《冰雪大作战》代码

Config = {}

------------------ t\_name ------------------

Config.t\_name = {}

function Config.parse\_t\_name()

local dbc\_name = dbc.New()

dbc\_name:load\_txt("t\_name")

local firname = {}

local lastname = {}

for i = 0, dbc\_name:get\_y() - 1 do

local fir = dbc\_name:get\_string(0, i)

local last = dbc\_name:get\_string(1, i)

if(fir ~= '') then

table.insert(firname, fir)

end

if(last ~= '') then

table.insert(lastname, last)

end

end

table.insert(Config.t\_name, firname)

table.insert(Config.t\_name, lastname)

end

function Config.get\_t\_name()

local fir = Config.t\_name[1][math.random(#Config.t\_name[1])]

local last = Config.t\_name[2][math.random(#Config.t\_name[2])]

return table.concat({fir, last})

end

function Config.get\_battle\_random\_name()

local fir = Config.t\_name[1][BattleOperation.random(1, #Config.t\_name[1] + 1)]

local last = Config.t\_name[2][BattleOperation.random(1, #Config.t\_name[2] + 1)]

return table.concat({fir, last})

end

----------------------------------------------

------------------ t\_skill ------------------

Config.t\_skill = {}

function Config.parse\_t\_skill()

local dbc\_skill = dbc.New()

dbc\_skill:load\_txt("t\_skill")

for i = 0, dbc\_skill:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_skill:get\_int(0, i)

tdic.name = dbc\_skill:get\_string(1, i)

tdic.level = dbc\_skill:get\_int(2, i)

tdic.type = dbc\_skill:get\_int(3, i)

tdic.cd = dbc\_skill:get\_int(4, i)

tdic.release\_type = dbc\_skill:get\_int(5, i)

tdic.range = dbc\_skill:get\_int(6, i)

tdic.range\_param = dbc\_skill:get\_int(7, i)

tdic.action = dbc\_skill:get\_string(8, i)

tdic.qy\_action\_time = dbc\_skill:get\_int(9, i)

tdic.hy\_action\_time = dbc\_skill:get\_int(10, i)

tdic.link\_effect = dbc\_skill:get\_int(11, i)

tdic.icon = dbc\_skill:get\_string(12, i)

tdic.get\_range = function(bp)

local range = tdic.range

if tdic.type == 1 then

range = bp.attr.range()

elseif tdic.type == 2 then

range = BattleOperation.toInt(range \* (1 + bp.get\_skill\_attr\_value(tdic.id, 2) / 100))

end

return range

end

tdic.get\_range\_param = function(bp)

local range\_param = tdic.range\_param

if tdic.skill\_type == 1 then

range\_param = range\_param + 1000 \* bp.player.level\_big

if range\_param > 20000 then

range\_param = 20000

end

elseif tdic.type == 2 and (tdic.release\_type == 2 or tdic.release\_type == 3) then

range\_param = BattleOperation.toInt(range\_param \* (1 + bp.get\_skill\_attr\_value(tdic.id, 3) / 100))

end

return range\_param

end

tdic.get\_cd = function(bp)

scd = tdic.cd

-- 减特殊雪球cd

if tdic.type == 2 then

local v = 1 - (bp.attr\_value[23] + bp.get\_skill\_attr\_value(tdic.id, 7)) / 100

scd = BattleOperation.toInt(scd \* v)

end

--

-- 减变雪人cd

if tdic.type == 3 then

local v = 1 - bp.attr\_value[24] / 100

scd = BattleOperation.toInt(scd \* v)

end

--

return scd

end

if Config.t\_skill[tdic.id] == nil then

Config.t\_skill[tdic.id] = {}

end

Config.t\_skill[tdic.id][tdic.level] = tdic

end

end

function Config.get\_t\_skill(id, level)

if Config.t\_skill[id] == nil then

return nil

else

if Config.t\_skill[id][level] == nil then

return nil

else

return Config.t\_skill[id][level]

end

end

end

----------------------------------------------

------------------ t\_skill\_effect ------------------

Config.t\_skill\_effect = {}

function Config.parse\_t\_skill\_effect()

local dbc\_skill\_effect = dbc.New()

dbc\_skill\_effect:load\_txt("t\_skill\_effect")

for i = 0, dbc\_skill\_effect:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_skill\_effect:get\_int(0, i)

tdic.name = dbc\_skill\_effect:get\_string(1, i)

tdic.skill\_id = dbc\_skill\_effect:get\_int(2, i)

tdic.skill\_type = dbc\_skill\_effect:get\_int(3, i)

tdic.type = dbc\_skill\_effect:get\_int(4, i)

tdic.range = dbc\_skill\_effect:get\_int(5, i)

tdic.range\_param = dbc\_skill\_effect:get\_int(6, i)

tdic.effect = dbc\_skill\_effect:get\_string(7, i)

tdic.xiaoshi\_effect = dbc\_skill\_effect:get\_string(8, i)

tdic.is\_zaxs = dbc\_skill\_effect:get\_int(9, i)

tdic.xiaoshi\_effect1 = dbc\_skill\_effect:get\_string(10, i)

tdic.is\_jzxs = dbc\_skill\_effect:get\_int(11, i)

tdic.xiaoshi\_effect2 = dbc\_skill\_effect:get\_string(12, i)

tdic.effect\_scale = dbc\_skill\_effect:get\_int(13, i) / 100

tdic.fx\_speed = dbc\_skill\_effect:get\_int(14, i)

tdic.fx\_hight = dbc\_skill\_effect:get\_int(15, i)

tdic.fx\_hight1 = dbc\_skill\_effect:get\_int(16, i)

tdic.fx\_hight2 = dbc\_skill\_effect:get\_int(17, i)

tdic.sh\_time = dbc\_skill\_effect:get\_int(18, i)

tdic.time = dbc\_skill\_effect:get\_int(19, i)

tdic.target\_type = dbc\_skill\_effect:get\_int(20, i)

tdic.dd\_type = dbc\_skill\_effect:get\_int(21, i)

tdic.dd\_jg = dbc\_skill\_effect:get\_double(22, i)

tdic.is\_zp = dbc\_skill\_effect:get\_int(23, i)

tdic.link\_xiaoguo = dbc\_skill\_effect:get\_int(24, i)

tdic.link\_effect = dbc\_skill\_effect:get\_int(25, i)

tdic.link\_effect1 = dbc\_skill\_effect:get\_int(26, i)

tdic.get\_range = function(bp)

local range = tdic.range

if tdic.skill\_type == 1 then

range = bp.attr.range()

elseif tdic.skill\_type == 2 then

range = BattleOperation.toInt(range \* (1 + bp.get\_skill\_attr\_value(tdic.skill\_id, 2) / 100))

end

return range

end

tdic.get\_range\_param = function(bp)

local range\_param = tdic.range\_param

if tdic.skill\_type == 1 then

range\_param = range\_param + 1000 \* bp.player.level\_big

if range\_param > 20000 then

range\_param = 20000

end

elseif tdic.skill\_type == 2 then

range\_param = BattleOperation.toInt(range\_param \* (1 + bp.get\_skill\_attr\_value(tdic.skill\_id, 3) / 100))

end

return range\_param

end

tdic.get\_effect\_scale = function(bp)

local scale = tdic.effect\_scale

if tdic.skill\_type == 1 then

scale = scale + 0.3 \* bp.player.level\_big

if scale > 3 then

scale = 3

end

elseif tdic.skill\_type == 2 then

scale = BattleOperation.toInt(scale \* (1 + bp.get\_skill\_attr\_value(tdic.skill\_id, 3) / 100))

end

return scale

end

tdic.get\_fx\_speed = function(bp)

local fx\_speed = tdic.fx\_speed

if tdic.skill\_type == 1 then

fx\_speed = fx\_speed + 10000 \* bp.player.level\_fspeed

if fx\_speed > 200000 then

fx\_speed = 200000

end

elseif tdic.type == 2 then

fx\_speed = BattleOperation.toInt(fx\_speed \* (1 + bp.get\_skill\_attr\_value(tdic.skill\_id, 4) / 100))

end

return fx\_speed

end

Config.t\_skill\_effect[tdic.id] = tdic

end

end

function Config.get\_t\_skill\_effect(id)

if Config.t\_skill\_effect[id] == nil then

return nil

else

return Config.t\_skill\_effect[id]

end

end

----------------------------------------------

------------------ t\_skill\_xiaoguo ------------------

Config.t\_skill\_xiaoguo = {}

function Config.parse\_t\_skill\_xiaoguo()

local dbc\_skill\_xiaoguo = dbc.New()

dbc\_skill\_xiaoguo:load\_txt("t\_skill\_xiaoguo")

for i = 0, dbc\_skill\_xiaoguo:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_skill\_xiaoguo:get\_int(0, i)

tdic.name = dbc\_skill\_xiaoguo:get\_string(1, i)

tdic.sj\_effect = dbc\_skill\_xiaoguo:get\_string(2, i)

tdic.dmg\_per = dbc\_skill\_xiaoguo:get\_double(3, i)

tdic.dmg\_gd = dbc\_skill\_xiaoguo:get\_double(4, i)

tdic.jf\_type = dbc\_skill\_xiaoguo:get\_int(5, i)

tdic.jf\_dis = dbc\_skill\_xiaoguo:get\_int(6, i)

tdic.jf\_speed = dbc\_skill\_xiaoguo:get\_int(7, i)

tdic.link\_buff = dbc\_skill\_xiaoguo:get\_int(8, i)

Config.t\_skill\_xiaoguo[tdic.id] = tdic

end

end

function Config.get\_t\_skill\_xiaoguo(id)

if Config.t\_skill\_xiaoguo[id] == nil then

return nil

else

return Config.t\_skill\_xiaoguo[id]

end

end

----------------------------------------------

------------------ t\_battle\_item ------------------

Config.t\_battle\_item = {}

function Config.parse\_t\_battle\_item()

local dbc\_battle\_item = dbc.New()

dbc\_battle\_item:load\_txt("t\_battle\_item")

for i = 0, dbc\_battle\_item:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_battle\_item:get\_int(0, i)

tdic.name = dbc\_battle\_item:get\_string(1, i)

tdic.type = dbc\_battle\_item:get\_int(2, i)

tdic.effect = dbc\_battle\_item:get\_string(3, i)

tdic.geffect = dbc\_battle\_item:get\_string(4, i)

tdic.skill = dbc\_battle\_item:get\_int(5, i)

tdic.max = dbc\_battle\_item:get\_int(6, i)

tdic.desc = dbc\_battle\_item:get\_string(8, i)

tdic.desc = Config.get\_t\_lang(tdic.desc)

Config.t\_battle\_item[tdic.id] = tdic

end

end

function Config.get\_t\_battle\_item(id)

if Config.t\_battle\_item[id] == nil then

return nil

else

return Config.t\_battle\_item[id]

end

end

----------------------------------------------

------------------ t\_battle\_exp ------------------

Config.t\_battle\_exp = {}

function Config.parse\_t\_battle\_exp()

local dbc\_battle\_exp = dbc.New()

dbc\_battle\_exp:load\_txt("t\_battle\_exp")

for i = 0, dbc\_battle\_exp:get\_y() - 1 do

local tdic = {}

tdic.level = dbc\_battle\_exp:get\_int(0, i)

tdic.exp = dbc\_battle\_exp:get\_int(1, i)

tdic.gexp = dbc\_battle\_exp:get\_int(2, i)

tdic.item\_num1 = dbc\_battle\_exp:get\_int(3, i)

tdic.item\_rate1 = dbc\_battle\_exp:get\_int(4, i)

tdic.item\_num2 = dbc\_battle\_exp:get\_int(5, i)

tdic.item\_rate2 = dbc\_battle\_exp:get\_int(6, i)

tdic.scale = dbc\_battle\_exp:get\_int(7, i)

tdic.dis = dbc\_battle\_exp:get\_int(8, i)

Config.t\_battle\_exp[tdic.level] = tdic

end

end

function Config.get\_t\_battle\_exp(level)

if Config.t\_battle\_exp[level] == nil then

return nil

else

return Config.t\_battle\_exp[level]

end

end

----------------------------------------------

------------------ t\_battle\_buff ------------------

Config.t\_battle\_buff = {}

function Config.parse\_t\_battle\_buff()

local dbc\_battle\_buff = dbc.New()

dbc\_battle\_buff:load\_txt("t\_battle\_buff")

for i = 0, dbc\_battle\_buff:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_battle\_buff:get\_int(0, i)

tdic.name = dbc\_battle\_buff:get\_string(1, i)

tdic.effect\_type = dbc\_battle\_buff:get\_int(2, i)

tdic.effect = dbc\_battle\_buff:get\_string(3, i)

tdic.effect\_end = dbc\_battle\_buff:get\_string(4, i)

tdic.effect\_bone = dbc\_battle\_buff:get\_string(5, i)

tdic.ptype = dbc\_battle\_buff:get\_int(6, i)

tdic.param1 = dbc\_battle\_buff:get\_int(7, i)

tdic.param2 = dbc\_battle\_buff:get\_int(8, i)

tdic.time = dbc\_battle\_buff:get\_int(9, i)

Config.t\_battle\_buff[tdic.id] = tdic

end

end

function Config.get\_t\_battle\_buff(id)

if Config.t\_battle\_buff[id] == nil then

return nil

else

return Config.t\_battle\_buff[id]

end

end

----------------------------------------------

---------------------t\_role\_skill----------------------------

Config.t\_role\_skill = {}

function Config.parse\_t\_role\_skill()

local dbc\_role\_skill = dbc.New()

dbc\_role\_skill:load\_txt("t\_role\_skill")

for i = 0, dbc\_role\_skill:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_role\_skill:get\_int(0, i)

tdic.name = dbc\_role\_skill:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.desc = dbc\_role\_skill:get\_string(4, i)

tdic.desc = Config.get\_t\_lang(tdic.desc)

tdic.type = dbc\_role\_skill:get\_int(5, i)

tdic.param1 = dbc\_role\_skill:get\_int(6, i)

tdic.param2 = dbc\_role\_skill:get\_int(7, i)

tdic.param3 = dbc\_role\_skill:get\_int(8, i)

tdic.param4 = dbc\_role\_skill:get\_int(9, i)

Config.t\_role\_skill[tdic.id] = tdic

tdic.param\_value = function(level)

return tdic.param3 + tdic.param4 \* (level - 1)

end

end

end

function Config.get\_t\_role\_skill(id)

if Config.t\_role\_skill[id] == nil then

return nil

else

return Config.t\_role\_skill[id]

end

end

-------------------------------------------------------------

---------------------t\_role\_buff-----------------------------

Config.t\_role\_buff = {}

function Config.parse\_t\_role\_buff()

local dbc\_role\_buff = dbc.New()

dbc\_role\_buff:load\_txt("t\_role\_buff")

for i = 0, dbc\_role\_buff:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_role\_buff:get\_int(0, i)

tdic.name = dbc\_role\_buff:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.desc = dbc\_role\_buff:get\_string(4, i)

tdic.desc = Config.get\_t\_lang(tdic.desc)

tdic.type = dbc\_role\_buff:get\_int(5, i)

tdic.param1 = dbc\_role\_buff:get\_int(6, i)

tdic.param2 = dbc\_role\_buff:get\_int(7, i)

tdic.param3 = dbc\_role\_buff:get\_int(8, i)

tdic.param4 = dbc\_role\_buff:get\_int(9, i)

Config.t\_role\_buff[tdic.id] = tdic

tdic.param\_value = function(level)

return tdic.param3 + tdic.param4 \* (level - 1)

end

end

end

function Config.get\_t\_role\_buff(id)

if Config.t\_role\_buff[id] == nil then

return nil

else

return Config.t\_role\_buff[id]

end

end

-------------------------------------------------------------

---------------------t\_role\_level----------------------------

Config.t\_role\_level = {}

Config.max\_role\_lev = 0

function Config.parse\_t\_role\_level()

local dbc\_role\_level = dbc.New()

dbc\_role\_level:load\_txt("t\_role\_level")

for i = 0, dbc\_role\_level:get\_y() - 1 do

local tdic = {}

tdic.level = dbc\_role\_level:get\_int(0, i)

tdic.suipian\_cost = dbc\_role\_level:get\_int(1, i)

tdic.gold\_cost = {}

tdic.exp = {}

for j = 1, 3 do

table.insert(tdic.gold\_cost, dbc\_role\_level:get\_int(j \* 2, i))

table.insert(tdic.exp, dbc\_role\_level:get\_int( j \* 2 + 1, i))

end

tdic.get\_gold\_cost = function(color)

local gold = tdic.gold\_cost[color]

if color == 2 then

gold = BattleOperation.toInt(gold \* (100 - PlayerData.out\_attr[3]) / 100)

elseif color == 3 then

gold = BattleOperation.toInt(gold \* (100 - PlayerData.out\_attr[4]) / 100)

end

return gold

end

Config.t\_role\_level[tdic.level] = tdic

Config.max\_role\_lev = Config.max\_role\_lev + 1

end

end

function Config.get\_t\_role\_level(level)

if Config.t\_role\_level[level] == nil then

return nil

else

return Config.t\_role\_level[level]

end

end

-------------------------------------------------------------

------------------ t\_item --------------------

Config.t\_item = {}

Config.t\_item\_ids = {}

function Config.parse\_t\_item()

local dbc\_item = dbc.New()

dbc\_item:load\_txt("t\_item")

for i = 0, dbc\_item:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_item:get\_int(0, i)

tdic.name = dbc\_item:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.color = dbc\_item:get\_int(3, i)

tdic.name = self.font\_color[tdic.color]..tdic.name.."[-]"

tdic.type = dbc\_item:get\_int(4, i)

tdic.level = dbc\_item:get\_int(5, i)

tdic.desc = dbc\_item:get\_string(7, i)

tdic.desc = Config.get\_t\_lang(tdic.desc)

tdic.icon = dbc\_item:get\_string(8, i)

tdic.price = dbc\_item:get\_int(9, i)

tdic.sell\_type = dbc\_item:get\_int(10, i)

tdic.sell = dbc\_item:get\_int(11, i)

tdic.def1 = dbc\_item:get\_int(12, i)

tdic.def2 = dbc\_item:get\_int(13, i)

tdic.def3 = dbc\_item:get\_int(14, i)

tdic.def4 = dbc\_item:get\_int(15, i)

Config.t\_item[tdic.id] = tdic

table.insert(Config.t\_item\_ids, tdic)

end

end

function Config.get\_t\_item(id)

if Config.t\_item[id] == nil then

return nil

else

return Config.t\_item[id]

end

end

----------------------------------------------

--------------------t\_item\_box----------------

Config.t\_item\_box = {}

function Config.parse\_t\_item\_box()

local dbc\_item\_box = dbc.New()

dbc\_item\_box:load\_txt("t\_itembox")

for i = 0, dbc\_item\_box:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_item\_box:get\_int(0, i)

tdic.index = dbc\_item\_box:get\_int(2, i)

Config.t\_item\_box[tdic.id] = tdic

end

end

function Config.get\_t\_item\_box(id)

if Config.t\_item\_box[id] == nil then

return nil

else

return Config.t\_item\_box[id]

end

end

-----------------------------------------------

----------------------t\_resource---------------

Config.t\_resource = {}

function Config.parse\_t\_resource()

local dbc\_resource = dbc.New()

dbc\_resource:load\_txt("t\_resource")

for i = 0, dbc\_resource:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_resource:get\_int(0, i)

tdic.name = dbc\_resource:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.color = dbc\_resource:get\_int(3, i)

tdic.name = self.font\_color[tdic.color]..tdic.name.."[-]"

tdic.icon = dbc\_resource:get\_string(5, i)

tdic.small\_icon = dbc\_resource:get\_string(7, i)

tdic.mid\_icon = dbc\_resource:get\_string(6, i)

tdic.desc = dbc\_resource:get\_string(9, i)

tdic.desc = Config.get\_t\_lang(tdic.desc)

