		- CRED UNREGISTERED UNF	COSTERED U					
		GameManager	_GISTERED U					
		-board: Board -maxTurnSeconds: int	GISTERED U					
		-currentRound: Round -controller: Controller	GISTERED U					
		-playerList: List <player> -movesHistory: List<movestatus></movestatus></player>	CICTEDED			User		
		-privateObjectiveCards: List <privateobjectivecards></privateobjectivecards>	GISTERED U			-username: String -positionInRanking: int		
		-toolCards: List <toolcards> -patternCards: List<patterncard></patterncard></toolcards>	GISTERED U	N <u>REGISTERED UNREG</u>	<u>IS</u> TERED (-active: boolean		
		-roundTrack: List <list<dice>> -noOfAck: AtomicInteger</list<dice>	GISTERED U	Player	TERED I	-ready: boolean -userObserver: UserObserver		
		-endRound: AtomicBoolean -stop: AtomicBoolean	GISTERED U	-user: User -score: int	TEDED I	-noOfWins: int -noOfLose: int		
		-doubleMove: AtomicBoolean		-favourToken: int -patternCard: PatternCard	LINED	-matchesPlayed: List <string> -stopWatch: StopWatch</string>		
		-cancelTimer: AtomicBoolean -turnInRound: AtomicInteger	GISTERED U	-privateObjectiveCard: PrivateObjectiveCard	TERED (-isStopWatchRunning: boolean		
«Interface»		-toolCardLock: AtomicBoolean -disconnectedPlayers: Set <player></player>	GISTERED U	+Player(User) +setScore(int): void	TERED (UNREGIST +User(String)		
RemoteSagrada		-playerBroadcaster: PlayerBroadcaster -endOfMatch: AtomicBoolean	GISTERED U	+getScore(): int	TERED I	-getPositionInRanking(): int -setPositionInRanking(int): void		
-setMaxTurnSeconds(int): void -setMaxJoinMatchSeconds(int): void		-controllerTimer: ControllerTimer		+setPatternCard(PatternCard): void +getPatternCard(): PatternCard	TEDED I	+getUsername(): String -getNoOfWins(): int		
~getConnectedUsers(): int		-toolCardThread: Thread -patternCardsChosen: AtomicBoolean	GISTERED U	+setPrivateObjectiveCard(PrivateObjectiveCard): vi +getPrivateObjectiveCard(): PrivateObjectiveCard	oid ERED	~incrementNoOfWins(): int ~getNoOfLose(): int		
~sendSelectedMatchHistory(String, String): void ~createUserStats(String): Map<>		-draftedDiceSet: AtomicBoolean -endGameDueToDisconnection: AtomicBoolean	GISTERED U	+getUser(): User +getUserObserver(): UservObserver	TERED (-incrementNoOfLose(): int		
~sendFinishedMatchesList(String): void ~loginUser(String, UserObserver): User		+GameManager(List <player>, int, Controller, ControllerTimer): void</player>	GISTERED U	+sendResponse(Response): void	TERED (-getMatchesPlayed(): List <string> +attachUserObserver(UserObserver): void</string>		
-logoutUser(String): void -createMatch(String): void		-setUpGameManager(): void -setUpPatternCards(): void	GISTERED U	-notifyDraft(): void +getFavourTokens(): int	TERED U	+getUserObserver(): UserObserver -setActive(boolean): void		
-loginPrexistentPlayer(String): void -broadcastUsersConnected(String): void		-setUpToolCards(): void -setUpPublicObjectiveCards(): void		+decreaseFavorTokens(int): void	LINED	~getActiveTime(): long ~getFormattedTime(): String		
~writeUsersStatsToFile(): void		-setUpPrivateObjectiveCards(): void	GISTERED U	/	LERED (+toString(): String		
~createAvailableMatchesList(): List <doublestring> ~loginUserToController(String, String): void</doublestring>		-chooseToolCards(): List <toolcard> -choosePublicObjectiveCards(): List<publicobjectivecards></publicobjectivecards></toolcard>	GISTERED /U		ISTERED (UNREGIST		
-sendBundleData(String): void -deactivateUser(String): void		+pickPatternCards(): HashMap +getNoOfCurrentRound(): int	GISTERFO U	Round	ISTERED I	UNREGIST		
dedelvateoser(onnig). Void		+getRoundTrack(): List <list<dice>> -deleteMatch(): void</list<dice>	GISTE/ED U	-player: Player	CTEDE	UNDECICLEBED UNDECICLEBED		
VED OWKE TO LEKED		-checkUserConnection(Set <player>): void</player>	GISTIFKED U	-hasMadeAMove: AtomicBoolean -gameManager: GameManager	STERELL			
