GameController

- connectedPlayers: int
- isFullLock: Object
- users: Map<String, User>
- actionNeededLock: Object
- expectedPlayers: int
- gameModel: GamePlatform
- round: int
- savePath: String
- activeUser: String
- subPhase: GameControllerState
- server: Server
- initialized: boolean
- id: String
- phase: Phase
- ended: boolean

+ removeUser(User): void
+ checkProfessor(String, String): void
+ run(): void
+ isFull(): boolean
+ saveGame(): void
+ getSavePath(): String
+ setBan(int): void
+ returnStudents(HouseColor): void
+ getRound(): int
+ isEnded(): boolean
+ getGameModel(): GamePlatform
- writeFile(JsonObject): void
+ playAssistantCard(String, int): void
+ moveStudent(JsonObject): void
+ endGame(): void
+ getUser(String): User
+ getUsernames(): List<String>
+ getActiveUser(): String
- moveStudentFrom(JsonObject): void
- checkForWinners(): List<Player>
+ getGameId(): String
+ chooseCloud(JsonObject): void
+ motherNatureAction(Island): void
- actionPhase(): void
- setSubPhase(GameControllerState): void
+ paySpecialCharacter(JsonObject): void
+ getExpectedPlayers(): int
+ getPhase(): Phase
+ getSubPhase(): GameControllerState
- planningPhase(): void
+ moveMotherNature(int, boolean): void
+ addUser(String, User): void
+ checkStartCondition(): void
- moveStudentTo(JsonObject): void
+ notifyUsers(JsonObject): void
+ getUsers(): List<User>
+ setIgnoredColor(HouseColor): void

User

- inputStream: BufferedReader
- connected: boolean
- connectedLock: Object
- logged: boolean
- username: String
- outputStream: PrintWriter
- server: Server
- ping: Ping
- gameController: GameController

- isLogged(): boolean
- manageGameCommand(JsonObject): void
+ run(): void
- manageEnterGame(JsonObject): void
+ getUsername(): String
+ getGameController(): GameController
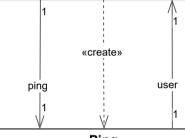- manageLogin(JsonObject): void
- removeFromGame(): void
+ disconnected(): void
- setLogged(boolean): void
+ sendMessage(JsonObject): void
- manageCommand(JsonObject): void
- getCommand(): JsonObject

1 ——— users ——— *

1 ——— gameController ——— 1

Ping

- inWaitingRoom: boolean
- user: User
- lock: Object
- stop: boolean

+ stopPing(): void
+ setInWaitingRoom(boolean): void
+ run(): void
+ isInWaitingRoom(): boolean

ping 1 / «create» / user 1

Matchmaking

~ gameCreation(JsonObject, Server): String
~ enterGame(String, Server): GameController
~ login(GameController, String, User): boolean