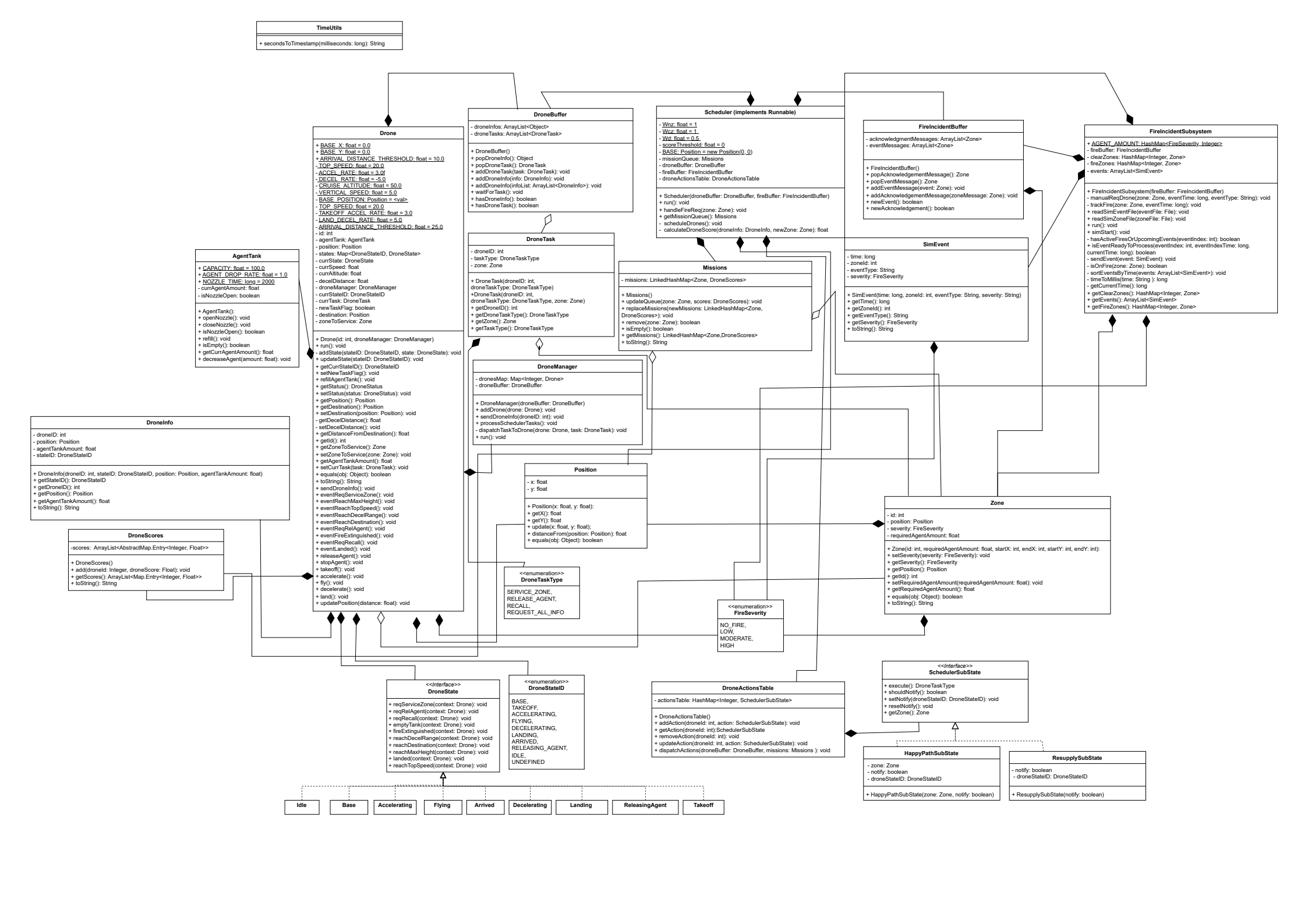
TimeUtils
+ secondsToTimestamp(milliseconds: long): String
Drone
+ BASE_X: float = 0.0
+ BASE_Y: float = 0.0
+ ARRIVAL_DISTANCE_THRESHOLD: float = 10.0
- TOP_SPEED: float = 20.0
- ACCEL_RATE: float = 3.0f
- DECEL_RATE: float = -5.0
- CRUISE_ALTITUDE: float = 50.0
- VERTICAL_SPEED: float = 5.0
- BASE_POSITION: Position = <val>
- TOP_SPEED: float = 20.0
- TAKEOFF_ACCEL_RATE: float = 3.0
- LAND_DECEL_RATE: float = 5.0
- ARRIVAL_DISTANCE_THRESHOLD: float = 25.0
- id: int
- agentTank: AgentTank
- position: Position
- states: Map<DroneStateID, DroneState>
- currState: DroneState
- currSpeed: float
- currAltitude: float
- decelDistance: float
- droneManager: DroneManager
- currStateID: DroneStateID
- currTask: DroneTask
- newTaskFlag: boolean
- destination: Position
- zoneToService: Zone
+ Drone(id: int, droneManager: DroneManager)
+ run(): void
- addState(stateID: DroneStateID, state: DroneState): void
+ updateState(stateID: DroneStateID): void
+ getCurrStateID(): DroneStateID
+ setNewTaskFlag(): void
+ refillAgentTank(): void
+ getStatus(): DroneStatus
+ setStatus(status: DroneStatus): void
+ getPosition(): Position
+ getDestination(): Position
+ setDestination(position: Position): void
- getDecelDistance(): float
- setDecelDistance(): void
+ getDistanceFromDestination(): float
+ getId(): int
+ getZoneToService(): Zone
+ setZoneToService(zone: Zone): void
+ getAgentTankAmount(): float
+ setCurrTask(task: DroneTask): void
+ equals(obj: Object): boolean
+ toString(): String
+ sendDroneInfo(): void
+ eventReqServiceZone(): void
+ eventReachMaxHeight(): void
+ eventReachTopSpeed(): void
+ eventReachDecelRange(): void
+ eventReachDestination(): void
+ eventReqRelAgent(): void
+ eventFireExtinguished(): void
+ eventReqRecall(): void
+ eventLanded(): void
+ releaseAgent(): void
+ stopAgent(): void
+ takeoff(): void
+ accelerate(): void
+ fly(): void
+ decelerate(): void
+ land(): void
+ updatePosition(distance: float): void
DroneBuffer
- droneInfos: ArrayList<Object>
- droneTasks: ArrayList<DroneTask>
+ DroneBuffer()
+ popDroneInfo(): Object
+ popDroneTask(): DroneTask
+ addDroneTask(task: DroneTask): void
+ addDroneInfo(info: DroneInfo): void
+ addDroneInfo(infoList: ArrayList<DroneInfo>): void
+ waitForTask(): void
+ hasDroneInfo(): boolean
+ hasDroneTask(): boolean
Scheduler (implements Runnable)
- Wnz: float = 1
- Wcz: float = 1
- Wd: float = 0.5
- scoreThreshold: float = 0
- BASE: Position = new Position(0, 0)
- missionQueue: Missions
- droneBuffer: DroneBuffer
- fireBuffer: FireIncidentBuffer
- droneActionsTable: DroneActionsTable
+ Scheduler(droneBuffer: DroneBuffer, fireBuffer: FireIncidentBuffer)
+ run(): void
+ handleFireReq(zone: Zone): void
+ getMissionQueue(): Missions