RED UNREGISTERED		-closeThreads(): void -listenForPlayerDisconnection(): void	GISTERED U	-playerEndedTurn: AtomicBoolean	ISTERED Y	UNREGISTERED UNREGISTERED		
RED UNREGISTERED		~getPlayerList(): List <player> +setPatternCardForPlayer(String, PatternCard): void</player>	STERED U	-blockedTurnPlayers: List <player> -noOfMoves: int</player>	STERED U	UNREGISTERED UNREGISTERED		
PED LIMPERISTERED		+getDraftedDice(): List <dice> +getTurnInRound(): int</dice>	CISTERED III	-avoidEndTurnNotification: AtomicBoolean +Round(GameManager)	ISTERED I	INPEGISTERED LINERGISTERED		
NED UNIXEDIOTERED		+waitForEveryPatternCard(Map <string, list<player="">>): void</string,>	DISTERED U	~startForPlayer(Player): void	OTENED	ONKEDIOTEKED ONKEDIOTEKED	IDEOLOTEDED	
RED UNREGISTERED	UNREGIST	+randomizePatternCards(Map <string, list<player="">>): void -setBoardAndStartMatch(): void</string,>	GISTERED U	+run(): voidendTurnNotification(): void	ISTERED ({abstract} PatternCard	IREGISTERED	
SagradaGame	UNREGIST	-waitForAck(): void +draftDiceFromBoard(): void	GISTERED U	-hasMadeAMove(): void -waitForMove(): void	ISTERED (-difficulty: int #grid: List <list<box>></list<box>	IREGISTERED	
-sagradaGameSingleton: SagradaGame -stop: boolean	INREGIST	-waitForDiceAck(): void +placeDiceForPlayer(Dice, int, int); void	GISTERED U		STERED I	+getDifficulty(): int	IREGISTERED	
-maxTurnSeconds: int -maxJoinMatchSeconds: int	INDECIST	+endTurn(String): void +stopTurn(): void	CISTEDED II	+hasPlayerEndedTurn(): AtomicBoolean -makeMove(Dice, int, int): void	ISTEDED I	-fillGrid(): void +getNoOfEmptyBoxes(): int	IDECISTEDED	
-userBroadcaster: UserBroadcaster	DINKEGIST	-resetAck(): void	DISTERED U	-makeMove(ToolCard): void +toolCardMoveDone(): void	OTENED (+computeAvailablePositionsDraftedDice(List <dice>): Map<string,boolean[][]: +computeAvailablePositionsLathekin(): Map<string,boolean[][]></string,boolean[][]></string,boolean[][]: </dice>	INCOISTENED	
+removeMatch(Controller): void +get(): SagradaGame	UNREGIST	+receiveAck(): void -notifyDraftToPlayer(Player): void	GISTERED U	+getCurrentPlayer(): Player +avoidEndTurnNotification(boolean): void	ISTERED (+computeAvailablePositionsTapWheel(Dice, boolean): Map<> -computePosition(Dice, boolean, boolean, boolean, boolean, int, int): Map<>	IREGISTERED	
-checkUserDisconnection(): void -rmiUsersListener(): void	UNREGIST	-startMatch(): void -writeHistoryToFile(List <movestatus>): void</movestatus>	GISTERED U	NF	STERED U	-boxToSwitchCheck(Dice, boolean, boolean, boolean[][], int, int, int): boolean	IREGISTERED	
-setMaxTurnSeconds(int): void -setMaxJoinMatchSeconds(int): void	INREGIST	-notifyWinner(): void -assignPointsToPlayers(): void	GISTERED II			-singleBoxCheck(Dice, boolean, boolean, boolean, boolean[][], int, int): boolean -checkAround(Dice, int, int, boolean[]): boolean	IREGISTERED	
-getConnectedUsers(); int	INDECIST	-evaluateBasicPoints(): Set <player></player>	NOTEBED II			-checkAroundSwitchBox(Dice, int, int, int, int, boolean[][]): boolean -hasDicesAround(int, int): boolean	IDECISTEDED	
-sendSelectedMatchHistory(String, String): void -createUserStats(String): Map<>	DINKEGIST	-evaluateBasicPrivateObjectiveCardPoints(): Set<player></player>-evaluateBasicFavourTokenPoints(): Set<player></player>	GILLERED OF	NREGISTERED UNKEG	ISTERED (-sameDiceColor(int, int, Dice): boolean -sameDiceValue(int, int, Dice): boolean	IKEGISTEKED	
-sendFinishedMatchesList(String): void -loginUser(String, UserObserver): User	UNREGIST	-evaluateWinner(): Player -startRound(): void	GISTERED U	-publicObjectiveCards: PublicObjectiveCards	KED (-sameGridColor(boolean, int, int): boolean	IREGISTERED	
