**LISTING PROGRAM**

**Objects->InGameElements->MyPlayer**

**Create Event:**

execute code:

{

object = -1;

team = TEAM\_SPECTATOR;

class = CLASS\_SOLDIER;

name = "My Bot";

destroy = 0// Gets set to one when you need to destroy a bot

reward =0//moving reward

totalreward=0 //just for movement finding optimal moving

shootreward=0//shooting reward

oldshootreward=0

domove=false

sql="" //sql syntax

sqlvalue[0]=0 //sqlvalue length

sqlvalue2[0]=0 //sqlvalue length for shooting sql

dowrite=true

prefermove="" //read from file

nomoving=0 //count stuck

shootingfps=0//shooting frame count

triggershoot=false //trigger was fight? if triggershoot fps counter +1

oldHP=0

enemyoldHP[0]=0

noshooting=0

inlist=false;

//\*\*\*\*\*\* I'm making statistic arrays

stats[KILLS] = 0;

stats[DEATHS] = 0;

stats[CAPS] = 0;

stats[ASSISTS] = 0;

stats[DESTRUCTION] = 0;

stats[STABS] = 0;

stats[HEALING] = 0;

stats[DEFENSES] = 0;

stats[INVULNS] = 0;

stats[BONUS] = 0;

stats[DOMINATIONS] = 0;

stats[REVENGE] = 0;

stats[POINTS] = 0;

//statistic array for single life/arena

roundStats[KILLS] = 0;

roundStats[DEATHS] = 0;

roundStats[CAPS] = 0;

roundStats[ASSISTS] = 0;

roundStats[DESTRUCTION] = 0;

roundStats[STABS] = 0;

roundStats[HEALING] = 0;

roundStats[DEFENSES] = 0;

roundStats[INVULNS] = 0;

roundStats[BONUS] = 0;

roundStats[DOMINATIONS] = 0;

roundStats[REVENGE] = 0;

roundStats[POINTS] = 0;

timesChangedCapLimit = 0;

lastKnownx=0;

lastKnowny=0;

humiliated=0;

//Arena mode - used by server to check if the player can spawn

canSpawn = 1;

if instance\_exists(ArenaHUD) {

if ArenaHUD.roundStart == 0 canSpawn = 0;

}

//Sentries for Engies

//sentryBuilt = 0;

sentry=-1;

isHaxxyWinner = false;

}

**Destroy Event:**

execute code:

if(object != -1) with(object) instance\_destroy();

with(Rocket) if(ownerPlayer == other.id) instance\_destroy();

with(Flame) if(ownerPlayer == other.id) instance\_destroy();

with(Shot) if(ownerPlayer == other.id) instance\_destroy();

with(Needle) if(ownerPlayer == other.id) instance\_destroy();

with(Sentry) if(ownerPlayer == other.id) instance\_destroy();

with(DeathCam) if(killedby == other.id) instance\_destroy();

**Alarm Event for alarm 0:**

execute code:

canSpawn = 0;

**Alarm Event for alarm 5:**

execute code:

if(global.isHost && global.mapchanging == 0 && team != TEAM\_SPECTATOR && canSpawn == 1) {

var group, spawnpointID, numSpawnPoints;

group = selectSpawnGroup(team);

if (group==-1) {

show\_message("This map does not contain valid spawn points");

}

if(team == TEAM\_RED) {

numSpawnPoints = ds\_list\_size(global.spawnPointsRed[0,group]);

} else {

numSpawnPoints = ds\_list\_size(global.spawnPointsBlue[0,group]);

}

spawnpointID = floor(random(numSpawnPoints));

sendEventSpawn(id, spawnpointID, group);

doEventSpawn(id, spawnpointID, group);

}

**Other Event: User Defined 12:**

execute code:

var subobjects;

if(global.updateType == FULL\_UPDATE) {

write\_ubyte(global.serializeBuffer, stats[KILLS]);

write\_ubyte(global.serializeBuffer, stats[DEATHS]);

write\_ubyte(global.serializeBuffer, stats[CAPS]);

write\_ubyte(global.serializeBuffer, stats[ASSISTS]);

write\_ubyte(global.serializeBuffer, stats[DESTRUCTION]);

write\_ubyte(global.serializeBuffer, stats[STABS]);

write\_ushort(global.serializeBuffer, stats[HEALING]);

write\_ubyte(global.serializeBuffer, stats[DEFENSES]);

write\_ubyte(global.serializeBuffer, stats[INVULNS]);

write\_ubyte(global.serializeBuffer, stats[BONUS]);

write\_ubyte(global.serializeBuffer, stats[POINTS]);

}

subobjects=0;

if(object != -1) subobjects |= $01;

if(sentry != -1) subobjects |= $02;

if(isHaxxyWinner) subobjects |= $04;

write\_ubyte(global.serializeBuffer, subobjects);

if(object != -1) with(object) event\_user(12);

if(sentry != -1) with(sentry) event\_user(12);

**Other Event: User Defined 13:**

execute code:

if(global.updateType == FULL\_UPDATE) {

receiveCompleteMessage(global.serverSocket,12,global.deserializeBuffer);

stats[KILLS] = read\_ubyte(global.deserializeBuffer);

stats[DEATHS] = read\_ubyte(global.deserializeBuffer);

stats[CAPS] = read\_ubyte(global.deserializeBuffer);

stats[ASSISTS] = read\_ubyte(global.deserializeBuffer);

stats[DESTRUCTION] = read\_ubyte(global.deserializeBuffer);

stats[STABS] = read\_ubyte(global.deserializeBuffer);

stats[HEALING] = read\_ushort(global.deserializeBuffer);

stats[DEFENSES] = read\_ubyte(global.deserializeBuffer);

stats[INVULNS] = read\_ubyte(global.deserializeBuffer);

stats[BONUS] = read\_ubyte(global.deserializeBuffer);

stats[POINTS] = read\_ubyte(global.deserializeBuffer);