Config.t\_resource[tdic.id] = tdic

end

end

function Config.get\_t\_resource(id)

if Config.t\_resource[id] == nil then

return nil

else

return Config.t\_resource[id]

end

end

-----------------------------------------------

------------------ t\_role --------------------

Config.t\_role = {}

Config.t\_role\_ids = {}

function Config.parse\_t\_role()

local dbc\_role = dbc.New()

dbc\_role:load\_txt("t\_role")

for i = 0, dbc\_role:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_role:get\_int(0, i)

tdic.name = dbc\_role:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.color = dbc\_role:get\_int(3, i)

tdic.name = self.font\_color[tdic.color]..tdic.name.."[-]"

tdic.sex = dbc\_role:get\_int(4, i)

tdic.icon = dbc\_role:get\_string(5, i)

tdic.res = dbc\_role:get\_string(6, i)

tdic.desc = dbc\_role:get\_string(7, i)

tdic.desc = Config.get\_t\_lang(tdic.desc)

tdic.yy = {}

for j = 0, 3 do

local yy = dbc\_role:get\_string(8 + j, i)

table.insert(tdic.yy, yy)

end

tdic.hp = dbc\_role:get\_int(12, i)

tdic.hp\_add = dbc\_role:get\_int(13, i)

tdic.atk = dbc\_role:get\_int(14, i)

tdic.atk\_add = dbc\_role:get\_int(15, i)

tdic.def = dbc\_role:get\_int(16, i)

tdic.def\_add = dbc\_role:get\_int(17, i)

tdic.range = dbc\_role:get\_int(18, i)

tdic.aspeed = dbc\_role:get\_int(19, i)

tdic.speed = dbc\_role:get\_int(20, i)

tdic.gskills = {}

for k = 0, 2 do

local bs = dbc\_role:get\_int(21 + k, i)

if bs > 0 then

table.insert(tdic.gskills, bs)

end

end

tdic.bskills = {}

for k = 0, 2 do

local bs = dbc\_role:get\_int(24 + k, i)

if bs > 0 then

table.insert(tdic.bskills, bs)

end

end

tdic.suipian\_id = dbc\_role:get\_int(27, i)

tdic.suipian\_cost = dbc\_role:get\_int(28, i)

tdic.type = dbc\_role:get\_string(29, i)

tdic.type = Config.get\_t\_lang(tdic.type)

tdic.exp = dbc\_role:get\_int(30, i)

Config.t\_role[tdic.id] = tdic

table.insert(Config.t\_role\_ids, tdic.id)

end

end

function Config.get\_t\_role(id)

if Config.t\_role[id] == nil then

return nil

else

return Config.t\_role[id]

end

end

function Config.get\_battle\_random\_role()

return Config.t\_role\_ids[BattleOperation.random(1, #Config.t\_role\_ids + 1)]

end

function Config.get\_random\_role()

if Config.t\_role[id] == nil then

return nil

else

return Config.t\_role[id]

end

end

----------------------------------------------

------------------ t\_avatar --------------------

Config.t\_avatar = {}

function Config.parse\_t\_avatar()

local dbc\_avatar = dbc.New()

dbc\_avatar:load\_txt("t\_avatar")

for i = 0, dbc\_avatar:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_avatar:get\_int(0, i)

tdic.name = dbc\_avatar:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.role\_id = dbc\_avatar:get\_int(3, i)

tdic.icon = dbc\_avatar:get\_string(4, i)

tdic.desc1 = dbc\_avatar:get\_string(5, i)

tdic.desc1 = Config.get\_t\_lang(tdic.desc1)

tdic.desc2 = dbc\_avatar:get\_string(6, i)

tdic.desc2 = Config.get\_t\_lang(tdic.desc2)

Config.t\_avatar[tdic.id] = tdic

end

end

function Config.get\_t\_avatar(id)

if Config.t\_avatar[id] == nil then

return nil

else

return Config.t\_avatar[id]

end

end

function Config.get\_t\_avatar\_id(role\_id)

for k, v in pairsByKeys(Config.t\_avatar) do

if(role\_id == v.role\_id) then

return v

end

end

return nil

end

----------------------------------------------

------------------ t\_toukuang --------------------

Config.t\_toukuang = {}

Config.t\_toukuang\_ids = {}

function Config.parse\_t\_toukuang()

local dbc\_toukuang = dbc.New()

dbc\_toukuang:load\_txt("t\_toukuang")

for i = 0, dbc\_toukuang:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_toukuang:get\_int(0, i)

tdic.name = dbc\_toukuang:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.big\_icon = dbc\_toukuang:get\_string(3, i)

tdic.color = dbc\_toukuang:get\_int(4, i)

tdic.desc1 = dbc\_toukuang:get\_string(5, i)

tdic.desc1 = Config.get\_t\_lang(tdic.desc1)

tdic.desc2 = dbc\_toukuang:get\_string(6, i)

tdic.desc2 = Config.get\_t\_lang(tdic.desc2)

tdic.type = dbc\_toukuang:get\_int(7, i)

tdic.param1 = dbc\_toukuang:get\_int(8, i)

tdic.param2 = dbc\_toukuang:get\_int(9, i)

tdic.param3 = dbc\_toukuang:get\_int(10, i)

tdic.desc = tdic.desc1.."\n"..tdic.desc2

tdic.icon = dbc\_toukuang:get\_string(11, i)

Config.t\_toukuang[tdic.id] = tdic

table.insert(Config.t\_toukuang\_ids, tdic.id)

end

end

function Config.get\_t\_toukuang(id)

if Config.t\_toukuang[id] == nil then

return nil

else

return Config.t\_toukuang[id]

end

end

function Config.get\_battle\_random\_toukuang()

return Config.t\_toukuang\_ids[BattleOperation.random(1, #Config.t\_toukuang\_ids + 1)]

end

----------------------------------------------

-------------------t\_lang-----------------------

Config.t\_lang = {}

function Config.parse\_t\_lang()

local dbc\_lang = dbc.New()

dbc\_lang:load\_txt("t\_lang")

for i = 0, dbc\_lang:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_lang:get\_string(0, i)

tdic.lang = {}

for j = 1, dbc\_lang:get\_x() - 1 do

local lang = string.gsub(dbc\_lang:get\_string(j, i), "##", "\n")

table.insert(tdic.lang, lang)

end

Config.t\_lang[tdic.id] = tdic

end

end

function Config.get\_t\_lang(id)

if Config.t\_lang[id] == nil then

return ""

else

return Config.t\_lang[id].lang[self.lang]

end

end

------------------------------------------------

-----------------t\_client\_lang----------------------

Config.t\_client\_lang = {}

function Config.parse\_t\_client\_lang()

local dbc\_client\_lang = dbc.New()

dbc\_client\_lang:load\_txt("t\_lang\_client")

for i = 0, dbc\_client\_lang:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_client\_lang:get\_string(0, i)

tdic.lang = {}

for j = 1, dbc\_client\_lang:get\_x() - 1 do

table.insert(tdic.lang, dbc\_client\_lang:get\_string(j, i))

end

Config.t\_client\_lang[tdic.id] = tdic

end

end

function Config.get\_t\_client\_lang(id)

if Config.t\_client\_lang[id] == nil then

return nil

else

return Config.t\_client\_lang[id]

end

end

----------------------------------------------------

----------------------t\_biaoqing--------------------------

Config.t\_biaoqing = {}

function Config.parse\_t\_biaoqing()

local dbc\_biaoqing = dbc.New()

dbc\_biaoqing:load\_txt('t\_biaoqing')

for i = 0, dbc\_biaoqing:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_biaoqing:get\_int(0, i)

tdic.desc = dbc\_biaoqing:get\_string(1, i)

tdic.icon = dbc\_biaoqing:get\_string(2, i)

Config.t\_biaoqing[tdic.id] = tdic

end

end

function Config.get\_t\_biaoqing(id)

if Config.t\_biaoqing[id] == nil then

return nil

else

return Config.t\_biaoqing[id]

end

end

----------------------------------------------

-----------------------t\_region----------------------------

Config.t\_province = {}

Config.t\_city = {}

function Config.parse\_t\_region()

local dbc\_region = dbc.New()

dbc\_region:load\_txt('t\_address')

for i = 0, dbc\_region:get\_y() - 1 do

local tdic = {}

tdic.country = dbc\_region:get\_string(0, i)

tdic.province = dbc\_region:get\_string(1, i)

tdic.city = dbc\_region:get\_string(2, i)

if(Config.t\_province[tdic.country] == nil) then

Config.t\_province[tdic.country] = {}

end

if(Config.t\_city[tdic.province] == nil) then

Config.t\_city[tdic.province] = {}

end

if(not table\_contain(Config.t\_province[tdic.country], tdic.province)) then

table.insert(Config.t\_province[tdic.country], tdic.province)

end

table.insert(Config.t\_city[tdic.province], tdic.city)

end

end

function Config.get\_t\_province(country)

if Config.t\_province[country] == nil then

return nil

else

return Config.t\_province[country]

end

end

function Config.get\_t\_city(province)

if Config.t\_city[province] == nil then

return nil

else

return Config.t\_city[province]

end

end

-----------------------------------------------------------

-----------------------t\_exp----------------------------------

Config.t\_exp = {}

function Config.parse\_t\_exp()

local dbc\_exp = dbc.New()

dbc\_exp:load\_txt('t\_exp')

for i = 0, dbc\_exp:get\_y() - 1 do

local tdic = {}

tdic.level = dbc\_exp:get\_int(0, i)

tdic.exp = dbc\_exp:get\_int(1, i)

tdic.level\_add = {}

for j = 0, 6 do

local add = {}

add.type = dbc\_exp:get\_int(j \* 4 + 2 , i)

if(add.type ~= 0) then

add.param1 = dbc\_exp:get\_int(j \* 4 + 3, i)

add.param2 = dbc\_exp:get\_int(j \* 4 + 4, i)

add.param3 = dbc\_exp:get\_int(j \* 4 + 5, i)

table.insert(tdic.level\_add, add)

end

end

Config.t\_exp[tdic.level] = tdic

end

end

function Config.get\_t\_exp(level)

if Config.t\_exp[level] == nil then

return nil

else

return Config.t\_exp[level]

end

end

function Config.get\_t\_acc\_add(type, id)

if(type == 1) then

return nil

elseif(type == 2) then

return nil

elseif(type == 3) then

return Config.get\_t\_attr(id)

end

end

---------------------------------------------------------------

-------------------------t\_cup----------------------------------

Config.t\_cup = {}

function Config.parse\_t\_cup()

local dbc\_cup = dbc.New()

dbc\_cup:load\_txt('t\_cup')

for i = 0, dbc\_cup:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_cup:get\_int(0, i)

tdic.name = dbc\_cup:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.num = dbc\_cup:get\_int(3, i)

tdic.icon = dbc\_cup:get\_string(4, i)

tdic.dai\_icon = dbc\_cup:get\_string(5, i)

tdic.star = dbc\_cup:get\_int(6, i)

tdic.down = dbc\_cup:get\_int(7, i)

tdic.sb = dbc\_cup:get\_int(8, i)

tdic.jb = dbc\_cup:get\_int(9, i)

tdic.rewards = {}

for j = 0, 2 do

local reward = {}

reward.type = dbc\_cup:get\_int(10 + j \* 4, i)

reward.value1 = dbc\_cup:get\_int(11 + j \* 4, i)

reward.value2 = dbc\_cup:get\_int(12 + j \* 4, i)

reward.value3 = dbc\_cup:get\_int(13 + j \* 4, i)

if reward.type ~= 0 then

table.insert(tdic.rewards, reward)

end

end

table.insert(Config.t\_cup, tdic)

end

end

function Config.get\_t\_cup\_by\_level(level)

if level <= 0 or level > #Config.t\_cup then

return nil

end

return Config.t\_cup[level]

end

function Config.get\_cup\_by\_cup(cup)

if cup <= 0 then

return Config.t\_cup[1]

end

for i = 1, #Config.t\_cup do

if Config.t\_cup[i].num > cup then

return Config.t\_cup[i - 1]

end

end

return Config.t\_cup[#Config.t\_cup]

end

function Config.get\_sj\_cup(cup, rank)

local t\_cup = Config.get\_cup\_by\_cup(cup, rank)

return t\_cup.sb - BattleOperation.toInt((t\_cup.sb - t\_cup.jb) \* (rank - 1) / 19)

end

----------------------------------------------------------------

--------------------------t\_shop--------------------------------

Config.t\_shop = {}

function Config.parse\_t\_shop()

local dbc\_shop = dbc.New()

dbc\_shop:load\_txt('t\_shop')

for i = 0, dbc\_shop:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_shop:get\_int(0, i)

tdic.type = dbc\_shop:get\_int(1, i)

tdic.item\_id = dbc\_shop:get\_int(2, i)

tdic.name = dbc\_shop:get\_string(3, i)

tdic.past\_type = dbc\_shop:get\_int(4, i)

tdic.price = dbc\_shop:get\_int(5, i)

tdic.icon = dbc\_shop:get\_string(6, i)

Config.t\_shop[tdic.id] = tdic

end

end

function Config.get\_t\_shop(id)

if Config.t\_shop[id] == nil then

return nil

else

return Config.t\_shop[id]

end

end

----------------------------------------------------------------

--------------------------t\_error--------------------------------

Config.t\_error = {}

function Config.parse\_t\_error()

local dbc\_error = dbc.New()

dbc\_error:load\_txt('t\_error')

for i = 0, dbc\_error:get\_y() - 1 do

local id = dbc\_error:get\_int(0, i)

local text = dbc\_error:get\_string(2, i)

text = Config.get\_t\_lang(text)

Config.t\_error[id] = text

end

end

function Config.get\_t\_error(id)

if Config.t\_error[id] == nil then

return ""

else

return Config.t\_error[id]

end

end

----------------------------------------------------------------

-------------------------t\_chest--------------------------------

Config.t\_chest = {}

function Config.parse\_t\_chest()

local dbc\_chest = dbc.New()

dbc\_chest:load\_txt('t\_chest')

for i = 0, dbc\_chest:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_chest:get\_int(0, i)

tdic.name = dbc\_chest:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.icon = dbc\_chest:get\_string(3, i)

tdic.time = dbc\_chest:get\_int(5, i)

tdic.gold\_min = dbc\_chest:get\_int(6, i)

tdic.gold\_max = dbc\_chest:get\_int(7, i)

tdic.item\_num = dbc\_chest:get\_int(8, i)

tdic.treasure\_num = dbc\_chest:get\_int(9, i)

tdic.get\_time = function()

return BattleOperation.toInt(tdic.time \* (100 - PlayerData.out\_attr[6]) / 100)

end

tdic.get\_gold\_min = function()

return tdic.gold\_min + PlayerData.out\_attr[5]

end

tdic.get\_gold\_max = function()

return tdic.gold\_max + PlayerData.out\_attr[5]

end

Config.t\_chest[tdic.id] = tdic

end

end

function Config.get\_t\_chest(id)

if Config.t\_chest[id] == nil then

return nil

else

return Config.t\_chest[id]

end

end

--------------------------------------------------------------------

-------------------------t\_attr-----------------------------------

Config.t\_attr = {}

function Config.parse\_t\_attr()

local dbc\_attr = dbc.New()

dbc\_attr:load\_txt('t\_attr')

for i = 0, dbc\_attr:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_attr:get\_int(0, i)

tdic.desc = dbc\_attr:get\_string(3, i)

tdic.desc = Config.get\_t\_lang(tdic.desc)

Config.t\_attr[tdic.id] = tdic

end

end

function Config.get\_t\_attr(id)

if Config.t\_attr[id] == nil then

return nil

else

return Config.t\_attr[id]

end

end

--------------------------------------------------------------------

---------------------------t\_achievement----------------------------

Config.t\_achievement = {}

function Config.parse\_t\_achievement()

local dbc\_achieve = dbc.New()

dbc\_achieve:load\_txt('t\_achievement')

local tags = {} --标签分类

local types = {} --类型分类

local achieveTrees = {} --成就树

Config.t\_achievement.achievesTB = {}

for i = 0, dbc\_achieve:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_achieve:get\_int(0, i)

tdic.name = dbc\_achieve:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.desc = dbc\_achieve:get\_string(4, i)

tdic.desc = Config.get\_t\_lang(tdic.desc)

tdic.dtype = dbc\_achieve:get\_int(5, i)

tdic.tag = dbc\_achieve:get\_int(6, i)

tdic.pre = dbc\_achieve:get\_int(7, i)

tdic.target\_num = dbc\_achieve:get\_int(8, i)

tdic.type\_id = dbc\_achieve:get\_int(9, i)

tdic.param1 = dbc\_achieve:get\_int(10,i)

tdic.param2 = dbc\_achieve:get\_int(11,i)

tdic.param3 = dbc\_achieve:get\_int(12,i)

tdic.param4 = dbc\_achieve:get\_int(13,i)

tdic.point = dbc\_achieve:get\_int(14,i)

tdic.rtype = dbc\_achieve:get\_int(15,i)

tdic.rvalue1 = dbc\_achieve:get\_int(16,i)

tdic.rvalue2 = dbc\_achieve:get\_int(17,i)

tdic.rvalue3 = dbc\_achieve:get\_int(18,i)

tdic.icon = dbc\_achieve:get\_string(19,i)

if tags[tdic.tag] == nil then

tags[tdic.tag] = {}

end

table.insert(tags[tdic.tag], tdic)

if types[tdic.type\_id] == nil then

types[tdic.type\_id] = {}

end

table.insert(types[tdic.type\_id], tdic)

if tdic.id == 0 then

print(tdic.name,tdic.desc)

end

Config.t\_achievement.achievesTB[tdic.id] = tdic

end

local searchState = {}

for k,v in pairs(Config.t\_achievement.achievesTB) do

if not searchState[k] then

local tmp = v

local list = {}

while (tmp.pre > 0) do

table.insert(list,1,tmp)

searchState[tmp.id] = true

tmp = Config.t\_achievement.achievesTB[tmp.pre]

end

if achieveTrees[tmp.id] ~= nil then

if #achieveTrees[tmp.id] < #list then

achieveTrees[tmp.id] = list

end

else

achieveTrees[tmp.id] = list

end

end

end

Config.t\_achievement.tags = tags

Config.t\_achievement.types = types

Config.t\_achievement.achieveTrees = achieveTrees

--每个技能的后续技能

for k,v in pairs(Config.t\_achievement.achieveTrees) do

if #v == 0 then

Config.get\_t\_achievement(k).next\_achieve\_id = nil

else

Config.get\_t\_achievement(k).next\_achieve\_id = v[1].id

end

for i = 1,#v do

if i ~= #v then

v[i].next\_achieve\_id = v[i + 1].id

else

v[i].next\_achieve\_id = nil

end

end

end

end

function Config.get\_t\_achievement(id)

if Config.t\_achievement.achievesTB[id] == nil then

return nil

else

return Config.t\_achievement.achievesTB[id]

end

end

--------------------------------------------------------------------

------------------------------t\_talent------------------------------

Config.t\_talent = {}

function Config.parse\_t\_talent()

local dbc\_talent = dbc.New()

dbc\_talent:load\_txt('t\_talent')

local c\_list = {} --后续技能表

for i = 0, dbc\_talent:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_talent:get\_int(0, i)

tdic.name = dbc\_talent:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.star = dbc\_talent:get\_int(3,i)

tdic.type = dbc\_talent:get\_int(4,i)

tdic.desc = dbc\_talent:get\_string(6, i)

tdic.desc = Config.get\_t\_lang(tdic.desc)

tdic.pre\_skill\_id\_1 = dbc\_talent:get\_int(7,i)

tdic.pre\_skill\_id\_2 = dbc\_talent:get\_int(8,i)

tdic.icon = dbc\_talent:get\_string(9, i)

tdic.sub = {}

for j = 0, 2 do

jsub = {}

jsub.desc = dbc\_talent:get\_string(11 + 14 \* j, i)

jsub.desc = Config.get\_t\_lang(jsub.desc)

jsub.sub = {}

for k = 0, 2 do

ksub = {}

ksub.type = dbc\_talent:get\_int(12 + 14 \* j + 4 \* k, i)

ksub.param1 = dbc\_talent:get\_int(13 + 14 \* j + 4 \* k, i)

ksub.param2 = dbc\_talent:get\_int(14 + 14 \* j + 4 \* k, i)

ksub.param3 = dbc\_talent:get\_int(15 + 14 \* j + 4 \* k, i)

table.insert(jsub.sub, ksub)

end

table.insert(tdic.sub, jsub)

end

if tdic.pre\_skill\_id\_1 ~= 0 then

if c\_list[tdic.pre\_skill\_id\_1] == nil then

c\_list[tdic.pre\_skill\_id\_1] = {}

end

table.insert(c\_list[tdic.pre\_skill\_id\_1],tdic.id)

end

if tdic.pre\_skill\_id\_2 ~= 0 then

if c\_list[tdic.pre\_skill\_id\_2] == nil then

c\_list[tdic.pre\_skill\_id\_2] = {}

end

table.insert(c\_list[tdic.pre\_skill\_id\_2],tdic.id)

end

Config.t\_talent[tdic.id] = tdic

end

for k,data in pairsByKeys(Config.t\_talent) do

if c\_list[k] == nil then

Config.t\_talent[k].nextOrList = {}

else

Config.t\_talent[k].nextOrList = c\_list[k]

end

end

end

function Config.get\_t\_talent(id)

if Config.t\_talent[id] == nil then

return nil

else

return Config.t\_talent[id]

end

end

--------------------------t\_recharge----------------------------------

Config.t\_recharge = {}

function Config.parse\_t\_recharge()

local dbc\_recharge = dbc.New()

dbc\_recharge:load\_txt("t\_recharge")

for i = 0, dbc\_recharge:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_recharge:get\_int(0, i)

tdic.name = dbc\_recharge:get\_string(2, i)

tdic.name = Config.get\_t\_lang(tdic.name)

tdic.price = dbc\_recharge:get\_int(3, i)

tdic.icon = dbc\_recharge:get\_string(4, i)

tdic.value = dbc\_recharge:get\_int(5, i)

tdic.code = dbc\_recharge:get\_string(6, i)

tdic.desc = dbc\_recharge:get\_string(7, i)

tdic.desc = Config.get\_t\_lang(tdic.desc)

