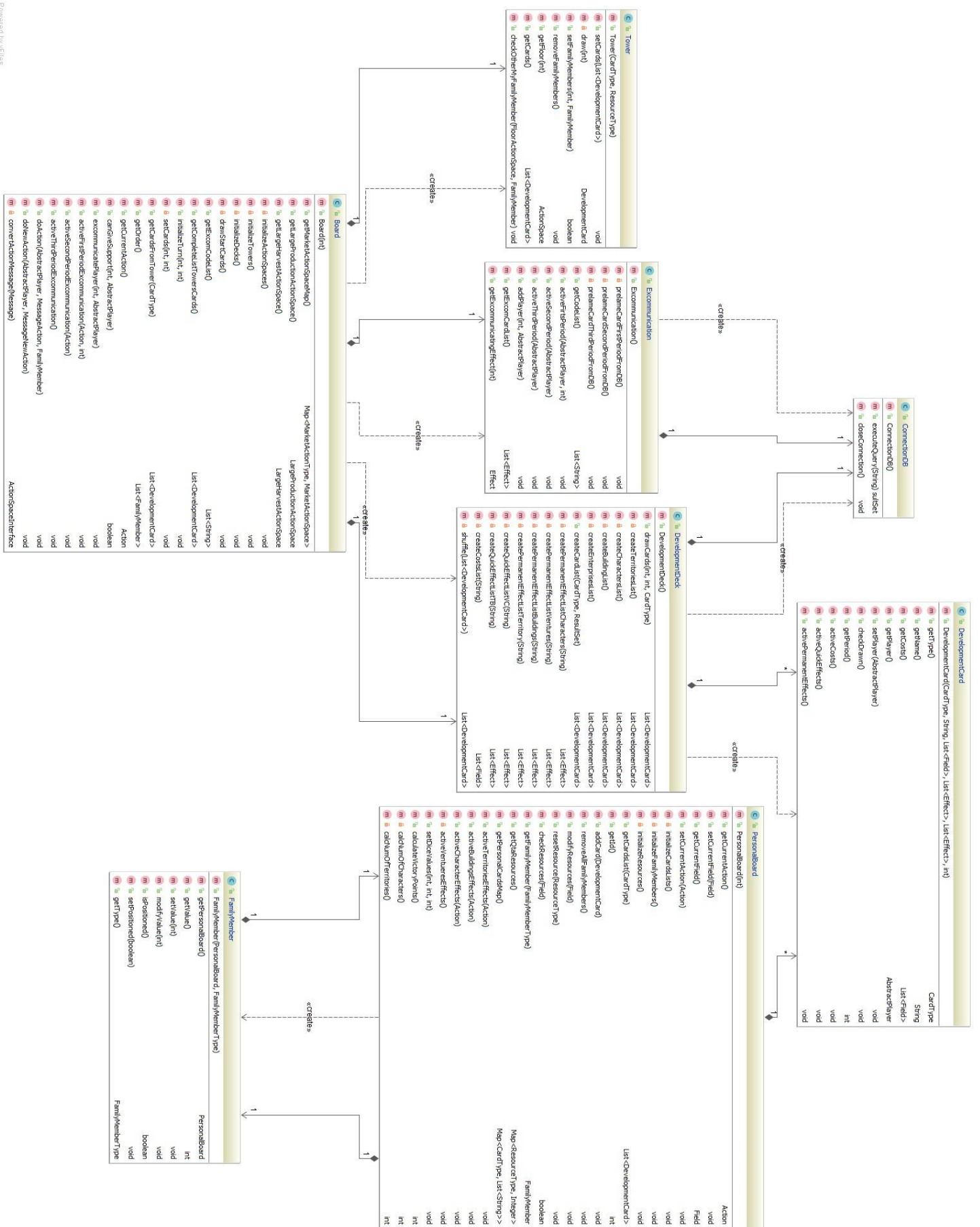
- FLOORACTIONSPACE: SPAZIO AZIONE DEI PIANI DELLE TORRI DI CUI AVRÀ UN RIFERIMENTO E CHE RICEVERÀ L' AZIONE DI RACCOLTA DI UNA CARTA.
- HARVESTACTIONSPACE & PRODUCTIONACTIONSPACE: CLASSI SIMILI CHE RAPPRESENTANO GLI SPAZI AZIONE DI RACCOLTA E PRODUZIONE, CHE DIFFERISCONO PERCHÉ CHIAMANO METODI IN BASE A DIVERSI TIPI DI CARTE.
- MARKETACTIONSPACE: CLASSE CHE RAPPRESENTA LO SPAZIO AZIONE DEL MERCATO, E CHE CONTIENE IL TIPO DETERMINATO DALLA ENUM NEL PACKAGE API. IL SUO METODO DOACTION ESEGUE SEMPRE UN INCREMENTO DI RISORSE.
- COUNCILACTIONSPACE: DEFINISCE LO SPAZIO AZIONE DEL CONSIGLIO.
- LARGEHARVEST & LARGEPRODUCTION: DEFINISCONO I DUE SPAZIAZIONE GRANDI, E HANNO LO STESSO COMPORTAMENTO DI QUELLI SINGOLI, SOLO POSSONO RICEVERE UNA LISTA DI FAMILIARI.