}

var charObj, subobjects;

receiveCompleteMessage(global.serverSocket,1,global.deserializeBuffer);

subobjects = read\_ubyte(global.deserializeBuffer);

// If the player has a character object on the server

if(subobjects & $01 != 0) {

if(object == -1) {

charObj = getCharacterObject(team, class);

if(charObj != -1) {

object = instance\_create(0,0,charObj);

object.player = id;

object.team = team;

with(object) {

event\_user(0);

}

} else {

show\_message("Invalid player object while deserializing");

}

}

with(object) event\_user(13);

} else {

if(object != -1) with(object) instance\_destroy();

object = -1;

}

// If the player has a sentry object on the server

if(subobjects & $02 != 0) {

if(sentry == -1) {

sentry = instance\_create(0,0,Sentry);

sentry.ownerPlayer=id;

sentry.team=team;

}

with(sentry) event\_user(13);

} else {

if sentry != -1 with sentry instance\_destroy();

sentry = -1;

}

isHaxxyWinner = (subobjects & $04 != 0)

**Script->MyBot->AvoidObstacle**

if(nomoving-1>30)

{

left=0;

jump=0;

right=1;

if(dir==1)

{

if(ds\_map\_exists(movedata2,pos))

{

list=ds\_map\_find\_value(movedata2,pos);

jlhmove=ds\_list\_size(list);

var i;

maxreward=0;

index=-1

for(i=0;i<jlhmove;i+=1)

{

if(real(ds\_list\_find\_value(ds\_list\_find\_value(list,i),1))>maxreward)

{

maxreward=real(ds\_list\_find\_value(ds\_list\_find\_value(list,i),1));

index=i;

}

}

ds\_list\_replace(ds\_list\_find\_value(list,index),1,0);

}

}

nomoving-=1;

dir=-1;

}

else

{

nomoving=0;

}

**Script->MyBot->Checkfailure**

if(shootreward<=0)

{

if(ds\_map\_exists(aimdata,lastKnownx))

{

tmppx=ds\_map\_find\_value(aimdata,lastKnownx);

if(ds\_map\_exists(tmppx,lastKnowny))

{

tmppy=ds\_map\_find\_value(tmppx,lastKnowny);

if(ds\_map\_exists(tmppy,nearestEnemy.x))

{

tmppex=ds\_map\_find\_value(tmppy,nearestEnemy.x);

if(ds\_map\_exists(tmppex,nearestEnemy.y))

{

tmppey=ds\_map\_find\_value(tmppex,nearestEnemy.y);

var tmpi;

tmpindex=-1;

for(tmpi=0;tmpi<ds\_list\_size(tmppey);tmpi+=1)

{

tmppr=ds\_list\_find\_value(tmppey,tmpi);

if(ds\_list\_find\_value(tmppr,0)>dr)

{

dr=ds\_list\_find\_value(tmppr,0);

tmpindex=tmpi;

}

}

tmppr=ds\_list\_find\_value(tmppey,tmpindex);

dr=ds\_list\_find\_value(tmppr,0);

ds\_list\_replace(tmppr,0,dr-5);

}

}

}

}

if(inlist)

{

show\_debug\_message(name);

show\_debug\_message('failed');

shootreward-=1;

inlist=false;

}

}

**Script->MyBot->CreateMyBot**

mybot = instance\_create(0, 0, MyPlayer)

ds\_list\_add(global.players, mybot)

with mybot

{

MyBotInit()

}

mybot.team = GetMyBotTeam(mybot)

//mybot.class = GetMyBotClass(mybot)

if global.mybotNamePrefix == ""

{

mybot.name = "My Bot "+string(global.mybotNameCounter)

}

else

{

mybot.name = global.mybotNamePrefix+string(global.mybotNameCounter)

}

if(global.mybotNameCounter==1)

{

mybot.class= CLASS\_HEAVY

}

else if(global.mybotNameCounter==2)

{

mybot.class= CLASS\_SOLDIER

}

else

{

mybot.class= CLASS\_SCOUT

}

global.mybotNameCounter += 1

mybot.alarm[5] = 1

ServerPlayerJoin(mybot, global.sendBuffer)

ServerPlayerChangeteam(ds\_list\_size(global.players)-1, mybot.team, global.sendBuffer)

ServerPlayerChangeclass(ds\_list\_size(global.players)-1, mybot.class, global.sendBuffer)

**Script->MyBot->** **GetMyBotInput**

// If you want to make your own bot off this, you need to define these 6 values:

left = 0

right = 0

jump = 0

LMB = 0

RMB = 0

aimDirection = 0

bubbleHP = 0

//BotMain();

//readfile();

//movementleftRL();//Kiri Learn

//movementleftjumpRL();//Kiri Jump Learn

//movementReactiveShoot();//Learn Shooting

//movement();

movementreadplay();

if pressed

{

jump = 0

pressed = 0

}

else if jump

{

pressed = 1

}

keybyte = 0 // keybyte converter.

if left == 1

{

keybyte |= $40

}

if right == 1

{

keybyte |= $20

}

if jump == 1

{

keybyte |= $80

}

if LMB == 1 and humiliated=0

{

keybyte |= $10

}

if RMB == 1 and humiliated=0

{

keybyte |= $08

}

if bubbleHP

{

keybyte |= $04

}

object.keyState = keybyte

object.aimDirection = aimDirection

object.netAimDirection = aimDirection\*65536/360

**Script->MyBot->** **GetMyBotTeam**

// argument0=the bot object

if global.mybotMode == 0

{

argument0.team = TEAM\_RED

}

else

{

argument0.team = TEAM\_BLUE

}

//return argument0.team

**Script->MyBot->Movement**

with GeneratorHUD

{

global.mytime=timer;

