model					
		Cond			
	ser	Card	«enume Card		«enumeration» CardAttribute
username: String password: String		- name: String - number: String		. 7 -	
nickname: String		- description: String	MONSTER		DARK
score: int		- type: CardType	SPELL TRAP		DIVINE EARTH
mainDeckID: int		7,7	IKAP		FIRE
getUserByUsername (u	sername : String): User				LIGHT
getUserByNickname (nickname : String): User					WATER
isPasswordCorrect (pas	sword : String): boolean				WIND
setPassword (password	l : String)				
getNickname (): String		T			
setNickname (nickname : String)		Monster	Spell		Trap
getUsername (): String		- level: int	- property:	Card	- property: Card
getScore (): int		- attackPoints: int	Property		Property
setScore (score : int)		- defensePoints: int			
		- attribute: CardAttribute - type: MonsterType			
«enumeration»	«enumeration»	- effect: EffectType			
MonsterType	EffectType	- summon: SummonType			
OHA	NORMAL	January Paris			
QUA EAST	CONTINUOUS				
EAST_WARRIOIR	IGNITION				
REATOR_GOD	TRIGGER	Player			Deck
YBERSE	QUICK	- user: User		- mainDec	k: List <card></card>
INOSAUR		- deck: GameDeck		- sideDeck	: List <card></card>
IVINE_BEAST		- field: Field		- usernam	e: String
RAGON	«enumeration»	- lifePoints: int		- id: int	
AIRY END	SummonType			- name: S	tring
SH					
ISECT	NORMAL				
ACHINE	TRIBUTE SPECIAL				
_ANT	RITUAL				
SYCHIC	RITOAL				
/RO		Field	_	_	
EPTILE OCK		Field			GameDeck
EA_SERPENT		- deck: List <gamecard></gamecard>		- mainDec	k: List <card></card>
PELLCASTER	GameCard	hand: List<gamecard></gamecard>graveyard: List<gameca< li=""></gameca<>	rds	- sideDeck	: List <card></card>
HUNDER	- card: Card	- banished: List <gamecar< td=""><td></td><td></td><td>eCard (int mainId, in</td></gamecar<>			eCard (int mainId, in
ARRIOR	- attackModifier: List< Integer>	- monsterZone: GameCard		sideId)	
INGED_BEAST	- defenseModifier: List<	- spellZone: GameCard [5]			
YRM OMBIE	Integer>	- fieldZone: GameCard			
ONBIE	- isFaceDown: boolean	+ drawCard (): GameCard			
	- isRevealed: boolean	()			
«enumeration»	- id: int				
Phase	- flip ()				ardLocation
	+ changePosition ()			- position:	
RAW TANDBY				- isInHand	
AIN1					emy: boolean onsterZone: boolean
ATTLE					pellZone: boolean
AIN2	«enumeration»	ShopCards			eldZone: boolean
ND	MenuState	- monsters: HashMap <monst< td=""><td>er, Integer></td><td></td><td>aveyard: boolean</td></monst<>	er, Integer>		aveyard: boolean
	Hendstate	- spells: HashMap <spell, inte<="" td=""><td>eger></td><td>1311011101</td><td>aveyara. Boolean</td></spell,>	eger>	1311011101	aveyara. Boolean
	LOGIN	- traps: HashMap <trap, integ<="" td=""><td>jer></td><td></td><td></td></trap,>	jer>		
«enumeration»	MAIN				
CardProperty	DUEL			Pla	yerInventory
ORMAL	SCOREBOARD			- money: i	
ONTINUOUS	PROFILE				nt ck: HashMap <string,< td=""></string,<>
QUIP	SHOP			Integer>	car riadiniap County,
IELD	CARD_FACTORY	RoundResult		- usernam	e: String
UICK_PLAY ITUAL		- isFirstPlayerWin: boolean			ntoryByUsername_
OUNTER		- firstPlayerLifePoints: int		(username	
311121		- secondPlayerLifePoints: int			