TUTTI QUESTI SPAZI AZIONE SONO GENERALIZZATI NEL PACKAGE API ATTRAVERSO LA ENUMERAZIONE ACTIONSPACEType.

PACKAGE BOARD (ATTRIBUTES)



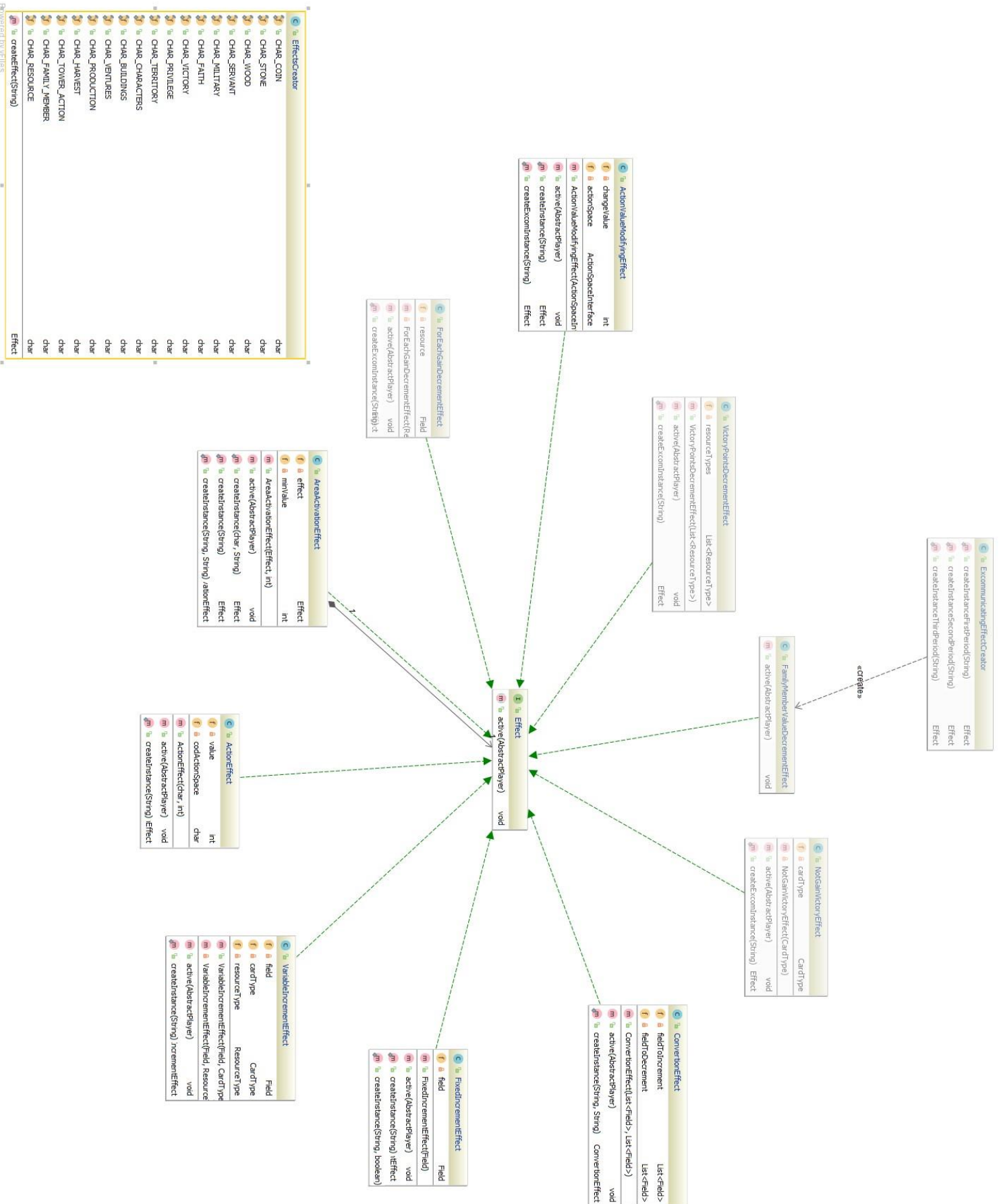
PACKAGE BOARD (METHODS)



PACKAGE BOARD: IL PACKAGE BOARD È CONTENUTO INSIEME AL PACKAGE ACTIONSPACE, EFFECTS E FIELD NEL PACKAGE MODEL, CHE SPECIFICA IL MODELLO FORMALE DEL GIOCO E TUTTI I SUOI COMPONENTI, E RAPPRESENTA IL TABELLONE DI GIOCO.

