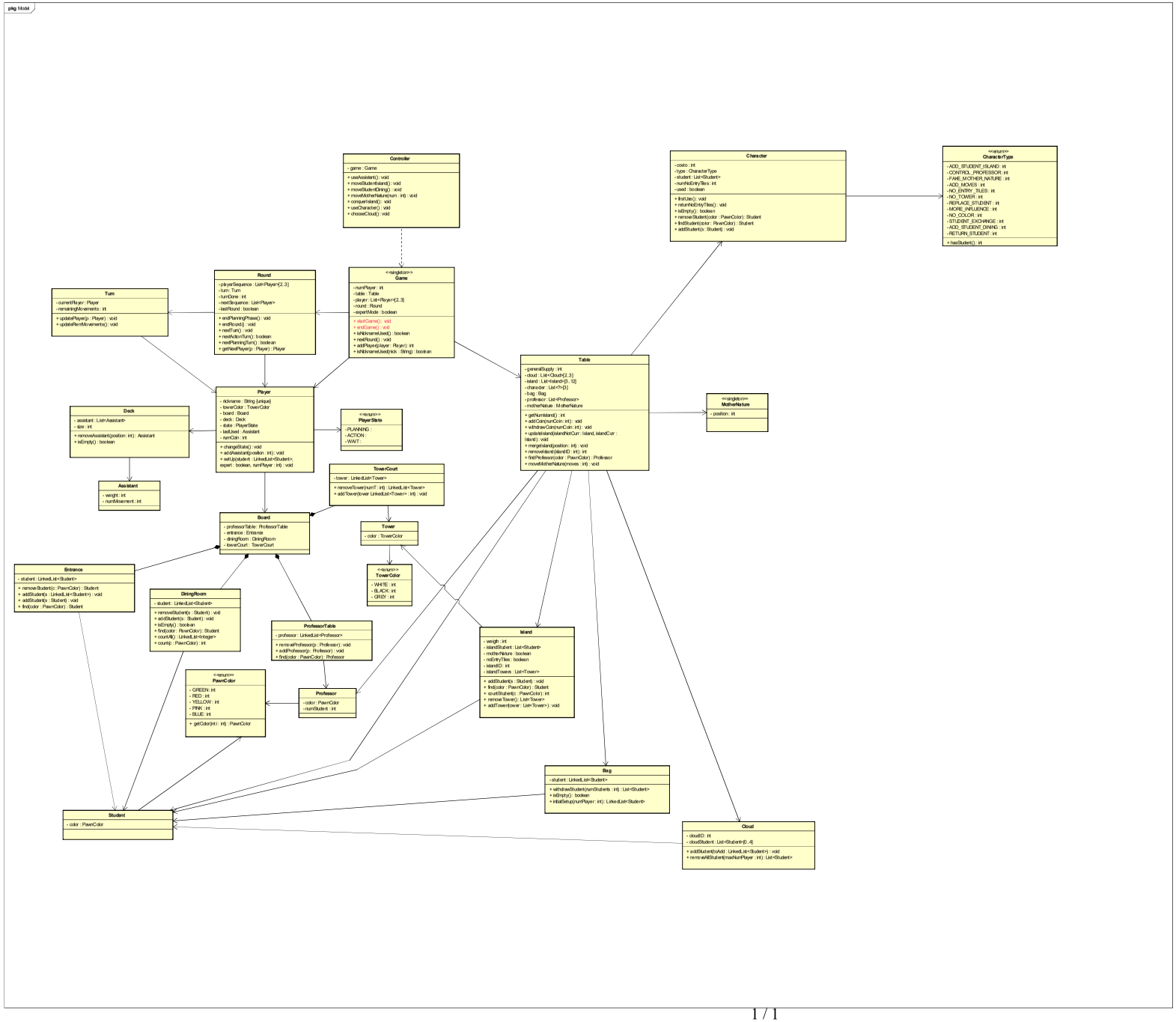
pkg Model
Controller
- game : Game
+ useAssistant() : void
+ moveStudentIsland() : void
+ moveStudentDining() : void
+ moveMotherNature(num : int) : void
+ conquerIsland() : void
+ useCharacter() : void
+ chooseCloud() : void
Character
- costo : int
- type : CharacterType
- student : List<Student>
- numNoEntryTiles : int
- used : boolean
+ firstUse() : void
+ returnNoEntryTiles() : void
+ isEmpty() : boolean
+ removeStudent(color : PawnColor) : Student
+ findStudent(color : PawnColor) : Student
+ addStudent(s : Student) : void
<<enum>>
CharacterType
- ADD_STUDENT_ISLAND : int
- CONTROL_PROFESSOR : int
- FAME_MOTHER_NATURE : int
- ADD_MOVES : int
- NO_ENTRY_TILES : int
- NO_TOWER : int
- REPLACE_STUDENT : int
- MORE_INFLUENCE : int
- NO_COLOR : int
- STUDENT_EXCHANGE : int
- ADD_STUDENT_DINING : int
- RETURN_STUDENT : int
+ hasStudent() : int
Turn
- currentPlayer : Player
- remainingMovements : int
+ updatePlayer(p : Player) : void
+ updateRemMovements() : void
Round
- playerSequence : List<Player>[2..3]
- turn : Turn
- turnDone : int
- nextSequence : List<Player>
- lastRound : boolean
+ endPlanningPhase() : void
+ endRound() : void
+ nextTurn() : void
+ nextActionTurn() : boolean
+ nextPlanningTurn() : boolean
+ getNextPlayer(p : Player) : Player
<<singleton>>
Game
- numPlayer : int
- table : Table
- player : List<Player>[2..3]
- round : Round
- expertMode : boolean
+ startGame() : void
+ endGame() : void
+ isNicknameUsed() : boolean
+ nextRound() : void
+ addPlayer(player : Player) : int
+ isNicknameUsed(nick : String) : boolean
Table
- generalSupply : int
- cloud : List<Cloud>[2..3]
- island : List<island>[3..12]
- character : List<?>[3]
- bag : Bag
- professor : List<Professor>
- motherNature : MotherNature
+ getNumIsland() : int
+ addCoin(numCoin : int) : void
+ withdrawCoin(numCoin : int) : void
+ updateIsland(islandNotCurr : Island, islandCurr : Island) : void
+ mergeIsland(position : int) : void
+ removeIsland(idandID : int) : int
+ findProfessor(color : PawnColor) : Professor
+ moveMotherNature(moves : int) : void
<<singleton>>
MotherNature
- position : int
Player
- nickname : String (unique)
- towerColor : TowerColor
- board : Board