controller					
ProgramController	CardFactoryController	_	DatabaseController		
	+ getInstance (): CardFactory	- db: Path	Databasecontroller		
	Controller		Map <string, path=""></string,>		
IIILIalize ()	+ importCards (backupName :	- userList: L			
- getinstance (): ProgramController	String) + exportCards (cardNames :		shopCards: ShopCards playerInventories: List <playerinventory></playerinventory>		
- getState (): MenuState	String)	- playerInve			
- setState (state : MenuState) - getCurrenctGameID (): int		+ initialize (
getcurrencedamero (). Inc			File (jsonDB : Path)		
5 16 1			(jsonDB : Path)		
DeckControl	er	+ updateUs	**		
user: User getInstance (): DeckController			st (): List <user> user : User)</user>		
create (name : String)			+ getShopCards (): ShopCards + getPlayerInventories (): List <playerinventory> + updateInventoriesToDB () + exportShopCards (cards : ShopCards) + importShopCards (cards : ShopCards) + getDecks (): List<deck></deck></playerinventory>		
delete (name : String)					
activate (name : String)					
addCard (deckName : String, cardName : Stri removeCard (deckName : String, cardName :					
showAllDecks ()	,,,,,,,				
showAllCards ()		+ updateDe			
showDeck (name : String)		+ addDeck ((deck : Deck)		
UserController		MainMenuController			
view: UserView		- user: User	MenuView ser : User)		
<pre>getInstance (): UserController registerUser (username : String, password : S</pre>	string nickname : String)	+ logout ()			
loginUser (username : String, password : Stri		+ setUser (use			
		+ getUser ():	User		
		+ getInstance	(): MainMenuController		
ScoreboardController		ProfileCo	ontroller		
- user: User	- user: User				
- view: ScoreboardView	- view: ProfileView				
+ setUser (user : User)	+ setUser (user : Us	ser)			
+ getUser (): User + showScoreboard ()	+ getUser (): User + changeNickname	(nickname : String			
+ getInstance (): ScoreboardController			: String, newPassword : String)		
	+ getInstance (): Pr	ofileController			
GameController					
gameControllers: List <gamecontroller></gamecontroller>	SelectionCo	antrollor.	SetController		
id: int		ontroller	- card: GameCard		
selectionController: SelectionController	- card: GameCard - location: CardLoca	tion	- gameControllerID: int		
gameTurnController: GameTurnController cheatController: CheatController	- gameControllerID:		+ set ()		
firstPlayer: Player	+ select (location : 0	CardLocation)	+ changePosition (isOffense :		
secondPlayer: Player			boolean): boolean		
isFirstPlayerTurn: boolean					
effectControllers: List <effectcontroller> numberOfRounds: int</effectcontroller>		111111111111			
roundReults: List <roundresult></roundresult>					
getGameControllerById (id : int): GameContr	oller_				
getCurrentPlayer (): Player					
getRivalPlayer (): Player		1 / / /			
- play () - select (location : CardLocation)					
set ()					
setPositon (isAttack : boolean)	SummonCo	ntroller	AttackController		
- summon ()	- card: GameCard		- attackingMonster: GameCard		
nextPhase ()	- gameControllerID:	int	- attackedMonster: GameCard		
endRound (isFirstPlayerWin : boolean) startRound ()	+ normalSummon ()		- gameControllerID: int		
attack (position : int)	+ tributeSummon ()		+ attack (): boolean		
directAttack ()	+ specialSummon () + ritualSummon ():		+ directAttack (): boolean		
- activateEffect ()	+ flipSummon (): bo				
flipSummon () surrender ()	()1 00	+ IIIpSullillioli (): booleali			
- cancel ()					
exchangeSideDeckCards ()					
		1 1			
GameTurnController					
phase: Phase		1 / / / /			
gameControllerID: int					
attackedMonsters: List <gamecard> chain: List<gamecard></gamecard></gamecard>	CheatCont	roller	EffectController		
isFirstTurn: boolean	- gameControllerID: in	t	- cardsAffected: List <gamecard></gamecard>		
drawPhase ()	+ increaseLifePoints (a	amount : int)	- cardsAffected: List<gamecard></gamecard>- gameControllerID: int		
standbyPhase ()	+ instantWin ()		- effectCard: GameCard		
firstMainPhase ()	+ increaseMoney (user amount : int)	: User,	- gameControllerID: int		
DATE IN THE SECOND SECO	+ forceDraw ()				
battlePhase () secondMainPhase ()					
secondMainPhase () endPhase ()					
secondMainPhase () endPhase () getPhase (): Phase		roller			
secondMainPhase () endPhase () getPhase (): Phase hasMonsterAttacked (monster : GameCard):	ShopContr				
secondMainPhase () endPhase () getPhase (): Phase hasMonsterAttacked (monster : GameCard): oolean	ShopContr - user: User				
secondMainPhase () endPhase () getPhase (): Phase hasMonsterAttacked (monster : GameCard):	- user: User + buy (name : String)				
secondMainPhase () endPhase () getPhase (): Phase hasMonsterAttacked (monster : GameCard): oolean makeChain ()	- user: User + buy (name : String) + showAllCards ()				
secondMainPhase () endPhase () getPhase (): Phase hasMonsterAttacked (monster : GameCard): oolean makeChain () AIController	- user: User + buy (name : String)				
secondMainPhase () endPhase () getPhase (): Phase hasMonsterAttacked (monster : GameCard): oolean makeChain () AIController gameControllerID: int	- user: User + buy (name : String) + showAllCards ()				
secondMainPhase () endPhase () getPhase (): Phase hasMonsterAttacked (monster : GameCard): colean makeChain () AIController gameControllerID: int play ()	- user: User + buy (name : String) + showAllCards ()				
secondMainPhase () endPhase () getPhase (): Phase hasMonsterAttacked (monster : GameCard): oolean makeChain () AIController gameControllerID: int	- user: User + buy (name : String) + showAllCards ()				
secondMainPhase () endPhase () getPhase (): Phase hasMonsterAttacked (monster : GameCard): oolean makeChain () AIController gameControllerID: int play () standbyPhase ()	- user: User + buy (name : String) + showAllCards ()				

Main

+ main (args : String [*]) + parseCommands ()

+ getCommands (): boolean

