Server

GameData
- ○ fallingPieces: Map[PlayerId, GamePiece]
- ○ progressState: int
- ○ timer: double
- ○ gameBoard: int[]
- ○ playerData: Player[]
- ○ gamePhase: GamePhase

Object broadcasted from server to client

SocketIOService
- ● generateRoom(): json
- ● connect(): void
- ● joinRoom(): json
- ● getChatHistory(roomCode): json
- ● sendChatMessage(roomCode,userID, msg): json
- ● broadcastNewMessage(userID, msg): json
- ● startGame(room): json
- ● changeSpeed(room, speed): void
- ● changeNickName(room, pID, nickName) : void
- ● changeColor(room, pID, color) : void
- ● getGameData(room): json
- ● broadcastGameData(): json
- ● applyGameInput(room, userID): void
- ● tryStartVoting(room, playerID): void
- ● captureVote(room, playerID, targetPlayerID): void

‹E› GamePhase
- STARTING
- PLAYING
- VOTING
- PAUSED
- FINISHED

‹E› PlayerType
- IMPOSTER
- CIVILIAN
- EXILED

uses 0..*

RoomSession
- ○ isFull: bool
- ○ code: string
- ● startGame(): void
- ● addPlayer(): void
- ● generateRoomCode(): void
- ● getChatHistory(): json

‹E› GamePieceType
- O
- J
- L
- T
- S
- Z
- I

contains 0..1

contains 0..1

contains 0..1

ChatSession
- ○ chatHistory: string[]
- ● addChatMessage(userID, msg): void
- ● getChatHistory(): void

WebRTCSession
- ● connectToHost(): void

GameSession
- ○ players: Player[]
- ○ gameState: GameState
- ○ phase: GamePhase
- ● getFallingPieces(): Map(PlayerId, GamePiece)
- ● isValid(GamePiece): bool
- ● run(): void
- ● updateFrame(): void
- ● startVoting(): void
- ● sabotage(): void
- ● sendGameData(): void
- ● getGameData(): void
- ● generateRandomValidPiece(): GamePiece
- ● inputEvent(PlayerId, eventType): void
- ● resumeGame(): void
- ● notifyChange(): void
- ● applyGameInput(userID, direction): void

main game loop

updates the game state given time passed

updates the game state given player input

contains 0..1

owns 0..5

owns 1

VoteSession
- ○ voteCount: object
- ○ isTie: bool
- ○ timer: double
- ● onTimerComplete(): void
- ● onVoteComplete(results: object): void
- ● captureVote(playerID, targetPlayerID): void

Player
- ○ id: int
- ○ type: PlayerType
- ○ playerName: string
- ○ playerColor: string
- ○ isHost: bool
- ○ hasEmergency: bool
- ○ isConnect: bool
- ○ currentPiece: GamePiece
- ○ nextPiece: GamePiece
- ● changePlayerType(PlayerType): void

owns 0..2

GameState
- ○ progressState: int
- ○ timer: double
- ○ speed: double
- ○ gameBoard: int[][]
- ● checkRowComplete(): void
- ● addFixedPiece(GamePiece, PlayerId): bool
- ● deleteRow(): void
- ● isValid(GamePiece): bool
- ● changeProgress(): void

GamePiece
- ○ type: GamePieceType
- ○ orientation: int
- ○ ofx: int
- ○ ofy: int
- ● getCells() : arr[[]]

Client

HomePage
- ● onCreateButtonClicked() : void
- ● onJoinButtonClicked() : void
- ● onHelpButtonClicked() : void
- ● requestJoinRoom(code): void

navigate

navigate

navigate

RoomPage
- ● onHelpButtonClicked() : void
- ● displayGameViewComponent() : void

navigate *displays* *displays* *displays* *displays*

HelpPage
- ○ tetrisRulesText: string
- ○ helpChatBoxText: string
- ○ helpVoiceChatText: string
- ○ helpSettingsText: string
- ● getHelpInfo(): string

ChatBoxComponent
- ● onSendButtonClicked() : void
- ● onMessageRecieved(): void
- ● render(): void

GameSettingsComponent
- ○ isHost: bool
- ● onSpeedChanged() : void
- ● onStartGamePressed() : void
- ● render(): void

GameViewComponent
- ● onKeyPressed(direction) : void
- ● sendInputEvent(playerID, eventType): void
- ● showVoteComponent(): void
- ● hideVoteComponent(): void
- ● disableInputs(): void
- ● enableInputs(): void
- ● render(): void

PlayerInfoComponent
- ● render(): void

displays *displays* *displays* *displays* *displays*

HostSettingsComponent
- ● disableForm(): void
- ● onSpeedChanged() : void
- ● onStartGamePressed() : void
- ● render(): void

PlayerSettingsComponent
- ● onNickNameChanged() : void
- ● onColorChanged() : void
- ● render(): void

GameCanvasComponent
- ● onGameDataReceived(gameData: Object): void
- ● draw(): void
- ● disableKeyInput(): void
- ● render(): void
- ● onKeyPressed(direction): void

GameButtonsComponent
- ○ isImposter: bool
- ● onEmergencyButtonClicked() : void
- ● onSabotageButtonClicked() : void
- ● disableButtons(): void
- ● showSabotageButton(): void
- ● render(): void

VoteComponent
- ○ remainingTime: int
- ● onPlayerSelected(): void
- ● render(): void