Config.t\_recharge[tdic.id] = tdic

end

end

function Config.get\_t\_recharge(id)

if Config.t\_recharge[id] == nil then

return nil

else

return Config.t\_recharge[id]

end

end

-----------------------------------------------------------------------

-------------------------t\_sign----------------------------------------

Config.t\_sign = {}

function Config.parse\_t\_sign()

local dbc\_sign = dbc.New()

dbc\_sign:load\_txt("t\_sign")

for i = 0, dbc\_sign:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_sign:get\_int(0, i)

tdic.day = dbc\_sign:get\_string(2, i)

tdic.day = Config.get\_t\_lang(tdic.day)

tdic.type = dbc\_sign:get\_int(3, i)

tdic.value1 = dbc\_sign:get\_int(4, i)

tdic.value2 = dbc\_sign:get\_int(5, i)

tdic.value3 = dbc\_sign:get\_int(6, i)

Config.t\_sign[tdic.id] = tdic

end

end

function Config.get\_t\_sign(id)

if Config.t\_sign[id] == nil then

return nil

else

return Config.t\_sign[id]

end

end

function Config.get\_t\_reward(type, id)

local reward\_temp = nil

if(type == 1) then

reward\_temp = Config.get\_t\_resource(id)

elseif(type == 2) then

reward\_temp = Config.get\_t\_item(id)

elseif(type == 3) then

reward\_temp = Config.get\_t\_role(id)

elseif(type == 4) then

reward\_temp = Config.get\_t\_avatar(id)

elseif(type == 5) then

reward\_temp = Config.get\_t\_chest(id)

elseif(type == 6) then

reward\_temp = Config.get\_t\_toukuang(id)

end

return reward\_temp

end

------------------------------------------------------------------------

-----------------------------t\_achievement\_reward-----------------------

Config.t\_achievement\_reward = {}

function Config.parse\_t\_achievement\_reward()

local dbc\_achieve\_reward = dbc.New()

dbc\_achieve\_reward:load\_txt("t\_achievement\_reward")

for i = 0, dbc\_achieve\_reward:get\_y() - 1 do

local tdic = {}

tdic.id = dbc\_achieve\_reward:get\_int(0, i)

tdic.point = dbc\_achieve\_reward:get\_int(1, i)

tdic.rewards = {}

local b\_pos = 2

local index = 1

for j = 1,3 do

local \_type = dbc\_achieve\_reward:get\_int(b\_pos + (j - 1)\*4,i)

if \_type > 0 then

local value1 = dbc\_achieve\_reward:get\_int(b\_pos + (j - 1)\*4 + 1,i)

local value2 = dbc\_achieve\_reward:get\_int(b\_pos + (j - 1)\*4 + 2,i)

local value3 = dbc\_achieve\_reward:get\_int(b\_pos + (j - 1)\*4 + 3,i)

tdic.rewards[index] = {\_type,value1,value2,value3}

index = index + 1

end

end

Config.t\_achievement\_reward[tdic.id] = tdic

end

end

function Config.get\_t\_achievement\_reward(id)

if Config.t\_achievement\_reward[id] == nil then

return nil

else

return Config.t\_achievement\_reward[id]

end

end

------------------------------------------------------------------------

function Config.Init()

Config.parse\_t\_lang()

Config.parse\_t\_client\_lang()

Config.parse\_t\_name()

Config.parse\_t\_skill()

Config.parse\_t\_skill\_effect()

Config.parse\_t\_skill\_xiaoguo()

Config.parse\_t\_battle\_item()

Config.parse\_t\_battle\_exp()

Config.parse\_t\_battle\_buff()

Config.parse\_t\_item()

Config.parse\_t\_role()

Config.parse\_t\_avatar()

Config.parse\_t\_role\_skill()

Config.parse\_t\_role\_level()

Config.parse\_t\_biaoqing()

Config.parse\_t\_region()

Config.parse\_t\_exp()

Config.parse\_t\_cup()

Config.parse\_t\_resource()

Config.parse\_t\_shop()

Config.parse\_t\_error()

Config.parse\_t\_chest()

Config.parse\_t\_attr()

Config.parse\_t\_achievement()

Config.parse\_t\_recharge()

Config.parse\_t\_talent()

Config.parse\_t\_role\_buff()

Config.parse\_t\_sign()

Config.parse\_t\_toukuang()

Config.parse\_t\_item\_box()

Config.parse\_t\_achievement\_reward()

end

BattleGrid = {}

local GRID\_SIZE = 5000

local BLOCK\_SIZE = 20000

local meshinfo\_ = nil

local meshinfo1\_ = nil

local width\_ = 0

local height\_ = 0

BattleGrid.et\_player = 0

BattleGrid.et\_effect = 1

BattleGrid.et\_item = 2

local obss\_ = {}

local objs\_ = {}

local locs\_ = {}

function BattleGrid.Init(name)

local obstancle = obstancle()

obstancle:load\_obs(name)

obss\_ = {}

for i = 0, obstancle:get\_x() - 1 do

obss\_[i] = {}

for j = 0, obstancle:get\_x() - 1 do

local a = obstancle:get(i, j)

if a >= 2 then

obss\_[i][j] = a - 1

else

obss\_[i][j] = 0

end

end

end

width\_ = BattleOperation.toInt(obstancle:get\_x() \* GRID\_SIZE / BLOCK\_SIZE)

height\_ = BattleOperation.toInt(obstancle:get\_y() \* GRID\_SIZE / BLOCK\_SIZE)

obstancle:clear()

meshinfo\_ = navMeshInfo()

meshinfo\_:load\_nav(name.. "\_nav")

meshinfo1\_ = navMeshInfo()

meshinfo1\_:load\_nav(name.. "\_nav1")

BattlePlayerAI.Init(meshinfo\_) --初始化寻路网格

-- 加载数据

objs\_ = {}

locs\_ = {}

for i = 1, width\_ \* height\_ do

table.insert(objs\_, {})

end

end

function BattleGrid.Fini(name)

if meshinfo\_ ~= nil then

meshinfo\_:clear()

meshinfo\_ = nil

end

if meshinfo1\_ ~= nil then

meshinfo1\_:clear()

meshinfo1\_ = nil

end

end

function BattleGrid.add(t, guid, x, y)

BattleGrid.del(t, guid)

local xx = BattleOperation.toInt(x / BLOCK\_SIZE)

local yy = BattleOperation.toInt(y / BLOCK\_SIZE)

if xx < 0 or xx >= width\_ or yy < 0 or yy >= height\_ then

return

end

local index = yy \* width\_ + xx + 1

if objs\_[index][t] == nil then

objs\_[index][t] = {}

end

objs\_[index][t][guid] = 1

locs\_[guid] = index

end

function BattleGrid.del(t, guid)

if locs\_[guid] == nil then

return

end

objs\_[locs\_[guid]][t][guid] = nil

locs\_[guid] = nil

end

function BattleGrid.get(t, x, y, guids, sz)

if sz == nil then

sz = 2

else

sz = BattleOperation.toInt(sz / BLOCK\_SIZE) + 1

end

local xx = BattleOperation.toInt(x / BLOCK\_SIZE)

local yy = BattleOperation.toInt(y / BLOCK\_SIZE)

for i = xx - sz, xx + sz do

for j = yy - sz, yy + sz do

if i >= 0 and i < width\_ and j >= 0 and j < height\_ then

local index = j \* width\_ + i + 1

local o = objs\_[index][t]

if o ~= nil then

for k, v in pairsByKeys(o) do

table.insert(guids, k)

end

end

end

end

end

end

function BattleGrid.get\_cao(x, y)

x = BattleOperation.toInt(x / GRID\_SIZE)

y = BattleOperation.toInt(y / GRID\_SIZE)

local a = obss\_[x][y]

if a == nil then

return 0

end

return a

end

function BattleGrid.can\_move\_type(t)

if t == -1 or t == 1 then

return false

end

return true

end

function BattleGrid.can\_move(xx, yy)

return meshinfo\_:can\_move(xx, yy)

end

function BattleGrid.can\_effect\_move(xx, yy)

return meshinfo1\_:can\_move(xx, yy)

end

function BattleGrid.get\_move\_point(x1, y1, xx, yy)

if BattleGrid.can\_move(xx, yy) then

return {xx, yy}

end

local x2 = xx

local y2 = yy

local dx = math.abs(x2 - x1)

local dy = math.abs(y2 - y1)

while dx > 1000 or dy > 1000 do

local mx = BattleOperation.toInt((x1 + x2) / 2)

local my = BattleOperation.toInt((y1 + y2) / 2)

if BattleGrid.can\_move(mx, my) then

x1 = mx

y1 = my

else

x2 = mx

y2 = my

end

dx = math.abs(x2 - x1)

dy = math.abs(y2 - y1)

end

return {x1, y1}

end

BattleSkillRange = {}

local obj\_ = nil

local obj\_sub\_ = nil

local materials\_ = {}

local name\_ = ""

local type\_ = 0

local range\_ = 0

local r\_ = 0

local x\_ = 0

local y\_ = 0

function BattleSkillRange.GetName()

return name\_

end

function BattleSkillRange.Create(name, rtype, range, param)

BattleSkillRange.Destroy()

name\_ = name

type\_ = rtype

range\_ = range

range = range / Battle.BL

if type\_ ~= 1 then

param = param / Battle.BL

end

if type\_ == 1 then

obj\_ = resMgr:CreateEffect("skill\_release\_di")

obj\_.transform.parent = resMgr.UnitRoot.transform

obj\_.transform.localScale = Vector3(range / 5, 1, range / 5)

local m = obj\_.transform:FindChild('zd\_fz001'):GetComponent("Renderer").material

table.insert(materials\_, m)

obj\_sub\_ = resMgr:CreateEffect("skill\_release\_shan")

obj\_sub\_.transform.parent = resMgr.UnitRoot.transform

obj\_sub\_.transform.localScale = Vector3(range / 5, 1, range / 5)

local left = obj\_sub\_.transform:FindChild('left')

m = left:FindChild('zd\_fz002c'):GetComponent("Renderer").material

table.insert(materials\_, m)

local right = obj\_sub\_.transform:FindChild('right')

m = right:FindChild('zd\_fz002a'):GetComponent("Renderer").material

table.insert(materials\_, m)

local center = obj\_sub\_.transform:FindChild('center')

local num = math.floor(param / 15 - 2)

left.localEulerAngles = Vector3(0, -7.5 - 7.5 \* num, 0)

right.localEulerAngles = Vector3(0, 7.5 + 7.5 \* num, 0)

for i = 0, num - 1 do

local tcenter = LuaHelper.Instantiate(center.gameObject)

tcenter.transform.parent = obj\_sub\_.transform

tcenter.transform.localPosition = Vector3.zero

tcenter.transform.localEulerAngles = Vector3(0, i \* 15 - 7.5 \* (num - 1), 0)

tcenter.transform.localScale = Vector3.one

tcenter:SetActive(true)

m = tcenter.transform:FindChild('zd\_fz002b'):GetComponent("Renderer").material

table.insert(materials\_, m)

end

elseif type\_ == 2 then

obj\_ = resMgr:CreateEffect("skill\_release\_di")

obj\_.transform.parent = resMgr.UnitRoot.transform

obj\_.transform.localScale = Vector3(range / 5, 1, range / 5)

local m = obj\_.transform:FindChild('zd\_fz001'):GetComponent("Renderer").material

table.insert(materials\_, m)

obj\_sub\_ = resMgr:CreateEffect("skill\_release\_jian")

obj\_sub\_.transform.parent = resMgr.UnitRoot.transform

obj\_sub\_.transform.localScale = Vector3(param / 2, 1, 1)

local head = obj\_sub\_.transform:FindChild('head')

head.localPosition = Vector3(0, 0, range - 2)

m = head:FindChild('zd\_fz003b'):GetComponent("Renderer").material

table.insert(materials\_, m)

local tail = obj\_sub\_.transform:FindChild('tail')

tail.localScale = Vector3(1, 1, (range - 2) / 3)

m = tail:FindChild('zd\_fz003a'):GetComponent("Renderer").material

table.insert(materials\_, m)

elseif type\_ == 3 then

obj\_ = resMgr:CreateEffect("skill\_release\_di")

obj\_.transform.parent = resMgr.UnitRoot.transform

obj\_.transform.localScale = Vector3(range / 5, 1, range / 5)

local m = obj\_.transform:FindChild('zd\_fz001'):GetComponent("Renderer").material

table.insert(materials\_, m)

obj\_sub\_ = resMgr:CreateEffect("skill\_release\_quan")

obj\_sub\_.transform.parent = resMgr.UnitRoot.transform

obj\_sub\_.transform.localScale = Vector3(param / 5, 1, param / 5)

m = obj\_sub\_.transform:FindChild('zd\_fz004'):GetComponent("Renderer").material

table.insert(materials\_, m)

elseif type\_ == 4 or type\_ == 5 then

obj\_ = resMgr:CreateEffect("skill\_release\_di")

obj\_.transform.parent = resMgr.UnitRoot.transform

obj\_.transform.localScale = Vector3(range / 5, 1, range / 5)

local m = obj\_.transform:FindChild('zd\_fz001'):GetComponent("Renderer").material

table.insert(materials\_, m)

end

end

function BattleSkillRange.Destroy()

if obj\_ ~= nil then

GameObject.Destroy(obj\_)

obj\_ = nil

end

if obj\_sub\_ ~= nil then

GameObject.Destroy(obj\_sub\_)

obj\_sub\_ = nil

end

materials\_ = {}

name\_ = ""

end

function BattleSkillRange.SetPosition(x, y, r, xx, yy)

x\_ = x + xx \* range\_

y\_ = y + yy \* range\_

r\_ = r

BattleSkillRange.SetPosition1(x, y, r, x\_, y\_)

end

function BattleSkillRange.SetPosition1(x, y, r, xx, yy)

x\_ = xx

y\_ = yy

r\_ = r

if obj\_ ~= nil then

obj\_.transform.localPosition = Vector3(x / Battle.BL, 0.1, y / Battle.BL)

end

if obj\_sub\_ ~= nil then

if type\_ == 1 then

obj\_sub\_.transform.localEulerAngles = Vector3(0, 90 - r, 0)

obj\_sub\_.transform.localPosition = Vector3(x / Battle.BL, 0.1, y / Battle.BL)

elseif type\_ == 2 then

obj\_sub\_.transform.localEulerAngles = Vector3(0, 90 - r, 0)

obj\_sub\_.transform.localPosition = Vector3(x / Battle.BL, 0.1, y / Battle.BL)

elseif type\_ == 3 then

obj\_sub\_.transform.localPosition = Vector3(x\_ / Battle.BL, 0.1, y\_ / Battle.BL)

end

end

end

function BattleSkillRange.GetPosition()

return {x\_, y\_, r\_}

end

function BattleSkillRange.SetColor(c)

for i = 1, #materials\_ do

materials\_[i]:SetColor("\_TintColor", c)

end

end

function BattleSkillRange.SetBlue()

BattleSkillRange.SetColor(Color(0, 0.165, 0.941, 0.5))

end

function BattleSkillRange.SetRed()

BattleSkillRange.SetColor(Color(1, 0, 0, 0.5))

end

State = {}

State.state = {

ss\_null = 0,

ss\_login = 1,

ss\_hall = 2,

ss\_battle = 3,

ss\_ofbattle = 4,

}

State.cur\_state = State.state.ss\_null

function State.ChangeState(state, param)

State.LeaveState(State.cur\_state, param)

State.cur\_state = state

State.EnterState(State.cur\_state, param)

end

function State.EnterState(state, param)

if state == State.state.ss\_login then

GUIRoot.ShowGUI("StartPanel")

soundMgr:play\_mus("hall")

elseif state == State.state.ss\_hall then

mapMgr:LoadScene("sf\_start", State.LoadEnd)

GUIRoot.ShowGUI("LoadPanel", {true})

soundMgr:play\_mus("hall")

elseif state == State.state.ss\_battle then

mapMgr:LoadScene("sf\_fight001", State.LoadEnd)

GUIRoot.ShowGUI("LoadPanel", {false})

soundMgr:play\_mus("fight\_00"..math.random(1, 2))

elseif state == State.state.ss\_ofbattle then

mapMgr:LoadScene("sf\_fight001", State.LoadEnd)

GUIRoot.ShowGUI("LoadPanel", {false})

soundMgr:play\_mus("fight\_00"..math.random(1, 2))

end

end

function State.LeaveState(state, param)

if state == State.state.ss\_login then

LoginPanel.Fini()

GUIRoot.HideGUI("StartPanel")

elseif state == State.state.ss\_hall then

GUIRoot.HideGUI("HallPanel")

HallScene.RemoveRole()

elseif state == State.state.ss\_battle then

Battle.OnDestroy()

elseif state == State.state.ss\_ofbattle then

Battle.OnDestroy()

end

end

function State.LoadEnd()

if State.cur\_state == State.state.ss\_hall then

mapMgr:SetInitCam(0, 0.6, 3, -5, 180, 0, 1)

GUIRoot.ShowGUI("HallPanel")

elseif State.cur\_state == State.state.ss\_battle then

mapMgr:SetInitCam(0, 17, -16.2, 45, 0, 0, 2)

