UML For Server Program

Application

- Game: StateOfGame
- Conn: NetworkConnection
- inOutString: String
- clientListString: String
- Gui buttons and text areas
- + createServerGUI()
- + main(args: String[])
- + start(primaryStage: Stage)
- + createServer(): Server
- + processInput(data: String)
- + connectuser(id: String, username: String)
- + addToClientList(clientNum: String, username: String)
- + addToInOutString(line: String)
- + sendStats()
- + processMoves()
- + sendTextForMoves(move: String)
- + processDefendMoves()
- + processItemMoves()
- + processAttackMoves()
- + sendHealth()
- + updateClientList()
- + updateClientNum()

StateOfGame

- p1: Player
- p2: Player
- p3: Player
- p4: Player
- currPlayer: String

StateOfGame()

- + rollWinner(): String
- + nextPlayer(playedlast: String): String
- + setUsername(id: String, username: String)
- + addRoll(id: String, roll: String)
- + everyoneRolled(): boolean
- + addChoice(id: String, itemType: String, level: String)
- + addMoveToPlayer(id: String, move: String)
- + allAliveMadeMove(): boolean
- + processAttack(attacker: String, attackee: String)
- + dealDmg(atk: int, attackee: String)
- + getAttackeeDefending(id: String): boolean
- + resetDefending(id: String)
- + processItem(id: String, item: String)
- + pickingOver(): int
- + calcStats()
- + weaponStr(level: String): int
- + helmetDef(level: String): int
- + chestDef(level: String): int
- + legsDef(level: String): int
- + calcDef(hLevel: String, cLevel: String, lLevel: String): int
- + getP1(): Player
- + getP2(): Player
- + getP3(): Player
- + getP4(): Player
- + getCurrPlayer(): String
- + setCurrPlayer(id: String)
- + getUsername(id: String): String
- + getItemName(id: String): String
- + checkForWinner(): int

Player

- username: String
- id: String
- roll: int
- weapon: String
- helmet: String
- chest: String
- legs: String
- item: String
- health: int
- atk: int
- def: int
- move: String
- defending: bool
- usedItem: boolean
- # Player()
- + getters and setters for all data members

