#### arrows

displayAttacks(areaId) displayArrow(mapX, mapY, unitIndex) displayAttackOptions(id) clearArrows()

## highLights

highlightArea(coords, highlightColor = PALE\_ORANGE) (coords, highlightColor = LIGHT GREY) highlightEneamy(coords, highlightColor = LIGHT\_PURPLE) clearHighlights()

# moving

var destinationId var originId var moveDestinationIds var fromAreas

startMove(controller) findUnitsAreas(control) hightlightUnitAreas(allOrigins) addMoveOriginListeners(allOrigins) moveOriginHandler(event) removeOriginListeners(allOrigins) addMoveDestinationListeners(allOrigins) rotateUnits(targetAreaId) rotateUnitsHandler() moveDestinationHandler(event) removeDestinationListeners(allOrigins clearUnits()

### attacking

var attackingAreaId var defendingAreaId var allFriendlyFrontLineAreas = [] var enemyNeighbours = []

findFrontline(friendly) startAttack(friendly) hightlightEnemyUnitAreas(allOrigins) addAttackOriginListeners(allOrigins) attackOriginHandler(event) removeAttackOriginListeners(allOrigins) addAttackedAreaListeners(allOrigins) attackDestinationHandler(event) removeAttackDestinationListeners(allOrigins

## intergrated2

loadHandler()

#### start

addSplashListerner() removeSplashListerner() continueCampaignHandler() startCampaignHandler() startGame()

# unitSetUp

var turn = 1

defaultSetUp() displayUnits() displayAllied(mapX, mapY, unitIndex) displayGerman(mapX, mapY, unitIndex)

#### <u>Area</u>

id:string terrain:string canSupplyAllied:boolean canSupplyGerman:boolean control:string centerX:int centerY:int

vp:int

fullName:string isAttacker:boolean isDefending:boolean

becomeAttacker(true) becomeDefender(true) cancelAttack(false)

cancelDefence(false)

addUnit(unit)

get areaName()

addNeighbour(aNeighbour) addAttackArrow(anAttackArrow)

findArrowOptions(id)

reportGermanUnits()

reportAlliedUnits()

findAdjacentEneamyAreas(targetAreaId) findAdjacentFriendlyAreas(targetAreaId)

reportAreas()

allUnits:[]

allAreas:[]

allMoves: []

allAttacks:[

getUnitsInArea(targetAreaId)

getArea(targetAreaId)

addNeighbours(targetAreaId, allNeighbourIds)

findAttackArrows(targetAreaId)

displayAttackArrows(attackAreaId, allNeighbourIds)

**Game** 

addUnit(id, name, army, training, branch, startTurn, startPosition)

addAttackArrows(attackAreaId, allNeighbourIds)

addAttack()

clearAllAttackers( allAreas)

clearAllDefenders( allAreas)

## Clock

currentTurn:int totalTurn:int allDates:[] allRules[] allDescriptions:[]

get turn() get date() get specialRules() get description() nextTurn()

#### Move

originId destinationId unitId

setOriginId(id) setDestinationId(id) setUnitId(id)

## <u>Unit</u>

id:string name:string army:string training:string branch:string startTurn:int startPosition:int

get name() get army()

## Attack

\_attackingAreaId:string defendingAreaId:string turn:int

setAttackingArea(id) setDefendingAreaId(id)