Battle.Awake("sf\_fight001", true)

elseif State.cur\_state == State.state.ss\_ofbattle then

mapMgr:SetInitCam(0, 17, -16.2, 45, 0, 0, 2)

Battle.Awake("sf\_fight001", false)

end

end

function State.Reset()

self.ClearData()

LuaHelper.restart()

end

Battle = {}

Battle.BL = 10000

Battle.name = ""

local move\_hold\_r\_ = 0

local is\_hold\_ = {}

local holds\_ = {"move", "attack", "jskill"}

local key\_hold\_r\_ = -1

Battle.is\_end = false

Battle.is\_online = false

Battle.bobjpool = nil

local offlinet\_

ready\_to\_start\_ = true

local ready\_time\_ = 0

function Battle.RegisterMessage()

Message.register\_net\_handle(opcodes.SMSG\_BATTLE\_LINK, Battle.SMSG\_BATTLE\_LINK)

Message.register\_net\_handle(opcodes.SMSG\_BATTLE\_OP, Battle.SMSG\_BATTLE\_OP)

Message.register\_net\_handle(opcodes.SMSG\_BATTLE\_ZHEN, Battle.SMSG\_BATTLE\_ZHEN)

Message.register\_net\_handle(opcodes.SMSG\_BATTLE\_FINISH, Battle.SMSG\_BATTLE\_FINISH)

end

function Battle.RemoveMessage()

Message.remove\_net\_handle(opcodes.SMSG\_BATTLE\_LINK, Battle.SMSG\_BATTLE\_LINK)

Message.remove\_net\_handle(opcodes.SMSG\_BATTLE\_OP, Battle.SMSG\_BATTLE\_OP)

Message.remove\_net\_handle(opcodes.SMSG\_BATTLE\_ZHEN, Battle.SMSG\_BATTLE\_ZHEN)

Message.remove\_net\_handle(opcodes.SMSG\_BATTLE\_FINISH, Battle.SMSG\_BATTLE\_FINISH)

end

function Battle.Awake(name, is\_online)

Battle.name = name

Battle.is\_end = false

Battle.is\_online = is\_online

skill\_hold\_ = false

move\_hold\_r\_ = -1

for i = 1, #holds\_ do

is\_hold\_[holds\_[i]] = false

end

offlinet\_ = 0

Battle.bobjpool = bobjpool()

GUIRoot.ShowGUI('BattlePanel')

BattleOperation.set\_random\_seed(0, 50)

Battle.RegisterMessage()

UpdateBeat:Add(Battle.Update, Battle)

ready\_to\_start\_ = true

ready\_time\_ = 0

Battle.send\_link()

end

function Battle.OnDestroy()

UpdateBeat:Remove(Battle.Update, Battle)

Battle.RemoveMessage()

GUIRoot.HideGUI('BattlePanel')

Battle.bobjpool:clear()

BattlePlayers.Fini()

end

function Battle.SMSG\_BATTLE\_LINK(message)

local msg = msg\_battle\_pb.smsg\_battle\_link()

msg:ParseFromString(message.luabuff)

BattlePlayers.Init()

BattlePlayers.max\_zhen = msg.zhen - 1

local state = msg.state

if msg.is\_state == 2 then

BattlePlayers.zhen = state.zhen

for i = 1, #state.players do

BattlePlayers.addplayer(state.players[i])

end

for i = 1, #state.effects do

BattlePlayers.add\_effect2(state.effects[i])

end

for i = 1, #state.items do

BattlePlayers.add\_item2(state.items[i])

end

for i = 1, #msg.ops do

BattlePlayers.AddOperation(msg.ops[i])

end

for i = 1, #msg.seeds do

BattlePlayers.seeds[BattlePlayers.zhen + i - 1] = msg.seeds[i]

end

elseif msg.is\_state == 1 then

BattlePlayers.zhen = state.zhen

for i = 1, #msg.ops do

BattlePlayers.AddOperation(msg.ops[i])

end

for i = 1, #msg.seeds do

BattlePlayers.seeds[BattlePlayers.zhen + i - 1] = msg.seeds[i]

end

else

BattlePlayers.zhen = state.zhen

end

BattlePlayers.tid = state.tid

BattlePlayers.init\_item = state.init\_item

BattlePlayers.dtime = 0

BattlePlayers.jiasu = true

BattleAchieve.Init()

end

function Battle.SMSG\_BATTLE\_OP(message)

local msg = msg\_battle\_pb.msg\_battle\_op()

msg:ParseFromString(message.luabuff)

BattlePlayers.AddOperation(msg)

end

function Battle.SMSG\_BATTLE\_ZHEN(message)

local msg = msg\_battle\_pb.smsg\_battle\_zhen()

msg:ParseFromString(message.luabuff)

BattlePlayers.max\_zhen = msg.zhen

BattlePlayers.seeds[msg.zhen] = msg.seed

end

function Battle.SMSG\_BATTLE\_FINISH(message)

BattlePanel.show\_end()

Battle.send\_result()

Battle.is\_end = true

Time.timeScale = 1

BattleAchieve.BattleFinish()

end

function Battle.GetNear(range)

if BattlePlayers.players[self.guid] == nil then

return

end

local bp = BattlePlayers.players[self.guid]

local flag = false

local d = 0

local tbp = nil

for guid, bp1 in pairsByKeys(BattlePlayers.players) do

if guid ~= self.guid and BattleOperation.can\_see(bp, bp1) and not bp.attr.is\_hunluan() and not bp.attr.is\_zhimang() and not bp1.is\_die and not bp1.player.is\_xueren then

local dd = BattleOperation.get\_distance2(bp.player.x, bp.player.y, bp1.player.x, bp1.player.y)

if dd <= range \* range then

if not flag then

flag = true

d = dd

tbp = bp1

elseif dd < d then

d = dd

tbp = bp1

end

end

end

end

local near = {}

if flag then

local dx = tbp.player.x - bp.player.x

local dy = tbp.player.y - bp.player.y

near.r = BattleOperation.toInt(math.atan2(dy, dx) \* 180 / math.pi)

near.x = tbp.player.x

near.y = tbp.player.y

else

near.r = bp.player.r

near.x = bp.player.x

near.y = bp.player.y

end

return near

end

function Battle.ChangeSkill()

if is\_hold\_["jskill"] then

if BattlePlayers.me ~= nil then

local bp = BattlePlayers.me

local t\_skill = Config.get\_t\_skill(bp.player.skill\_id, bp.player.skill\_level)

if t\_skill ~= nil then

BattleSkillRange.Create("jskill", t\_skill.release\_type, t\_skill.get\_range(bp), t\_skill.get\_range\_param(bp))

BattleSkillRange.SetBlue()

end

end

end

end

function Battle.Update()

if ready\_to\_start\_ then

if ready\_time\_ < 5 then

ready\_time\_ = ready\_time\_ + Time.deltaTime / Time.timeScale

if ready\_time\_ >= 5 then

BattleUdp.Disconnect3()

return

end

end

if BattlePlayers.start and not BattlePlayers.jiasu then

ready\_to\_start\_ = false

LoadPanel.close(Battle.send\_in)

end

end

if not Battle.is\_online then

offlinet\_ = offlinet\_ + Time.deltaTime / Time.timeScale

while offlinet\_ >= BattlePlayers.TICK / 1000 do

offlinet\_ = offlinet\_ - BattlePlayers.TICK / 1000

if BattlePlayers.max\_zhen >= BattlePlayers.TNUM \* 900 then

if not Battle.is\_end then

BattlePanel.show\_end()

Battle.is\_end = true

end

BattlePanel.Update()

return

end

BattlePlayers.max\_zhen = BattlePlayers.max\_zhen + 1

BattlePlayers.seeds[BattlePlayers.max\_zhen] = tonumber(tostring(os.time()):reverse():sub(1, 7))

end

end

BattlePlayers.Update()

BattlePanel.Update()

if BattlePlayers.me ~= nil then

local bp = BattlePlayers.me

--自己移动

local joy\_mode = false

local joy = Joy.Get("move")

if joy ~= nil then

if joy:IsHolding() and not joy:IsStart() then

joy\_mode = true

local r = joy:Axis2Angle(true)

r = BattleOperation.toInt(r / 6) \* 6

local flag = false

if not is\_hold\_["move"] then

is\_hold\_["move"] = true

move\_hold\_r\_ = r

flag = true

elseif r ~= move\_hold\_r\_ then

move\_hold\_r\_ = r

flag = true

end

if flag then

Battle.send\_move(move\_hold\_r\_)

end

else

if is\_hold\_["move"] then

joy\_mode = true

is\_hold\_["move"] = false

Battle.send\_stop()

end

end

end

--键盘移动

if not joy\_mode then

local key = -1

if Input.GetKey(KeyCode.I) and Input.GetKey(KeyCode.L) then

key = 45

elseif Input.GetKey(KeyCode.J) and Input.GetKey(KeyCode.I) then

key = 135

elseif Input.GetKey(KeyCode.K) and Input.GetKey(KeyCode.J) then

key = 225

elseif Input.GetKey(KeyCode.K) and Input.GetKey(KeyCode.L) then

key = 315

elseif Input.GetKey(KeyCode.I) then

key = 90

elseif Input.GetKey(KeyCode.K) then

key = 270

elseif Input.GetKey(KeyCode.J) then

key = 180

elseif Input.GetKey(KeyCode.L) then

key = 0

end

if key ~= key\_hold\_r\_ then

key\_hold\_r\_ = key

if key == -1 then

Battle.send\_stop()

else

Battle.send\_move(key\_hold\_r\_)

end

end

end

--普通攻击，技能

for i = 2, #holds\_ do

local xm = holds\_[i]

local skill = 100101

local skill\_level = 1

if i == 3 then

skill = bp.player.skill\_id

skill\_level = bp.player.skill\_level

end

joy = Joy.Get(xm)

if joy ~= nil and skill > 0 and skill\_level > 0 then

local mode = joy:Mode()

if joy:IsHolding() then

local r = 0

local x = 0

local y = 0

local t\_skill = Config.get\_t\_skill(skill, skill\_level)

if t\_skill ~= nil then

if not is\_hold\_[xm] then

is\_hold\_[xm] = true

if mode ~= 2 then

BattleSkillRange.Create(xm, t\_skill.release\_type, t\_skill.get\_range(bp), t\_skill.get\_range\_param(bp))

BattleSkillRange.SetBlue()

else

--Battle.send\_release(t\_skill.id, 0, 0, 0)

end

end

if mode ~= 2 then

if BattleSkillRange.GetName() == xm then

if joy:IsStart() or t\_skill.release\_type > 3 then

local near = Battle.GetNear(t\_skill.get\_range(bp))

r = near.r

x = near.x

y = near.y

BattleSkillRange.SetPosition1(bp.player.x, bp.player.y, r, x, y)

else

r = joy:Axis2Angle(true)

local pos = joy:GetPos()

x = pos.x

y = pos.y

BattleSkillRange.SetPosition(bp.player.x, bp.player.y, r, x, y)

end

end

end

end

else

if is\_hold\_[xm] then

is\_hold\_[xm] = false

if mode ~= 2 then

if BattleSkillRange.GetName() == xm then

local t\_skill = Config.get\_t\_skill(skill, skill\_level)

if t\_skill then

cs = BattleSkillRange.GetPosition()

BattleSkillRange.Destroy()

Battle.send\_release(t\_skill.id, cs[1], cs[2], cs[3])

end

end

end

end

end

end

end

end

end

function Battle.send\_link()

BattlePlayers.Fini()

if Battle.is\_online then

local msg = msg\_battle\_pb.cmsg\_battle\_link()

msg.code = self.battle\_code

local data = msg:SerializeToString()

BattleUdp.Send(opcodes.CMSG\_BATTLE\_LINK, data)

else

BattlePlayers.Init()

BattlePlayers.max\_zhen = -1

BattlePlayers.zhen = 0

BattlePlayers.tid = 0

BattlePlayers.init\_item = false

BattlePlayers.dtime = 0

BattlePlayers.jiasu = true

end

end

function Battle.send\_in()

if Battle.is\_online then

BattleUdp.Send(opcodes.CMSG\_BATTLE\_IN)

else

local msg = msg\_battle\_pb.msg\_battle\_op()

msg.zhen = BattlePlayers.zhen + 1

msg.opcode = opcodes.MSG\_BATTLE\_IN

msg.guid = self.guid

msg.param\_strings:append(self.player.name)

if self.player.avatar\_on < 1000 then

msg.param\_strings:append(self.player.avatar\_url[self.player.avatar\_on])

else

msg.param\_strings:append("")

end

local role = self.get\_role(self.player.role\_on)

msg.param\_ints:append(role.template\_id)

msg.param\_ints:append(role.level)

msg.param\_ints:append(self.player.sex)

msg.param\_ints:append(self.player.avatar\_on)

msg.param\_ints:append(self.player.cup)

msg.param\_ints:append(self.player.toukuang\_on)

local sid = 0

local slevel = 0

local t\_item = Config.get\_t\_item(self.player.battle\_item\_id)

if t\_item ~= nil then

sid = t\_item.def1

slevel = t\_item.def2

end

msg.param\_ints:append(sid)

msg.param\_ints:append(slevel)

local index = #msg.param\_ints + 1

msg.param\_ints:append(0)

local num = 0

for i = 1, #self.roles do

local t\_role = Config.get\_t\_role(self.roles[i].template\_id)

for j = 1, #t\_role.gskills do

local t\_role\_buff = Config.get\_t\_role\_buff(t\_role.gskills[j])

if t\_role\_buff.type < 3 then

num = num + 1

msg.param\_ints:append(t\_role\_buff.type)

msg.param\_ints:append(t\_role\_buff.param1)

msg.param\_ints:append(t\_role\_buff.param2)

msg.param\_ints:append(t\_role\_buff.param3 + t\_role\_buff.param4 \* (self.roles[i].level - 1))

end

end

end

for i = 1, #self.player.toukuang do

local t\_toukuang = Config.get\_t\_toukuang(self.player.toukuang[i])

if t\_toukuang.type < 3 then

num = num + 1

msg.param\_ints:append(t\_toukuang.type)

msg.param\_ints:append(t\_toukuang.param1)

msg.param\_ints:append(t\_toukuang.param2)

msg.param\_ints:append(t\_toukuang.param3)

end

end

msg.param\_ints[index] = num

BattlePlayers.AddOperation(msg)

end

end

function Battle.send\_reset()

BattlePlayers.Fini()

if Battle.is\_online then

BattleUdp.Send(opcodes.CMSG\_BATTLE\_RESET)

end

end

function Battle.send\_move(r)

if Battle.is\_online then

local msg = msg\_battle\_pb.msg\_battle\_op()

msg.opcode = opcodes.MSG\_BATTLE\_MOVE

msg.guid = self.guid

msg.param\_ints:append(r)

local data = msg:SerializeToString()

BattleUdp.Send(opcodes.CMSG\_BATTLE\_OP, data)

else

BattlePlayers.Move1(BattlePlayers.me, r)

end

end

function Battle.send\_stop()

if Battle.is\_online then

local msg = msg\_battle\_pb.msg\_battle\_op()

msg.opcode = opcodes.MSG\_BATTLE\_STOP

msg.guid = self.guid

local data = msg:SerializeToString()

BattleUdp.Send(opcodes.CMSG\_BATTLE\_OP, data)

else

BattlePlayers.Stop1(BattlePlayers.me)

end

end

function Battle.send\_release(id, x, y, r)

if Battle.is\_online then

local msg = msg\_battle\_pb.msg\_battle\_op()

msg.opcode = opcodes.MSG\_BATTLE\_RELEASE

msg.guid = self.guid

msg.param\_ints:append(id)

msg.param\_ints:append(x)

msg.param\_ints:append(y)

msg.param\_ints:append(r)

local data = msg:SerializeToString()

BattleUdp.Send(opcodes.CMSG\_BATTLE\_OP, data)

else

BattlePlayers.Release1(BattlePlayers.me, id, x, y, r)

end

end

function Battle.send\_fuhuo()

if Battle.is\_online then

local msg = msg\_battle\_pb.msg\_battle\_op()

msg.opcode = opcodes.MSG\_BATTLE\_FUHUO

msg.guid = self.guid

local data = msg:SerializeToString()

BattleUdp.Send(opcodes.CMSG\_BATTLE\_OP, data)

else

BattlePlayers.Fuhuo1(BattlePlayers.me)

end

end

function Battle.send\_talent(talent\_id)

if Battle.is\_online then

local msg = msg\_battle\_pb.msg\_battle\_op()

msg.opcode = opcodes.MSG\_BATTLE\_TALENT

msg.guid = self.guid

msg.param\_ints:append(talent\_id)

local data = msg:SerializeToString()

BattleUdp.Send(opcodes.CMSG\_BATTLE\_OP, data)

else

BattlePlayers.Talent1(BattlePlayers.me, talent\_id)

end

end

function Battle.send\_change\_skill()

if Battle.is\_online then

local msg = msg\_battle\_pb.msg\_battle\_op()

msg.opcode = opcodes.MSG\_BATTLE\_CHANGE\_SKILL

msg.guid = self.guid

local data = msg:SerializeToString()

BattleUdp.Send(opcodes.CMSG\_BATTLE\_OP, data)

else

BattlePlayers.ChangeSkill1(BattlePlayers.me)

end

end

function Battle.send\_state()

if Battle.is\_online then

local msg = msg\_battle\_pb.msg\_battle\_state()

msg.zhen = BattlePlayers.zhen

msg.tid = BattlePlayers.tid

for guid, bp in pairsByKeys(BattlePlayers.players) do

msg.players:append(bp.player)

end

for tid, be in pairsByKeys(BattlePlayers.effects) do

msg.effects:append(be.effect)

end

for tid, bi in pairsByKeys(BattlePlayers.items) do

msg.items:append(bi.item)

end

msg.init\_item = BattlePlayers.init\_item

local data = msg:SerializeToString()

BattleUdp.Send(opcodes.CMSG\_BATTLE\_STATE, data)

end

end

function Battle.send\_result()

if Battle.is\_online then

local msg = battle\_result\_pb.battle\_result\_t()

msg.guid = 0

local players = {}

for guid, bp in pairsByKeys(BattlePlayers.players) do

table.insert(players, bp)

end

local comps = function(a, b)

return a.player.exp > b.player.exp

end

table.sort(players, comps)

for i = 1, #players do

local bp = players[i]

msg.player\_guids:append(bp.player.guid)

msg.names:append(bp.player.name)

msg.role\_ids:append(bp.player.role\_id)

msg.sexs:append(bp.player.sex)

msg.avatars:append(bp.player.avatar)

msg.avatar\_urls:append(bp.player.avatar\_url)

msg.ranks:append(i)

msg.shas:append(bp.player.sha)

msg.dies:append(bp.player.die)

msg.scores:append(bp.player.exp)

msg.cups:append(bp.player.cup)

msg.cup\_adds:append(1)

end

local data = msg:SerializeToString()

