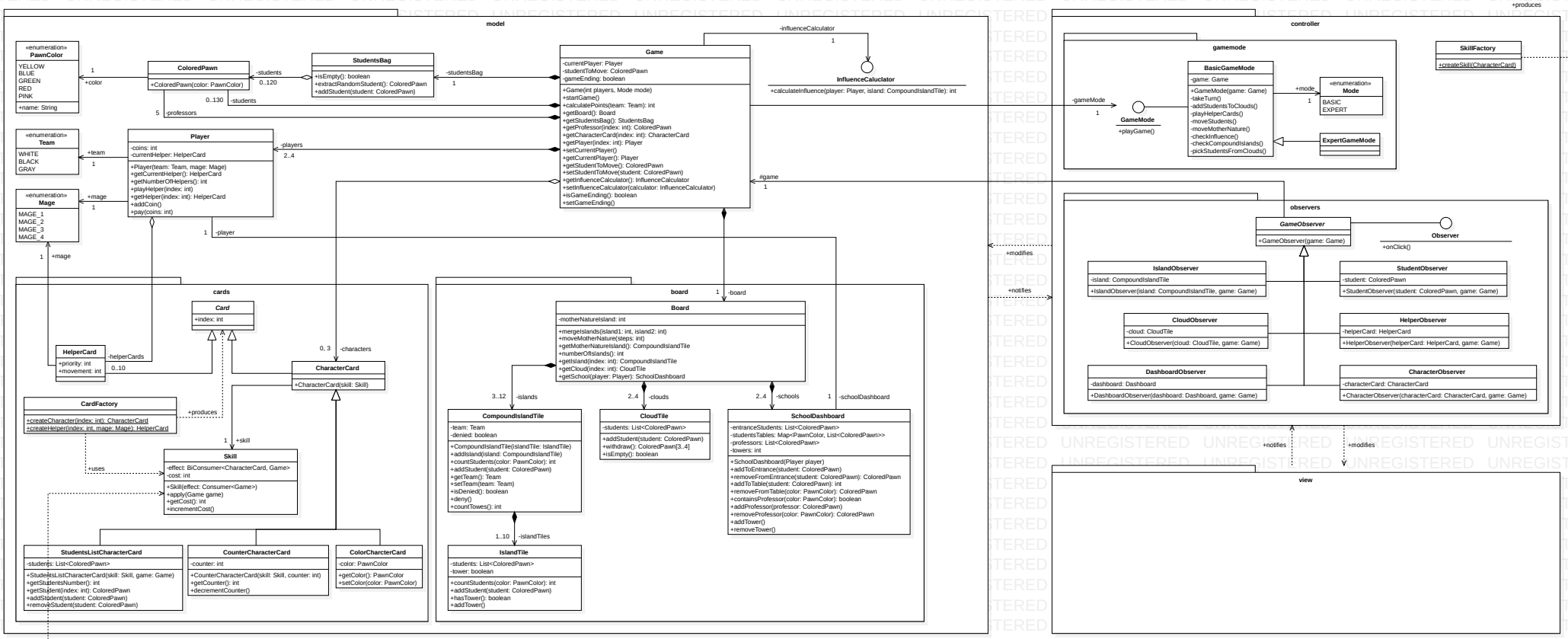
model
«enumeration»
PawnColor
YELLOW
BLUE
GREEN
RED
PINK
+name: String
ColoredPawn
+ColoredPawn(color: PawnColor)
StudentsBag
+isEmpty(): boolean
+extractRandomStudent(): ColoredPawn
+addStudent(student: ColoredPawn)
Game
-currentPlayer: Player
-studentToMove: ColoredPawn
-gameEnding: boolean
+Game(int players, Mode mode)
+startGame()
+calculatePoints(team: Team): int
+getBoard(): Board
+getStudentsBag(): StudentsBag
+getProfessor(index: int): ColoredPawn
+getCharacterCard(index: int): CharacterCard
+getPlayer(index: int): Player
+setCurrentPlayer()
+getCurrentPlayer(): Player
+getStudentToMove(): ColoredPawn
+setStudentToMove(student: ColoredPawn)
+setInfluenceCalculator(): InfluenceCalculator
+setInfluenceCalculator(calculator: InfluenceCalculator)
+isGameEnding(): boolean
+setGameEnding()
-influenceCalculator
InfluenceCaluclator
+calculateInfluence(player: Player, island: CompoundIslandTile): int
«enumeration»
Team
WHITE
BLACK
GRAY
Player
-coins: int
-currentHelper: HelperCard
+Player(team: Team, mage: Mage)
+getCurrentHelper(): HelperCard
+getNumberOfHelpers(): int
+playHelper(index: int)
+getHelper(index: int): HelperCard
+addCoin()
+pay(coins: int)
«enumeration»
Mage
MAGE_1
MAGE_2
MAGE_3
MAGE_4
cards
Card
+index: int
HelperCard
+priority: int
+movement: int
CharacterCard
+CharacterCard(skill: Skill)
CardFactory
+createCharacter(index: int): CharacterCard
+createHelper(index: int, mage: Mage): HelperCard
Skill
-effect: BiConsumer<CharacterCard, Game>
-cost: int
+Skill(effect: Consumer<Game>)
+apply(Game game)
+getCost(): int
+incrementCost()
StudentsListCharacterCard
-students: List<ColoredPawn>
+StudentsListCharacterCard(skill: Skill, game: Game)
+getStudentsNumber(): int
+getStudent(index: int): ColoredPawn
+addStudent(student: ColoredPawn)
+removeStudent(student: ColoredPawn)
CounterCharacterCard
-counter: int
+CounterCharacterCard(skill: Skill, counter: int)
+getCounter(): int
+decrementCounter()
ColorCharterCard
-color: PawnColor
+getColor(): PawnColor
+setColor(color: PawnColor)
board
Board
-motherNatureIsland: int
+mergeIslands(island1: int, island2: int)
+moveMotherNature(steps: int)
+getMotherNatureIsland(): CompoundIslandTile
+numberOfIslands(): int
+getIsland(index: int): CompoundIslandTile
+getCloud(index: int): CloudTile
