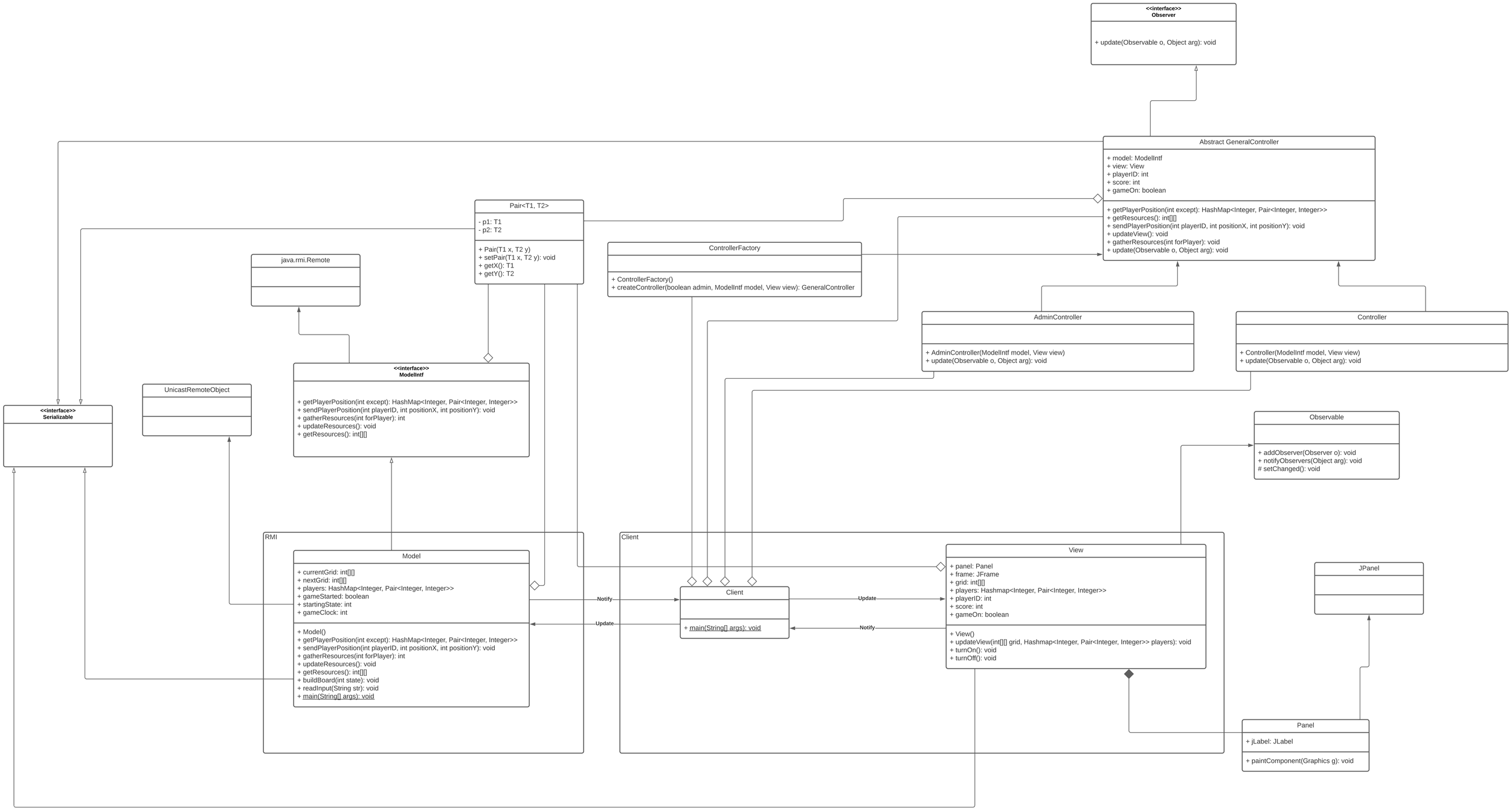
<<interface>>
Observer
+ update(Observable o, Object arg): void
Abstract GeneralController
+ model: ModelIntf
+ view: View
+ playerID: int
+ score: int
+ gameOn: boolean
+ getPlayerPosition(int except): HashMap<Integer, Pair<Integer, Integer>>
+ getResources(): int[][]
+ sendPlayerPosition(int playerID, int positionX, int positionY): void
+ updateView(): void
+ gatherResources(int forPlayer): void
+ update(Observable o, Object arg): void
Pair<T1, T2>
- p1: T1
- p2: T2
+ Pair(T1 x, T2 y)
+ setPair(T1 x, T2 y): void
+ getX(): T1
+ getY(): T2
ControllerFactory
+ ControllerFactory()
+ createController(boolean admin, ModelIntf model, View view): GeneralController
java.rmi.Remote
AdminController
+ AdminController(ModelIntf model, View view)
+ update(Observable o, Object arg): void
Controller
+ Controller(ModelIntf model, View view)
+ update(Observable o, Object arg): void
<<interface>>
ModelIntf
+ getPlayerPosition(int except): HashMap<Integer, Pair<Integer, Integer>>
+ sendPlayerPosition(int playerID, int positionX, int positionY): void
+ gatherResources(int forPlayer): int
+ updateResources(): void
+ getResources(): int[][]
UnicastRemoteObject
<<interface>>
Serializable
Observable
+ addObserver(Observer o): void
+ notifyObservers(Object arg): void
# setChanged(): void
RMI
Model
+ currentGrid: int[][]
+ nextGrid: int[][]
+ players: HashMap<Integer, Pair<Integer, Integer>>
+ gameStarted: boolean
+ startingState: int
+ gameClock: int
+ Model()
+ getPlayerPosition(int except): HashMap<Integer, Pair<Integer, Integer>>
+ sendPlayerPosition(int playerID, int positionX, int positionY): void
+ gatherResources(int forPlayer): int
+ updateResources(): void
+ getResources(): int[][]
+ buildBoard(int state): void
+ readInput(String str): void
+ main(String[] args): void
Notify
Update
Client
Client
+ main(String[] args): void
Update
Notify
View
+ panel: Panel
+ frame: JFrame
+ grid: int[][]
+ players: Hashmap<Integer, Pair<Integer, Integer>>
+ playerID: int
+ score: int
+ gameOn: boolean
+ View()
+ updateView(int[][] grid, Hashmap<Integer, Pair<Integer, Integer>> players): void
+ turnOn(): void
+ turnOff(): void
JPanel
Panel
+ jLabel: JLabel
+ paintComponent(Graphics g): void