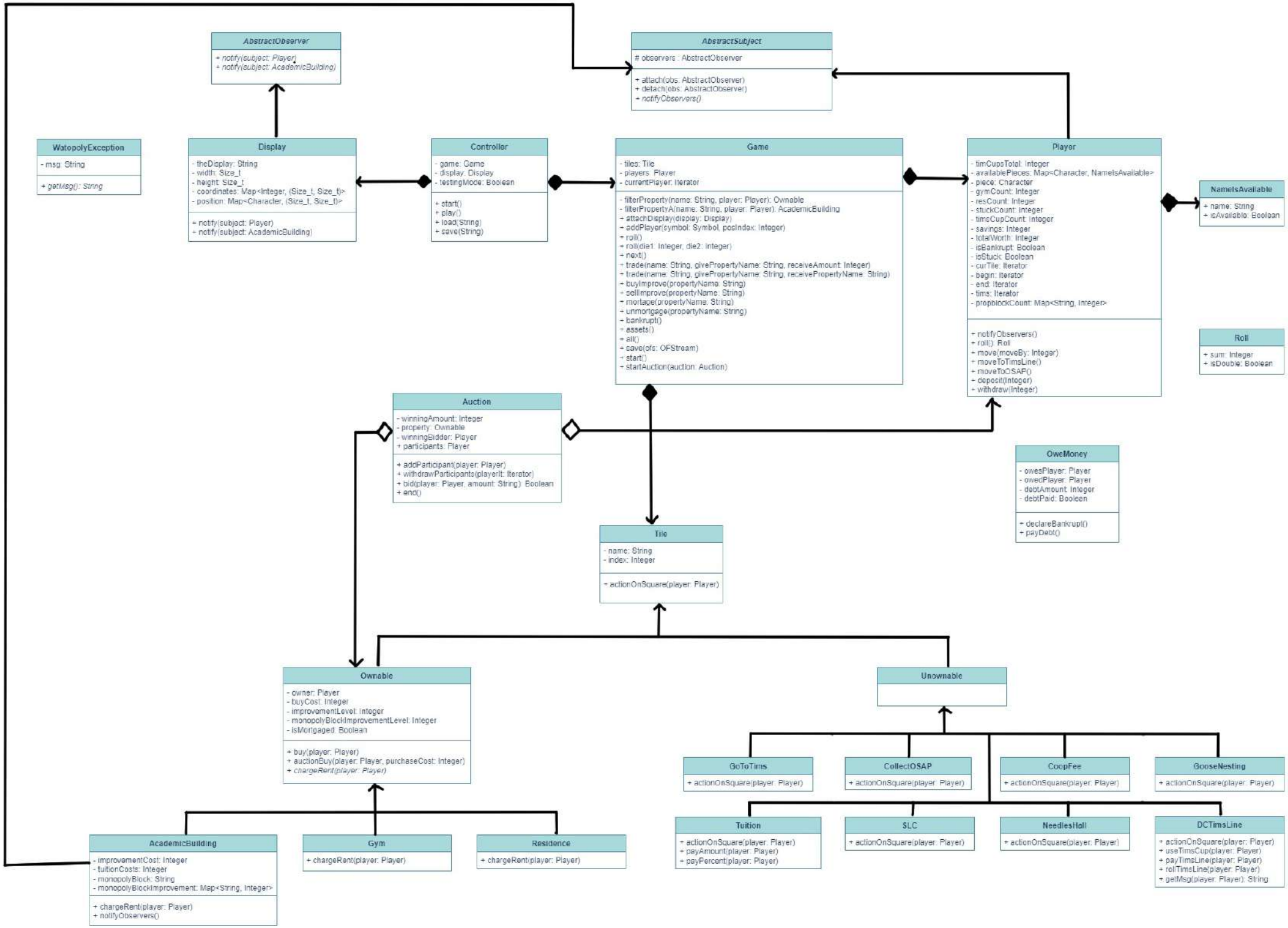
AbstractObserver
+ notify(subject: Player)
+ notify(subject: AcademicBuilding)
AbstractSubject
observers : AbstractObserver
+ attach(obs: AbstractObserver)
+ detach(obs: AbstractObserver)
+ notifyObservers()
WatopolyException
- msg: String
+ getMsg(): String
Display
- theDisplay: String
- width: Size_t
- height: Size_t
- coordinates: Map<Integer, (Size_t, Size_t)>
- position: Map<Character, (Size_t, Size_t)>
+ notify(subject: Player)
+ notify(subject: AcademicBuilding)
Controller
- game: Game
- display: Display
- testingMode: Boolean
+ start()
+ play()
+ load(String)
+ save(String)
Game
- tiles: Tile
- players: Player
- currentPlayer: Iterator
- filterProperty(name: String, player: Player): Ownable
- filterPropertyA(name: String, player: Player): AcademicBuilding
+ attachDisplay(display: Display)
+ addPlayer(symbol: Symbol, posIndex: Integer)
+ roll()
+ roll(die1: Integer, die2: Integer)
+ next()
+ trade(name: String, givePropertyName: String, receiveAmount: Integer)
+ trade(name: String, givePropertyName: String, receivePropertyName: String)
+ buyImprove(propertyName: String)
+ sellImprove(propertyName: String)
+ mortage(propertyName: String)
+ unmortgage(propertyName: String)
+ bankrupt()
+ assets()
+ all()
+ save(ofs: OFStream)
+ start()
+ startAuction(auction: Auction)
Player
- timCupsTotal: Integer
- availablePieces: Map<Character, NameIsAvailable>
- piece: Character
- gymCount: Integer
- resCount: Integer
- stuckCount: Integer
- timsCupCount: Integer
- savings: Integer
- totalWorth: Integer
- isBankrupt: Boolean
- isStuck: Boolean
- curTile: Iterator
- begin: Iterator
- end: Iterator
- tims: Iterator
- propblockCount: Map<String, Integer>
