Restaurant Simulation - Quick Object Reference

## 1. Restaurant

* Attributes:

- name: String

- tables: List<Table>

- menu: List<Platter>

- staff: List<Staff>

- revenue: double

- avgWalkTime: double

- globals: GlobalVariables

- decisionEngine: GlobalMethods

* Methods:

+ registerLoggables()

+ triggerLog()

+ simulateDay()

+ decideStaffing()

+ decideTableAssignments()

+ decideInventoryRestock()

+ decideStaffing (uses Probability: + RushHourOverloadChance)

decideInventoryRestock (uses Probability: IngredientSpoilageRate)

+ getGlobals(): GlobalVariables

+ getDecisionEngine(): GlobalMethods

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 2. Hostess

* Attributes:

- name: String

* Methods:

+ registerLoggables()

+ triggerLog()

+ greetCustomer(Customer)

+ seatCustomer(Customer, Table)

+ decideTableAssignment(Customer)

+ releaseTableDecision(Table, Customer)

+ decideSeating()

+ decideQueueOrder()

decideSeating (uses Probability: OptimalTableSelectionProbability)

decideQueueOrder (uses Probability: QueueBalkingProbability)

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 3. CashRegister

* Attributes:

- totalStoreIncome: double

- totalTips: double

* Methods:

+ registerLoggables()

+ triggerLog()

+ processPayment(Server, Order, double tip)

+ recordTip(Server, double tip)

+ decidePaymentHandling()

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 4. Table

* Attributes:

- id: int

- capacity: int

- location: String

- occupied: boolean

* Methods:

+ registerLoggables()

+ triggerLog()

+ assignCustomer(Customer)

+ clearTable()

+ decideClearTable()

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 5. Platter

* Attributes:

- name: String

- ingredients: List<Ingredient>

- cookingTime: int

- price: double

* Methods:

+ registerLoggables()

+ triggerLog()

+ prepare()

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 6. Ingredient

* Attributes:

- name: String

- quantity: double

- costPerUnit: double

- expiryDate: Date

* Methods:

+ registerLoggables()

+ triggerLog()

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 7. Cook

* Attributes:

- name: String

- skillLevel: int

* Methods:

+ registerLoggables()

+ triggerLog()

+ cook(Platter)

+ prep(Platter)

+ announceOrder(Table)

+ decideNextOrder()

+ decideCookingPriority()

decideNextOrder (uses Probability: PrepTimeVariance)

decideCookingPriority (uses Probability: CookingDelayRate)

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 8. Server

* Attributes:

- name: String

- tablesAssigned: List<Table>

* Methods:

+ registerLoggables()

+ triggerLog()

+ serve(Table)

+ takeOrder(Customer)

+ deliverCheck(Customer)

+ returnPayment(CashRegister)

+ decideDeliverySequence()

+ decideTipHandling()

decideDeliverySequence (uses Probability: ServiceDelayVariance)

decideTipHandling (uses Probability: TipLikelihood)

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 9. ServerTable

* Attributes:

- bell: boolean

- orderQueue: Queue<Order>

* Methods:

+ registerLoggables()

+ triggerLog()

+ announceOrder(Order)

+ pickUpOrder(Server)

+ viewNextOrder()

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 10. Order

* Attributes:

- platter: Platter

- table: Table

- timePlaced: int

* Methods:

+ registerLoggables()

+ triggerLog()

+ getTable()

+ getPlatter()

+ calculateTip(happiness: double)

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 11. Customer

* Attributes:

- groupSize: int

- patience: int

- satisfaction: int

- tablePreference: String

* Methods:

+ registerLoggables()

+ triggerLog()

+ requestTable(String)

+ order(Platter)

+ leave()

+ decideExit()

+ decideReorder()

decideExit (uses Probability: PatienceThreshold)

decideReorder (uses Probability: ReorderLikelihood)

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 12. Inventory

* Attributes:

- ingredients: List<Ingredient>

* Methods:

+ registerLoggables()

+ triggerLog()

+ restock()

+ checkStock(Platter)

+ decideRestockTiming()

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 13. Refrigerator

* Attributes:

- capacity: double

* Methods:

+ registerLoggables()

+ triggerLog()

+ store(Ingredient)

+ remove(Ingredient)

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 14. Stove

* Attributes:

- slots: int

* Methods:

+ registerLoggables()

+ triggerLog()

+ startCooking(Platter)

+ finishCooking(Platter)

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 15. Finance

* Attributes:

- fixedCosts: double

- laborCosts: double

* Methods:

+ registerLoggables()

+ triggerLog()

+ calculateNetRevenue()

+ decideExpenseApproval()

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 16. SimulationClock

* Attributes:

- currentTime: int

- tickRate: int

* Methods:

+ advanceTime()

+ scheduleEvent()

+ getTime()

+ decideNextEventTrigger()

+ evaluate(): boolean

+ commit(): void

+ getState(): String[][]

## 19. SimulationManager

* Methods:

+ registerLoggables(objectName: String, logMap: Map<String, Supplier<Object>>)

+ logObject(objectName: String)

+ triggerLog(objectName: String)

+ loadInitFile(filename: String)

+ resetSimulation()

+ saveState(filename: String)

+ setVCD(filename: String)

+ run(int): void

+ stepOnce(): void

+ getClock(): SimulationClock

+ getProcesses(): List<SimProcess>

+ startConsole(): void

+ listObjects(): void

+ showObject(String): void

+ setValue(String, String, String): void

+ startLogging(String): void

+ stopLogging(): void

+ logTick(int): void

+ displayStates(): void