- **BOARD:** CLASSE CHE RACCOGLIE ISTANZE DELLE ALTRE CLASSI PRESENTI NEL PACKAGE MODEL, COME TOWER, DEVELOPMENTCARD E PERSONALBOARD. FORNISCE I METODI DI INIZIALIZZAZIONE DEL TABELLONE, COME IL SETTAGGIO DELLE 16 CARTE INIZIALI O DEGLI SPAZI AZIONE.
- **CONNECTIONDB:** CLASSE CHE SI INTERFACCIA CON IL DATABASE DELLE CARTE E DELLE CARTE SCOMUNICA. TUTTE LE CARTE SONO CODIFICATE DA UNA STRINGA CHE NE DETERMINA L' EFFETTO DI ATTIVAZIONE, IL COSTO E L' EFFETTO PERMANENTE.
- **DEVELOPMENTCARD & DEVELOPMENTCARDDECK:** CLASSI CHE RISPETTIVAMENTE MODELLANO LE CARTE SVILUPPO E SI OCCUPANO DELLA GESTIONE DEL MAZZO, PESCANDO CASUALMENTE LE 16 CARTE DALLA LISTA FORNITAGLI DAL DB E INSERENDOLE NEL TABELLONE.
- **EXCOMMUNICATION:** CLASSE CHE MODELLA LE CARTE SCOMUNICA, ANCHE QUESTA UTILIZZA LA CLASSE CONNECTIONDB E PESCA CASUALMENTE TRE TESSERE (UNA PER PERIODO) E DI ATTIVARE GLI EFFETTI DELLE CARTE QUALORA NECESSARIO.
- **FAMILYMEMBER:** MODELLA IL SINGOLO FAMILIARE E CARATTERIZZATO DALL' ENUMERAZIONE FAMILYMEMBERTYPE.
- **PERSONALBOARD:** CARATTERIZZA LE RISORSE DEL SINGOLO GIOCATORE, AD ESEMPIO QUANTITÀ DI LEGNA, LISTA DELLE CARTE, FAMILIARI DISPONIBILI. SARÀ QUESTA CLASSE A COMUNICARE, ATTRAVERSO LA CLASSE BOARD E GAME, CON IL CLIENT, E SARANNO I SUOI ATTRIBUTI A ESSERE MODIFICATI DURANTE LA PARTITA.
- **TOWER:** CARATTERIZZATA DA UN TIPO (CARDTYPE DEL PACKAGE API) E DA UNA LISTA DI CARTE LA TORRE HA METODI CHE PERMETTONO DI SETTARE E PESCARE LE CARTE. HA INOLTRE UN ARRAY DI FLOORACTIONSPACES CHE RAPPRESENTANO I PIANI.

PACKAGE EFFECTS:



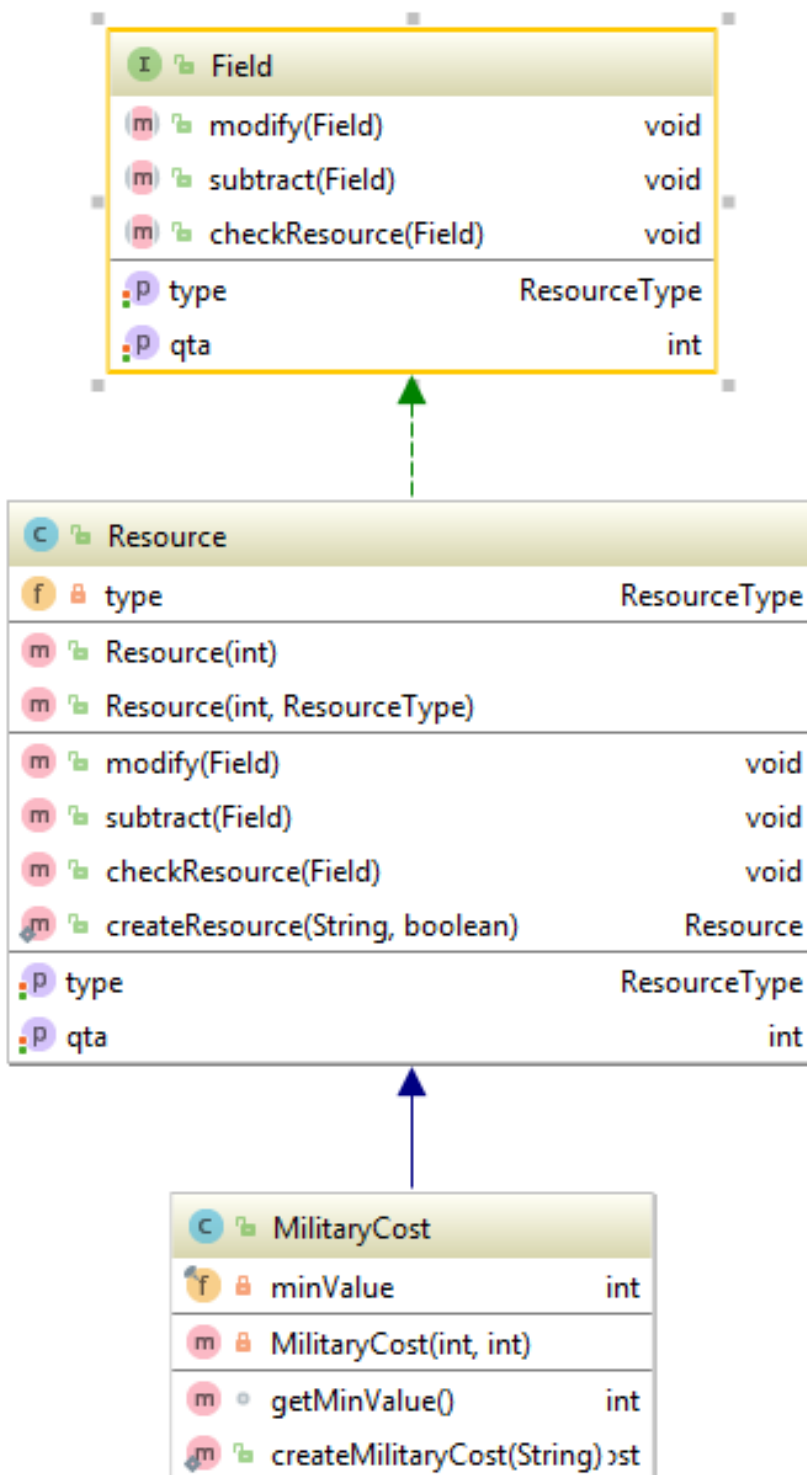
PACKAGE EFFECTS: CONTIENE TUTTI I POSSIBILI EFFETTI DELLE CARTE SVILUPPO E SCOMUNICA, IN DUE DIFFERENTI PACKAGES. TUTTI GLI EFFETTI IMPLEMENTANO L' INTERFACCIA EFFECT, CHE FORNISCE IL METODO ACTIVE(ABSTRACTPLAYER ABSTRACPLAYER) COMUNE PRENDENDO COME PARAMETRO IL GIOCATORE CHE LI ATTIVA.