+ notifyObservers()
+ roll(): Roll
+ move(moveBy: Integer)
+ moveToTimsLine()
+ moveToOSAP()
+ deposit(Integer)
+ withdraw(Integer)
NameIsAvailable
+ name: String
+ isAvailable: Boolean
Roll
+ sum: Integer
+ isDouble: Boolean
Auction
- winningAmount: Integer
- property: Ownable
- winningBidder: Player
+ participants: Player
+ addParticipant(player: Player)
+ withdrawParticipants(playerIt: Iterator)
+ bid(player: Player, amount: String): Boolean
+ end()
OweMoney
- owesPlayer: Player
- owedPlayer: Player
- debtAmount: Integer
- debtPaid: Boolean
+ declareBankrupt()
+ payDebt()
Tile
- name: String
- index: Integer
+ actionOnSquare(player: Player)
Ownable
- owner: Player
- buyCost: Integer
- improvementLevel: Integer
- monopolyBlockImprovementLevel: Integer
- isMortgaged: Boolean
+ buy(player: Player)
+ auctionBuy(player: Player, purchaseCost: Integer)
+ chargeRent(player: Player)
Unownable
GoToTims
+ actionOnSquare(player: Player)
CollectOSAP
+ actionOnSquare(player: Player)
CoopFee
+ actionOnSquare(player: Player)
GooseNesting
+ actionOnSquare(player: Player)
Tuition
+ actionOnSquare(player: Player)
+ payAmount(player: Player)
+ payPercent(player: Player)
SLC
+ actionOnSquare(player: Player)
NeedlesHall
+ actionOnSquare(player: Player)
DCTimsLine
+ actionOnSquare(player: Player)
+ useTimsCup(player: Player)
+ payTimsLine(player: Player)
+ rollTimsLine(player: Player)
+ getMsg(player: Player): String
AcademicBuilding
- improvementCost: Integer
- tuitionCosts: Integer
- monopolyBlock: String
- monopolyBlockImprovement: Map<String, Integer>
+ chargeRent(player: Player)
+ notifyObservers()
Gym
+ chargeRent(player: Player)
Residence
+ chargeRent(player: Player)