}

pos=real(round(lastKnownx));

if(ds\_map\_exists(movedata2,pos))

{

show\_debug\_message(name);

list=ds\_map\_find\_value(movedata2,pos);

jlhmove=ds\_list\_size(list);

var i;

maxreward=0;

for(i=0;i<jlhmove;i+=1)

{

if(real(ds\_list\_find\_value(ds\_list\_find\_value(list,i),1))>maxreward)

{

maxreward=real(ds\_list\_find\_value(ds\_list\_find\_value(list,i),1));

prefermove=ds\_list\_find\_value(ds\_list\_find\_value(list,i),0);

}

}

show\_debug\_message(prefermove);

}

if(global.mytime>0)

{ if(lastKnownx>2500)

{

left=1;

}

if(prefermove=="L")

{

left=1;

}

if(prefermove=="LJ")

{

left=1;

jump=1;

}

mov=prefermove;

reward=reward-lastKnownx;

if(sqlvalue[0]==0)

{

reward=0;

}

if(reward<-50.0)

{

reward=-10;

}

if(reward>50.0)

{

reward=10;

}

if(reward<=0&& reward>-1)

{

nomoving+=1

}

if (nomoving=15)

{

nomoving+=random(30);

}

if(nomoving<15 && reward >0)

{

dir=1;

nomoving=0

}

if(nomoving>15)

{

AvoidObstacle();

mov="R"

}

totalreward+=reward;

if(sqlvalue[0]!=0)

{

ds\_list\_add(sqlvalue[sqlvalue[0]],string(reward));//reward

}

global.playerx=0;

global.playery=0;

with Player

{

global.playerx=lastKnownx;

global.playery=lastKnowny;

}

sqlvalue[0]+=1;

sqlvalue[sqlvalue[0]]=ds\_list\_create();

ds\_list\_add(sqlvalue[sqlvalue[0]],string(name));//Nama

ds\_list\_add(sqlvalue[sqlvalue[0]],string(class));//Class

ds\_list\_add(sqlvalue[sqlvalue[0]],string(lastKnownx));//Pos\_x

ds\_list\_add(sqlvalue[sqlvalue[0]],string(lastKnowny));//Pos\_y

ds\_list\_add(sqlvalue[sqlvalue[0]],string(object.hp));//HP

ds\_list\_add(sqlvalue[sqlvalue[0]],string(mov));//movement

ds\_list\_add(sqlvalue[sqlvalue[0]],string(reward));//previous reward

reward=lastKnownx;

}

else if(global.mytime==0&&dowrite)

{

var i;

for(i=1;i<=sqlvalue[0];i+=1)

{

ds\_list\_add(sqlvalue[i],string(totalreward));//total reward

currentreward=ds\_list\_find\_value(sqlvalue[i],ds\_list\_size(sqlvalue[i])-2);

totalreward=real(totalreward)-real(currentreward);

}

sql='INSERT INTO `GerakRL`(`Nama`,`Class`,`Pos\_x`,`Pos\_y`,`HP`,`Movement`,`PreviousReward`,`Reward`,`TotalReward`,`GamePlay`) VALUES (';

var j;

for(i=1;i<sqlvalue[0];i+=1)

{

sqlcmd=sql;

for(j=0;j<ds\_list\_size(sqlvalue[i]);j+=1)

{

if(j>0)

{

sqlcmd+=",";

}

sqlcmd+='"'+ds\_list\_find\_value(sqlvalue[i],j)+'"';

}

sqlcmd+=',"'+string(global.mybotgameplay-1)+'"';

sqlcmd+=');';

file\_text\_write\_string(global.sqldump, sqlcmd);

file\_text\_writeln(global.sqldump);

}

dowrite=false;

}

**Script->MyBot->MovementleftRL**

with GeneratorHUD

{

global.mytime=timer;

}

if(global.mytime>0)

{

left=1

reward=reward-lastKnownx;

mov='L';

if(sqlvalue[0]==0)

{

reward=0;

}

if(reward<-50.0)

{

reward=-10;

}

if(reward>50.0)

{

reward=10;

}

totalreward+=reward;

if(sqlvalue[0]!=0)

{

ds\_list\_add(sqlvalue[sqlvalue[0]],string(reward));//reward

}

global.playerx=0;

global.playery=0;

with Player

{

global.playerx=lastKnownx;

global.playery=lastKnowny;

}

sqlvalue[0]+=1;

sqlvalue[sqlvalue[0]]=ds\_list\_create();

ds\_list\_add(sqlvalue[sqlvalue[0]],string(name));//Nama

ds\_list\_add(sqlvalue[sqlvalue[0]],string(class));//Class

ds\_list\_add(sqlvalue[sqlvalue[0]],string(lastKnownx));//Pos\_x

ds\_list\_add(sqlvalue[sqlvalue[0]],string(lastKnowny));//Pos\_y

ds\_list\_add(sqlvalue[sqlvalue[0]],string(object.hp));//HP

ds\_list\_add(sqlvalue[sqlvalue[0]],string(mov));//movement

ds\_list\_add(sqlvalue[sqlvalue[0]],string(reward));//previous reward

reward=lastKnownx;

}

else if(global.mytime==0&&dowrite)

{

var i;

for(i=1;i<=sqlvalue[0];i+=1)

{

ds\_list\_add(sqlvalue[i],string(totalreward));//total reward

currentreward=ds\_list\_find\_value(sqlvalue[i],ds\_list\_size(sqlvalue[i])-2);

totalreward=real(totalreward)-real(currentreward);