- deck : Deck
- state : PlayerState
- lastUsed : Assistant
- numCoin : int
+ changeState() : void
+ addAssistant(position : int) : void
+ setUpStudent : LinkedList<Student>, expert : boolean, numPlayer : int) : void
<<enum>>
PlayerState
- PLANNING :
- ACTION :
- WAIT :
Deck
- assistant : List<Assistant>
- size : int
+ removeAssistant(position : int) : Assistant
+ isEmpty() : boolean
Assistant
- weight : int
- numMovement : int
TowerCourt
- tower : LinkedList<Tower>
+ removeTower(numT : int) : LinkedList<Tower>
+ addTower(tower LinkedList<Tower> : int) : void
Board
- professorTable : ProfessorTable
- entrance : Entrance
- diningRoom : DiningRoom
- towerCourt : TowerCourt
Tower
- color : TowerColor
<<enum>>
TowerColor
- WHITE : int
- BLACK : int
- GREY : int
Entrance
- student : LinkedList<Student>
+ removeStudent(c : PawnColor) : Student
+ addStudent(s : LinkedList<Student>) : void
+ addStudent(s : Student) : void
+ find(color : PawnColor) : Student
DiningRoom
- student : LinkedList<Student>
+ removeStudent(s : Student) : void
+ addStudent(s : Student) : void
+ isEmpty() : boolean
+ find(color : PawnColor) : Student
+ countM() : LinkedList<Integer>
+ countp : PawnColor) : int
ProfessorTable
- professor : LinkedList<Professor>
+ removeProfessor(p : Professor) : void
+ addProfessor(p : Professor) : void
+ find(color : PawnColor) : Professor
Island
- weigth : int
- islandStudent : List<Student>
- motherNature : boolean
- noEntryTiles : boolean
- islandID : int
- islandTowers : List<Tower>
+ addStudent(s : Student) : void
+ find(color : PawnColor) : Student
+ countStudent(c : PawnColor) : int
+ removeTower() : List<Tower>
+ addTower(tower : List<Tower>) : void
<<enum>>
PawnColor
- GREEN : int
- RED : int
- YELLOW : int
- PINK : int
- BLUE : int
+ getColor(int i : int) : PawnColor
Professor
- color : PawnColor
- numStudent : int
Bag
- student : LinkedList<Student>
+ withdrawStudent(numStudents : int) : List<Student>
+ isEmpty() : boolean
+ initialSetup(numPlayer : int) : LinkedList<Student>
Student
- color : PawnColor
Cloud
- cloudID : int
- cloudStudent : List<Student>[0..4]
+ addStudent(toAdd : LinkedList<Student>) : void
+ removeAllStudent(maxNumPlayer : int) : List<Student>