- scheduleDrones(): void
- calculateDroneScore(droneInfo: DroneInfo, newZone: Zone): float
FireIncidentBuffer
- acknowledgmentMessages: ArrayList<Zone>
- eventMessages: ArrayList<Zone>
+ FireIncidentBuffer()
+ popAcknowledgementMessage(): Zone
+ popEventMessage(): Zone
+ addEventMessage(event: Zone): void
+ addAcknowledgementMessage(zoneMessage: Zone): void
+ newEvent(): boolean
+ newAcknowledgement(): boolean
FireIncidentSubsystem
+ AGENT_AMOUNT: HashMap<FireSeverity, Integer>
- fireBuffer: FireIncidentBuffer
- clearZones: HashMap<Integer, Zone>
- fireZones: HashMap<Integer, Zone>
- events: ArrayList<SimEvent>
+ FireIncidentSubsystem(fireBuffer: FireIncidentBuffer)
- manualReqDrone(zone: Zone, eventTime: long, eventType: String): void
- trackFire(zone: Zone, eventTime: long): void
+ readSimEventFile(eventFile: File): void
+ readSimZoneFile(zoneFile: File): void
+ run(): void
+ simStart(): void
- hasActiveFiresOrUpcomingEvents(eventIndex: int): boolean
+ isEventReadyToProcess(eventIndex: int, eventIndexTime: long, currentTime: long): boolean
- sendEvent(event: SimEvent): void
- isOnFire(zone: Zone): boolean
- sortEventsByTime(events: ArrayList<SimEvent>): void
- timeToMillis(time: String ): long
- getCurrentTime(): long
+ getClearZones(): HashMap<Integer, Zone>
+ getEvents(): ArrayList<SimEvent>
+ getFireZones(): HashMap<Integer, Zone>
AgentTank
+ CAPACITY: float = 100.0
+ AGENT_DROP_RATE: float = 1.0
+ NOZZLE_TIME: long = 2000
- currAgentAmount: float
- isNozzleOpen: boolean
+ AgentTank():
+ openNozzle(): void
+ closeNozzle(): void
+ isNozzleOpen(): boolean
+ refill(): void
+ isEmpty(): boolean
+ getCurrAgentAmount(): float
+ decreaseAgent(amount: float): void
DroneTask
- droneID: int
- taskType: DroneTaskType
- zone: Zone
+ DroneTask(droneID: int, droneTaskType: DroneTaskType)
+DroneTask(droneID: int, droneTaskType: DroneTaskType, zone: Zone)
+ getDroneID(): int
+ getDroneTaskType(): DroneTaskType
+ getZone(): Zone
+ getTaskType(): DroneTaskType
Missions
- missions: LinkedHashMap<Zone, DroneScores>
+ Missions()
+ updateQueue(zone: Zone, scores: DroneScores): void
+ replaceMissions(newMissions: LinkedHashMap<Zone, DroneScores>): void
+ remove(zone: Zone): boolean
+ isEmpty(): boolean
+ getMissions(): LinkedHashMap<Zone,DroneScores>
+ toString(): String
SimEvent
- time: long
- zoneId: int
- eventType: String
- severity: FireSeverity
+ SimEvent(time: long, zoneId: int, eventType: String, severity: String)
+ getTime(): long
+ getZoneId(): int
+ getEventType(): String
+ getSeverity(): FireSeverity
+ toString(): String
DroneManager
- dronesMap: Map<Integer, Drone>
- droneBuffer: DroneBuffer
+ DroneManager(droneBuffer: DroneBuffer)
+ addDrone(drone: Drone): void
+ sendDroneInfo(droneID: int): void
+ processSchedulerTasks(): void