- **DEVELOPMENTEFFECTS:** QUESTI EFFETTI VENGONO CREATI DALLA CLASSE EFFECTSCREATOR E DAL SUO METODO STATICOCREATEEFFECT, IN BASE AL CODICE STRINGA CHE GLI SI PASSA.
- **ACTIONEFFECT:** EFFETTO SENZA FAMILIARE CARATTERIZZATO SOLO DA CODICE DELLO SPAZIO AZIONE E VALORE.
- **ACTIONVALUEMODIFYEFFECT:** EFFETTO CHE PERMETTE DI MODIFICARE IL VALORE DI UN' AZIONE IN MANIERA PERMANENTE.
- **AREAACTIVATIONEFFECT:** EFFETTO CHE SPECIFICA L' ATTIVAZIONE DI UNA DELLE DUE AREE RACCOLTA O PRODUZIONE.
- **CONVERSIONEFFECT:** PRENDE UNA LISTA DI RISORSE DA AUMENTARE E UNA DA DECREMENTARE E ATTIVA IL METODO DOACTIVE SUL GIOCATORE E QUINDI SULLA SUA PLANCIA.
- **FIXEDINCREMENTEFFECT & VARIABLEINCREMENTEFFECT:** INCREMENTO FISSO E VARIABILE DI UNA RISORSA
- **EXCOMMUNICATING_EFFECTS:** CONTIENE TUTTI GLI EFFETTI SCOMUNICA E HA UNA CLASSE EXCOMMUNICATINGEFFECTCREATOR CHE FUNZIONA IN MANIERA SIMILARE A EFFECTSCREATOR.
- **FAMILYMEMBERVALUEDECREMENTEFFECT:** CLASSE SPECIFICA PER L' EFFETTO CHE TOGLIE 1 AL VALORE DI OGNI FAMILIARE (TRANNE IL NEUTRO).
- **FOREACHGAINDECREMENTEFFECT:** EFFETTO CHE DECREMENTA IL VALORE DI UNA RISORSA SPECIFICA DI 1 OGNI VOLTA CHE SE NE GUADAGNANO, ANDANDO A MODIFICARE LE RISORSE DEL GIOCATORE CHE LO ATTIVA.
- **NOTGAINVICTORYEFFECT:** CLASSE CHE PRENDE PARAMETRO IL TIPO DI CARTA E RESETTA LA LISTA DELLE CARTE DEL TIPO PASSATO A FINE PARTITA, IN MODO CHE IL PLAYER NON GUADAGNI I PUNTI VITTORIA CORRISPONDENTI.
- **VICTORYPOINTSDECREMENTEFFECT:** CLASSE CHE DECREMENTA I PUNTI VITTORIA OTTENUTI IN BASE AL NUMERO DI RISORSE (OGNI 5) PRESENTI NELLA SUA PLANCIA.

OGNUNO DI QUESTI EFFETTI È CREATO IN BASE A UN CODICE, UNA STRINGA SALVATA NEL DB CHE VIENE ANALIZZATA ATTRAVERSO LE ISTRUZIONI SUBSTRING E CHARAT, E ATTIVATI DAL GIOCATORE QUANDO PESCA UNA CARTA, (IN BASE AL CODICE SI CREA UN EFFETTO DIVERSO), E ALLA FINE DEL 2,4,6 TURNO DI GIOCO PER LE SCOMUNICHE.