BattleUdp.Send(opcodes.CMSG\_BATTLE\_END, data)

end

end

BattlePlayerAI = {}

local msin = {0, 175, 349, 523, 698, 872, 1045, 1219, 1392, 1564, 1736, 1908, 2079, 2250, 2419, 2588, 2756, 2924, 3090, 3256, 3420, 3584, 3746, 3907, 4067, 4226, 4384, 4540, 4695, 4848, 5000, 5150, 5299, 5446, 5592, 5736, 5878, 6018, 6157, 6293, 6428, 6561, 6691, 6820, 6947, 7071, 7193, 7314, 7431, 7547, 7660, 7771, 7880, 7986, 8090, 8192, 8290, 8387, 8480, 8572, 8660, 8746, 8829, 8910, 8988, 9063, 9135, 9205, 9272, 9336, 9397, 9455, 9511, 9563, 9613, 9659, 9703, 9744, 9781, 9816, 9848, 9877, 9903, 9925, 9945, 9962, 9976, 9986, 9994, 9998, 10000, 9998, 9994, 9986, 9976, 9962, 9945, 9925, 9903, 9877, 9848, 9816, 9781, 9744, 9703, 9659, 9613, 9563, 9511, 9455, 9397, 9336, 9272, 9205, 9135, 9063, 8988, 8910, 8829, 8746, 8660, 8572, 8480, 8387, 8290, 8192, 8090, 7986, 7880, 7771, 7660, 7547, 7431, 7314, 7193, 7071, 6947, 6820, 6691, 6561, 6428, 6293, 6157, 6018, 5878, 5736, 5592, 5446, 5299, 5150, 5000, 4848, 4695, 4540, 4384, 4226, 4067, 3907, 3746, 3584, 3420, 3256, 3090, 2924, 2756, 2588, 2419, 2250, 2079, 1908, 1736, 1564, 1392, 1219, 1045, 872, 698, 523, 349, 175, 0, -175, -349, -523, -698, -872, -1045, -1219, -1392, -1564, -1736, -1908, -2079, -2250, -2419, -2588, -2756, -2924, -3090, -3256, -3420, -3584, -3746, -3907, -4067, -4226, -4384, -4540, -4695, -4848, -5000, -5150, -5299, -5446, -5592, -5736, -5878, -6018, -6157, -6293, -6428, -6561, -6691, -6820, -6947, -7071, -7193, -7314, -7431, -7547, -7660, -7771, -7880, -7986, -8090, -8192, -8290, -8387, -8480, -8572, -8660, -8746, -8829, -8910, -8988, -9063, -9135, -9205, -9272, -9336, -9397, -9455, -9511, -9563, -9613, -9659, -9703, -9744, -9781, -9816, -9848, -9877, -9903, -9925, -9945, -9962, -9976, -9986, -9994, -9998, -10000, -9998, -9994, -9986, -9976, -9962, -9945, -9925, -9903, -9877, -9848, -9816, -9781, -9744, -9703, -9659, -9613, -9563, -9511, -9455, -9397, -9336, -9272, -9205, -9135, -9063, -8988, -8910, -8829, -8746, -8660, -8572, -8480, -8387, -8290, -8192, -8090, -7986, -7880, -7771, -7660, -7547, -7431, -7314, -7193, -7071, -6947, -6820, -6691, -6561, -6428, -6293, -6157, -6018, -5878, -5736, -5592, -5446, -5299, -5150, -5000, -4848, -4695, -4540, -4384, -4226, -4067, -3907, -3746, -3584, -3420, -3256, -3090, -2924, -2756, -2588, -2419, -2250, -2079, -1908, -1736, -1564, -1392, -1219, -1045, -872, -698, -523, -349, -175}

local mcos = {10000, 9998, 9994, 9986, 9976, 9962, 9945, 9925, 9903, 9877, 9848, 9816, 9781, 9744, 9703, 9659, 9613, 9563, 9511, 9455, 9397, 9336, 9272, 9205, 9135, 9063, 8988, 8910, 8829, 8746, 8660, 8572, 8480, 8387, 8290, 8192, 8090, 7986, 7880, 7771, 7660, 7547, 7431, 7314, 7193, 7071, 6947, 6820, 6691, 6561, 6428, 6293, 6157, 6018, 5878, 5736, 5592, 5446, 5299, 5150, 5000, 4848, 4695, 4540, 4384, 4226, 4067, 3907, 3746, 3584, 3420, 3256, 3090, 2924, 2756, 2588, 2419, 2250, 2079, 1908, 1736, 1564, 1392, 1219, 1045, 872, 698, 523, 349, 175, 0, -175, -349, -523, -698, -872, -1045, -1219, -1392, -1564, -1736, -1908, -2079, -2250, -2419, -2588, -2756, -2924, -3090, -3256, -3420, -3584, -3746, -3907, -4067, -4226, -4384, -4540, -4695, -4848, -5000, -5150, -5299, -5446, -5592, -5736, -5878, -6018, -6157, -6293, -6428, -6561, -6691, -6820, -6947, -7071, -7193, -7314, -7431, -7547, -7660, -7771, -7880, -7986, -8090, -8192, -8290, -8387, -8480, -8572, -8660, -8746, -8829, -8910, -8988, -9063, -9135, -9205, -9272, -9336, -9397, -9455, -9511, -9563, -9613, -9659, -9703, -9744, -9781, -9816, -9848, -9877, -9903, -9925, -9945, -9962, -9976, -9986, -9994, -9998, -10000, -9998, -9994, -9986, -9976, -9962, -9945, -9925, -9903, -9877, -9848, -9816, -9781, -9744, -9703, -9659, -9613, -9563, -9511, -9455, -9397, -9336, -9272, -9205, -9135, -9063, -8988, -8910, -8829, -8746, -8660, -8572, -8480, -8387, -8290, -8192, -8090, -7986, -7880, -7771, -7660, -7547, -7431, -7314, -7193, -7071, -6947, -6820, -6691, -6561, -6428, -6293, -6157, -6018, -5878, -5736, -5592, -5446, -5299, -5150, -5000, -4848, -4695, -4540, -4384, -4226, -4067, -3907, -3746, -3584, -3420, -3256, -3090, -2924, -2756, -2588, -2419, -2250, -2079, -1908, -1736, -1564, -1392, -1219, -1045, -872, -698, -523, -349, -175, 0, 175, 349, 523, 698, 872, 1045, 1219, 1392, 1564, 1736, 1908, 2079, 2250, 2419, 2588, 2756, 2924, 3090, 3256, 3420, 3584, 3746, 3907, 4067, 4226, 4384, 4540, 4695, 4848, 5000, 5150, 5299, 5446, 5592, 5736, 5878, 6018, 6157, 6293, 6428, 6561, 6691, 6820, 6947, 7071, 7193, 7314, 7431, 7547, 7660, 7771, 7880, 7986, 8090, 8192, 8290, 8387, 8480, 8572, 8660, 8746, 8829, 8910, 8988, 9063, 9135, 9205, 9272, 9336, 9397, 9455, 9511, 9563, 9613, 9659, 9703, 9744, 9781, 9816, 9848, 9877, 9903, 9925, 9945, 9962, 9976, 9986, 9994, 9998}

local navTrisInit = false --寻路的网格数据

local estimate -- 误差

BattlePlayerAI.SimpleTNUM = nil

BattlePlayerAI.ComplexTNUM = nil

function BattlePlayerAI.Init(meshinfo)

if not navTrisInit then

NavUtil.InitNavTriInfoFromNavMesh(meshinfo)

BattlePlayerAI.SimpleTNUM = BattleOperation.toInt(BattlePlayers.TNUM / 5)

BattlePlayerAI.ComplexTNUM = BattleOperation.toInt(BattlePlayers.TNUM \* 2)

end

end

function BattlePlayerAI.IsCd(pt,sk)

if pt == nil then

return

end

local rz = -1

for i = 1, #pt.player.save\_re\_id do

if pt.player.save\_re\_id[i] == sk.id then

rz = pt.player.save\_re\_zhen[i]

break

end

end

if rz == -1 then

return false

else

local stime = BattleOperation.toInt((BattlePlayers.zhen - rz) \* BattlePlayers.TICK / sk.get\_cd(pt))

return (stime <= 1)

end

end

function BattlePlayerAI.GetMaxAttackDis(pt) --获得该机器人的最大攻击距离

if pt == nil then

return

end

--自身存在两个技能 如果第二个技能存在的话 比较取其中距离大的

local normal\_skill = Config.get\_t\_skill(100101,1)

local exp\_skill = nil

if pt.player.skill\_id ~= 0 and pt.player.skill\_level ~= 0 then

exp\_skill = Config.get\_t\_skill(pt.player.skill\_id,pt.player.skill\_level)

end

if exp\_skill == nil then

return {normal\_skill.id,normal\_skill.get\_range(pt)}

else

local nor\_dis = normal\_skill.get\_range(pt)

if BattlePlayerAI.IsCd(pt,exp\_skill) then

return {normal\_skill.id,nor\_dis}

else

local exp\_dis = exp\_skill.get\_range(pt)

if nor\_dis > exp\_dis then

return {normal\_skill.id,nor\_dis}

else

return {exp\_skill.id,exp\_dis}

end

end

end

end

function BattlePlayerAI.FindOtherPlayers(bp) --寻找 其他的 玩家(包括机器人)

local playerList = {} --在范围内的人物

local guids = {}

BattleGrid.get(BattleGrid.et\_player, bp.player.x, bp.player.y, guids, bp.player.eyeRange)

for i = 1, #guids do

local g = guids[i]

local bp1 = BattlePlayers.players[g]

if bp1 ~= nil and bp.player.guid ~= bp1.player.guid and not bp1.is\_die and not bp1.player.is\_xueren then

table.insert(playerList, bp1)

end

end

return playerList

end

function BattlePlayerAI.GetAvailablePoint(x,y,r) --获得一个有效的点 在 中心点(x,y)为中心的 矩形中

local x\_min,x\_max,y\_min,y\_max,xx,yy;

if (x - r) >= 0 then

x\_min = x - r

else

x\_min = 0

end

if (x + r)<= 700000 then

x\_max = x + r

else

x\_max = 700000

end

if (y - r)>= 0 then

y\_min = y - r

else

y\_min = 0

end

if (y + r)<= 700000 then

y\_max = y + r

else

y\_max = 700000

end

xx = BattleOperation.random(x\_min, x\_max)

yy = BattleOperation.random(y\_min, y\_max)

while not BattleGrid.can\_move(xx, yy)do

xx = BattleOperation.random(x\_min, x\_max)

yy = BattleOperation.random(y\_min, y\_max)

end

return {xx, yy}

end

function BattlePlayerAI.GetAvailablePathPoint(pathlist,x,y) -- path中的有效点

if pathlist == nil then

return

end

while pathlist.Count > 0 do

if BattlePlayerAI.IsEqual(pathlist[0],pathlist[1],x,y) then

pathlist:RemoveAt(1)

pathlist:RemoveAt(0)

else

if pathlist.Count < 2 then

return

else

return {pathlist[0],pathlist[1]}

end

end

end --while

end

function BattlePlayerAI.GetPatrolPoint(pt) --获得下一个 巡逻点

local tpoint = BattlePlayerAI.GetAvailablePoint(pt.player.x,pt.player.y,pt.player.eyeRange)

local result = NavUtil.GetPathFromNavMesh(pt.player.x,pt.player.y,tpoint[1],tpoint[2])

return BattlePlayerAI.GetAvailablePathPoint(result,pt.player.x,pt.player.y)

--return {tpoint[1], tpoint[2]}

end

function BattlePlayerAI.GetRunPoint(x1,y1,x2,y2) --获得 向 point走去的第一个拐点

local result = NavUtil.GetPathFromNavMesh(x1,y1,x2,y2)

return BattlePlayerAI.GetAvailablePathPoint(result,x1,y1)

--return {x2, y2}

end

function BattlePlayerAI.GetEscapePoint(pt) --获取一个逃跑的位置

if pt == nil then

return

end

--逃跑的方向 取当前视野范围内 敌人数最少的那一边

local areaPlayList = BattlePlayerAI.FindOtherPlayers(pt)

local areaList = {}

local \_index = 1

for i = 1,12,1 do

if areaList[i] == nil then

areaList[i] = {}

end

end

while \_index <= #areaPlayList do --将可见玩家 放进 区域中

local x = areaPlayList[\_index].player.x

local y = areaPlayList[\_index].player.y

local dx = x - pt.player.x

local dy = y - pt.player.y

local angle = BattleOperation.toInt(BattleOperation.get\_r(x, y, pt.player.x, pt.player.y))--BattleOperation.toInt(math.atan2(dy, dx) \* 180 / math.pi) --得到角度 (-180,180)

local order

angle = BattleOperation.checkr(angle) -- 0 到 360

order = math.ceil((angle + 1) / 30)

table.insert(areaList[order],areaPlayList[\_index])

\_index = \_index + 1

end

-- 先判断 是向上 还是 向下 走 然后 判断向左还是向右 最后判断3个区域中那个区域

local topAreaCount = 0

local bottomAreaCount = 0

local leftAreaCount = 0

local rightAreaCount = 0

local sign = 0

local startIndex = 0

local endIndex = 0

for i=1,12,1 do

if #areaList[i] < 7 then

topAreaCount = topAreaCount + 1

else

bottomAreaCount = bottomAreaCount + 1

end

end

if topAreaCount > bottomAreaCount then

sign = 0

for i = 1,6,1 do

if #areaList[i] > 3 then

leftAreaCount = leftAreaCount + 1

else

rightAreaCount = rightAreaCount + 1

end

end

else

sign = 6

for i = 7,12,1 do

if #areaList[i] < 10 then

leftAreaCount = leftAreaCount + 1

else

rightAreaCount = rightAreaCount + 1

end

end

end

if (leftAreaCount > rightAreaCount) and sign == 0 then

startIndex = 4

endIndex = 6

elseif (leftAreaCount <= rightAreaCount) and sign == 0 then

startIndex = 1

endIndex = 3

elseif (leftAreaCount > rightAreaCount) and sign == 6 then

startIndex = 7

endIndex = 9

else

startIndex = 10

endIndex = 12

end

local finalIndex = nil

for i = startIndex,endIndex,1 do

if finalIndex == nil then

finalIndex = i

else

if #areaList[i] < #areaList[finalIndex] then

finalIndex = i

end

end

end

--根据 逃跑方向 求出 一个随机逃跑的点

local r\_x = BattleOperation.random(100000, 700000)

local r\_y = BattleOperation.random(100000, 700000)

while not BattleGrid.can\_move(r\_x, r\_y) do

local ag = BattleOperation.random((finalIndex - 1)\*30,finalIndex \* 30)

local r\_r = BattleOperation.random(-50000,50000)

r\_x = pt.player.x + BattleOperation.toInt(mcos[ag + 1] \* r\_r / 10000)

r\_y = pt.player.y + BattleOperation.toInt(msin[ag + 1] \* r\_r / 10000)

end

return {r\_x,r\_y}

end

function BattlePlayerAI.IsEscape(pt,otherPt) --是否逃跑

--return true

--剩余血量的 百分比

local pt\_percent = BattleOperation.toInt(pt.player.hp / pt.attr.max\_hp() \* 100)

local otherPt\_percent = BattleOperation.toInt(otherPt.player.hp / otherPt.attr.max\_hp() \* 100)

return pt\_percent < 30 and pt\_percent < otherPt\_percent

end

function BattlePlayerAI.GetFoodPoint(pt) --获得食物的 位置

local tids = {} --在范围内的人物

BattleGrid.get(BattleGrid.et\_item, pt.player.x, pt.player.y, tids, pt.player.eyeRange)

if #tids > 0 then

local t = tids[1]

local bi = BattlePlayers.items[t]

if bi ~= nil then

return {bi.item.x, bi.item.y}

end

end

return nil

end

function BattlePlayerAI.ResetRobotState(pt)

if pt == nil then

return

end

pt.player.ai\_state = 0

pt.player.nextPoint\_x = 0

pt.player.nextPoint\_y = 0

pt.player.deathTime = 0

pt.player.totalZhen = 0

end

function BattlePlayerAI.GetR(x,y,x1,y1)

local dx = x1 - x

local dy = y1 - y

local r = math.deg(math.atan2(pt.player.nextPoint\_y - pt.player.y,pt.player.nextPoint\_x - pt.player.x))

return BattleOperation.checkr(r)

end

function BattlePlayerAI.RobotAI(pt)

if pt == nil then

return

end

local speed = BattleOperation.toInt(pt.attr.speed() / 1.5)

estimate = BattleOperation.toInt(speed \* BattlePlayers.TICK / 1000) --每帧的误差

if pt.player.hp <= 0 and pt.player.ai\_state ~= 4 then

pt.player.ai\_state = 4

pt.player.nextPoint\_x = 0

pt.player.nextPoint\_y = 0

pt.player.deathTime = 0

pt.player.totalZhen = 0

return

elseif pt.player.is\_xueren and pt.player.ai\_state ~= 7 then

pt.player.ai\_state = 7

pt.player.nextPoint\_x = 0

pt.player.nextPoint\_y = 0

pt.player.deathTime = 0

pt.player.totalZhen = 0

return

end

--进行简单思考

if pt.player.ai\_state == 0 or pt.player.ai\_state == 2 and pt.player.ai\_state == 7 then --正在进行攻击

if pt.player.totalZhen >= 0 and pt.player.totalZhen < BattlePlayerAI.SimpleTNUM then

pt.player.totalZhen = pt.player.totalZhen + 1

return

end

--进行复杂思考

elseif pt.player.ai\_state == 1 or pt.player.ai\_state == 5 or pt.player.ai\_state == 3 or pt.player.ai\_state == 6 then

if pt.player.totalZhen >= 0 and pt.player.totalZhen < BattlePlayerAI.ComplexTNUM then

pt.player.totalZhen = pt.player.totalZhen + 1

return

end

end

-- 查看非法状态

if pt.player.ai\_state == 4 then

if pt.player.hp > 0 then

BattlePlayerAI.ResetRobotState(pt)

elseif pt.player.deathTime >= 5000 then

BattlePlayers.Fuhuo1(pt)

BattlePlayerAI.ResetRobotState(pt)

else

pt.player.deathTime = pt.player.deathTime + BattlePlayers.TICK

return

end

elseif pt.player.ai\_state == 7 then -- 保证状态7的条件下 人物是雪人

if not pt.player.is\_xueren then

BattlePlayerAI.ResetRobotState(pt)

end

end

-- 雪人状态

if pt.player.ai\_state == 7 then

-- 假如血量超过50%变回来

if pt.player.hp >= pt.attr.max\_hp() \* 0.5 then

BattlePlayers.Release1(pt,200101,0,0,0)

BattlePlayerAI.ResetRobotState(pt)

else

return

end

end

--休息时间过了之后 根据具体情况 进行不同的操作

local areaPlayerList = BattlePlayerAI.FindOtherPlayers(pt)

if #areaPlayerList == 0 then

local np = BattlePlayerAI.GetFoodPoint(pt) --看视野内是否有食物

if np == nil then --没有食物 随机巡逻

np = BattlePlayerAI.GetPatrolPoint(pt)

pt.player.ai\_state = 1

else

--print("食物点:",pt.player.x,pt.player.y,np[1],np[2])

local fpt = NavUtil.GetPathFromNavMesh(pt.player.x,pt.player.y,np[1],np[2])

np = BattlePlayerAI.GetAvailablePathPoint(fpt,pt.player.x,pt.player.y)

pt.player.ai\_state = 6 -- 食物

end

if np == nil then

BattlePlayerAI.ResetRobotState(pt)

return

end

--print("抉择的state:",pt.player.ai\_state,pt.player.x,pt.player.y,np[1],np[2])

pt.player.nextPoint\_x = np[1]

pt.player.nextPoint\_y = np[2]

pt.player.deathTime = 0

pt.player.totalZhen = 0

local r = BattlePlayerAI.GetR(pt.player.x,pt.player.y,pt.player.nextPoint\_x,pt.player.nextPoint\_y)

BattlePlayers.Move1(pt,r)

return

else

local escapeState = BattlePlayerAI.IsEscape(pt, areaPlayerList[1]) --是否逃跑

if escapeState then

--如果逃跑并且血量低于 30% 变成雪人状态 ai\_state = 7

local bloodPert = BattleOperation.toInt(pt.player.hp \* 100 / pt.attr.max\_hp())

if bloodPert <= 30 then

local cds = BattlePlayerAI.IsCd(pt,Config.get\_t\_skill(200101,1))

if not cds then

pt.player.ai\_state = 7 -- 雪人状态

pt.player.nextPoint\_x = 0

pt.player.nextPoint\_y = 0

pt.player.deathTime = 0

pt.player.totalZhen = 0

BattlePlayers.Release1(pt,200101,0,0,0)

return

end

end

local np = BattlePlayerAI.GetEscapePoint(pt)

if np ~= nil then

local fpt = NavUtil.GetPathFromNavMesh(pt.player.x,pt.player.y,np[1],np[2])

np = BattlePlayerAI.GetAvailablePathPoint(fpt,pt.player.x,pt.player.y)

if np == nil then

BattlePlayerAI.Stop(pt)