}

sql='INSERT INTO `GerakRL`(`Nama`,`Class`,`Pos\_x`,`Pos\_y`,`HP`,`Movement`,`PreviousReward`,`Reward`,`TotalReward`,`GamePlay`) VALUES (';

var j;

for(i=1;i<sqlvalue[0];i+=1)

{

sqlcmd=sql;

for(j=0;j<ds\_list\_size(sqlvalue[i]);j+=1)

{

if(j>0)

{

sqlcmd+=",";

}

sqlcmd+='"'+ds\_list\_find\_value(sqlvalue[i],j)+'"';

}

sqlcmd+=',"'+string(global.mybotgameplay-1)+'"';

sqlcmd+=');';

file\_text\_write\_string(global.sqldump, sqlcmd);

file\_text\_writeln(global.sqldump);

}

dowrite=false;

}

**Script->MyBot->MovementleftjumpRL**

with GeneratorHUD

{

global.mytime=timer;

}

if(global.mytime>0)

{

left=1

reward=reward-lastKnownx;

mov='L';

if(random(40)>20)

{

jump=1

mov+='J'

}

if(sqlvalue[0]==0)

{

reward=0;

}

if(reward<-50.0)

{

reward=-10;

}

if(reward>50.0)

{

reward=10;

}

totalreward+=reward;

if(sqlvalue[0]!=0)

{

ds\_list\_add(sqlvalue[sqlvalue[0]],string(reward));//reward

}

global.playerx=0;

global.playery=0;

with Player

{

global.playerx=lastKnownx;

global.playery=lastKnowny;

}

sqlvalue[0]+=1;

sqlvalue[sqlvalue[0]]=ds\_list\_create();

ds\_list\_add(sqlvalue[sqlvalue[0]],string(name));//Nama

ds\_list\_add(sqlvalue[sqlvalue[0]],string(class));//Class

ds\_list\_add(sqlvalue[sqlvalue[0]],string(lastKnownx));//Pos\_x

ds\_list\_add(sqlvalue[sqlvalue[0]],string(lastKnowny));//Pos\_y

ds\_list\_add(sqlvalue[sqlvalue[0]],string(object.hp));//HP

ds\_list\_add(sqlvalue[sqlvalue[0]],string(mov));//movement

ds\_list\_add(sqlvalue[sqlvalue[0]],string(reward));//previous reward

reward=lastKnownx;

}

else if(global.mytime==0&&dowrite)

{

var i;

for(i=1;i<=sqlvalue[0];i+=1)

{

ds\_list\_add(sqlvalue[i],string(totalreward));//total reward

currentreward=ds\_list\_find\_value(sqlvalue[i],ds\_list\_size(sqlvalue[i])-2);

totalreward=real(totalreward)-real(currentreward);

}

sql='INSERT INTO `GerakRL`(`Nama`,`Class`,`Pos\_x`,`Pos\_y`,`HP`,`Movement`,`PreviousReward`,`Reward`,`TotalReward`,`GamePlay`) VALUES (';

var j;

for(i=1;i<sqlvalue[0];i+=1)

{

sqlcmd=sql;

for(j=0;j<ds\_list\_size(sqlvalue[i]);j+=1)

{

if(j>0)

{

sqlcmd+=",";

}

sqlcmd+='"'+ds\_list\_find\_value(sqlvalue[i],j)+'"';

}

sqlcmd+=',"'+string(global.mybotgameplay-1)+'"';

sqlcmd+=');';

file\_text\_write\_string(global.sqldump, sqlcmd);

file\_text\_writeln(global.sqldump);

}

dowrite=false;

}

**Script->MyBot->MovementReactiveShoot**

BotGetNearestEnemy();

with GeneratorHUD

{

global.mytime=timer;

}

target = Generator

// Aiming and Fighting

if nearestEnemy != -1

{

if(!triggershoot)

{

var p;

p=1;

rwd=0;

Generator.oldhp=Generator.hp;

Generator.oldshield=Generator.shieldHp;

with(BotPlayer)

{

if global.botNamePrefix == ""

{

if(name == "Tempest Bot "+string(p))

{

if(object!=-1)

{

other.rwd=oldHP-object.hp;

oldHP=object.hp;

}

else

{

other.rwd=oldHP;

oldHP=0;

//other.eoldHP[other.p]=0;

}

}

}

else

{

if(name == global.botNamePrefix+string(p))

{

if(object!=-1)

{

other.rwd=oldHP-object.hp;

oldHP=object.hp;

//show\_debug\_message(object.hp);

}

}

}

p+=1;

}

}

triggershoot=true

BotAim()

if point\_distance(object.x, object.y, nearestEnemy.x, nearestEnemy.y) < 350

{

if !wasFighting

{

wasFighting = 1

}

if task != 'objective' and target != nearestEnemy and !target\_in\_sight

{

ds\_list\_clear(directionList)

target = nearestEnemy

}

mov="S"

//some random left

if random(50)<5

{

left=1

mov="L"

}

// Some random jumping...

if random(40)<4

{

jump = 1

if(mov=="S")

{

mov="J"

}

else

{

mov+="J"

}

}

switch (class)

{

case CLASS\_SCOUT:

ScoutFight()

break

case CLASS\_SOLDIER:

SoldierFight()

break

case CLASS\_HEAVY:

HeavyFight()

break

case CLASS\_ENGINEER:

EngineerFight()

break

case CLASS\_PYRO:

PyroFight()

break

case CLASS\_SNIPER:

SniperFight()

break

}

}

else if wasFighting

{

if task != 'objective'

{

ds\_list\_clear(directionList)

target = -1

}

wasFighting = 0

}

}

else if wasFighting

{

if task != 'objective'

{

ds\_list\_clear(directionList)

target = -1

}

wasFighting = 0

}

if(triggershoot)

{

shootingfps+=1//shooting frame count

if(rwd>0)

{

shootreward+=rwd;

