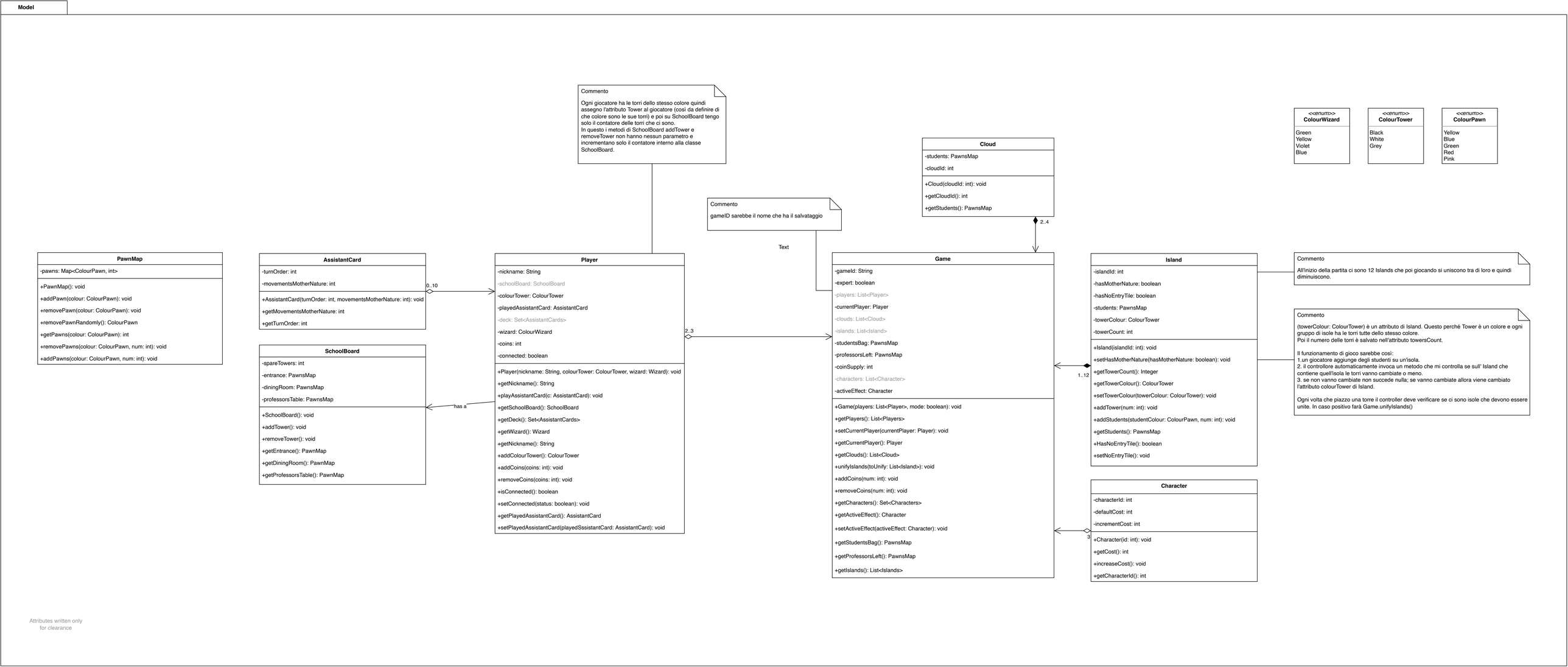
Model
Commento
Ogni giocatore ha le torri dello stesso colore quindi assegno l'attributo Tower al giocatore (così da definire di che colore sono le sue torri) e poi su SchoolBoard tengo solo il contatore delle torri che ci sono.
In questo i metodi di SchoolBoard addTower e removeTower non hanno nessun parametro e incrementano solo il contatore interno alla classe SchoolBoard.