BattlePlayerAI.ResetRobotState(pt)

return

end

pt.player.ai\_state = 5 --逃跑状态

pt.player.nextPoint\_x = np[1] --逃跑的位置

pt.player.nextPoint\_y = np[2] --逃跑的位置

pt.player.deathTime = 0

pt.player.totalZhen = 0

local r = BattlePlayerAI.GetR(pt.player.x,pt.player.y,np[1],np[2])

BattlePlayers.Move1(pt,r)

return

else

BattlePlayerAI.Stop(pt)

BattlePlayerAI.ResetRobotState(pt)

return

end

end

local skillInfo = BattlePlayerAI.GetMaxAttackDis(pt) --技能 和 范围

local dvalue = BattleOperation.get\_distance(areaPlayerList[1].player.x,areaPlayerList[1].player.y,pt.player.x,pt.player.y)

if skillInfo[2] >= dvalue then --如果 进入可以释放技能的范围

BattlePlayerAI.Stop(pt) --停止行走动作

pt.player.ai\_state = 2 --攻击状态

pt.player.nextPoint\_x = areaPlayerList[1].player.x --攻击位置

pt.player.nextPoint\_y = areaPlayerList[1].player.y --攻击位置

pt.player.deathTime = 0

pt.player.totalZhen = 0

local r = BattlePlayerAI.GetR(pt.player.x,pt.player.y,areaPlayerList[1].player.x,areaPlayerList[1].player.y)

BattlePlayers.Release1(pt,skillInfo[1],areaPlayerList[1].player.x,areaPlayerList[1].player.y,r)

else

local np = BattlePlayerAI.GetRunPoint(pt.player.x,pt.player.y,areaPlayerList[1].player.x,areaPlayerList[1].player.y)

if np == nil then

BattlePlayerAI.Stop(pt)

BattlePlayerAI.ResetRobotState(pt)

return

end

pt.player.ai\_state = 3 --朝目标走去

pt.player.nextPoint\_x = np[1]

pt.player.nextPoint\_y = np[2]

pt.player.deathTime = 0

pt.player.totalZhen = 0

local r = BattlePlayerAI.GetR(pt.player.x,pt.player.y,np[1],np[2])

BattlePlayers.Move1(pt,r)

end

end

end

function BattlePlayerAI.IsEqual(x1,y1,x2,y2) --比较是否相等

return (not BattleOperation.check\_distance(x1,y1,x2,y2,estimate))

end

function BattlePlayerAI.GetR(x1,y1,x2,y2)

return BattleOperation.get\_r(x1, y1, x2, y2)

end

function BattlePlayerAI.Stop(pt)

if pt == nil then

return

end

if pt.is\_die then

return

end

BattlePlayers.Stop1(pt)

end

BattlePlayers = {}

BattlePlayers.players = {}

BattlePlayers.effects = {}

BattlePlayers.items = {}

BattlePlayers.item\_follows = {}

BattlePlayers.item\_speeds = {}

BattlePlayers.item\_num = {}

BattlePlayers.init\_item = false

BattlePlayers.operations\_head = nil

BattlePlayers.operations\_tail = nil

BattlePlayers.zhen = 0

BattlePlayers.max\_zhen = -1

BattlePlayers.dtime = 0

BattlePlayers.ptime = 0

BattlePlayers.start = false

BattlePlayers.jiasu = false

BattlePlayers.seeds = {}

BattlePlayers.TICK = 50 --每帧的ms

BattlePlayers.TNUM = 20 --每秒跑的帧数

BattlePlayers.STATE\_TIME = 5

BattlePlayers.me = nil

BattlePlayers.dis = 170

function BattlePlayers.Init()

BattlePlayers.players = {}

BattlePlayers.effects = {}

BattlePlayers.items = {}

BattlePlayers.item\_follows = {}

BattlePlayers.item\_speeds = {}

BattlePlayers.item\_num = {0, 0, 0, 0}

BattlePlayers.init\_item = false

BattlePlayers.operations\_head = nil

BattlePlayers.operations\_tail = nil

BattlePlayers.zhen = 0

BattlePlayers.max\_zhen = -1

BattlePlayers.dtime = 0

BattlePlayers.ptime = 0

BattlePlayers.start = true

BattlePlayers.jiasu = false

BattlePlayers.seeds = {}

BattlePlayers.me = nil

BattlePlayers.dis = 170

BattleGrid.Init(Battle.name)

end

function BattlePlayers.Fini()

Battle.bobjpool:clear()

BattlePanel.clear\_min\_pro()

BattleGrid.Fini()

for guid, bp in pairsByKeys(BattlePlayers.players) do

GameObject.Destroy(bp.posobj)

end

BattlePlayers.players = {}

for tid, be in pairsByKeys(BattlePlayers.effects) do

resMgr:DeleteEffect(be.obj)

end

BattlePlayers.effects = {}

for tid, bi in pairsByKeys(BattlePlayers.items) do

resMgr:DeleteEffect(bi.obj)

end

BattlePlayers.items = {}

BattlePlayers.start = false

Time.timeScale = 1

end

function BattlePlayers.init\_player(player)

local p = BattleOperation.get\_avilialbe\_xy()

player.x = p[1]

player.y = p[2]

player.r = 0

player.r\_py = 0

player.is\_move = false

player.re\_state = 0

player.re\_id = 0

player.re\_time = 0

player.re\_x = 0

player.re\_y = 0

player.re\_r = 0

player.save\_re\_id:clear()

player.save\_re\_zhen:clear()

player.is\_xueren = false

player.is\_jf = false

player.jf\_sx = 0

player.jf\_sy = 0

player.jf\_xx = 0

player.jf\_yy = 0

player.jf\_speed = 0

player.jf\_r = 0

player.level = 1

player.exp = 0

player.skill\_id = 0

player.skill\_level = 0

player.level\_duo = 0

player.level\_range = 0

player.level\_attack = 0

player.level\_big = 0

player.level\_fspeed = 0

player.level\_speed = 0

player.hp = 0

player.buffs:clear()

player.buffs\_time:clear()

player.lattr\_value:clear()

for i = 1, 9 do

player.lattr\_value:append(0)

end

player.talent\_id:clear()

player.talent\_level:clear()

player.talent\_point = 0

end

function BattlePlayers.In(msg)

local player = msg\_battle\_pb.msg\_battle\_player()

BattlePlayers.init\_player(player)

player.guid = msg.guid

player.name = msg.param\_strings[1]

player.role\_id = msg.param\_ints[1]

player.role\_level = msg.param\_ints[2]

player.sex = msg.param\_ints[3]

player.avatar = msg.param\_ints[4]

player.cup = msg.param\_ints[5]

player.toukuang = msg.param\_ints[6]

player.skill\_id = msg.param\_ints[7]

player.skill\_level = msg.param\_ints[8]

player.avatar\_url = msg.param\_strings[2]

player.re\_tid = 0

player.is\_ai = 0

local num = msg.param\_ints[9]

for i = 0, num - 1 do

player.attr\_type:append(msg.param\_ints[10 + i \* 4])

player.attr\_param1:append(msg.param\_ints[11 + i \* 4])

player.attr\_param2:append(msg.param\_ints[12 + i \* 4])

player.attr\_param3:append(msg.param\_ints[13 + i \* 4])

end

local new\_index = num \* 4 + 8

BattlePlayers.addplayer(player, true)

end

function BattlePlayers.InitAIBaseInfo(player)

player.eyeRange = BattleOperation.random(10000, 40000)\*2

player.ai\_state = 0

player.ai\_type = 0

player.nextPoint\_x = 1

player.nextPoint\_y = 1

player.totalZhen = 0

player.deathTime = 0

end

function BattlePlayers.add\_ai(id)

local player = msg\_battle\_pb.msg\_battle\_player()

BattlePlayers.init\_player(player)

player.guid = tostring(id)

player.name = Config.get\_battle\_random\_name()

player.role\_id = Config.get\_battle\_random\_role()

player.role\_level = 1

player.sex = Config.get\_t\_role(player.role\_id).sex

player.avatar = 1010000 + player.role\_id

player.cup = BattleOperation.random(0, 2500)

player.avatar\_url = ""

player.toukuang = Config.get\_battle\_random\_toukuang()

player.re\_tid = 0

player.is\_ai = 1

BattlePlayers.InitAIBaseInfo(player)

BattlePlayers.addplayer(player, true)

end

function BattlePlayers.addplayer(player, is\_new)

is\_new = is\_new or false

player.r = BattleOperation.checkr(player.r)

if is\_new and player.is\_ai == 0 then

if BattlePlayers.players[player.guid] ~= nil then

--顶替

local bp = BattlePlayers.players[player.guid]

bp.player.is\_ai = 0

bp.player.is\_move = false

if self.guid == bp.player.guid then

BattlePlayers.me = bp

BattlePanel.InitBattlePanel()

end

return

else

--删除

for guid, bp in pairsByKeys(BattlePlayers.players) do

if tonumber(guid) < 100 then

BattlePlayers.delplayer(guid)

break

end

end

end

end

local t\_role = Config.get\_t\_role(player.role\_id)

if t\_role == nil then

return

end

local obj = resMgr:CreateUnit(t\_role.res, false)

local objid = Battle.bobjpool:add(obj)

local posobj = GameObject.New()

local posobjid = Battle.bobjpool:add(posobj)

local posobjt = posobj.transform

posobjt.parent = resMgr.UnitRoot

Battle.bobjpool:set\_localPosition(posobjid, player.x / Battle.BL, 0, player.y / Battle.BL)

Battle.bobjpool:set\_localEulerAngles(posobjid, 10, 0, 0)

Battle.bobjpool:set\_localScale(posobjid, 1, 1, 1)

local objt = obj.transform

obj.transform.parent = posobj.transform

Battle.bobjpool:set\_localPosition(objid, 0, 0, 0)

Battle.bobjpool:set\_localEulerAngles(objid, 0, 90 - player.r, 0)

Battle.bobjpool:set\_localScale(objid, 1, 1, 1)

local cao = BattleGrid.get\_cao(player.x, player.y)

local unit = obj:GetComponent("unit")

local achieveRecords = {}

local accept = unit:get\_bone("accept")

-- mtk 做标示表示 在血量20% 以上被主角攻击 设置 true 死亡后刷新

BattlePlayers.players[player.guid] = {player = player, action = "", action\_speed = 1, jfr = 0, alpha = 1, last = {{player.x, player.y, player.is\_jf, nil}},

obj = obj, objt = obj.transform, objid = objid, posobj = posobj, posobjt = posobjt, posobjid = posobjid, accept = accept,

cao = cao, unit = unit, ur\_ = unit:get\_round(), is\_die = false, achieveRecords = achieveRecords,mtk = false}

BattleGrid.add(BattleGrid.et\_player, player.guid, player.x, player.y)

local bp = BattlePlayers.players[player.guid]

BattleAttr.init\_attr(bp)

if is\_new then

bp.set\_hp(bp.attr.max\_hp())

end

if bp.player.is\_xueren then

BattlePlayers.MakeXueren(bp, true, true)

end

BattlePlayers.CalcCao(bp)

if self.guid == bp.player.guid then

BattlePlayers.me = bp

BattlePanel.InitBattlePanel()

end

if bp.player.hp <= 0 then

bp.is\_die = true

if BattlePlayers.me ~= nil then

if BattlePlayers.me.player.guid == bp.player.guid then

BattlePanel.show\_die(nil)

end

end

end

BattlePanel.add\_min\_pro(bp)

end

function BattlePlayers.Out(msg)

local guid = msg.guid

if BattlePlayers.players[guid] == nil then

return

end

local bp = BattlePlayers.players[guid]

bp.player.is\_ai = 1

BattlePlayers.InitAIBaseInfo(bp.player)

end

function BattlePlayers.delplayer(guid)

if BattlePlayers.players[guid] == nil then

return

end

local bp = BattlePlayers.players[guid]

Battle.bobjpool:remove(bp.objid)

Battle.bobjpool:remove(bp.posobjid)

GameObject.Destroy(bp.posobj)

BattlePlayers.players[guid] = nil

BattleGrid.del(BattleGrid.et\_player, guid)

BattlePanel.del\_min\_pro(guid)

if self.guid == guid then

BattlePlayers.me = nil

end

end

function BattlePlayers.add\_effect(re\_tid, id, x, y, xx, yy, r, re\_guid, re\_ur)

r = BattleOperation.checkr(r)

local re\_bp = BattlePlayers.players[re\_guid]

if re\_bp == nil then

return

end

local effect = msg\_battle\_pb.msg\_battle\_effect()

effect.tid = BattlePlayers.tid

BattlePlayers.tid = BattlePlayers.tid + 1

effect.re\_tid = re\_tid

effect.id = id

effect.sx = x

effect.sy = y

effect.x = x

effect.y = y

effect.xx = xx

effect.yy = yy

effect.r = r

effect.re\_guid = re\_guid

effect.re\_ur = re\_ur

effect.time = BattlePlayers.zhen

effect.dd\_time = BattlePlayers.zhen

effect.state = 0

local t\_skill\_effect = Config.get\_t\_skill\_effect(effect.id)

if t\_skill\_effect == nil then

return

end

effect.len = 0

if t\_skill\_effect.type == 1 then

effect.len = t\_skill\_effect.get\_range(re\_bp) - effect.re\_ur

elseif t\_skill\_effect.type == 2 then

effect.len = BattleOperation.get\_distance(effect.sx, effect.sy, effect.xx, effect.yy)

if effect.len > t\_skill\_effect.get\_range(re\_bp) then

effect.len = t\_skill\_effect.get\_range(re\_bp)

end

end

if t\_skill\_effect.type == 7 or (t\_skill\_effect.type < 2 and t\_skill\_effect.fx\_speed == 0) then

effect.sx = effect.xx

effect.sy = effect.yy

effect.x = effect.xx

effect.y = effect.yy

end

--如果当前的技能是自己释放 且 技能id不是普攻 设置状态

if self.guid == re\_guid and id ~= 10010101 and BattlePlayers.me.achieveRecords.IsUseSkill == nil then

BattlePlayers.me.achieveRecords.IsUseSkill = true

end

BattlePlayers.add\_effect2(effect)

end

function BattlePlayers.add\_effect2(effect)

effect.r = BattleOperation.checkr(effect.r)

local t\_skill\_effect = Config.get\_t\_skill\_effect(effect.id)

if t\_skill\_effect == nil then

return

end

local re\_bp = BattlePlayers.players[effect.re\_guid]

local obj = nil

local objt = nil

local objid = nil

if t\_skill\_effect.effect ~= "" then

obj = resMgr:CreateEffect(t\_skill\_effect.effect)

objid = Battle.bobjpool:add(obj)

objt = obj.transform

objt.parent = resMgr.UnitRoot

Battle.bobjpool:set\_localPosition(objid, effect.x / Battle.BL, t\_skill\_effect.fx\_hight / Battle.BL, effect.y / Battle.BL)

Battle.bobjpool:set\_localEulerAngles(objid, 0, 90 - effect.r, 0)

Battle.bobjpool:set\_localScale(objid, t\_skill\_effect.get\_effect\_scale(re\_bp), t\_skill\_effect.get\_effect\_scale(re\_bp), t\_skill\_effect.get\_effect\_scale(re\_bp))

end

if t\_skill\_effect.is\_zp == 1 then

if BattlePlayers.me ~= nil then

local d = BattleOperation.get\_distance(effect.x, effect.y, BattlePlayers.me.player.x, BattlePlayers.me.player.y)

if d < 120000 \* BattlePlayers.me.get\_scale() then

mapMgr:shake\_cam(0.5)

end

end

end

local destroyEffects = {} --添加 记录当前这个特效 摧毁其他的特效

local effect\_hums = {} --添加 记录 当前特效 击杀和伤害的人

BattlePlayers.effects[effect.tid] = {effect = effect, obj = obj, objt = objt, objid = objid, last = {{effect.x, effect.y, nil}},

destroyEffects = destroyEffects,effect\_hums = effect\_hums, re\_bp = re\_bp}

local be = BattlePlayers.effects[effect.tid]

if obj ~= nil and t\_skill\_effect.type <= 2 then

be.p0 = Vector3(effect.sx / Battle.BL, t\_skill\_effect.fx\_hight / Battle.BL, effect.sy / Battle.BL)

if t\_skill\_effect.type == 1 then

local p = BattleOperation.add\_distance2(effect.sx, effect.sy, effect.r, effect.len)

be.p3 = Vector3(p[1] / Battle.BL, 0, p[2] / Battle.BL)

elseif t\_skill\_effect.type == 2 then

local p = BattleOperation.add\_distance2(effect.sx, effect.sy, effect.r, effect.len)

be.p3 = Vector3(p[1] / Battle.BL, 0, p[2] / Battle.BL)

end

be.p1 = be.p0 + (be.p3 - be.p0) / 2

be.p1.y = t\_skill\_effect.fx\_hight1 / Battle.BL

be.p2 = be.p1 + (be.p3 - be.p1) / 2

be.p2.y = t\_skill\_effect.fx\_hight2 / Battle.BL

local pos = BattleOperation.get\_distance(effect.sx, effect.sy, effect.x, effect.y)

be.pt = pos / effect.len

be.last[1][3] = be.pt

end

BattleGrid.add(BattleGrid.et\_effect, effect.tid, effect.x, effect.y)

end

function BattlePlayers.del\_effect(tid)

if BattlePlayers.effects[tid] == nil then

return

end

local be = BattlePlayers.effects[tid]

--删除的时候 遍历下特效 看看有没有后续的 技能 如果有 则暂时不结算

if be.effect.re\_guid == self.guid then

--在删除这个特效的时候 判断 这个技能有没有达到某个成就

local skill\_id = Config.get\_t\_skill\_effect(be.effect.id).skill\_id

local sign = false

for k,data in pairsByKeys(BattlePlayers.effects) do

if k ~= tid and data.effect.re\_tid == be.effect.re\_tid then

for skd,snum in pairsByKeys(be.destroyEffects) do

if data.destroyEffects[skd] ~= nil then

data.destroyEffects[skd] = data.destroyEffects[skd] + snum

else

data.destroyEffects[skd] = snum

end

end

sign = true

end

end

if not sign then

for k,v in pairsByKeys(be.destroyEffects) do

BattleAchieve.OnlyBattleSkillRelationForSkill(skill\_id,k,v)

end

be.destroyEffects = {}

end

--判断该技能杀人数 或者 击中数 有没有达到要求

sign = false

for k,data in pairsByKeys(BattlePlayers.effects) do

if k ~= tid and data.effect.re\_tid == be.effect.re\_tid then

for ink,inv in pairsByKeys(be.effect\_hums) do

if (inv == 1 and data.effect\_hums[ink] == 0) or (inv == 0 and data.effect\_hums[ink] == 1) then

data.effect\_hums[ink] = 0

else

data.effect\_hums[ink] = inv

end

end

sign = true

end

end

if not sign then

BattleAchieve.OnlyBattleSkillRelationForMan(skill\_id,be.effect\_hums)

end

end

if be.obj ~= nil then

Battle.bobjpool:remove(be.objid)

resMgr:DeleteEffect(be.obj)

end

BattlePlayers.effects[tid] = nil

BattleGrid.del(BattleGrid.et\_effect, tid)

end

function BattlePlayers.do\_check\_item()

while BattlePlayers.item\_num[1] < 80 do

local p = BattleOperation.get\_avilialbe\_xy1()