PACKAGE FIELDS:

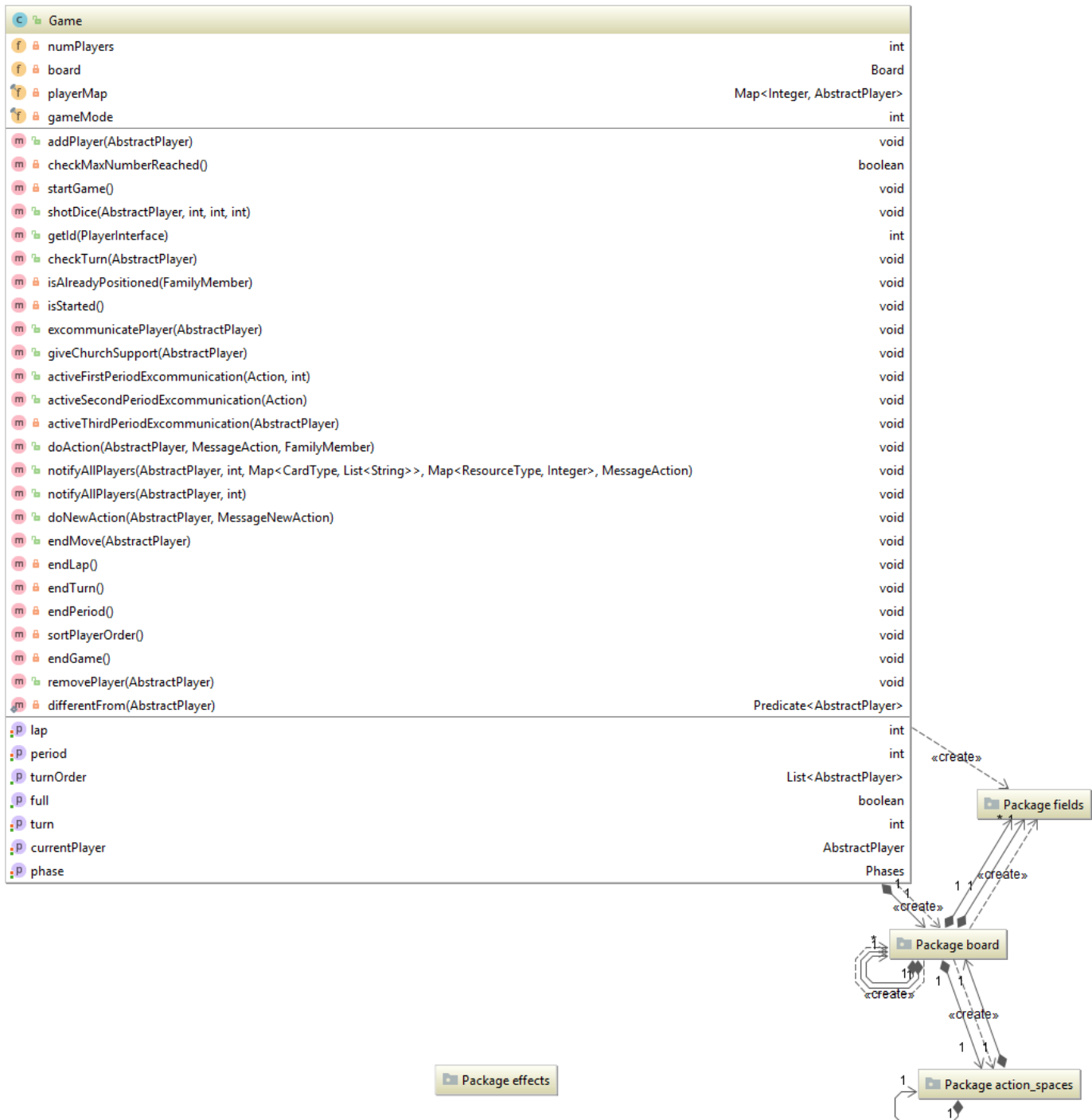
– CONTIENE LE RISORSE CHE VERRANNO UTILIZZATE DAL GIOCATORE, ED È CARATTERIZZATO DALLA TIPOLOGIA DELLA ENUMERAZIONE RESOURCEType DEL PACKAGE API



