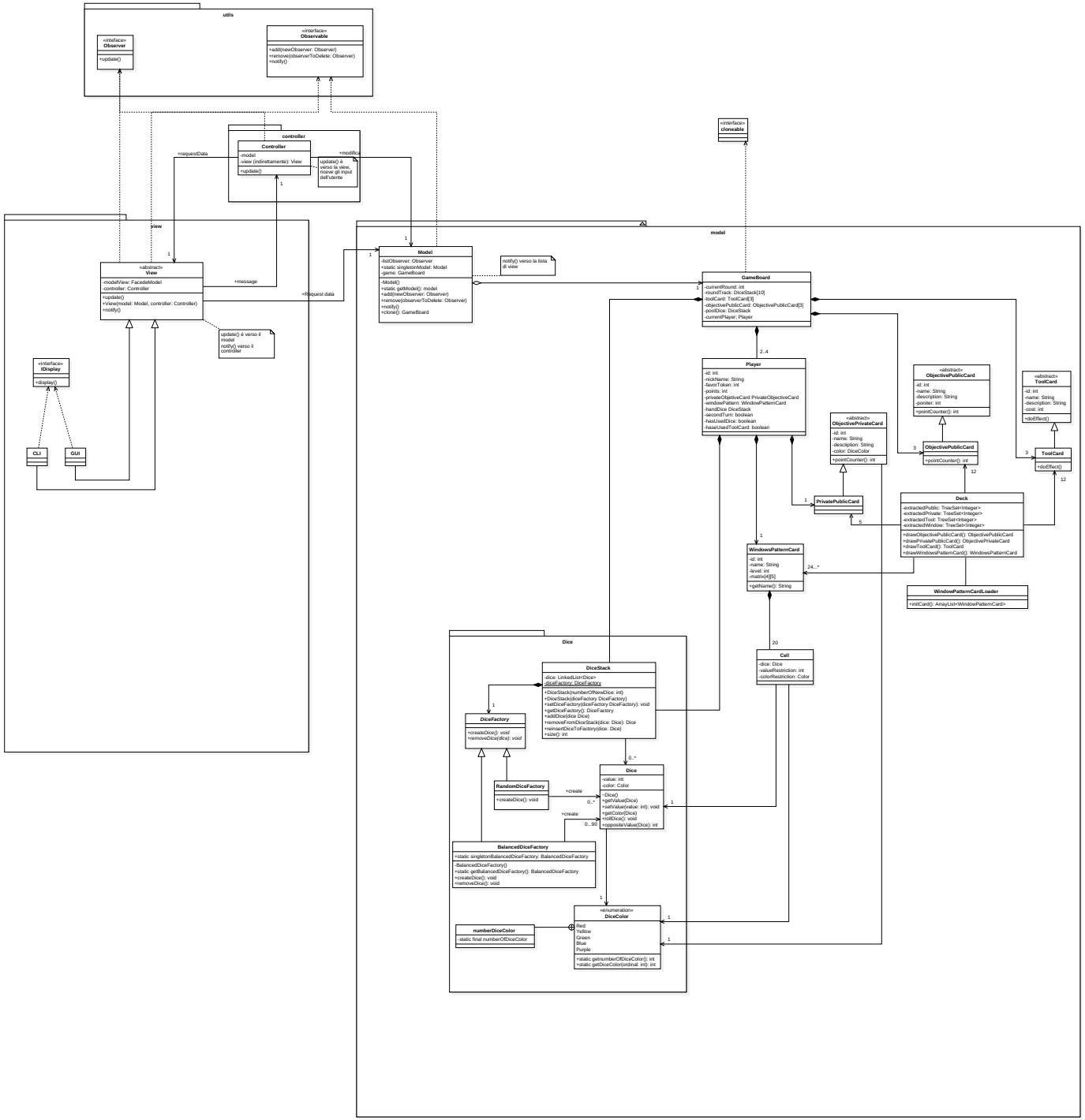
Model::Main


utils
«interface»
Observer
+update()
«interface»
Observable
+add(newObserver: Observer)
+remove(observerToDelete: Observer)
+notify()
controller
Controller
-model
-view (indirettamente): View
+update()
+requestData
+modifica
update() è verso la view, riceve gli input dell'utente
view
«abstract»
View
-modelView: FacadeModel
-controller: Controller
+update()
+View(model: Model, controller: Controller)
+notify()
+message
+Request data
update() è verso il model
notify() verso il controller
«interface»
IDisplay
+display()
CLI
GUI
model
Model
-listObserver: Observer
+static singletonModel: Model
-game: GameBoard
-Model()
+static getModel(): model
+add(newObserver: Observer)
+remove(observerToDelete: Observer)
+notify()
+clone(): GameBoard
notify() verso la lista di view
«interface»
cloneable
GameBoard
-currentRound: int
-roundTrack: DiceStack[10]
-lootCard: ToolCard[3]
-objectivePublicCard: ObjectivePublicCard[3]
-poolDice: DiceStack
-currentPlayer: Player
Player
-id: int
-nickName: String
-favorToken: int
-points: int
-privateObjetiveCard PrivateObjectiveCard
-windowPattern: WindowPatternCard
-handDice DiceStack
-secondTurn: boolean
-hasUsedDice: boolean
-haseUsedToolCard: boolean
«abstract»
ObjectivePublicCard
-id: int
-name: String
-description: String
-pointer: int
+pointCounter(): int
«abstract»
ToolCard
-id: int
-name: String
-description: String
-cost: int
+doEffect()
«abstract»
ObjectivePrivateCard
-id: int
-name: String
-description: String
-color: DiceColor
+pointCounter(): int
ObjectivePublicCard
+pointCounter(): int
ToolCard
+doEffect()
PrivatePublicCard
Deck
-extractedPublic: TreeSet<Integer>
-extractedPrivate: TreeSet<Integer>
-extractedTool: TreeSet<Integer>
-extractedWindow: TreeSet<Integer>
+drawObjectivePublicCard(): ObjectivePublicCard
+drawPrivatePublicCard(): ObjectivePrivateCard
+drawToolCard(): ToolCard
+drawWindowsPatternCard(): WindowsPatternCard
WindowsPatternCard
-id: int
-name: String
-level: int
-matrix[4][5]
+getName(): String
WindowPatternCardLoader
+initCard(): ArrayList<WindowPatternCard>
Cell
-dice: Dice
-valueRestriction: int
-colorRestriction: Color
Dice
DiceStack
-dice: LinkedList<Dice>
-diceFactory: DiceFactory
+DiceStack(numberOfNewDice: int)
+DiceStack(diceFactory DiceFactory)
+setDiceFactory(diceFactory DiceFactory): void
+getDiceFactory(): DiceFactory
+addDice(dice Dice)
+removeFromDiceStack(dice: Dice): Dice
+reinsertDiceToFactory(dice: Dice)
+size(): int
DiceFactory
+creataDice(): void
+removeDice(dice): void
RandomDiceFactory
+createDice(): void
+create
Dice
-value: int
-color: Color
-Dice()
+getValue(Dice)
+setValue(value: int): void
+getColor(Dice)
+rollDice(): void
+oppositeValue(Dice): int
BalancedDiceFactory
+static singletonBalancedDiceFactory: BalancedDiceFactory
-BalancedDiceFactory()
+static getBalancedDiceFactory(): BalancedDiceFactory
+createDice(): void
+removeDice(): void
«enumeration»
DiceColor
Red
Yellow
Green
Blue
Purple
+static getnumberOfDiceColor(): int
+static getDiceColor(ordinal: int): int
numberDiceColor
-static final numberOfDiceColor