BattlePlayers.add\_item(0, p[1], p[2])

end

while BattlePlayers.item\_num[2] < 6 do

local p = BattleOperation.get\_avilialbe\_xy1()

local r = BattleOperation.random(0, 99 + 1)

local id = 1

if r < 20 then

id = 2

end

BattlePlayers.add\_item(id, p[1], p[2])

end

while BattlePlayers.item\_num[3] < 6 do

local p = BattleOperation.get\_avilialbe\_xy1()

local id = BattleOperation.random(100, 105 + 1)

BattlePlayers.add\_item(id, p[1], p[2])

end

while BattlePlayers.item\_num[4] < 6 do

local p = BattleOperation.get\_avilialbe\_xy1()

local id = BattleOperation.random(200, 207 + 1)

BattlePlayers.add\_item(id, p[1], p[2])

end

end

function BattlePlayers.init\_ai()

for i = 1, 20 do

BattlePlayers.add\_ai(i)

end

end

function BattlePlayers.add\_item(id, x, y)

local item = msg\_battle\_pb.msg\_battle\_item()

item.tid = BattlePlayers.tid

BattlePlayers.tid = BattlePlayers.tid + 1

item.id = id

item.x = x

item.y = y

local param = 1

if id == 0 then

local r = BattleOperation.random(0, 100)

if r < 5 then

param = 2

elseif r < 20 then

param = 5

end

end

item.param = param

item.zhen = BattlePlayers.zhen

return BattlePlayers.add\_item2(item)

end

function BattlePlayers.add\_item2(item)

local t\_battle\_item = Config.get\_t\_battle\_item(item.id)

if t\_battle\_item == nil then

return

end

local obj = resMgr:CreateEffect("rongqi")

local objid = Battle.bobjpool:add(obj)

local objt = obj.transform

objt.parent = resMgr.UnitRoot

Battle.bobjpool:set\_localPosition(objid, item.x / Battle.BL, 0.1, item.y / Battle.BL)

Battle.bobjpool:set\_localEulerAngles(objid, 0, 0, 0)

local obj1 = resMgr:CreateEffect(t\_battle\_item.effect)

local objt1 = obj1.transform

objt1.parent = objt:FindChild("sub")

objt1.localPosition = Vector3.zero

objt1.localEulerAngles = Vector3(0, math.random(0, 359), 0)

objt1.localScale = Vector3.one

local scale = 0.5 + item.param \* 0.5

if scale > 2 then

scale = 2

end

Battle.bobjpool:set\_localScale(objid, scale, scale, scale)

BattlePlayers.items[item.tid] = {item = item, obj = obj, objt = objt, objid = objid}

BattleGrid.add(BattleGrid.et\_item, item.tid, item.x, item.y)

BattlePlayers.item\_num[t\_battle\_item.type] = BattlePlayers.item\_num[t\_battle\_item.type] + 1

return BattlePlayers.items[item.tid]

end

function BattlePlayers.add\_random\_item(id, x, y)

local p = BattleOperation.get\_avilialbe\_xy2(x, y)

local bi = BattlePlayers.add\_item(id, p[1], p[2])

bi.xx = x / Battle.BL

bi.yy = y / Battle.BL

bi.speed = BattleOperation.get\_distance(x, y, p[1], p[2]) / 10000 / 0.7

local at = bi.obj:GetComponent("Animator")

at:Play("rongqi\_put")

table.insert(BattlePlayers.item\_speeds, bi)

end

function BattlePlayers.del\_item(tid, bp)

if BattlePlayers.items[tid] == nil then

return

end

local bi = BattlePlayers.items[tid]

local t\_battle\_item = Config.get\_t\_battle\_item(bi.item.id)

if t\_battle\_item == nil then

return

end

if bi.obj ~= nil and bp ~= nil then

bi.follow = bp.player.guid

bi.follow\_time = 0

local at = bi.obj:GetComponent("Animator")

at:Play("rongqi\_get")

bi.xx = bi.item.x / Battle.BL

bi.yy = bi.item.y / Battle.BL

bi.follow\_speed = 1 + bp.attr\_value[33] / 100

BattlePlayers.item\_follows[tid] = bi

end

BattlePlayers.item\_num[t\_battle\_item.type] = BattlePlayers.item\_num[t\_battle\_item.type] - 1

BattlePlayers.items[tid] = nil

BattleGrid.del(BattleGrid.et\_item, tid)

end

function BattlePlayers.Move(msg)

local guid = msg.guid

local r = msg.param\_ints[1]

if BattlePlayers.players[guid] == nil then

return

end

local bp = BattlePlayers.players[guid]

BattlePlayers.Move1(bp, r)

end

function BattlePlayers.Move1(bp, r)

if bp.attr.is\_hunluan() then

r = r + 180

end

r = BattleOperation.checkr(r)

bp.player.r = r

bp.player.r\_py = 0

bp.player.is\_move = true

end

function BattlePlayers.Stop(msg)

local guid = msg.guid

if BattlePlayers.players[guid] == nil then

return

end

local bp = BattlePlayers.players[guid]

BattlePlayers.Stop1(bp)

end

function BattlePlayers.Stop1(bp)

bp.player.is\_move = false

bp.player.r\_py = 0

if bp.player.re\_state > 0 then

bp.player.r = bp.player.re\_r

end

end

function BattlePlayers.Release(msg)

local guid = msg.guid

local id = msg.param\_ints[1]

local x = msg.param\_ints[2]

local y = msg.param\_ints[3]

local r = msg.param\_ints[4]

if BattlePlayers.players[guid] == nil then

return

end

local bp = BattlePlayers.players[guid]

BattlePlayers.Release1(bp, id, x, y, r)

end

function BattlePlayers.Release1(bp, id, x, y, r)

if bp.attr.is\_hunluan() then

r = r + 180

x = bp.player.x \* 2 - x

y = bp.player.y \* 2 - y

end

r = BattleOperation.checkr(r)

if bp.player.re\_state > 0 then --如果上一帧是其他状态 不能释放技能

return

end

local t\_skill = Config.get\_t\_skill(id, bp.get\_skill\_level(id))

if t\_skill == nil then

return

end

if not bp.can\_do() then

return

end

if bp.player.is\_xueren and t\_skill.type ~= 3 then

return

end

-- 判断技能位置对不对

if t\_skill.release\_type == 3 then

if BattleOperation.check\_distance(x, y, bp.player.x, bp.player.y, t\_skill.get\_range(bp) + 5000) then

return

end

end

if t\_skill.link\_effect > 0 then

local t\_skill\_effect = Config.get\_t\_skill\_effect(t\_skill.link\_effect)

if t\_skill\_effect ~= nil and t\_skill.release\_type == 2 and t\_skill\_effect.type == 2 then

local p = BattleOperation.add\_distance2(bp.player.x, bp.player.y, r, t\_skill.get\_range(bp))

x = p[1]

y = p[2]

end

end

--

local index = -1

for i = 1, #bp.player.save\_re\_id do

if bp.player.save\_re\_id[i] == id then

index = i

break

end

end

if index == -1 then

index = #bp.player.save\_re\_id + 1

bp.player.save\_re\_id:append(id)

bp.player.save\_re\_zhen:append(-999999)

end

local ttime = (BattlePlayers.zhen - bp.player.save\_re\_zhen[index]) \* BattlePlayers.TICK

if t\_skill.type == 1 then

ttime = BattleOperation.toInt(ttime \* (10000 + bp.attr.aspeed()) / 10000)

end

if ttime <= t\_skill.get\_cd(bp) and t\_skill.type ~= 3 then

return

end

if t\_skill.type == 3 then

if not bp.player.is\_xueren then

BattlePlayers.MakeXueren(bp, true)

bp.player.xueren\_zhen = BattlePlayers.zhen

if bp.player.guid == self.guid then

bp.achieveRecords.useXueren = true

end

bp.player.save\_re\_zhen[index] = BattlePlayers.zhen

else

BattlePlayers.MakeXueren(bp, false)

--bp.player.save\_re\_zhen[index] = BattlePlayers.zhen

end

return

-- 释放技能回血

elseif t\_skill.type == 2 then

bp.yy(2)

if BattleOperation.random(0, 100) < bp.attr\_value[27] then

bp.set\_hp(bp.attr.max\_hp(), true)

else

bp.set\_hp(bp.player.hp + bp.attr\_value[26], true)

end

if BattleOperation.random(0, 100) >= bp.attr\_value[30] then

bp.player.save\_re\_zhen[index] = BattlePlayers.zhen

end

else

if BattleOperation.random(0, 100) < 20 then

bp.yy(4)

end

bp.player.save\_re\_zhen[index] = BattlePlayers.zhen

end

bp.player.re\_id = id

bp.player.re\_state = 1

bp.player.re\_time = 0

bp.player.re\_x = x

bp.player.re\_y = y

bp.player.re\_r = r

bp.player.re\_tid = bp.player.re\_tid + 1

end

function BattlePlayers.MakeXueren(bp, flag, is\_init)

bp.player.is\_xueren = flag

bp.alpha = -1

if bp.xobj ~= nil then

Battle.bobjpool:remove(bp.xobjid)

GameObject.Destroy(bp.xobj)

bp.xobj = nil

bp.xobjt = nil

bp.xunit = nil

end

if flag then

local s = "snowman\_f01"

if bp.sex() == 0 then

s = "snowman\_m01"

end

local xobj = resMgr:CreateUnit(s, false)

local xobjid = Battle.bobjpool:add(xobj)

local xobjt = xobj.transform

xobjt.parent = bp.posobjt

Battle.bobjpool:set\_localPosition(xobjid, 0, 0, 0)

Battle.bobjpool:set\_localEulerAngles(xobjid, 0, 90 - bp.player.r, 0)

Battle.bobjpool:set\_localScale(xobjid, 1, 1, 1)

bp.xobj = xobj

bp.xobjt = xobjt

bp.xunit = bp.xobj:GetComponent("unit")

bp.xobjid = xobjid

bp.unit:pause\_action()

Battle.bobjpool:set\_localScale(bp.objid, 0, 0, 0)

if not is\_init then

bp.add\_buff(1001)

end

BattlePlayers.Attach1(bp, "accept", "Unit\_Release\_snowman")

else

Battle.bobjpool:set\_localScale(bp.objid, 1, 1, 1)

bp.unit:continue\_action()

bp.remove\_buff(1001)

BattlePlayers.Attach1(bp, "accept", "Unit\_die\_snowman")

end

end

function BattlePlayers.Fuhuo(msg)

local guid = msg.guid

if BattlePlayers.players[guid] == nil then

return

end

local bp = BattlePlayers.players[guid]

BattlePlayers.Fuhuo1(bp)

end

function BattlePlayers.Attach(bp, bone, name, yj)

if yj == nil then

yj = false

end

return bp.unit:Attach(bone, name, yj)

end

function BattlePlayers.Attach1(bp, bone, name, yj)

if yj == nil then

yj = false

end

local eff = bp.unit:Attach1(bone, name, yj)

if eff ~= nil then

local s = bp.get\_scale()

eff.transform.localScale = Vector3(s, s, s)

end

return eff

end

function BattlePlayers.Disappear(x, y, h, r, s, name)

if mapMgr:in\_view(x / Battle.BL, h, y / Battle.BL) == -1 then

return

end

local eff = resMgr:CreateEffect(name)

local efft = eff.transform

efft.parent = resMgr.UnitRoot

efft.localPosition = Vector3(x / Battle.BL, h, y / Battle.BL)

efft.localEulerAngles = Vector3(0, 90 - r, 0)

efft.localScale = Vector3(s, s, s)

resMgr:DeleteEffect(eff, 2)

end

function BattlePlayers.Attack(bp, be)

local re\_bp = be.re\_bp

local t\_skill\_effect = Config.get\_t\_skill\_effect(be.effect.id)

if t\_skill\_effect == nil then

return

end

if t\_skill\_effect.xiaoshi\_effect2 ~= "" then

local h = 0.1

if be.obj ~= nil then

local ox, oy, oz = Battle.bobjpool:get\_localPosition(be.objid, nil, nil, nil)

h = oy

end

BattlePlayers.Disappear(be.effect.x, be.effect.y, h, be.effect.r, t\_skill\_effect.get\_effect\_scale(be.re\_bp), t\_skill\_effect.xiaoshi\_effect2)

end

local t\_skill\_xiaoguo = Config.get\_t\_skill\_xiaoguo(t\_skill\_effect.link\_xiaoguo)

if t\_skill\_xiaoguo == nil then

return

end

if bp.player.is\_xueren then

if t\_skill\_xiaoguo.sj\_effect ~= "" then

local h = 0.1

if be.obj ~= nil then

local ox, oy, oz = Battle.bobjpool:get\_localPosition(be.objid, nil, nil, nil)

h = oy

end

BattlePlayers.Disappear(be.effect.x, be.effect.y, h, be.effect.r, t\_skill\_effect.get\_effect\_scale(be.re\_bp), t\_skill\_xiaoguo.sj\_effect)

end

bp.xunit:action("injured")

if t\_skill\_xiaoguo.jf\_type > 0 then

bp.set\_jf(be.effect.r, t\_skill\_xiaoguo.jf\_dis, t\_skill\_xiaoguo.jf\_speed)

end

return

end

if t\_skill\_xiaoguo.sj\_effect ~= "" then

BattlePlayers.Attach(bp, "accept", t\_skill\_xiaoguo.sj\_effect)

end

if BattlePlayers.me ~= nil then

if be.effect.re\_guid == BattlePlayers.me.player.guid then

bp.unit:white()

end

end

-- 普攻免疫

if t\_skill\_effect.skill\_type == 1 then

if BattleOperation.random(0, 100) < bp.attr\_value[22] then

BattlePanel.add\_text(bp.accept.position, "免疫", 4)

return

end

end

-- 无敌

if bp.attr\_value[105] > 0 then

BattlePanel.add\_text(bp.accept.position, "免疫", 4)

return

end

-- 盾挡

if bp.player.lattr\_value[9] > 0 then

bp.player.lattr\_value[9] = bp.player.lattr\_value[9] - 1

if bp.player.lattr\_value[9] == 0 then

bp.remove\_buff(3001)

end

BattlePanel.add\_text(bp.accept.position, "免疫", 4)

return

end

-- 对身上携带的技能免疫

if t\_skill\_effect.skill\_type == 2 and re\_bp.player.skill\_id ~= 0 and bp.player.skill\_id == re\_bp.player.skill\_id and BattleOperation.random(0, 100) < bp.attr\_value[89] then

BattlePanel.add\_text(bp.accept.position, "免疫", 4)

return

end

-- 伤害计算

local cri = false

if BattleOperation.random(0, 100) < re\_bp.attr\_value[9] then

cri = true

end

local attack = re\_bp.attr.atk()

attack = t\_skill\_xiaoguo.dmg\_per \* attack / 100 + t\_skill\_xiaoguo.dmg\_gd

local zjs = 100

-- 增伤减伤

zjs = zjs + re\_bp.attr.zs() - bp.attr.js()

-- 技能增伤减伤

if t\_skill\_effect.skill\_type ~= 1 then

zjs = zjs + re\_bp.attr\_value[16] - bp.attr\_value[17] + re\_bp.get\_skill\_attr\_value(t\_skill\_effect.skill\_id, 1)

end

-- 普攻加伤害

if t\_skill\_effect.skill\_type == 1 then

attack = attack + re\_bp.attr\_value[72]

zjs = zjs + re\_bp.attr\_value[46] + re\_bp.player.level\_attack \* 5

end

-- 普攻飞行距离算伤害

if t\_skill\_effect.skill\_type == 1 and re\_bp.attr\_value[42] ~= 0 then

local dis = BattleOperation.get\_distance(bp.player.x, bp.player.y, re\_bp.player.x, re\_bp.player.y)

zjs = zjs + dis / Battle.BL \* re\_bp.attr\_value[42]

end

-- 对方携带相同技能，伤害增加百分比

if re\_bp.player.skill\_id ~= 0 and bp.player.skill\_id == re\_bp.player.skill\_id then

zjs = zjs + re\_bp.attr\_value[80]

end

-- 技能冷却期间减伤

if bp.attr\_value[28]~= 0 and bp.is\_skillcd(bp.player.skill\_id) then

zjs = zjs - bp.attr\_value[28]

end

attack = attack \* zjs / 100

local def = bp.attr.def()

def = def / (100 + def)

-- 无视防御

if BattleOperation.random(0, 100) < re\_bp.attr\_value[18] then

def = 0

end

attack = BattleOperation.toInt(attack \* (1 - def))

if cri then

attack = attack \* 2

end

if attack <= 1 then

attack = 1

end

if be.effect.re\_guid == self.guid and bp.player.guid ~= self.guid then --如果是被自己攻击

local pert = BattleOperation.toInt(bp.player.hp \* 100 / bp.attr.max\_hp())

if pert > 20 then

bp.mtk = true

end

end

if cri then

BattlePanel.add\_text(bp.accept.position, attack, 2)

else

BattlePanel.add\_text(bp.accept.position, attack, 1)

end

bp.set\_hp(bp.player.hp - attack)

if t\_skill\_effect.skill\_type == 1 and not re\_bp.is\_die then

-- 普攻吸血

local xx = re\_bp.attr\_value[74] + BattleOperation.toInt(attack \* re\_bp.attr\_value[39] / 100)

re\_bp.set\_hp(re\_bp.player.hp + xx, true)

elseif t\_skill\_effect.skill\_type ~= 1 and not re\_bp.is\_die then

-- 技能吸血

local xx = BattleOperation.toInt(attack \* re\_bp.attr\_value[40] / 100)

re\_bp.set\_hp(re\_bp.player.hp + xx, true)

end

-- 反弹技能

local is\_ft = false

if bp.attr\_value[29] ~= 0 and t\_skill\_effect.skill\_type ~= 1 and not re\_bp.is\_die and re\_bp.attr\_value[105] == 0 then

local fantan = attack \* bp.attr\_value[29] / 100

re\_bp.set\_hp(re\_bp.player.hp - fantan)

BattlePanel.add\_attack(re\_bp.accept.position, fantan, false)

is\_ft = true

end

-- 触发无敌

if is\_ft and re\_bp.player.hp < 0 and re\_bp.attr\_value[106] == 0 then

local t = re\_bp.get\_talent\_value(3)

if t ~= nil then

re\_bp.add\_buff(2006, 0, t)

re\_bp.add\_buff(3002)

re\_bp.set\_hp(1)

end

end

if bp.player.hp <= 0 and bp.attr\_value[106] == 0 then

local t = bp.get\_talent\_value(3)

if t ~= nil then

bp.add\_buff(2006, 0, t)

bp.add\_buff(3002)

bp.set\_hp(1)

end

end

if is\_ft and bp.player.hp <= 0 and re\_bp.player.hp < 0 then

--如果是自己 判断当前击杀的人是否是排行榜第一

BattlePlayers.BeforeKills(re\_bp, bp)

BattlePlayers.BeforeKills(bp, re\_bp)

bp.is\_die = true

bp.yy(3)

bp.player.die = bp.player.die + 1

bp.player.lsha = 0

bp.player.exp = BattleOperation.toInt(bp.player.exp \* bp.attr\_value[38] / 100)

re\_bp.is\_die = true

re\_bp.yy(3)

re\_bp.player.die = bp.player.die + 1

re\_bp.player.lsha = 0

re\_bp.player.exp = BattleOperation.toInt(re\_bp.player.exp \* re\_bp.attr\_value[38] / 100)

BattlePanel.show\_kill(bp, re\_bp, re\_bp.player.lsha)

