

variables

{

    global:

        0: BotPositions

        1: noofbots

        2: Timer

        3: HighScore

        4: Starttimer

        5: noofbotres

        6: Bot

    player:

        0: position

}

rule("WHAT IS BOT----- change character in variable to change hero")

{

    event

    {

        Ongoing - Global;

    }

    actions

    {

        Set Global Variable(Bot, Hero(Soldier: 76));

    }

}

rule("----Setting up Team----")

```
{  
    event  
    {  
        Ongoing - Global;  
    }  
}  
  
rule("Hold Team")  
{  
    event  
    {  
        Ongoing - Each Player;  
        Team 2;  
        All;  
    }  
  
    conditions  
    {  
        Is Alive(Event Player) == True;  
    }  
  
    actions  
    {  
        Set Status(Event Player, Null, Rooted, 9999);  
        Set Respawn Max Time(Event Player, 0.500);  
        Set Damage Received(Event Player, 600);  
        Destroy All Dummy Bots;  
        Set Ultimate Charge(All Living Players(Team Of(Event Player)), 0);  
    }  
}
```

```
}
```

```
rule("Start")
```

```
{
```

```
    event
```

```
    {
```

```
        Ongoing - Each Player;
```

```
        Team 2;
```

```
        All;
```

```
    }
```

```
    conditions
```

```
    {
```

```
        Number of Players(Opposite Team Of(Team Of(Event Player))) == 0;
```

```
        Global Variable(BotPositions) != 0;
```

```
    }
```

```
    actions
```

```
    {
```

```
        Big Message(Event Player, Custom String("Click Interact to start", Null, Null, Null));
```

```
        Wait(5, Ignore Condition);
```

```
        Loop If Condition Is True;
```

```
    }
```

```
}
```

```
rule("Wrong team notification")
```

```
{
```

```
    event
```

```
    {
```

```

        Ongoing - Each Player;
        Team 1;
        All;
    }

    conditions
    {
        Is Dummy Bot(Event Player) == False;
        Host Player == False;
    }

    actions
    {
        Big Message(All Players(Team Of(Event Player)), Custom String("WRONG TEAM!!!", Null,
Null, Null));
        Wait(5, Abort When False);
        Loop If Condition Is True;
    }
}

rule("HUD")
{
    event
    {
        Ongoing - Global;
    }

    actions
    {

```

```

        Create HUD Text(All Players(All Teams), String("{0} {1} {2}", Hero Icon String(Global
Variable(Bot)), Custom String("X", Null,
            Null, Null), Global Variable(noofbots)), Null, Null, Left, 0, White, White, White,
Visible To and String, Visible Always);

        disabled Create HUD Text(All Players(All Teams), String("{0}: {1}", Custom
String("Number of Targets", Null, Null, Null), Global Variable(
            noofbots), Null), Null, Null, Left, 0, White, White, White, Visible To and String,
Visible Always);

        Create HUD Text(All Players(All Teams), String("{0}: {1}", Custom String("High Score",
Null, Null, Null), Global Variable(
            HighScore), Null), Null, Null, Top, 0, White, White, White, Visible To and String,
Visible Always);
    }
}

```

```

rule("Timer")
{
    event
    {
        Ongoing - Global;
    }

    conditions
    {
        Global Variable(Starttimer) == True;
    }

    actions
    {
        Modify Global Variable(Timer, Add, 0.016);
        Wait(0.016, Abort When False);
    }
}

```

```
        Loop If Condition Is True;
    }
}

rule("----BOT----")
{
    event
    {
        Ongoing - Global;
    }
}

rule("bots init")
{
    event
    {
        Ongoing - Each Player;
        All;
        All;
    }

    conditions
    {
        Is Dummy Bot(Event Player) == True;
    }

    actions
    {
        Set Damage Dealt(Event Player, 500);
    }
}
```

```

        Set Max Health(Event Player, 200);
        Wait(0.016, Ignore Condition);
        Heal(Event Player, Null, Max Health(Event Player));
    }
}

rule("Set bot spawn notification")
{
    event
    {
        Ongoing - Each Player;
        Team 1;
        All;
    }

    conditions
    {
        Is Dummy Bot(Event Player) == False;
        Global Variable(noofbots) < 6;
        Host Player == Event Player;
    }

    actions
    {
        Big Message(Event Player, Custom String("Press Interact to set bot location, Press
Crouch to reset", Null, Null, Null));
        Wait(5, Abort When False);
        Loop If Condition Is True;
    }
}