Powered by yFiles

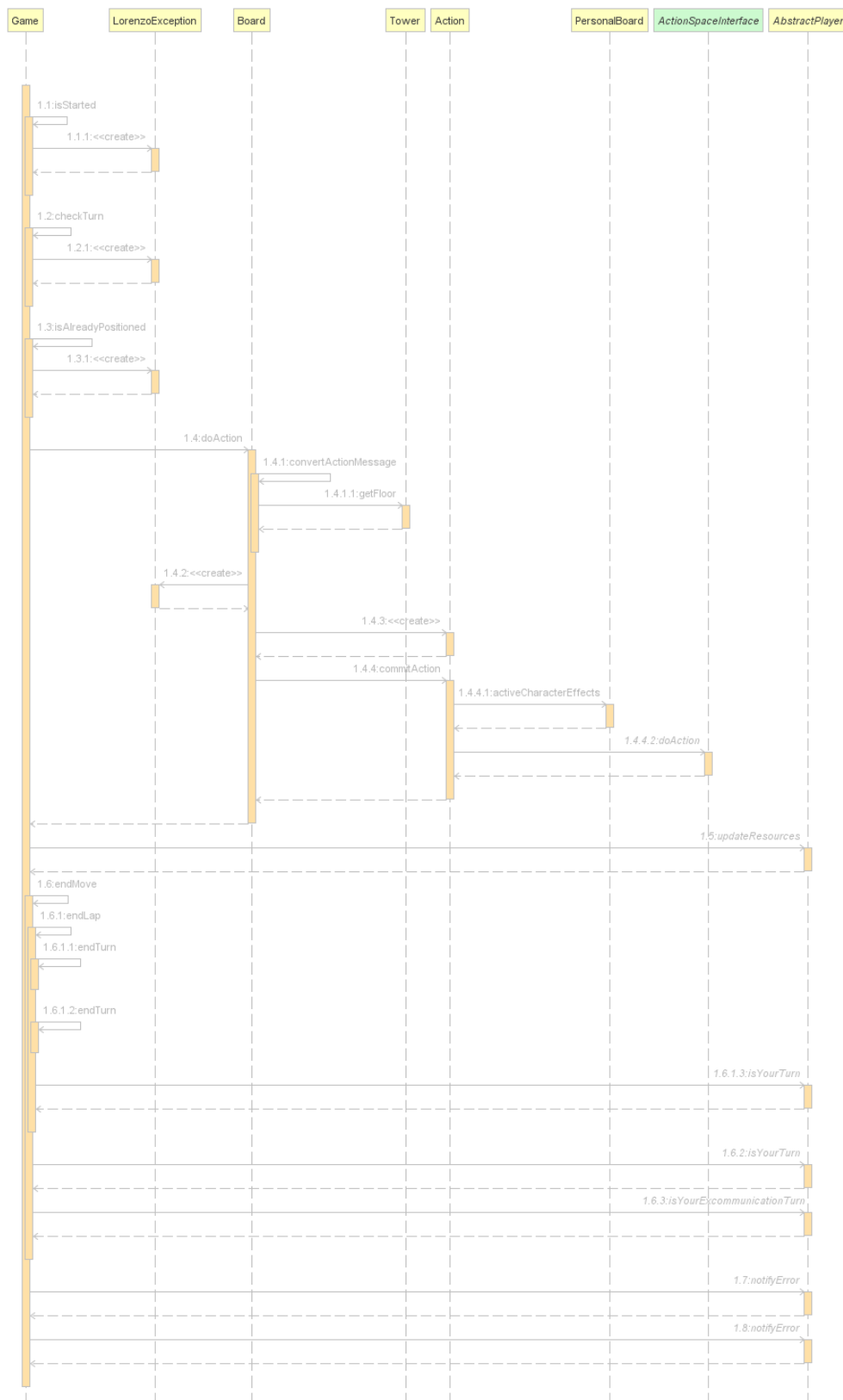
PACKAGE MODEL E CLASSE GAME:

- LA CLASSE **GAME** RACCHIUDE IL COMPORTAMENTO DELLA PARTITA, INIZIALIZZA IL TABELLONE, AGGIUNGE GIOCATORI ALLA PARTITA, ORGANIZZA I TURNI, LE FASI E CHIAMA I METODI DI **CLIENTINTERFACE** PER COMUNICARE CON IL CLIENT.
- UNA NUOVA ISTANZA DI GAME VIENE CREATA OGNI QUAL VOLTA UN GIOCATORE VOGLIA GIOCARE UN' ALTRA PARTITA, PRENDENDO IN INGRESSO LA MODALITÀ DI GIOCO (IL NUMERO DI GIOCATORI) E INIZIALIZZANDO, UNA VOLTA ARRIVATA LA RICHIESTA DI GIOCO, LA PLANCIA DEL GIOCATORE.