--BattlePanel.show\_kill(re\_bp, bp, bp.player.lsha)

if BattlePlayers.me ~= nil then

if BattlePlayers.me.player.guid == bp.player.guid then

BattlePanel.show\_die(re\_bp)

elseif BattlePlayers.me.player.guid == re\_bp.player.guid then

BattlePanel.show\_die(bp)

end

end

elseif is\_ft and re\_bp.player.hp <= 0 then

BattlePlayers.jisha(bp, re\_bp)

elseif bp.player.hp <= 0 then

BattlePlayers.jisha(re\_bp, bp)

else

if BattleOperation.random(0, 100) >= bp.attr\_value[21] then

if t\_skill\_xiaoguo.link\_buff > 0 then

local yc = re\_bp.attr\_value[25]

yc = yc + re\_bp.get\_skill\_attr\_value(t\_skill\_effect.skill\_id, 5)

bp.add\_buff(t\_skill\_xiaoguo.link\_buff, yc)

end

-- 普攻加buff

if t\_skill\_effect.skill\_type == 1 then

local v = re\_bp.get\_bskill\_value(3) + re\_bp.get\_skill\_attr\_value(t\_skill\_effect.skill\_id, 5)

bp.add\_buff(v, re\_bp.attr\_value[25])

end

end

end

if t\_skill\_xiaoguo.jf\_type > 0 then

local dis = BattleOperation.toInt(t\_skill\_xiaoguo.jf\_dis \* (1 + re\_bp.get\_skill\_attr\_value(t\_skill\_effect.skill\_id, 6) / 100))

bp.set\_jf(be.effect.r, dis, t\_skill\_xiaoguo.jf\_speed)

end

end

function BattlePlayers.jisha(re\_bp, bp)

-- 击杀吸血

if not re\_bp.is\_die then

local v = re\_bp.attr\_value[19] + re\_bp.attr.max\_hp() \* re\_bp.attr\_value[20] / 100

if v > 0 then

re\_bp.set\_hp(re\_bp.player.hp, true)

end

end

-- 击杀击杀加属性

re\_bp.jisha\_value()

--如果是自己 判断当前击杀的人是否是排行榜第一

BattlePlayers.BeforeKills(re\_bp,bp)

local t\_exp = Config.get\_t\_battle\_exp(bp.player.level)

if t\_exp ~= nil then

local gexp = t\_exp.gexp

-- 击杀得分增加

gexp = BattleOperation.toInt(gexp \* (1 + re\_bp.attr\_value[71] / 100))

re\_bp.add\_exp(gexp)

for i = 1, t\_exp.item\_num1 do

local r = BattleOperation.random(0, 100)

if r < t\_exp.item\_rate1 then

BattlePlayers.add\_random\_item(0, bp.player.x, bp.player.y)

end

end

for i = 1, t\_exp.item\_num2 do

local r = BattleOperation.random(0, 100)

if r < t\_exp.item\_rate2 then

local iid = BattleOperation.random(200, 207 + 1)

BattlePlayers.add\_random\_item(iid, bp.player.x, bp.player.y)

end

end

end

re\_bp.player.sha = re\_bp.player.sha + 1

re\_bp.player.lsha = re\_bp.player.lsha + 1

bp.is\_die = true

bp.yy(3)

bp.player.die = bp.player.die + 1

bp.player.lsha = 0

bp.player.exp = BattleOperation.toInt(bp.player.exp \* bp.attr\_value[38] / 100)

BattlePanel.show\_kill(bp, re\_bp, re\_bp.player.lsha)

if BattlePlayers.me ~= nil then

if BattlePlayers.me.player.guid == bp.player.guid then

BattlePanel.show\_die(re\_bp)

end

end

end

function BattlePlayers.AddOperation(msg)

if not BattlePlayers.start then

return

end

local op = {msg = msg, nextp = nil}

if BattlePlayers.operations\_head == nil then

BattlePlayers.operations\_head = op

BattlePlayers.operations\_tail = op

else

BattlePlayers.operations\_tail.nextp = op

BattlePlayers.operations\_tail = op

end

end

function BattlePlayers.RemoveOperation()

if BattlePlayers.operations\_head == nil then

return

end

if BattlePlayers.operations\_head.nextp == nil then

BattlePlayers.operations\_head = nil

BattlePlayers.operations\_tail = nil

else

BattlePlayers.operations\_head = BattlePlayers.operations\_head.nextp

end

end

-----------------------------------------------------------------------

function BattlePlayers.do\_iteminit()

BattlePlayers.init\_item = true

BattlePlayers.do\_check\_item()

BattlePlayers.init\_ai()

end

function BattlePlayers.do\_operation()

while BattlePlayers.operations\_head ~= nil and BattlePlayers.operations\_head.msg.zhen == BattlePlayers.zhen do

local op = BattlePlayers.operations\_head.msg.opcode

local msg = BattlePlayers.operations\_head.msg

if op == opcodes.MSG\_BATTLE\_MOVE then

BattlePlayers.Move(msg)

elseif op == opcodes.MSG\_BATTLE\_STOP then

BattlePlayers.Stop(msg)

elseif op == opcodes.MSG\_BATTLE\_IN then

BattlePlayers.In(msg)

elseif op == opcodes.MSG\_BATTLE\_OUT then

BattlePlayers.Out(msg)

elseif op == opcodes.MSG\_BATTLE\_RELEASE then

BattlePlayers.Release(msg)

elseif op == opcodes.MSG\_BATTLE\_FUHUO then

BattlePlayers.Fuhuo(msg)

elseif op == opcodes.MSG\_BATTLE\_TALENT then

BattlePlayers.Talent(msg)

elseif op == opcodes.MSG\_BATTLE\_CHANGE\_SKILL then

BattlePlayers.ChangeSkill(msg)

end

BattlePlayers.RemoveOperation()

end

end

local max\_num = BattlePlayers.TNUM / 10

local speed = BattlePlayers.TICK / 1000

local tt = 1

if BattlePlayers.jiasu then

tt = 8

elseif dzhen > BattlePlayers.TNUM / 5 then

tt = 4

elseif dzhen > 1 then

tt = 2

end

max\_num = max\_num \* tt

speed = speed / tt

local num = 0

while BattlePlayers.dtime > 0 and num <= max\_num do

num = num + 1

dzhen = BattlePlayers.max\_zhen - BattlePlayers.zhen + 1

if dzhen == 0 or num > max\_num then

BattlePlayers.dtime = 0

if dzhen == 0 then

BattlePlayers.jiasu = false

end

break

end

if not BattlePlayers.Play() then

break

end

BattlePlayers.dtime = BattlePlayers.dtime - speed

end

BattlePlayers.Pos()

BattlePlayers.ItemPos()

end

function BattlePlayers.Pos()

local p = BattlePlayers.ptime

BattlePlayers.ptime = BattlePlayers.ptime + Time.deltaTime / Time.timeScale

local num = 0

for guid, bp in pairsByKeys(BattlePlayers.players) do

local num1 = #bp.last - 2

if num1 > num then

num = num1

end

end

if num < 0 then

num = 0

end

local tt = 1

if num >= 20 then

tt = 8

elseif num >= 10 then

tt = 4

elseif num >= 5 then

tt = 2

else

tt = 0.95 + 0.05 \* num

end

Time.timeScale = tt

local pj = 0

local per = BattlePlayers.ptime \* Time.timeScale / (BattlePlayers.TICK / 1000)

if num == 0 and per > 1 then

BattlePlayers.ptime = p

return

end

while per >= 1 do

pj = pj + 1

per = per - 1

BattlePlayers.ptime = BattlePlayers.ptime - BattlePlayers.TICK / 1000 / Time.timeScale

end

for tid, be in pairsByKeys(BattlePlayers.effects) do

for i = 1, pj do

if #be.last > 2 then

table.remove(be.last, 1)

else

break

end

end

if be.obj ~= nil then

local index = 1

if #be.last > 1 then

index = 2

end

local v = Vector3(be.last[index][1] / Battle.BL, 0, be.last[index][2] / Battle.BL)

local v1 = Vector3(be.last[1][1] / Battle.BL, 0, be.last[1][2] / Battle.BL)

if be.last[index][3] ~= nil then

v.y = BattleOperation.CubicBezierCurve(be.p0, be.p1, be.p2, be.p3, be.last[index][3]).y

end

if be.last[1][3] ~= nil then

v1.y = BattleOperation.CubicBezierCurve(be.p0, be.p1, be.p2, be.p3, be.last[1][3]).y

end

v = v1 + (v - v1) \* per

Battle.bobjpool:set\_localPosition(be.objid, v.x, v.y, v.z)

end

end

for guid, bp in pairsByKeys(BattlePlayers.players) do

for i = 1, pj do

if #bp.last > 2 then

table.remove(bp.last, 1)

else

break

end

end

if #bp.last > 0 then

local index = 1

if #bp.last > 1 then

index = 2

end

local v = Vector3(bp.last[index][1] / Battle.BL, 0, bp.last[index][2] / Battle.BL)

local v1 = Vector3(bp.last[1][1] / Battle.BL, 0, bp.last[1][2] / Battle.BL)

local r = 90 - bp.player.r

if bp.player.is\_jf then

if bp.last[index][4] then

v.y = BattleOperation.QuadBezierCurve(bp.jf\_p0, bp.jf\_p1, bp.jf\_p2, bp.last[index][4]).y

end

if bp.last[1][4] then

v1.y = BattleOperation.QuadBezierCurve(bp.jf\_p0, bp.jf\_p1, bp.jf\_p2, bp.last[1][4]).y

end

bp.jfr = bp.jfr + Time.deltaTime \* 360

r = bp.jfr

Battle.bobjpool:set\_localEulerAngles(bp.objid, 0, r, 0)

else

if bp.player.re\_state > 0 then

r = 90 - bp.player.re\_r

elseif bp.can\_do() then

if bp.player.is\_move then

BattlePlayers.action(bp, "run")

else

BattlePlayers.action(bp, "ready")

end

elseif bp.is\_die then

BattlePlayers.action(bp, "death")

else

BattlePlayers.action(bp, "ready")

end

if bp.lastr ~= r and bp.can\_do() then

bp.lastr = r

Battle.bobjpool:set\_localEulerAngles(bp.objid, 0, r, 0)

end

end

v = v1 + (v - v1) \* per

Battle.bobjpool:set\_localPosition(bp.posobjid, v.x, v.y, v.z)

end

end

for guid, bp in pairsByKeys(BattlePlayers.players) do

BattlePlayers.CalcCao(bp, me)

end

if BattlePlayers.me ~= nil then

local ox, oy, oz = Battle.bobjpool:get\_localPosition(BattlePlayers.me.posobjid, nil, nil, nil)

mapMgr:SetCurCam(ox, oy, oz)

local t\_battle\_exp = Config.get\_t\_battle\_exp(BattlePlayers.me.player.level)

if t\_battle\_exp ~= nil then

if t\_battle\_exp.dis <= BattlePlayers.dis then

BattlePlayers.dis = t\_battle\_exp.dis

else

local d1 = t\_battle\_exp.dis - BattlePlayers.dis

local d = d1 \* Time.deltaTime \* 2

if d1 < 0.01 or Time.deltaTime \* 2 >= 1 then

BattlePlayers.dis = t\_battle\_exp.dis

else

BattlePlayers.dis = BattlePlayers.dis + d

end

end

mapMgr:SetVCam(0, BattlePlayers.dis / 10, -BattlePlayers.dis / 10 + 0.8)

end

end

end

function BattlePlayers.ItemPos()

for i = #BattlePlayers.item\_speeds, 1, -1 do

local bi = BattlePlayers.item\_speeds[i]

local v = Vector3(bi.item.x / Battle.BL, 0, bi.item.y / Battle.BL)

local v1 = Vector3(bi.xx, 0, bi.yy)

local dv = v - v1

local dis = dv.sqrMagnitude

local l = bi.speed \* Time.deltaTime

if dis < l then

bi.speed = nil

Battle.bobjpool:set\_localPosition(bi.objid, v.x, 0.1, v.z)

table.remove(BattlePlayers.item\_speeds, i)

else

v1 = v1 + dv:Normalize() \* l

bi.xx = v1.x

bi.yy = v1.z

Battle.bobjpool:set\_localPosition(bi.objid, v1.x, 0.1, v1.z)

end

end

local idels = {}

for tid, bi in pairsByKeys(BattlePlayers.item\_follows) do

local bp = BattlePlayers.players[bi.follow]

if bp == nil then

table.insert(idels, tid)

else

bi.follow\_time = bi.follow\_time + Time.deltaTime

if bi.follow\_time >= 0.3 then

t\_battle\_item = Config.get\_t\_battle\_item(bi.item.id)

if t\_battle\_item ~= nil then

BattlePlayers.Attach(bp, "accept", t\_battle\_item.geffect)

end

table.insert(idels, tid)

else

local dv = Vector3(bp.player.x / Battle.BL - bi.xx , 0, bp.player.y / Battle.BL - bi.yy)

local dis = dv.sqrMagnitude

if dis > 0.01 then

dv = dv:Normalize() \* Time.deltaTime \* 2 \* bi.follow\_speed

local dis1 = dv.sqrMagnitude

if dis1 < dis then

bi.xx = dv.x + bi.xx

bi.yy = dv.y + bi.yy

Battle.bobjpool:set\_localPosition(bi.objid, bi.xx, 0.1, bi.yy)

end

end

end

end

end

for i = 1, #idels do

local bi = BattlePlayers.item\_follows[idels[i]]

resMgr:DeleteEffect(bi.obj)

BattlePlayers.item\_follows[idels[i]] = nil

end

end

function BattlePlayers.CalcCao(bp)

if BattlePlayers.me == nil then

if bp.cao == 0 then

BattlePlayers.set\_alpha(bp, 1)

else

BattlePlayers.set\_alpha(bp, 0)

end

return

end

if bp.player.guid ~= BattlePlayers.me.player.guid and not BattleOperation.can\_see(BattlePlayers.me, bp) then

BattlePlayers.set\_alpha(bp, 0)

return

end

if bp.cao == 0 then

BattlePlayers.set\_alpha(bp, 1)

else

BattlePlayers.set\_alpha(bp, 0.5)

end

end

function BattlePlayers.set\_alpha(bp, a)

if bp.alpha ~= a then

bp.alpha = a

bp.unit:set\_alpha(a)

if bp.xobj ~= nil then

bp.xunit:set\_alpha(a)

end

end

end

function BattlePlayers.action(bp, name, speed)

if speed == nil then

if name == "run" then

speed = bp.attr.speed() / 30000

if speed > 3 then

speed = 3

end

speed = speed / ((bp.get\_scale() - 1) \* 0.5 + 1)

local d = bp.action\_speed - speed

if bp.action == "run" and d < 0.001 and d > -0.001 then

return

end

else

speed = 1

end

end

if bp.action == name and name ~= "run" then

return

end

bp.action = name

bp.action\_speed = speed

bp.unit:action(name, speed)

end

function BattlePlayers.PlayersAI()

-- 在寻路过程中 如果 遇到更近的人 不能行动状态(can\_do)->变雪人状态->释放技能->移动状态

for k,v in pairsByKeys(BattlePlayers.players) do

-- is\_ai ==1 机器人

if v.player.is\_ai == 1 then

BattlePlayerAI.RobotAI(v)

end

end

end

function BattlePlayers.fillEffectInfo(bp, be) --记录 被这个 技能影响的人物

if be.effect.re\_guid ~= self.guid then

return

end

local re\_bp = be.re\_bp

if bp.is\_die then --如果被击杀 标示 0

be.effect\_hums[bp.player.guid] = 0

if bp.player.sex == 1 then

if re\_bp.achieveRecords.killFemale == nil then

re\_bp.achieveRecords.killFemale = 1

else

re\_bp.achieveRecords.killFemale = re\_bp.achieveRecords.killFemale + 1

end

elseif bp.player.sex == 0 then

if re\_bp.achieveRecords.killMale == nil then

re\_bp.achieveRecords.killMale = 1

else

re\_bp.achieveRecords.killMale = re\_bp.achieveRecords.killMale + 1

end

end

if be.effect.id == 10010101 then

re\_bp.achieveRecords.IsUseNormalToKill = true --使用普攻杀人

else

re\_bp.achieveRecords.IsUseSkillToKill = true

end

if not bp.mtk then --记录被偷偷击杀的 人数

if re\_bp.achieveRecords.quietKills == nil then

re\_bp.achieveRecords.quietKills = 1

else

re\_bp.achieveRecords.quietKills = re\_bp.achieveRecords.quietKills + 1

end

end

--记录这一局击杀的guid

if re\_bp.achieveRecords.kills == nil then

re\_bp.achieveRecords.kills = {}

end

if re\_bp.achieveRecords.kills[bp.player.guid] == nil then

re\_bp.achieveRecords.kills[bp.player.guid] = 1

else

re\_bp.achieveRecords.kills[bp.player.guid] = re\_bp.achieveRecords.kills[bp.player.guid] + 1

end

--记录自己 是在什么血量状态 杀人的

local pert = BattleOperation.toInt(re\_bp.player.hp \* 100 / bp.attr.max\_hp())

if pert <= 10 then

if re\_bp.achieveRecords.blood\_ten\_pert == nil then

re\_bp.achieveRecords.blood\_ten\_pert = 1

else

re\_bp.achieveRecords.blood\_ten\_pert = re\_bp.achieveRecords.blood\_ten\_pert + 1

end

end

if re\_bp.cao == 0 and bp.cao > 0 then

if re\_bp.achieveRecords.killCaos == nil then

re\_bp.achieveRecords.killCaos = 1

else

re\_bp.achieveRecords.killCaos = re\_bp.achieveRecords.killCaos + 1

end

end

if re\_bp.player.hp <= 0 or re\_bp.is\_die then

if re\_bp.achieveRecords.killDeath == nil then

re\_bp.achieveRecords.killDeath = 1

else

re\_bp.achieveRecords.killDeath = re\_bp.achieveRecords.killDeath + 1

end

end

--看看当前击杀人的时候 有没有buff

BattlePlayers.AccumKillInBuffer(re\_bp)

BattleAchieve.OnlyBattleKillRelation(re\_bp,bp)

else --如果受伤 标示 1

be.effect\_hums[bp.player.guid] = 1

end

local sk\_id = Config.get\_t\_skill\_effect(be.effect.id).skill\_id

if bp.player.is\_xueren and sk\_id == 100101 then

BattleAchieve.AttackXueRen()

end

end

function BattlePlayers.BeforeKills(me,armyPt)

if me.player.guid ~= self.guid then

return

end

if armyPt.player.hp > 0 then

return

end

local players = {}

for guid, bp in pairsByKeys(BattlePlayers.players) do

table.insert(players, bp)

end

local rank = 30

for i = 1, #players do

if players[i].player.guid == armyPt.player.guid then

rank = i

break

end

end

if rank == 1 then

if me.achieveRecords.killHeader == nil then

me.achieveRecords.killHeader = 1

else

me.achieveRecords.killHeader = me.achieveRecords.killHeader + 1

end

end

--击杀人判断等级差距

BattleAchieve.KillArmyInLevelLimit(me,armyPt)

end

function BattlePlayers.AccumKillInBuffer(bp)

if bp.player.guid ~= self.guid then

return

end

for i = #bp.player.buffs,1,-1 do

t\_battle\_buff = Config.get\_t\_battle\_buff(bp.player.buffs[i])

if t\_battle\_buff ~= nil then

if BattlePlayers.zhen < bp.player.buffs\_time[i] and t\_battle\_buff.time ~= -1 then

if bp.achieveRecords.bufferKill == nil then

bp.achieveRecords.bufferKill = {}

end

if bp.achieveRecords.bufferKill[t\_battle\_buff.id] == nil then

bp.achieveRecords.bufferKill[t\_battle\_buff.id] = 1

else

bp.achieveRecords.bufferKill[t\_battle\_buff.id] = bp.achieveRecords.bufferKill[t\_battle\_buff.id] + 1

end

end

end

end

end