```

```
}
```

```
rule("Setting up Bot location")
```

```
{
```

```
    event
```

```
    {
```

```
        Ongoing - Each Player;
```

```
        Team 1;
```

```
        All;
```

```
    }
```

```
    conditions
```

```
    {
```

```
        Is Button Held(Event Player, Interact) == True;
```

```
        Is Dummy Bot(Event Player) == False;
```

```
        Global Variable(noofbots) < 6;
```

```
        Is In Spawn Room(Event Player) == False;
```

```
    }
```

```
    actions
```

```
    {
```

```
        Modify Global Variable(BotPositions, Append To Array, Position Of(Event Player));
```

```
        Modify Global Variable(BotPositions, Remove From Array By Value, 0);
```

```
        Create Effect(All Players(Team 1), Light Shaft, Green, Value In Array(Global  
Variable(BotPositions), Global Variable(noofbotres)),
```

```
            1, Visible To);
```

```
        Modify Global Variable(noofbots, Add, 1);
```

```
        Set Global Variable(noofbotres, Global Variable(noofbots));
```

```
        Set Global Variable(HighScore, 0);
```



```
    }  
}  
  
rule("Reset bot location")  
{  
    event  
    {  
        Ongoing - Each Player;  
        Team 1;  
        All;  
    }  
  
    conditions  
    {  
        Is Button Held(Event Player, Crouch) == True;  
        Is Dummy Bot(Event Player) == False;  
    }  
  
    actions  
    {  
        Set Global Variable(BotPositions, 0);  
        Set Global Variable(noofbots, 0);  
        Set Global Variable(noofbotres, 0);  
        Destroy All Effects;  
        Set Global Variable(HighScore, 0);  
        Destroy All Dummy Bots;  
    }  
}
```

```

rule("Spawn Bots")
{
    event
    {
        Ongoing - Each Player;
        Team 2;
        All;
    }

    conditions
    {
        Is Button Held(Event Player, Interact) == True;
        Is In Spawn Room(Event Player) == True;
        Number of Players(Opposite Team Of(Team Of(Event Player))) == 0;
        Global Variable(BotPositions) != 0;
    }

    actions
    {
        Destroy All Dummy Bots;
        Wait(0.016, Ignore Condition);
        Set Global Variable(noofbots, Global Variable(noofbotres));
        Create Dummy Bot(Global Variable(Bot), Opposite Team Of(Team Of(Event Player)), -
1.000, Value In Array(Global Variable(
        BotPositions), 0), Vector(0, 0, 0));
        Skip If(Compare(Value In Array(Global Variable(BotPositions), 1), ==, 0), 1);
        Create Dummy Bot(Global Variable(Bot), Opposite Team Of(Team Of(Event Player)), -
1.000, Value In Array(Global Variable(
        BotPositions), 1), Vector(0, 0, 0));
    }
}

```

```

        Skip If(Compare(Value In Array(Global Variable(BotPositions), 2), ==, 0), 1);
        Create Dummy Bot(Global Variable(Bot), Opposite Team Of(Team Of(Event Player)), -
1.000, Value In Array(Global Variable(
            BotPositions), 2), Vector(0, 0, 0));
        Skip If(Compare(Value In Array(Global Variable(BotPositions), 3), ==, 0), 1);
        Create Dummy Bot(Global Variable(Bot), Opposite Team Of(Team Of(Event Player)), -
1.000, Value In Array(Global Variable(
            BotPositions), 3), Vector(0, 0, 0));
        Skip If(Compare(Value In Array(Global Variable(BotPositions), 4), ==, 0), 1);
        Create Dummy Bot(Global Variable(Bot), Opposite Team Of(Team Of(Event Player)), -
1.000, Value In Array(Global Variable(
            BotPositions), 4), Vector(0, 0, 0));
        Skip If(Compare(Value In Array(Global Variable(BotPositions), 5), ==, 0), 1);
        Create Dummy Bot(Global Variable(Bot), Opposite Team Of(Team Of(Event Player)), -
1.000, Value In Array(Global Variable(
            BotPositions), 5), Vector(0, 0, 0));
        Big Message(All Players(All Teams), Custom String("GO!", Null, Null, Null));
        Clear Status(All Players(Team Of(Event Player)), Rooted);
        Set Global Variable(Timer, 0);
        Set Global Variable(Starttimer, True);
    }
}

```

```
rule("Bots control")
```

```

{
    event
    {
        Ongoing - Each Player;
        All;
        All;
    }
}