-logoutUser(String): void -createMatch(String): void	UNREGIST	+executeTurn(int, String): void ~getToolCardLock(): AtomicBoolean	GISTERED &	-toolCards: List <toolcard></toolcard>	RED (-sameGridValue(boolean, int int, dice) +noDice(int, int): boolean	IREGISTERED	
-loginPrexistentPlayer(String): void -broadcastUsersConnected(String): void	INREGIST	-getControllerTimer(): ControllerTimer -shiftPlayerList(): void	GISTERED U	-diceBag: List <dice> -draftedDice: List<dice></dice></dice>	RED I	-isBlank(int, int): boolean -noDice(int, int): boolean	IREGISTERED	
~writeUsersStatsToFile(): void	INDECICE	-notifyUpdatedRoundTrack(): void	CICTERED II	+Board(List <publicobjectivecard>, List<toolci< th=""><th>ard>)</th><th>-isGridEmpty(): boolean -getNoOfDice(): int</th><th>IDECISTEDED</th><th></th></toolci<></publicobjectivecard>	ard>)	-isGridEmpty(): boolean -getNoOfDice(): int	IDECISTEDED	
~createAvailableMatchesList(): List <doublestring> ~loginUserToController(String, String): void</doublestring>	DINKEGIST	-waitEndTurn(): void -sendAvailablePositions(Player): Map <string, boolean[[[]=""></string,>	GISTERED U	-setupDiceBag(): void #draftDice(int): List <dice></dice>	LED (A	REGISTERED	
-sendBundleData(String): void -deactivateUser(String): void	UNREGIST	~makeMove(Dice, int, int): boolean +useToolCard(String): void	GISTERED U	~setDraftedDice(List <dice>): void ~draftOneDice(): Dice</dice>	RED (UNREGISTERED UNREGISTERED		
RED UNREGISTERED	UNREGIST	+addMoveToHistoryAndNotify(MoveStatus): void +getCurrentRound(): Round	GISTERED U	~addDiceToBag(Dice): void ~getDraftedDice(): List <dice></dice>	RED (UNREGISTERED UNREGISTERED	VNREGISTERED	
		-placeDiceToolCard(Dice, int, int): void +avoidToolCardUse(): void	GISTERED III	+getPublicObjectiveCards(): List <publicobjecti< th=""><th>iveCards></th><th>UNREGISTERED_UNREGISTERED</th><th>LINREGISTERED</th><th></th></publicobjecti<>	iveCards>	UNREGISTERED_UNREGISTERED	LINREGISTERED	
DED UNDECLOTEDED	HADECICT	-wakeUpToolCardThread(): void	CICTEDED	ND	JED I	Aurorae Magnificus	others	
Dice	REGIST	+glazingHammerResponse(): void +grozingPliersMove(Dice, Boolean): void	DISTERED U	INICE	IED !	JNREGISTERE -json: String ERED	KED	
-faceUpValue: int -diceColor: Color	REGIST	+grozingPliersResponse(): void +fluxBrushMove(Dice): void	GISTERED U	NKt	KED (UNREGISTERE	RED	
RF +Attribute1	REGIST	+fluxBrushMove(Dice, int, int): void +fluxBrushMove()	GISTERED U			UNREGISTERE D UNKEGIS ERED "	UNREGISTERED	
+Dice(Color) +Dice(int. Color)	REGIST	+fluxBrushResponse(): void	GISTERED U					
+roll(): void +getFaceUpValue(): int	REGIST	+fluxRemoverMove(Dice): void +fluxRemoverMove(Dice, int): void	GISTERED U					
+setFaceUpValue(int): void		+fluxRemoverMove(Dice, int, int): void -getSelectedToolCard(String): ToolCard +fluxRemoverMove(): void						
	REGIST	+fluxRemoverMove(): void +fluxRemoverResponse(): void	GISTERED U					
-setOppositeFace(): void +getDiceColor(): Dice	REGIST	+grindingStoneMove(Dice): void +grindingStoneResponse(): void	GISTERED U					
	REGIST	+copperFoilBurnisherMove(Tuple, Tuple): void	GISTERED U					
		+copperFoilBurnisherResponse(): void +corkBackedStraightedgeMove(Dice, int, int): void						
	REGIST	+corkBackedStraightedgeResponse(): void +lensCutterMove(int, String, String): void						
		+lensCutterResponse(): void +eglomiseBrushMove(Tuple, Tuple): void	GISTERED U					
		+eglomiseBrushResponse(): void	GISTERED U					
		+lathekinMove(Tuple, Tuple, boolean): void +lathekinResponse(): void						
		-copyPatternCard(): List <list<box>> +runningPliersMove(Dice, int, int): void</list<box>						
		+runningPilersResponse(): void +getDoubleMove(): boolean	GISTERED U					
		+setDoubleMove(boolean): void	GISTERED U					
		+tapWheelMove(Dice, int, Tuple, Tuple, boolean): void -tapWheelResponse(Map <string, boolean][]="">, int): void</string,>	GISTERFD II					
		EDED LINDEGISTEDED LINE	REGISTERED U					