+getSchool(player: Player): SchoolDashboard
CompoundIslandTile
-team: Team
-denied: boolean
+CompoundIslandTile(islandTile: IslandTile)
+addIsland(island: CompoundIslandTile)
+countStudents(color: PawnColor): int
+addStudent(student: ColoredPawn)
+getTeam(): Team
+setTeam(team: Team)
+isDenied(): boolean
+deny()
+countTowes(): int
CloudTile
-students: List<ColoredPawn>
+addStudent(student: ColoredPawn)
+withdraw(): ColoredPawn[3..4]
+isEmpty(): boolean
SchoolDashboard
-entranceStudents: List<ColoredPawn>
-studentsTables: Map<PawnColor, List<ColoredPawn>>
-professors: List<ColoredPawn>
-towers: int
+SchoolDashboard(Player player)
+addToEntrance(student: ColoredPawn)
+removeFromEntrance(student: ColoredPawn): ColoredPawn
+addToTable(student: ColoredPawn): int
+removeFromTable(color: PawnColor): ColoredPawn
+containsProfessor(color: PawnColor): boolean
+addProfessor(professor: ColoredPawn)
+removeProfessor(color: PawnColor): ColoredPawn
+addTower()
+removeTower()
IslandTile
-students: List<ColoredPawn>
-tower: boolean
+countStudents(color: PawnColor): int
+addStudent(student: ColoredPawn)
+hasTower(): boolean
+addTower()
controller
gamemode
BasicGameMode
-game: Game
+GameMode(game: Game)
-takeTurn()
-addStudentsToClouds()
-playHelperCards()
-moveStudents()
-moveMotherNature()
-checkInfluence()
-checkCompoundIslands()
-pickStudentsFromClouds()
«enumeration»
Mode
BASIC
EXPERT
ExpertGameMode
GameMode
+playGame()
SkillFactory
+createSkill(CharacterCard)
observers
GameObserver
+GameObserver(game: Game)
Observer
+onClick()
IslandObserver
-island: CompoundIslandTile
+IslandObserver(island: CompoundIslandTile, game: Game)
StudentObserver
-student: ColoredPawn
+StudentObserver(student: ColoredPawn, game: Game)
CloudObserver
-cloud: CloudTile
+CloudObserver(cloud: CloudTile, game: Game)
HelperObserver
-helperCard: HelperCard
+HelperObserver(helperCard: HelperCard, game: Game)
DashboardObserver
-dashboard: Dashboard
+DashboardObserver(dashboard: Dashboard, game: Game)
CharacterObserver
-characterCard: CharacterCard
+CharacterObserver(characterCard: CharacterCard, game: Game)
view
+produces
+modifies
+notifies
+uses
+skill
+mage
+team
+mage
-players
-students
-studentsBag
-professors
-gameMode
-game
-board
-characters
-helperCards
-player
-islands
-clouds
-schools
-schoolDashboard
-islandTiles
0..120
0..130
2..4
0..10
0, 3
3..12
1..10