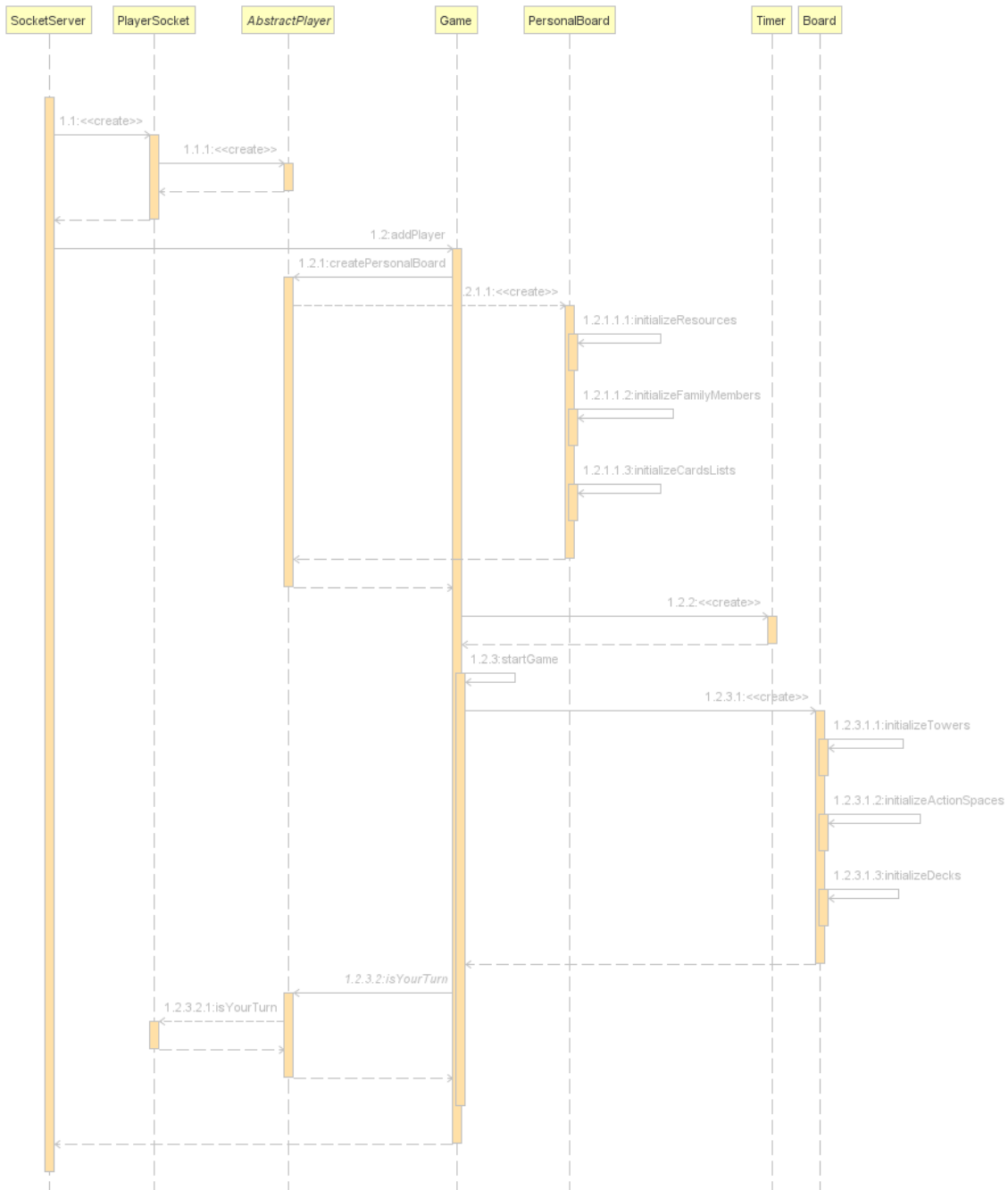


= DIAGRAMMI DI SEQUENZA =

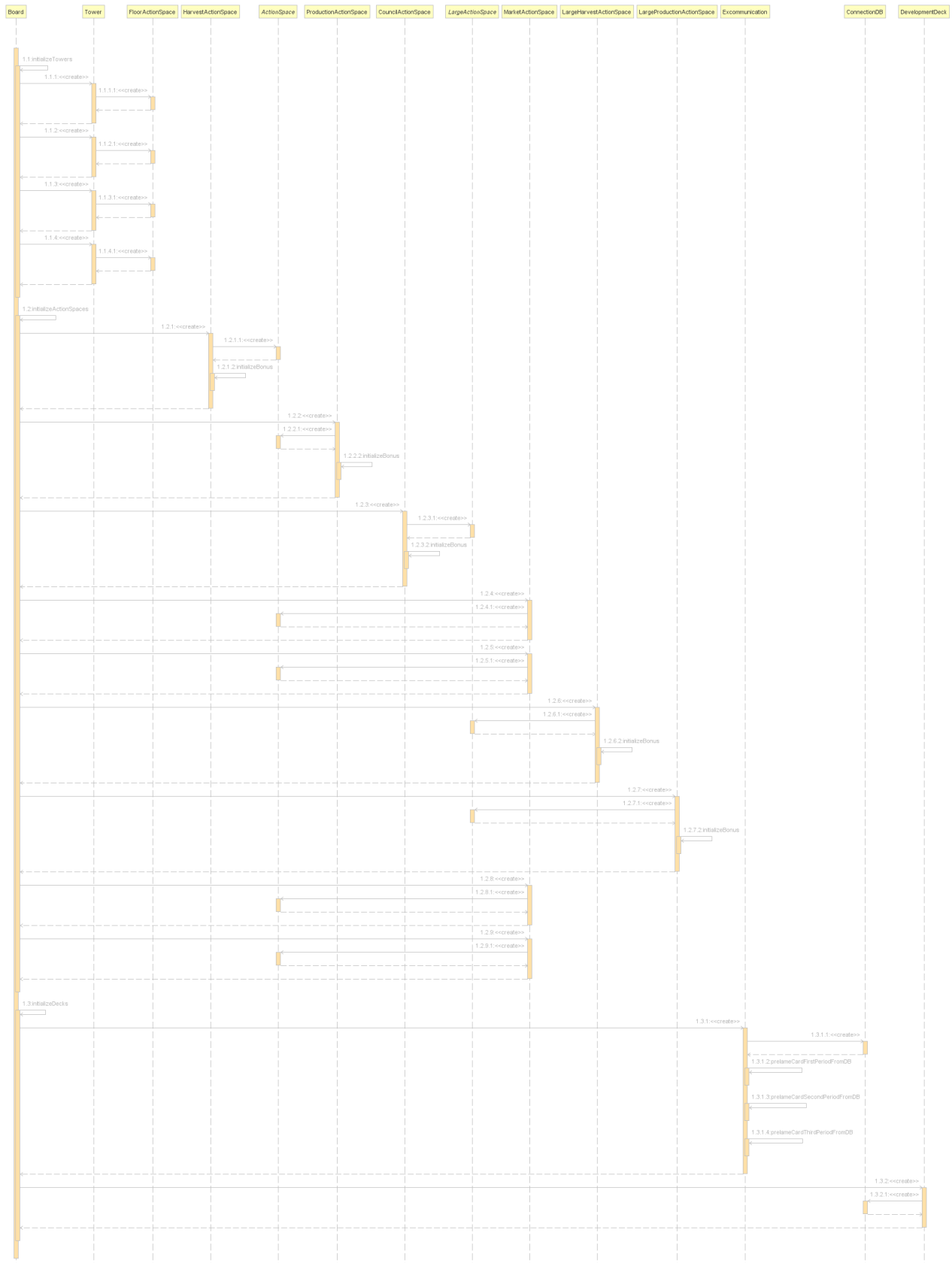
DoAction();