Commento
gameID sarebbe il nome che ha il salvataggio
Text
Cloud
-students: PawnsMap
-cloudId: int
+Cloud(cloudId: int): void
+getCloudId(): int
+getStudents(): PawnsMap
2..4
<<enum>>
ColourWizard
Green
Yellow
Violet
Blue
<<enum>>
ColourTower
Black
White
Grey
<<enum>>
ColourPawn
Yellow
Blue
Green
Red
Pink
PawnMap
-pawns: Map<ColourPawn, int>
+PawnMap(): void
+addPawn(colour: ColourPawn): void
+removePawn(colour: ColourPawn): void
+removePawnRandomly(): ColourPawn
+getPawns(colour: ColourPawn): int
+removePawns(colour: ColourPawn, num: int): void
+addPawns(colour: ColourPawn, num: int): void
AssistantCard
-turnOrder: int
-movementsMotherNature: int
+AssistantCard(turnOrder: int, movementsMotherNature: int): void
+getMovementsMotherNature: int
+getTurnOrder: int
0..10
SchoolBoard
-spareTowers: int
-entrance: PawnsMap
-diningRoom: PawnsMap
-professorsTable: PawnsMap
+SchoolBoard(): void
+addTower(): void
+removeTower(): void
+getEntrance(): PawnMap
+getDiningRoom(): PawnMap
+getProfessorsTable(): PawnMap
has a
Player
-nickname: String
-schoolBoard: SchoolBoard
-colourTower: ColourTower
-playedAssistantCard: AssistantCard
-deck: Set<AssistantCards>
-wizard: ColourWizard
-coins: int
-connected: boolean
+Player(nickname: String, colourTower: ColourTower, wizard: Wizard): void
+getNickname(): String
+playAssistantCard(c: AssistantCard): void
+getSchoolBoard(): SchoolBoard
+getDeck(): Set<AssistantCards>
+getWizard(): Wizard
+getNickname(): String
+addColourTower(): ColourTower
+addCoins(coins: int): void
+removeCoins(coins: int): void
+isConnected(): boolean
+setConnected(status: boolean): void
+getPlayedAssistantCard(): AssistantCard
+setPlayedAssistantCard(playedSssistantCard: AssistantCard): void
2..3
Game
-gameId: String
-expert: boolean
-players: List<Player>
-currentPlayer: Player
-clouds: List<Cloud>
-islands: List<Island>
-studentsBag: PawnsMap
-professorsLeft: PawnsMap
-coinSupply: int
-characters: List<Character>
-activeEffect: Character
+Game(players: List<Player>, mode: boolean): void
+getPlayers(): List<Players>
+setCurrentPlayer(currentPlayer: Player): void
+getCurrentPlayer(): Player
+getClouds(): List<Cloud>
+unifyIslands(toUnify: List<Island>): void
+addCoins(num: int): void
+removeCoins(num: int): void
+getCharacters(): Set<Characters>
+getActiveEffect(): Character
+setActiveEffect(activeEffect: Character): void
+getStudentsBag(): PawnsMap
+getProfessorsLeft(): PawnsMap
+getIslands(): List<Islands>
1..12
Island
-islandId: int
-hasMotherNature: boolean
-hasNoEntryTile: boolean
-students: PawnsMap
-towerColour: ColourTower
-towerCount: int
+Island(islandId: int): void
+setHasMotherNature(hasMotherNature: boolean): void
+getTowerCount(): Integer
+getTowerColour(): ColourTower
+setTowerColour(towerColour: ColourTower): void
+addTower(num: int): void
+addStudents(studentColour: ColourPawn, num: int): void
+getStudents(): PawnsMap
+HasNoEntryTile(): boolean
+setNoEntryTile(): void
Commento
All'inizio della partita ci sono 12 Islands che poi giocando si uniscono tra di loro e quindi diminuiscono.
Commento
(towerColour: ColourTower) è un attributo di Island. Questo perché Tower è un colore e ogni gruppo di isole ha le torri tutte dello stesso colore.
Poi il numero delle torri è salvato nell'attributo towersCount.
Il funzionamento di gioco sarebbe così:
1.un giocatore aggiunge degli studenti su un'isola.
2. il controllore automaticamente invoca un metodo che mi controlla se sull' Island che contiene quell'isola le torri vanno cambiate o meno.
3. se non vanno cambiate non succede nulla; se vanno cambiate allora viene cambiato l'attributo colourTower di Island.
Ogni volta che piazzo una torre il controller deve verificare se ci sono isole che devono essere unite. In caso positivo farà Game.unifyIslands()
Character
-characterId: int
-defaultCost: int
-incrementCost: int
+Character(id: int): void
+getCost(): int
+increaseCost(): void
+getCharacterId(): int
3
Attributes written only for clearance