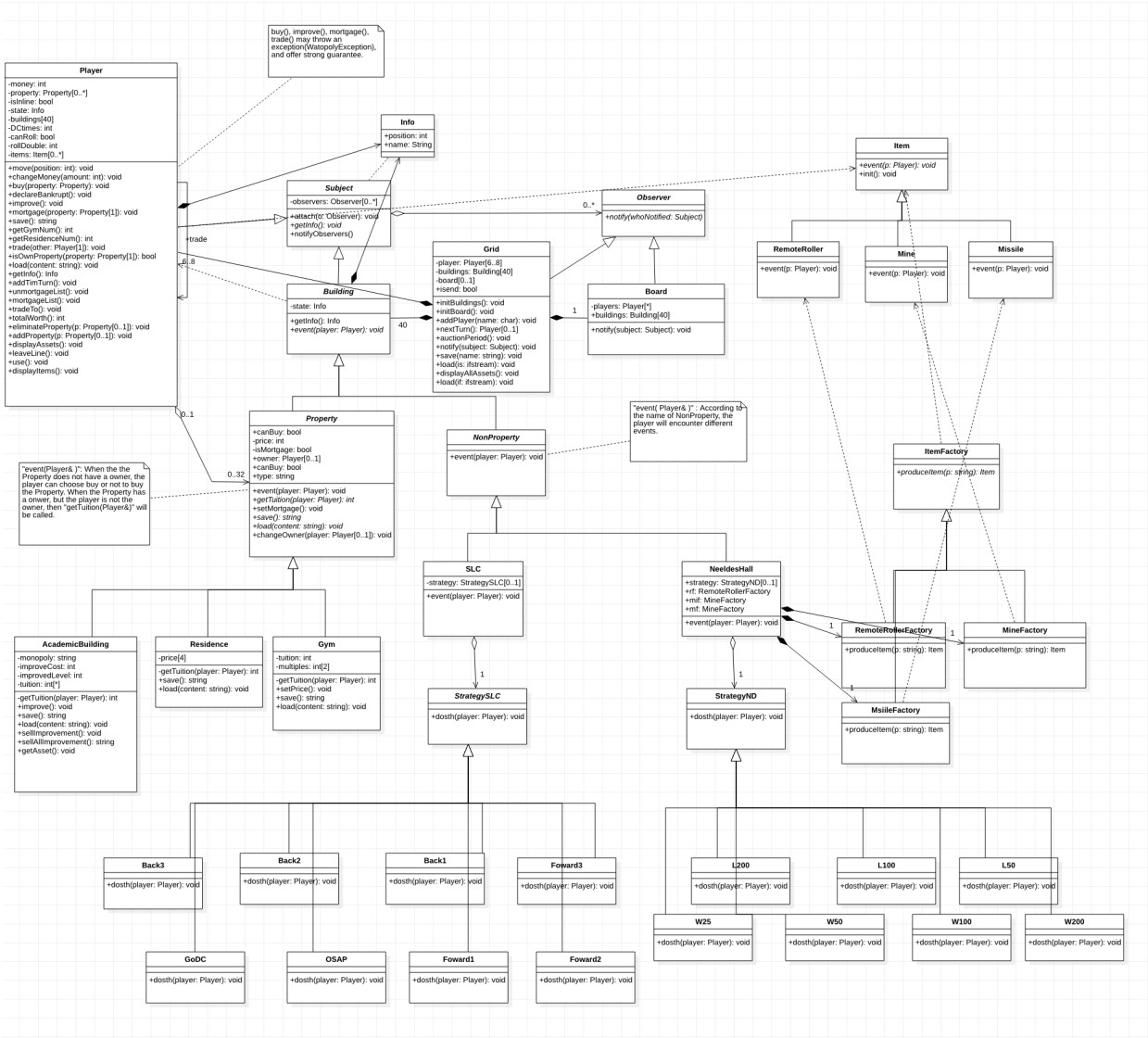
buy(), improve(), mortgage(), trade() may throw an exception(WatopolyException), and offer strong guarantee.