- dispatchTaskToDrone(drone: Drone, task: DroneTask): void
+ run(): void
DroneInfo
- droneID: int
- position: Position
- agentTankAmount: float
- stateID: DroneStateID
+ DroneInfo(droneID: int, stateID: DroneStateID, position: Position, agentTankAmount: float)
+ getStateID(): DroneStateID
+ getDroneID(): int
+ getPosition(): Position
+ getAgentTankAmount(): float
+ toString(): String
Position
- x: float
- y: float
+ Position(x: float, y: float):
+ getX(): float
+ getY(): float
+ update(x: float, y: float);
+ distanceFrom(position: Position): float
+ equals(obj: Object): boolean
Zone
- id: int
- position: Position
- severity: FireSeverity
- requiredAgentAmount: float
+ Zone(id: int, requiredAgentAmount: float, startX: int, endX: int, startY: int, endY: int):
+ setSeverity(severity: FireSeverity): void
+ getSeverity(): FireSeverity
+ getPosition(): Position
+ getId(): int
+ setRequiredAgentAmount(requiredAgentAmount: float): void
+ getRequiredAgentAmount(): float
+ equals(obj: Object): boolean
+ toString(): String
DroneScores
-scores: ArrayList<AbstractMap.Entry<Integer, Float>>
+ DroneScores()
+ add(droneId: Integer, droneScore: Float): void
+ getScores(): ArrayList<Map.Entry<Integer, Float>>
+ toString(): String
<<enumeration>>
DroneTaskType
SERVICE_ZONE,
RELEASE_AGENT,
RECALL,
REQUEST_ALL_INFO
<<enumeration>>
FireSeverity
NO_FIRE,
LOW,
MODERATE,
HIGH
<<Interface>>
DroneState
+ reqServiceZone(context: Drone): void
+ reqRelAgent(context: Drone): void
+ reqRecall(context: Drone): void
+ emptyTank(context: Drone): void
+ fireExtinguished(context: Drone): void
+ reachDecelRange(context: Drone): void
+ reachDestination(context: Drone): void
+ reachMaxHeight(context: Drone): void
+ landed(context: Drone): void
+ reachTopSpeed(context: Drone): void
<<enumeration>>
DroneStateID
BASE,
TAKEOFF,
ACCELERATING,
FLYING,
DECELERATING,
LANDING,
ARRIVED,
RELEASING_AGENT,
IDLE,
UNDEFINED
DroneActionsTable
- actionsTable: HashMap<Integer, SchedulerSubState>
+ DroneActionsTable()
+ addAction(droneId: int, action: SchedulerSubState): void
+ getAction(droneId: int):SchedulerSubState
+ removeAction(droneId: int): void
+ updateAction(droneId: int, action: SchedulerSubState): void
+ dispatchActions(droneBuffer: DroneBuffer, missions: Missions ): void
<<Interface>>
SchedulerSubState
+ execute(): DroneTaskType
+ shouldNotify(): boolean
+ setNotify(droneStateID: DroneStateID): void
+ resetNotify(): void
+ getZone(): Zone
HappyPathSubState
- zone: Zone
- notify: boolean
- droneStateID: DroneStateID
+ HappyPathSubState(zone: Zone, notify: boolean)
ResupplySubState
- notify: boolean
- droneStateID: DroneStateID
+ ResupplySubState(notify: boolean)
Idle
Base
Accelerating
Flying
Arrived
Decelerating
Landing
ReleasingAgent
Takeoff