UML Class Diagram "Moving Maze" - Client

<<enumeration>> TileType

Tile
- -openTop : boolean
- -openRight : boolean
- -openBottom : boolean
- -openLeft : boolean
- -item : Item
- -type : TileType
- -rotationInDegrees : int
- +rotateClockwise() : void
- +rotateCounterClockwise() : void
- +getBinaryString() : string
- +setByBinaryString(binaryString : string) : void
- +isOpenTop() : boolean
- +isOpenRight() : boolean
- +isOpenBottom() : boolean
- +isOpenLeft() : boolean
- +getItem() : Item
- +setItem(item : Item) : Item
- +getTileType() : TileType

TileFactory
- +getLTile() : Tile
- +getITile() : Tile
- +getTTile() : Tile
- +getTileByType(type : TileType) : Tile

GameBoardFactory
- +getStandardGameBoard() : GameBoard
- +getEasyGameBoard() : GameBoard

Item
- -name
- -pos : Position

Position
- -x : int
- -y : int
- -MAX_X : int
- -MAX_Y : int
- +getX() : int
- +getY() : int
- +setX(x : int) : void
- +setY(y : int) : int
- +setPos(x : int, y : int) : void

Card
- -item : Item

GameBoard
- -mapSize : int
- -tileMap : Tile [ ][ ]

Player
- -name : string
- -pos : Position
- -cardsToFind : Card [ ]
- -cardsFound : Card [ ]
- -currentCard : Card
- -role : PlayerRole
- +move(dX : int, dY : int) : Position

<<enumeration>> PlayerRole

GameStateHandler
- -sessionCode : string
- -players : Player [ ]
- -currentPlayerOnTurn : Player
- -board : GameBoard
- -gameState : GameState
- -chat : ChatMessage [ ]
- +submitUpdatedGameState() : boolean
- +receiveUpdatedGameState()
- +initGame()
- +finishTurn()

<<enumeration>> GameState

ChatMessage (a)
- -sender : string
- -message : string
- -id : int

NetworkService
- +sendUpdateToServer()
- +receiveUpdateFromServer()

Turn
- -sequenceNumber : int
- -player : Player
- -executedActions : TurnAction [ ]
- -availableActions : TurnAction [ ]
- +executeAction(action : TurnAction) : boolean

<<Interface>> TurnAction
- +execute() : void
- +validate() : boolean

DrawNewCard

MovePlayer

InsertTile
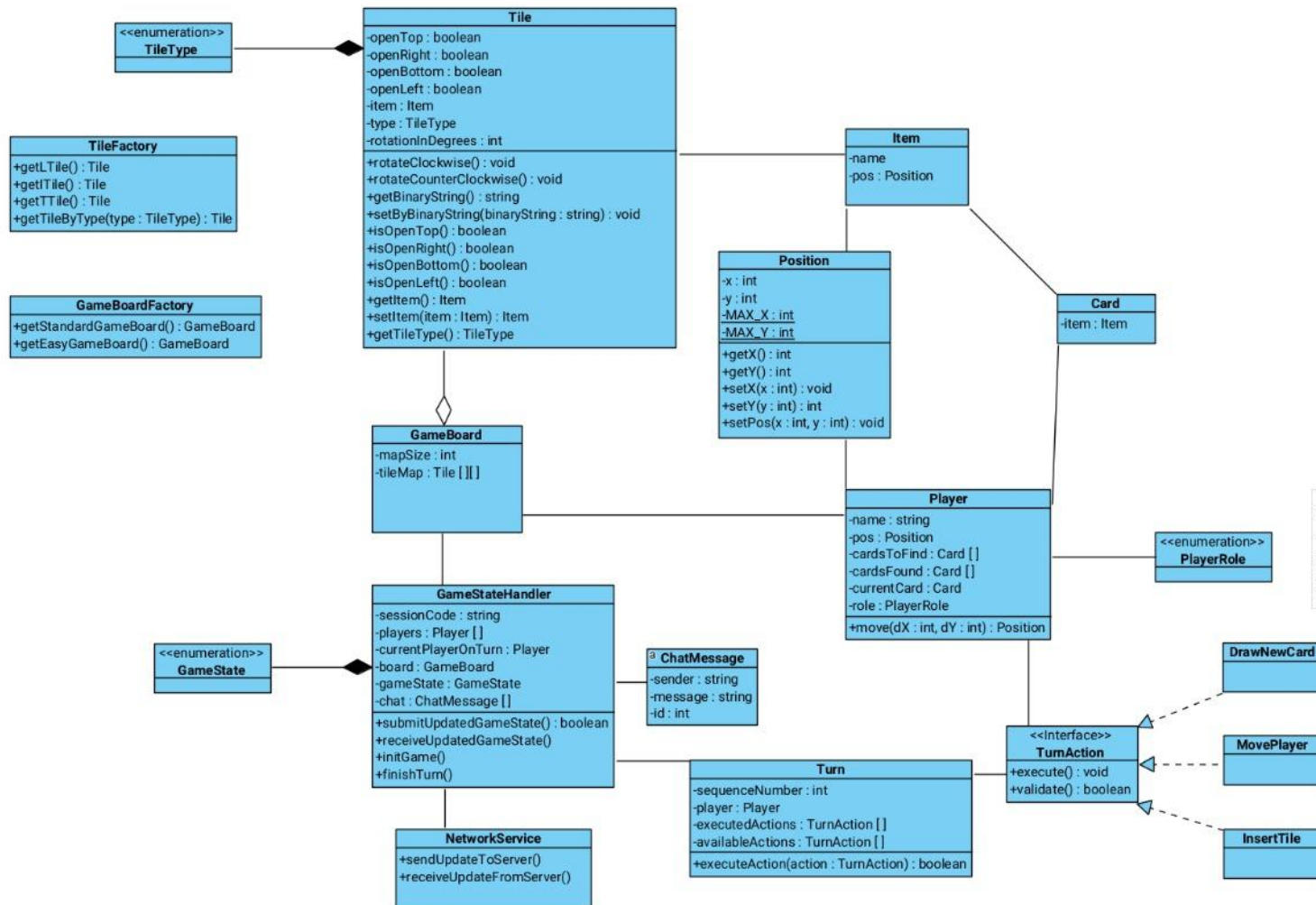
Jeder Player kann, wenn er am Zug ist, eine TurnAction ausführen. TurnActions sind jegliche in einem Zug ausführbare Aktionen. TurnActions validieren sich selbst und updaten den GameState. Der GameStateHandler vergibt die Turns und beeinflusst so die View des Players. (wenn am Zug -> zeig Menü zum verschieben an, erlaube Bewegung der Spielfigur usw...)