}

var p;

p=1;

rwd=0;

with(BotPlayer)

{

if(object!=-1)

{

if(object.lastDamageDealer!=-1)

{

//show\_debug\_message(other.name);

if(object.lastDamageDealer.name==other.name)

{

if global.botNamePrefix == ""

{

if(name == "Tempest Bot "+string(p))

{

if(object!=-1)

{

other.rwd=oldHP-object.hp;

oldHP=object.hp;

//rwd=eoldHP[p]-object.hp;

}

else

{

other.rwd=oldHP;

oldHP=0;

//eoldHP[p]=0;

//rwd=eoldHP[p];

}

}

}

else

{

if(name == global.botNamePrefix+string(p))

{

if(object!=-1)

{

other.rwd=oldHP-object.hp;

oldHP=object.hp;

//eoldHP[p]=object.hp;

//rwd=eoldHP[p]-object.hp;

}

else

{

other.rwd=oldHP;

oldHP=0;

//eoldHP[p]=0;

//rwd=eoldHP[p];

}

}

}

}

}

}

p+=1;

}

if(rwd>0)

{

shootreward+=rwd;

}

if(Generator.lastDamageDealer!=-1)

{

if(Generator.lastDamageDealer.name==name)

{

if((Generator.oldhp-Generator.hp)>0)

{

shootreward+=2\*(Generator.oldhp-Generator.hp)

}

if((Generator.oldshield-Generator.shieldHp)>0)

{

shootreward+=(Generator.oldshield-Generator.shieldHp);

}

Generator.oldhp=Generator.hp;

Generator.oldshield=Generator.shieldHp;

}

}

if(object.hp-oldHP<0)

{

shootreward+=object.hp-oldHP

}

sqlvalue2[0]+=1;

sqlvalue2[sqlvalue2[0]]=ds\_list\_create();

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(name));//Nama

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(class));//Class

if(object!=-1)

{

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(object.hp));//HP

}

else

{

ds\_list\_add(sqlvalue2[sqlvalue2[0]],"0");//HP

}

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(lastKnownx));//Pos\_x

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(lastKnowny));//Pos\_y

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(mov));//movement

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(aimDirection));//aimradius

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(LMB));//shoot left?

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(RMB));//shoot right?

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(nearestEnemy.x));//enemy pos\_x

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(nearestEnemy.y));//enemy pos\_y

if(shootingfps>15)

{

sql='INSERT INTO `AimingRL`(`Nama`,`Class`,`HP`,`Pos\_x`,`Pos\_y`,`Movement`,`AimRadius`,`LMB`,`RMB`,`enemy\_x`,`enemy\_y`,`Reward`,`GamePlay`) VALUES (';

var i;

var j;

for(i=1;i<sqlvalue2[0];i+=1)

{

sqlcmd=sql;

for(j=0;j<ds\_list\_size(sqlvalue2[i]);j+=1)

{

if(j>0)

{

sqlcmd+=",";

}

sqlcmd+='"'+ds\_list\_find\_value(sqlvalue2[i],j)+'"';

}

sqlcmd+=',"'+string(shootreward)+'"';

sqlcmd+=',"'+string(global.mybotgameplay-1)+'"';

sqlcmd+=');';

file\_text\_write\_string(global.sqldump, sqlcmd);

file\_text\_writeln(global.sqldump);

}

sqlvalue2[0]=0;

shootreward=0;

shootingfps=0;

triggershoot=false;

}

}

oldHP=object.hp

if(wasFighting!=1)

{

pos=real(round(lastKnownx));

if(ds\_map\_exists(movedata2,pos))

{

list=ds\_map\_find\_value(movedata2,pos);

jlhmove=ds\_list\_size(list);

var i;

maxreward=0;

for(i=0;i<jlhmove;i+=1)

{

if(real(ds\_list\_find\_value(ds\_list\_find\_value(list,i),1))>maxreward)

{

maxreward=real(ds\_list\_find\_value(ds\_list\_find\_value(list,i),1));

prefermove=ds\_list\_find\_value(ds\_list\_find\_value(list,i),0);

}

}

}

if(global.mytime>0)

{

if(prefermove=="L")

{

left=1;

}

if(prefermove=="LJ")

{

left=1;

jump=1;

}

mov=prefermove;

reward=reward-lastKnownx;

if(sqlvalue[0]==0)

{

reward=0;

}

if(dir==-1)

{

reward\*=-1;

reward/=2;

}

if(reward<-50.0)

{

reward=-10;

}

if(reward>50.0)

{

reward=10;

}

if(reward<=0&& reward>-1)

{

nomoving+=1

}

if (nomoving=15)

{

nomoving+=random(30);

}

if(nomoving<15 && reward >0)

{

dir=1;

nomoving=0

}

if(nomoving>15)

{

AvoidObstacle();

mov="R"

}

reward=lastKnownx;

}

}

if reloadCounter > 0

{

reloadCounter -= 1

}

**Script->MyBot->Movementreadplay**

BotGetNearestEnemy();

with GeneratorHUD

{

global.mytime=timer;

}

target = Generator

// Aiming and Fighting

if nearestEnemy != -1

{

if(!triggershoot)

{

var p;

p=1;

rwd=0;

Generator.oldhp=Generator.hp;

Generator.oldshield=Generator.shieldHp;

with(BotPlayer)

{

if global.botNamePrefix == ""

{

if(name == "Tempest Bot "+string(p))

{

if(object!=-1)

{

other.rwd=oldHP-object.hp;

oldHP=object.hp;

}

else

{

other.rwd=oldHP;

oldHP=0;

}

}

}

else

{

if(name == global.botNamePrefix+string(p))

{

if(object!=-1)

{

other.rwd=oldHP-object.hp;

oldHP=object.hp;

}

}

}

p+=1;

