

arrows

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

moving

var destinationId  
var originId  
var moveDestinationIds  
var fromAreas

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

highLights

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

unitSetUp

var turn = 1

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

attacking

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

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

start

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

intergrated2

loadHandler()

Area

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

addUnit(unit)  
get areaName()  
addNeighbour(aNeighbour)  
addAttackArrow(anAttackArrow)  
findArrowOptions(id)

Clock

\_currentTurn:int  
\_totalTurn:int  
\_allDates:[]  
\_allRules[]  
\_allDescriptions:[]

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

Attack

\_attackingAreaId:string  
\_defendingAreaId:string

setAttackingArea(id)  
setDefendingAreaId(id)

Game

\_allUnits:[]  
\_allAreas:[]  
\_allMoves: []  
\_allAttacks:[]  
allReinforcementUnits = []  
\_clock

addUnit(id, name, army, training, branch, startTurn, startPosition)  
addReinforcementUnit(id, name, army, training, branch, startTurn, startPosition)  
reportAlliedUnits()  
reportGermanUnits()  
findAdjacentEneamyAreas(targetAreaId)  
findAdjacentFriendlyAreas(targetAreaId)  
addArea(id, terrain, canSupplyAllied, canSupplyGerman, control, centerX, centerY, vp = 0, fullName = "")  
reportAreas()  
getUnitsInArea(targetAreaId)  
getArea(targetAreaId)  
addNeighbours(targetAreaId, allNeighbourIds)  
findAttackArrows(targetAreaId)  
displayAttackArrows(attackAreaId, allNeighbourIds)  
addAttackArrows(attackAreaId, allNeighbourIds)  
addMove(originId, destinationId, unitId)

Unit

id:string  
name:string  
army:string  
training:string  
branch:string  
startTurn:int  
startPosition:string  
currentPosition:string  
hasMoved :boolean false

get name()  
get army()  
clearMove()  
makeMove()

Move

\_turn  
\_originId  
\_destinationId  
\_unitId