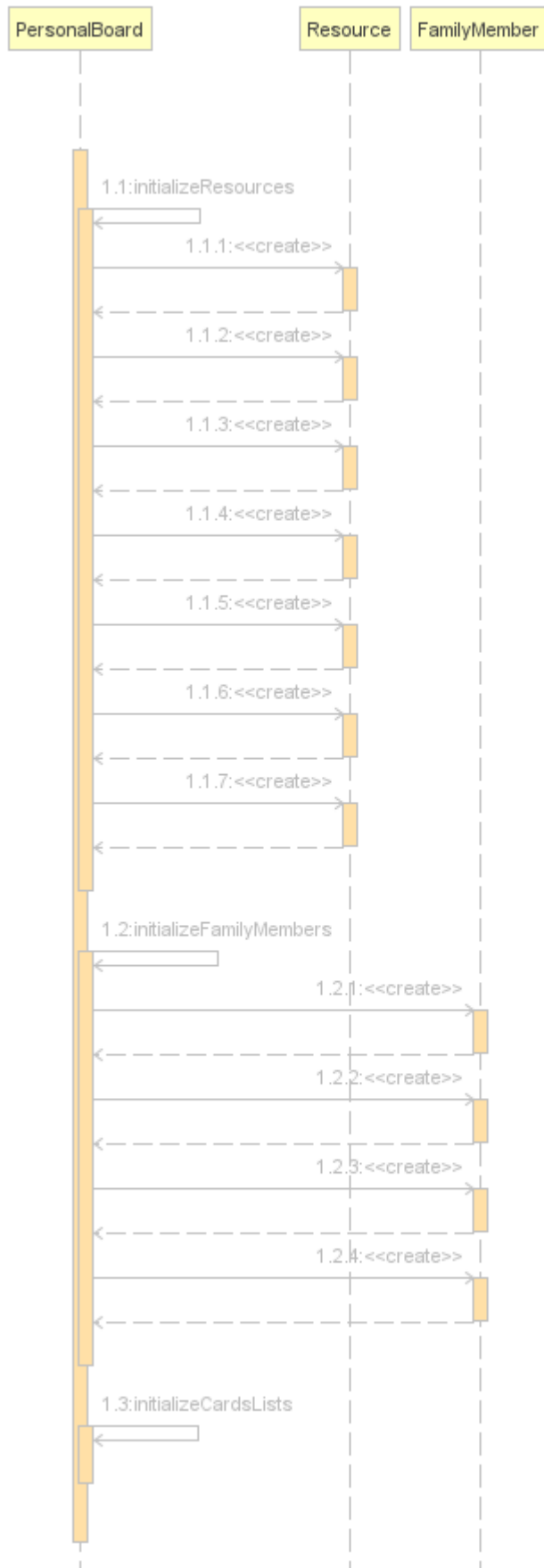
StartGame() (Socket)



InitializeBoard();



InitializePersonalBoard();



EndGame();

