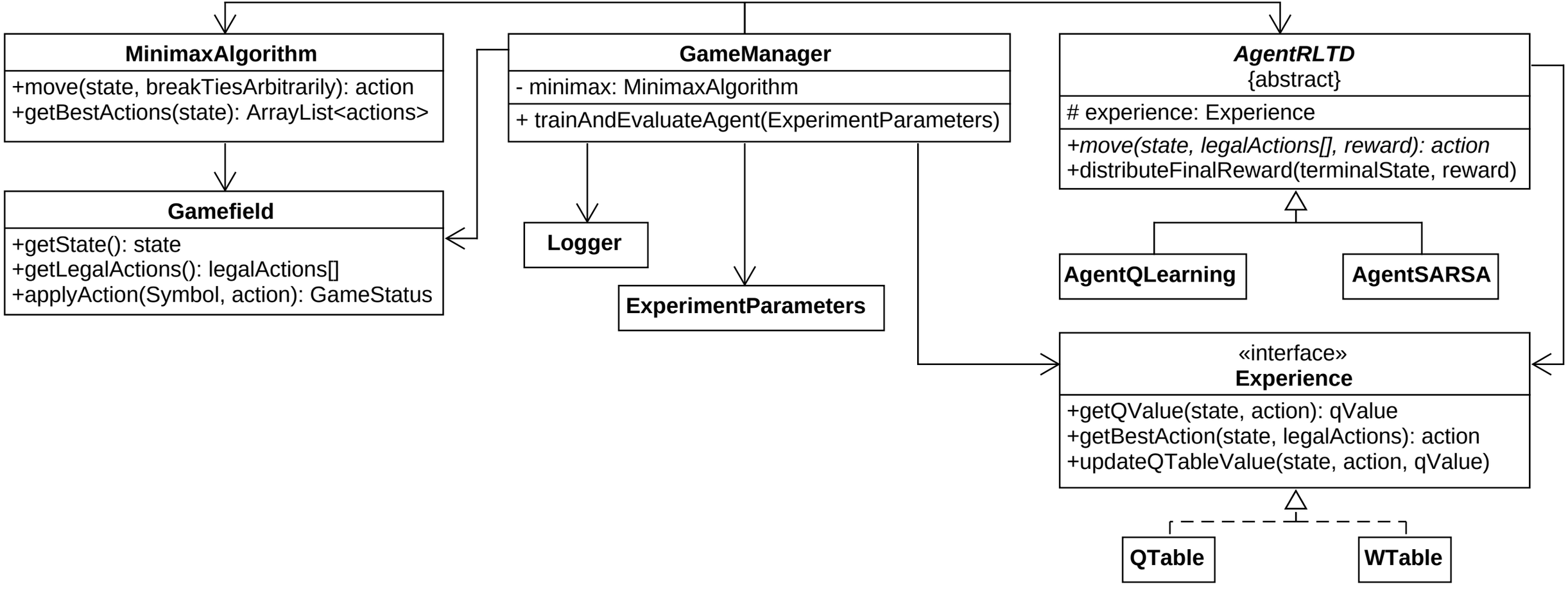
MinimaxAlgorithm
+move(state, breakTiesArbitrarily): action
+getBestActions(state): ArrayList<actions>
GameManager
- minimax: MinimaxAlgorithm
+ trainAndEvaluateAgent(ExperimentParameters)
AgentRLTD
{abstract}
experience: Experience
+move(state, legalActions[], reward): action
+distributeFinalReward(terminalState, reward)
Gamefield
+getState(): state
+getLegalActions(): legalActions[]
+applyAction(Symbol, action): GameStatus
Logger
ExperimentParameters
AgentQLearning
AgentSARSA
«interface»
Experience
+getQValue(state, action): qValue
+getBestAction(state, legalActions): action
+updateQTableValue(state, action, qValue)
QTable
WTable