}

}

triggershoot=true

BotAim()

if point\_distance(object.x, object.y, nearestEnemy.x, nearestEnemy.y) < 350

{

if !wasFighting

{

wasFighting = 1

}

if task != 'objective' and target != nearestEnemy and !target\_in\_sight

{

ds\_list\_clear(directionList)

target = nearestEnemy

}

mov="S"

//some random left

if random(50)<5

{

left=1

mov="L"

}

// Some random jumping...

if random(40)<4

{

jump = 1

if(mov=="S")

{

mov="J"

}

else

{

mov+="J"

}

}

switch (class)

{

case CLASS\_SCOUT:

ScoutFight()

break

case CLASS\_SOLDIER:

SoldierFight()

break

case CLASS\_HEAVY:

HeavyFight()

break

case CLASS\_ENGINEER:

EngineerFight()

break

case CLASS\_PYRO:

PyroFight()

break

case CLASS\_SNIPER:

SniperFight()

break

}

}

else if wasFighting

{

if task != 'objective'

{

ds\_list\_clear(directionList)

target = -1

}

wasFighting = 0

}

}

else if wasFighting

{

if task != 'objective'

{

ds\_list\_clear(directionList)

target = -1

}

wasFighting = 0

}

if(triggershoot)

{

if(LMB==1)

{

noshooting=0;

dx=real(round(lastKnownx));

dy=real(round(lastKnowny));

dex=real(round(nearestEnemy.x));

dey=real(round(nearestEnemy.y));

dr=-1;

dm="";

da=-1;

if(ds\_map\_exists(aimdata,dx))

{

tmppx=ds\_map\_find\_value(aimdata,dx);

if(ds\_map\_exists(tmppx,dy))

{

tmppy=ds\_map\_find\_value(tmppx,dy);

if(ds\_map\_exists(tmppy,dex))

{

tmppex=ds\_map\_find\_value(tmppy,dex);

if(ds\_map\_exists(tmppex,dey))

{

tmppey=ds\_map\_find\_value(tmppex,dey);

var tmpi;

tmpindex=-1;

for(tmpi=0;tmpi<ds\_list\_size(tmppey);tmpi+=1)

{

tmppr=ds\_list\_find\_value(tmppey,tmpi);

if(ds\_list\_find\_value(tmppr,0)>dr)

{

dr=ds\_list\_find\_value(tmppr,0);

tmpindex=tmpi;

}

}

if(shootingfps>15&& shootreward<0)

{

dr2=ds\_list\_find\_value(tmppr,0);

ds\_list\_replace(tmppr,0,dr2-5);

}

tmppr=ds\_list\_find\_value(tmppey,tmpindex);

dm=ds\_list\_find\_value(tmppr,1);

da=ds\_list\_find\_value(tmppr,2);

}

else

{

show\_debug\_message(name);

show\_debug\_message("Not in list");

}

}

else

{

show\_debug\_message(name);

show\_debug\_message("Not in list");

}

}

else

{

show\_debug\_message(name);

show\_debug\_message("Not in list");

}

}

else

{

show\_debug\_message(name);

show\_debug\_message("Not in list");

}

if(dr!=-1)

{

if(dm=="L")

{

left=1;

jump=0;

right=0;

}

if(dm=="LJ")

{

left=1;

jump=1;

right=0;

}

if(dm=="J")

{

left=0;

jump=1;

right=0;

}

if(dm=="S")

{

left=0;

jump=0;

right=0;

}

mov=dm;

aimDirection=da;

inlist=true;

show\_debug\_message("Read");

show\_debug\_message(name);

show\_debug\_message(mov);

show\_debug\_message(aimDirection);

}

}

shootingfps+=1//shooting frame count

if(rwd>0)

{

shootreward+=rwd;

}

var p;

p=1;

rwd=0;

with(BotPlayer)

{

if(object!=-1)

{

if(object.lastDamageDealer!=-1)

{

if(object.lastDamageDealer.name==other.name)

{

if global.botNamePrefix == ""

{

if(name == "Tempest Bot "+string(p))

{

if(object!=-1)

{

other.rwd=oldHP-object.hp;

oldHP=object.hp;

}

else

{

other.rwd=oldHP;

oldHP=0;

}

}

}

else

{

if(name == global.botNamePrefix+string(p))

{

if(object!=-1)

{

other.rwd=oldHP-object.hp;

oldHP=object.hp;

}

else

{

other.rwd=oldHP;

oldHP=0;

}

}

}

}

}

}

p+=1;

}

if(rwd>0)

{

shootreward+=rwd;

}

if(Generator.lastDamageDealer!=-1)

{

if(Generator.lastDamageDealer.name==name)

{

if((Generator.oldhp-Generator.hp)>0)

{

shootreward+=2\*(Generator.oldhp-Generator.hp)

}

if((Generator.oldshield-Generator.shieldHp)>0)

{

shootreward+=(Generator.oldshield-Generator.shieldHp);

}

Generator.oldhp=Generator.hp;

Generator.oldshield=Generator.shieldHp;

}

}

if(object.hp-oldHP<0)

{

shootreward+=object.hp-oldHP

}

sqlvalue2[0]+=1;

sqlvalue2[sqlvalue2[0]]=ds\_list\_create();

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(name));//Nama

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(class));//Class

if(object!=-1)

{

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(object.hp));//HP

}

else

{

ds\_list\_add(sqlvalue2[sqlvalue2[0]],"0");//HP

}

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(lastKnownx));//Pos\_x

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(lastKnowny));//Pos\_y

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(mov));//movement

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(aimDirection));//aimradius

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(LMB));//shoot left?

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(RMB));//shoot right?