```

```

    }

    conditions
    {
        Is Dummy Bot(Event Player) == True;
        Ray Cast Hit Player(Eye Position(Event Player), Eye Position(Closest Player To(Event
Player, All Teams))), Players in View Angle(
            Event Player, All Teams, 360), All Dead Players(All Teams), True) == True;
    }

    actions
    {
        disabled Set Facing(Event Player, Direction Towards(Eye Position(Event Player), Eye
Position(Closest Player To(Event Player,
            Players in View Angle(Event Player, Opposite Team Of(Team Of(Event Player)),
360))), To World);
        disabled Set Facing(Event Player, Direction Towards(Eye Position(Event Player), Eye
Position(Closest Player To(Event Player,
            Opposite Team Of(Team Of(Event Player))))) , To World);
        Set Facing(Event Player, Direction Towards(Eye Position(Event Player), Add(Eye
Position(Closest Player To(Event Player,
            Opposite Team Of(Team Of(Event Player))), Vector(0, -0.500, 0))), To World);
        Start Holding Button(Event Player, Primary Fire);
        Wait(0.016, Abort When False);
        Loop If Condition Is True;
    }
}

rule("Bots control past shield")
{

```

event

{

Ongoing - Each Player;

All;

All;

}

conditions

{

Is Dummy Bot(Event Player) == True;

Ray Cast Hit Player(Eye Position(Event Player), Eye Position(Closest Player To(Event Player, All Teams)), Players in View Angle(

Event Player, All Teams, 360), All Dead Players(All Teams), True) != True;

Ray Cast Hit Player(Eye Position(Event Player), Eye Position(Closest Player To(Event Player, All Teams)), Players in View Angle(

Event Player, All Teams, 360), All Dead Players(All Teams), False) == True;

}

actions

{

disabled Set Facing(Event Player, Direction Towards(Eye Position(Event Player), Eye Position(Closest Player To(Event Player,

Players in View Angle(Event Player, Opposite Team Of(Team Of(Event Player)), 360))), To World);

Set Facing(Event Player, Direction Towards(Eye Position(Event Player), Eye Position(Closest Player To(Event Player,

Opposite Team Of(Team Of(Event Player))))) , To World);

Start Holding Button(Event Player, Primary Fire);

Wait(0.016, Abort When False);

Loop If Condition Is True;

```

    }
}

rule("Bots control off")
{
    event
    {
        Ongoing - Each Player;
        All;
        All;
    }

    conditions
    {
        Is Dummy Bot(Event Player) == True;
        Ray Cast Hit Player(Eye Position(Event Player), Eye Position(Closest Player To(Event
Player, All Teams)), Players in View Angle(
            Event Player, All Teams, 360), All Dead Players(All Teams), True) != True;
        Ray Cast Hit Player(Eye Position(Event Player), Eye Position(Closest Player To(Event
Player, All Teams)), Players in View Angle(
            Event Player, All Teams, 360), All Dead Players(All Teams), False) != True;
    }

    actions
    {
        Stop Holding Button(Event Player, Primary Fire);
    }
}

```

```
rule("Bots control off deflecting")
{
    event
    {
        Ongoing - Each Player;
        All;
        All;
    }

    conditions
    {
        Is Dummy Bot(Event Player) == True;
        Ray Cast Hit Player(Eye Position(Event Player), Eye Position(Closest Player To(Event
Player, All Teams)), Players in View Angle(
        Event Player, All Teams, 360), All Dead Players(All Teams), False) == True;
    }

    actions
    {
        Stop Holding Button(Event Player, Primary Fire);
    }
}
```

```
rule("bot died")
```

```
{
    event
    {
        Player Died;
        All;
    }
}
```

```
        All;
    }

    conditions
    {
        Is Dummy Bot(Event Player) == True;
    }

    actions
    {
        Modify Global Variable(noofbots, Subtract, 1);
        Play Effect(All Players(All Teams), Good Pickup Effect, White, Event Player, 100);
        Play Effect(All Players(All Teams), Good Explosion, Lime Green, Event Player, 5);
    }
}
```

```
rule("Bot position reminder")
{
    event
    {
        Ongoing - Each Player;
        All;
        All;
    }

    conditions
    {
        Is Dummy Bot(Event Player) == True;
        Has Spawned(Event Player) == True;
    }
}
```



```

    }

    actions
    {
        Set Player Variable(Event Player, position, Position Of(Event Player));
    }
}

rule("bot position return")
{
    event
    {
        Ongoing - Each Player;
        All;
        All;
    }

    conditions
    {
        Is Dummy Bot(Event Player) == True;
        Distance Between(Player Variable(Event Player, position), Event Player) >= 1;
    }

    actions
    {
        disabled Start Throttle In Direction(Event Player, Direction Towards(Position Of(Event
Player), Player Variable(Event Player, position)), 1,
            To World, Replace existing throttle, Direction and Magnitude);
    }
}

