

```

classDiagram
    class Teacher {
        color: String
        id: int
        +moveTeacherToDashboard()
    }
    class Student {
        color: String
        id: int
    }
    class StudentsBag {
        numberOFStudents: int
        +studentsBag(int number)
    }
    class Dashboard {
        id: int
        position_x: int
        position_y: int
        position_x_status: bool
        position_y_status: bool
        towerNumber: int
        studentList: Student
        teacherList: Teacher
        coinPosition_x: int
        +Dashboard(studentList: Student, int coinPosition_x): void
        +giveCoinToPlayer(wallet: Wallet, position: Dashboard): int
        +moveTowerToIsland(tower: Tower, island: Island): void
    }
    class Tower {
        id: int
        color: String
        positionInGame: bool
        +Tower(int id, String color, bool positionInGame)
    }
    class GameModel {
        id: int
        gameStatus: bool
        round: int
        numberOFPlayer: int
        orderOfGame: int
        numberOFIsland: int
        +modifyPlayer(g: Player): void
        +newRound(): int
        +checkIslandGroup(island1: Island, island2: Island): bool
        +buildNewGame(n: numberOFPlayer)
        +moveTower(Island, Tower)
        +checkOrderOfGame()
    }
    class Player {
        id: int
        order: int
        moveOFMadrenatura: int
        statusPlayer: bool
        numberOFTower: int
        towerColor: String
        wallet: int
        winner: bool
        +Player(int id, bool statusPlayer, int order): void
        +chooseDeck(): Integer deck Deck(): void
        +playAssistentCard(int Integer card, AssistentCard): void
        +moveStudentsToIsland(island: Island): void
        +moveStudentsToDashboard(Dashboard d: Dashboard): void
        +playSpecialCard(SpecialCard, Wallet w): void
        +moveMadrenatura(move: AssistentCard): void
        +chooseNumberOFPlayer(int number, g: GameModel): void
        +chooseCloudCard(CloudCard c)
    }
    class Wallet {
        numberOFCoins
        +wallet(int numberOFCoins)
        +giveCoin( Player): int
    }
    class Card {
        id: int
        hasChosen: bool
    }
    class Deck {
        id: int
        hasChosen: bool
    }
    class SpecialCard {
        cost: int
        effect()
    }
    class AssistentCard {
        orderGame: int
        moveMadrenatura: int
        hasPlayer: bool
        +giveOrder(order: Player): int
        +calculateAvailableMove(move: Player): int
    }
    class CloudCard {
        id: int
        side: bool
        studentList: Student
        +CloudCard(int id, bool side, studentList: Student)
        +moveStudentsToDashboard(Dashboard): void
        +randomExtraction(StudentBag)
    }
    class Island {
        id: int
        isConquered: bool
        idGroup: int
        towerStatus: bool
        towerColor: String
        studentList: Student
        +getTowerColor(): String
        +getStudentList()
        +Island(int id, bool isConquered, studentList: Student)
        +calculateInfluence(Student studentList)
    }
    class Table {
        +communicate
    }
    class Visualizer {
        +displayWinner(): void
        +displayOrderOfPlayer(): void
    }
    class GameController {
        roundNumber: int
        studentNumber: int
        towerPlayer: int
        +checkEndOfGame(status: GameModel, s: StudentsBag): bool
        +calculateWinner( Player): int
        +modifyOrderOfGame( Player, GameModel): void
        +connectAndIsland1_Island2( GameModel): void
    }
    class Server {
        playerId: int
        +startNewGame( GameModel): void
        +closeGame( GameModel): void
        +addNewPlayer( Player): void
        +removePlayer( Player): void
        +blockNumberOFPlayer( GameModel): void
        +closeServer()
    }
    class PlayersList {
        +Array<int> playerList
        +addPlayer()
        +getPlayerNumber()
    }

    Teacher --> Dashboard : +has
    Student --> Dashboard : +has
    StudentsBag --> Dashboard : +has
    Dashboard --> Tower : +has
    Dashboard --> GameModel : +has
    GameModel --> Player : +has
    Player --> Wallet : +has
    Player --> Card : +has
    Card --> Deck : +has
    Deck --> SpecialCard : +has
    Deck --> AssistentCard : +has
    AssistentCard --> AssistentCard : +has
    CloudCard --> CloudCard : +has
    Island --> Island : +has
    Table --> Table : +has
    Visualizer --> Visualizer : +has
    GameController --> GameController : +has
    Server --> Server : +has
    PlayersList --> PlayersList : +has

    Teacher --> Dashboard : +moveTeacherToDashboard()
    Dashboard --> Tower : +Tower(int id, String color, bool positionInGame)
    Dashboard --> GameModel : +Dashboard(studentList: Student, int coinPosition_x): void
    GameModel --> Player : +Player(int id, bool statusPlayer, int order): void
    Player --> Wallet : +wallet(int numberOFCoins)
    Player --> Card : +Card(int id, bool statusPlayer, int order): void
    Card --> Deck : +Deck(int id, bool statusPlayer, int order): void
    Deck --> SpecialCard : +SpecialCard(int id, bool statusPlayer, int order): void
    Deck --> AssistentCard : +AssistentCard(int id, bool statusPlayer, int order): void
    AssistentCard --> AssistentCard : +AssistentCard(int id, bool statusPlayer, int order): void
    CloudCard --> CloudCard : +CloudCard(int id, bool statusPlayer, int order): void
    Island --> Island : +Island(int id, bool statusPlayer, int order): void
    Table --> Table : +Table(int id, bool statusPlayer, int order): void
    Visualizer --> Visualizer : +Visualizer(int id, bool statusPlayer, int order): void
    GameController --> GameController : +GameController(int id, bool statusPlayer, int order): void
    Server --> Server : +Server(int id, bool statusPlayer, int order): void
    PlayersList --> PlayersList : +PlayersList(int id, bool statusPlayer, int order): void

```