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(nearestEnemy.x));//enemy pos\_x

ds\_list\_add(sqlvalue2[sqlvalue2[0]],string(nearestEnemy.y));//enemy pos\_y

if(shootingfps>15)

{

checkfailure();

sql='INSERT INTO `AimingRL`(`Nama`,`Class`,`HP`,`Pos\_x`,`Pos\_y`,`Movement`,`AimRadius`,`LMB`,`RMB`,`enemy\_x`,`enemy\_y`,`Reward`,`GamePlay`) VALUES (';

var i;

var j;

for(i=1;i<sqlvalue2[0];i+=1)

{

sqlcmd=sql;

for(j=0;j<ds\_list\_size(sqlvalue2[i]);j+=1)

{

if(j>0)

{

sqlcmd+=",";

}

sqlcmd+='"'+ds\_list\_find\_value(sqlvalue2[i],j)+'"';

}

sqlcmd+=',"'+string(shootreward)+'"';

sqlcmd+=',"'+string(global.mybotgameplay-1)+'"';

sqlcmd+=');';

file\_text\_write\_string(global.sqldump, sqlcmd);

file\_text\_writeln(global.sqldump);

}

sqlvalue2[0]=0;

shootreward=0;

shootingfps=0;

triggershoot=false;

}

}

oldHP=object.hp

if(LMB==0)

{

noshooting+=1;

}

if(wasFighting!=1)

{

pos=real(round(lastKnownx));

if(ds\_map\_exists(movedata2,pos))

{

list=ds\_map\_find\_value(movedata2,pos);

jlhmove=ds\_list\_size(list);

var i;

maxreward=0;

for(i=0;i<jlhmove;i+=1)

{

if(real(ds\_list\_find\_value(ds\_list\_find\_value(list,i),1))>maxreward)

{

maxreward=real(ds\_list\_find\_value(ds\_list\_find\_value(list,i),1));

prefermove=ds\_list\_find\_value(ds\_list\_find\_value(list,i),0);

}

}

}

if(global.mytime>0)

{ if(noshooting>60&& lastKnownx>2500)

{

left=1;

}

if(prefermove=="L" && noshooting>60)

{

left=1;

}

if(prefermove=="LJ" && noshooting>60)

{

left=1;

jump=1;

}

mov=prefermove;

reward=reward-lastKnownx;

if(sqlvalue[0]==0)

{

reward=0;

}

if(reward<-50.0)

{

reward=-10;

}

if(reward>50.0)

{

reward=10;

}

if(reward<=0&& reward>-1)

{

nomoving+=1

}

if (nomoving=15)

{

nomoving+=random(30);

}

if(nomoving<15 && noshooting>60)

{

dir=1;

nomoving=0

}

if(nomoving>15 &&triggershoot==false)

{

AvoidObstacle();

mov="R"

}

totalreward+=reward;

if(sqlvalue[0]!=0)

{

ds\_list\_add(sqlvalue[sqlvalue[0]],string(reward));//reward

}

global.playerx=0;

global.playery=0;

with Player

{

global.playerx=lastKnownx;

global.playery=lastKnowny;

}

if(triggershoot)

{

nomoving=0;

}

sqlvalue[0]+=1;

sqlvalue[sqlvalue[0]]=ds\_list\_create();

ds\_list\_add(sqlvalue[sqlvalue[0]],string(name));//Nama

ds\_list\_add(sqlvalue[sqlvalue[0]],string(class));//Class

ds\_list\_add(sqlvalue[sqlvalue[0]],string(lastKnownx));//Pos\_x

ds\_list\_add(sqlvalue[sqlvalue[0]],string(lastKnowny));//Pos\_y

ds\_list\_add(sqlvalue[sqlvalue[0]],string(mov));//movement

ds\_list\_add(sqlvalue[sqlvalue[0]],string(reward));//previous reward

reward=lastKnownx;

}

else if(global.mytime==0&&dowrite==false)

{

var i;

for(i=1;i<=sqlvalue[0];i+=1)

{

ds\_list\_add(sqlvalue[i],string(totalreward));//total reward

currentreward=ds\_list\_find\_value(sqlvalue[i],ds\_list\_size(sqlvalue[i])-2);

totalreward=real(totalreward)-real(currentreward);

}

sql='INSERT INTO `GerakRL`(`Nama`,`Class`,`Pos\_x`,`Pos\_y`,`Movement`,`PreviousReward`,`Reward`,`TotalReward`) VALUES (';

var j;

for(i=1;i<sqlvalue[0];i+=1)

{

sqlcmd=sql;

for(j=0;j<ds\_list\_size(sqlvalue[i]);j+=1)

{

if(j>0)

{

sqlcmd+=",";

}

sqlcmd+='"'+ds\_list\_find\_value(sqlvalue[i],j)+'"';

}

sqlcmd+=');';

file\_text\_write\_string(global.sqldump, sqlcmd);

file\_text\_writeln(global.sqldump);

}

dowrite=false;

}

}

if reloadCounter > 0

{

reloadCounter -= 1

}

**Script->MyBot->MyBotInit**

pressed = 0

target = -1

aimModifier = 0

isHealing = 0

patient = -1

if variable\_local\_exists("directionList")

{

ds\_list\_clear(directionList)

}

else

{

directionList = ds\_list\_create()

}

wasFighting = 0

dir = 1

stuckTimer = 0

oldX = 0

oldY = 0

reloadCounter = 0

// Task selecting

task = choose("roam", "roam", "roam", "objective", "hunt")

if class == CLASS\_MEDIC

{

task = 'roam'

}

**Script->MyBot->readlearn**

with MyPlayer

{

movedata2=ds\_map\_create();

num = 0;

file = file\_text\_open\_read(working\_directory + "\move.csv");

while (!file\_text\_eof(file))