Player
-money: int
-property: Property[0..*]
-isInline: bool
-state: Info
-buildings[40]
-DCtimes: int
-canRoll: bool
-rollDouble: int
-items: Item[0..*]
+move(position: int): void
+changeMoney(amount: int): void
+buy(property: Property): void
+declareBankrupt(): void
+improve(): void
+mortgage(property: Property[1]): void
+save(): string
+getGymNum(): int
+getResidenceNum(): int
+trade(other: Player[1]): void
+isOwnProperty(property: Property[1]): bool
+load(content: string): void
+getInfo(): Info
+addTimTurn(): void
+unmortgageList(): void
+mortgageList(): void
+tradeTo(): void
+totalWorth(): int
+eliminateProperty(p: Property[0..1]): void
+addProperty(p: Property[0..1]): void
+displayAssets(): void
+leaveLine(): void
+use(): void
+displayItems(): void
Info
+position: int
+name: String
Subject
-observers: Observer[0..*]
+attach(o: Observer): void
+getInfo(): void
+notifyObservers()
Observer
+notify(whoNotified: Subject)
Item
+event(p: Player): void
+init(): void
RemoteRoller
+event(p: Player): void
Mine
+event(p: Player): void
Missile
+event(p: Player): void
Grid
-player: Player[6..8]
-buildings: Building[40]
-board[0..1]
+send: bool
+initBuildings(): void
+initBoard(): void
+addPlayer(name: char): void
+nextTurn(): Player[0..1]
+auctionPeriod(): void
+notify(subject: Subject): void
+save(name: string): void
+load(is: ifstream): void
+displayAllAssets(): void
+load(if: ifstream): void
Building
-state: Info
+getInfo(): Info
+event(player: Player): void
Board
-players: Player[*]
+buildings: Building[40]
+notify(subject: Subject): void
+trade
6..8
40
1
0..*
0..1
0..32
Property
+canBuy: bool
-price: int
-isMortgage: bool
+owner: Player[0..1]
+canBuy: bool
+type: string
+event(player: Player): void
+getTuition(player: Player): int
+setMortgage(): void
+save(): string
+load(content: string): void
+changeOwner(player: Player[0..1]): void
"event(Player&)": When the the Property does not have a owner, the player can choose buy or not to buy the Property. When the Property has a onwer, but the player is not the owner, then "getTuition(Player&)" will be called.
NonProperty
+event(player: Player): void
"event(Player&)" : According to the name of NonProperty, the player will encounter different events.
ItemFactory
+produceItem(p: string): Item
SLC
-strategy: StrategySLC[0..1]
+event(player: Player): void
NeeldesHall
+strategy: StrategyND[0..1]
+rf: RemoteRollerFactory
+mf: MineFactory
+mf: MineFactory
+event(player: Player): void
RemoteRollerFactory
+produceItem(p: string): Item
MineFactory
+produceItem(p: string): Item
MsiileFactory
+produceItem(p: string): Item
AcademicBuilding
-monopoly: string
-improveCost: int
-improvedLevel: int
-tuition: int[*]
-getTuition(player: Player): int
+improve(): void
+save(): string
+load(content: string): void
+sellImprovement(): void
+sellAllImprovement(): string
+getAsset(): void
Residence
-price[4]
-getTuition(player: Player): int
+save(): string
+load(content: string): void
Gym
-tuition: int
-multiples: int[2]
-getTuition(player: Player): int
+setPrice(): void
+save(): string
+load(content: string): void
StrategySLC
+dosth(player: Player): void
StrategyND
+dosth(player: Player): void
Back3
+dosth(player: Player): void
Back2
+dosth(player: Player): void
Back1
+dosth(player: Player): void
Foward3
+dosth(player: Player): void
L200
+dosth(player: Player): void
L100
+dosth(player: Player): void
L50
+dosth(player: Player): void
W25
+dosth(player: Player): void
W50
+dosth(player: Player): void
W100
+dosth(player: Player): void
W200
+dosth(player: Player): void
GoDC
+dosth(player: Player): void
OSAP
+dosth(player: Player): void
Foward1
+dosth(player: Player): void
Foward2
+dosth(player: Player): void