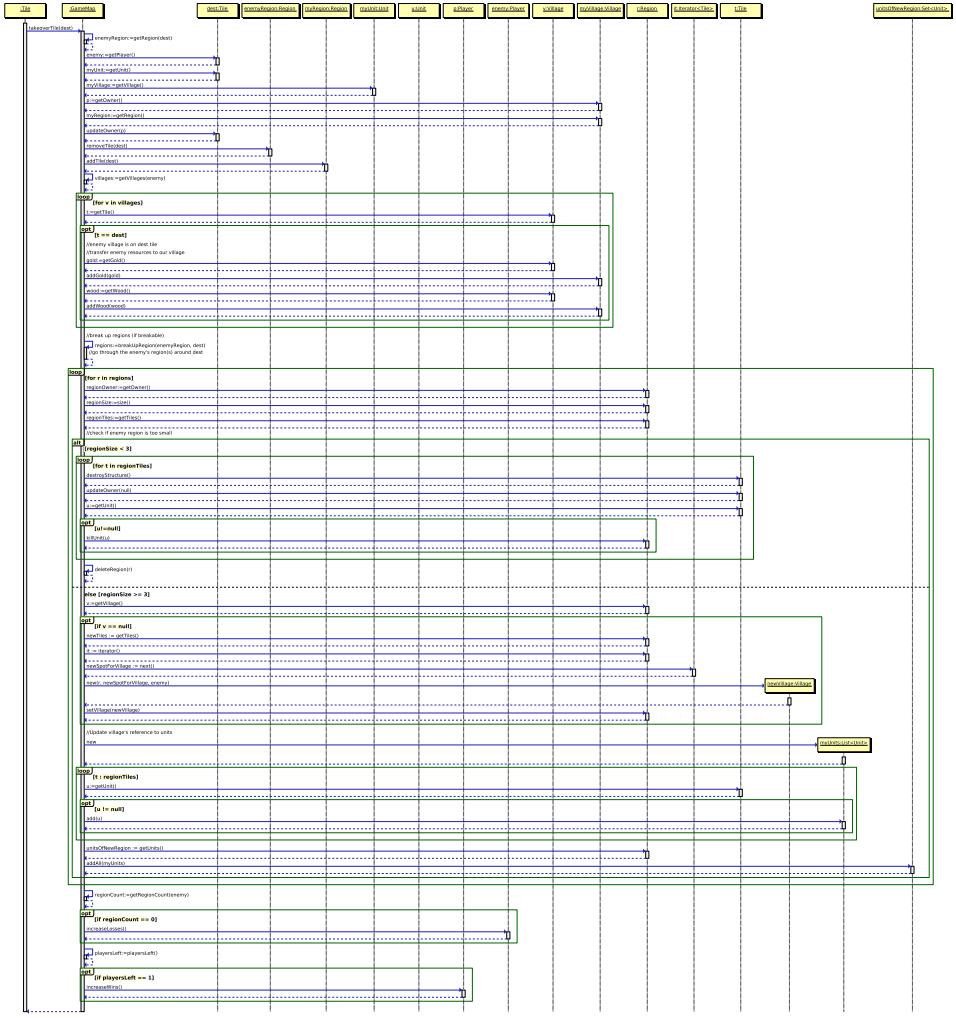
:Tile
:GameMap
dest:Tile
enemyRegion:Region
myRegion:Region
myUnit:Unit
u:Unit
e:Player
enemy:Player
v:Village
myVillage:Village
r:Region
tileIterator:Tile
t:Tile
unitsOfNewRegion:Set<Unit>
takeover(Tile dest)
enemyRegion:=getRegion(dest)
enemy:=getPlayer()
myUnit:=getUnit()
myVillage:=getVillage()
e:=getOwner()
myRegion:=getRegion()
updateOwner(p)
removeTile(dest)
addTile(dest)
villages:=getVillages(enemy)
loop
[for v in villages]
t:=getTiles()
opt
[t == dest]
//enemy village is on dest tile
//transfer enemy resources to our village
gold:=getGold()
addGold(gold)
wood:=getWood()
addWood(wood)
//break up regions (if breakable)
regions:=breakUpRegion(enemyRegion, dest)
//go through the enemy's region(s) around dest
loop
[for r in regions]
regionOwner:=getOwner()
regionSize:=size()
regionTiles:=getTiles()
//check if enemy region is too small
alt
[regionSize < 3]
loop
[for t in regionTiles]
destroyStructure()
updateOwner(null)
u:=getUnit()
opt
[u!=null]
killUnit(u)
deleteRegion()
else [regionSize >= 3]
v:=getVillage()
opt
[if v == null]
newTiles := getTiles()
t := Iterator()
newSpotForVillage := next()
new(r, newSpotForVillage, enemy)
:NewVillage:Village
addVillage(newVillage)
//Update village's reference to units
save
:myUnits:List<Unit>
loop
[t : regionTiles]
u:=getUnit()
opt
[u != null]
add(u)
unitsOfNewRegion. := getUnits()
addAll(myUnits)
regionCount:=getRegionCount(enemy)
opt
[if regionCount == 0]
increaseLosses()
playersLeft:=playersLeft()
opt
[if playersLeft == 1]
increaseWins()