{

text=file\_text\_read\_string(file);

splitBy='|';

slot = 0;

bn='';

var split;

str2 = "";

for (i = 1; i < (string\_length(text)+1); i+=1) {

currStr = string\_copy(text, i, 1);

if (currStr == splitBy) {

if(slot==0)

{

if(str2!=name)

{

break;

}

}

if(slot==1)

{

str2=real(str2);

if(!ds\_map\_exists(movedata2,str2))

{

ds\_map\_add(movedata2,str2,ds\_list\_create());

}

bn=str2;

}

else if(slot==2)

{

lst=ds\_map\_find\_value(movedata2,bn);

ds\_list\_add(lst,ds\_list\_create());

listrw=ds\_list\_find\_value(lst,ds\_list\_size(lst)-1);

ds\_list\_add(listrw,str2);

}

else if(slot==3)

{

lst=ds\_map\_find\_value(movedata2,bn);

listrw=ds\_list\_find\_value(lst,ds\_list\_size(lst)-1);

ds\_list\_add(listrw,str2);

}

slot+=1;

str2 = "";

} else {

str2 = str2 + currStr;

}

}

if(slot==3)

{

lst=ds\_map\_find\_value(movedata2,bn);

listrw=ds\_list\_find\_value(lst,ds\_list\_size(lst)-1);

ds\_list\_add(listrw,str2);

}

file\_text\_readln(file);

num+=1;

}

file\_text\_close(file);

}

**Script->MyBot->readlearnshoot**

with MyPlayer

{

aimdata=ds\_map\_create();

num = 0;

file = file\_text\_open\_read(working\_directory + "\aim"+string(name)+".csv");

show\_debug\_message("aim"+string(name)+".csv");

while (!file\_text\_eof(file))

{

text=file\_text\_read\_string(file);

splitBy='|';

slot = 0;

bn='';

var split;

str2 = "";

var px,py,pex,pey,pr,pm,pa;

for (i = 1; i < (string\_length(text)+1); i+=1) {

currStr = string\_copy(text, i, 1);

if (currStr == splitBy) {

if(slot==-1)

{

}

if(slot==0)

{

px=real(str2);

}

else if(slot==1)

{

py=real(str2);

}

else if(slot==2)

{

pex=real(str2);

}

else if(slot==3)

{

pey=real(str2);

}

else if(slot==4)

{

pr=real(str2);

}

else if(slot==5)

{

pm=str2;

}

else if(slot==6)

{

pa=real(str2);

}

slot+=1;

str2 = "";

}

else {

str2 = str2 + currStr;

}

}

file\_text\_readln(file);

if(!ds\_map\_exists(aimdata,px))

{

ds\_map\_add(aimdata,px,ds\_map\_create());

}

tmppx=ds\_map\_find\_value(aimdata,px);

if(!ds\_map\_exists(tmppx,py))

{

ds\_map\_add(tmppx,py,ds\_map\_create());

}

tmppy=ds\_map\_find\_value(tmppx,py);

if(!ds\_map\_exists(tmppy,pex))

{

ds\_map\_add(tmppy,pex,ds\_map\_create());

}

tmppex=ds\_map\_find\_value(tmppy,pex);

if(!ds\_map\_exists(tmppex,pey))

{

ds\_map\_add(tmppex,pey,ds\_list\_create());

}

tmppey=ds\_map\_find\_value(tmppex,pey);

ds\_list\_add(tmppey,ds\_list\_create());

tmppr=ds\_list\_find\_value(tmppey,ds\_list\_size(tmppey)-1);

ds\_list\_add(tmppr,pr);

ds\_list\_add(tmppr,pm);

ds\_list\_add(tmppr,pa);

num+=1;

}

file\_text\_close(file);

}

**Table Aiming**

CREATE TABLE AimingRL (

Nama TEXT,

Class INT,

HP REAL,

pos\_x REAL,

pos\_y REAL,

Movement TEXT,

AimRadius REAL,

LMB INT,

RMB INT,

enemy\_x REAL,

enemy\_y REAL,

Reward REAL,

GamePlay INTEGER

);

**Table Gerak**

CREATE TABLE GerakRL (

Nama TEXT,

Class INT,

Pos\_x REAL,

Pos\_y REAL,

HP REAL,

Movement TEXT,

PreviousReward REAL,

Reward REAL,

TotalReward REAL,

GamePlay INTEGER

);

**Table Normalize Gerak**

CREATE VIEW RLGerakbot AS

SELECT RL2.nama,

printf('%d', RL2.pos\_x) AS pos\_x,

RL2.movement,

RL2.totalreward

FROM (

SELECT round(pos\_x) AS pos\_x,

movement,

max(totalreward) AS totalreward

FROM GerakRL

WHERE reward > 0

GROUP BY nama,

round(pos\_x),

movement

)

RL1

JOIN

(

SELECT round(pos\_x) AS pos\_x,

movement,

nama,

totalreward

FROM GerakRL

WHERE reward > 0

GROUP BY round(pos\_x),

movement,

nama,

totalreward

)

RL2 ON RL1.pos\_x = RL2.pos\_x

WHERE RL1.totalreward = RL2.totalreward AND

RL1.movement = RL2.movement;

**Table Normalisasi Aiming**

CREATE VIEW aimMyBot1 AS

SELECT \*

FROM (

SELECT printf('%d', pos\_x) AS pos\_x,

printf('%d', pos\_y) AS pos\_y,

printf('%d', enemy\_x) AS enemy\_x,

printf('%d', enemy\_y) AS enemy\_y,

avg(Reward) AS reward,

movement,

aimRadius,

count(movement) AS counter

FROM AimingRL

WHERE LMB = 1 AND

nama = nama

GROUP BY nama,

round(pos\_x),

round(pos\_y),

movement,

aimradius,

round(enemy\_x),

round(enemy\_y)

)

WHERE reward > 0

ORDER BY counter DESC;