```

```
        Start Throttle In Direction(Event Player, Direction Towards(Event Player, Nearest
Walkable Position(Player Variable(Event Player,
            position))), 1, To World, Replace existing throttle, Direction and Magnitude);
        Wait(1, Abort When False);
        Loop If Condition Is True;
    }
}
```

```
rule("bot position returned")
```

```
{
    event
    {
        Ongoing - Each Player;
        All;
        All;
    }

    conditions
    {
        Is Dummy Bot(Event Player) == True;
        Distance Between(Player Variable(Event Player, position), Event Player) < 1;
    }

    actions
    {
        Stop Throttle In Direction(Event Player);
    }
}
```

```
disabled rule("Bot clear knocked down")
```

```
{
```

```
    event
```

```
    {
```

```
        Ongoing - Each Player;
```

```
        All;
```

```
        All;
```

```
    }
```

```
    conditions
```

```
    {
```

```
        Is Dummy Bot(Event Player) == True;
```

```
        Has Status(Event Player, Knocked Down) == True;
```

```
    }
```

```
    actions
```

```
    {
```

```
        Wait(0.016, Ignore Condition);
```

```
        Clear Status(Event Player, Knocked Down);
```

```
    }
```

```
}
```

```
disabled rule("Bot clear Asleep")
```

```
{
```

```
    event
```

```
    {
```

```
        Ongoing - Each Player;
```

```
        All;
```

```
        All;
```

```
}
```

```
conditions
```

```
{
```

```
    Is Dummy Bot(Event Player) == True;
```

```
    Has Status(Event Player, Asleep) == True;
```

```
}
```

```
actions
```

```
{
```

```
    Wait(0.016, Ignore Condition);
```

```
    disabled Clear Status(Event Player, Asleep);
```

```
    Big Message(All Players(All Teams), Custom String("afdafsd", Null, Null, Null));
```

```
    Damage(Event Player, Null, 1);
```

```
}
```

```
}
```

```
disabled rule("Bot clear stun")
```

```
{
```

```
    event
```

```
    {
```

```
        Ongoing - Each Player;
```

```
        All;
```

```
        All;
```

```
    }
```

```
conditions
```

```
{
```

```
    Is Dummy Bot(Event Player) == True;
```

```
        Has Status(Event Player, Stunned) == True;
    }

    actions
    {
        Wait(0.016, Ignore Condition);
        Clear Status(Event Player, Stunned);
    }
}
```

```
rule("---others---")
{
    event
    {
        Ongoing - Global;
    }
}
```

```
rule("death reset")
{
    event
    {
        Player Died;
        All;
        All;
    }

    conditions
    {
```

```
        Is Dummy Bot(Event Player) == False;
    }

    actions
    {
        Kill(All Living Players(Team Of(Event Player)), Event Player);
        Set Ultimate Charge(All Living Players(Team Of(Event Player)), 0);
        Destroy All Dummy Bots;
        Set Global Variable(Starttimer, 0);
        disabled Set Status(All Living Players(Team Of(Event Player)), Null, Rooted, 9999);
    }
}
```

```
rule("Timer stop")
```

```
{
    event
    {
        Ongoing - Global;
    }

    conditions
    {
        Global Variable(noofbots) == 0;
        Global Variable(noofbotres) != 0;
        Number of Players(Team 1) > 0;
    }

    actions
    {
```

```
        Set Global Variable(Starttimer, 0);
        Big Message(All Players(All Teams), String("{0}:{1}", Custom String("Score", Null, Null,
Null), Global Variable(Timer), Null));
        Wait(5, Ignore Condition);
        Kill(All Living Players(Team 2), Null);
    }
}
```

```
rule("Timer stop high score")
```

```
{
    event
    {
        Ongoing - Global;
    }

    conditions
    {
        Global Variable(noofbots) == 0;
        Number of Players(Team 1) > 0;
        Global Variable(HighScore) > Global Variable(Timer);
        Global Variable(Timer) != 0;
    }

    actions
    {
        Set Global Variable(Starttimer, 0);
        Set Global Variable(HighScore, Global Variable(Timer));
    }
}
```

```
rule("Timer stop high score 0")
```

```
{
```

```
    event
```

```
    {
```

```
        Ongoing - Global;
```

```
    }
```

```
    conditions
```

```
    {
```

```
        Global Variable(noofbots) == 0;
```

```
        Number of Players(Team 1) > 0;
```

```
        Global Variable(HighScore) == 0;
```

```
    }
```

```
    actions
```

```
    {
```

```
        Set Global Variable(Starttimer, 0);
```

```
        Set Global Variable(HighScore, Global Variable(Timer));
```

```
    }
```

```
}
```

```
rule("deflecting")
```

```
{
```

```
    event
```

```
    {
```

```
        Ongoing - Each Player;
```

```
        All;
```

```
        Genji;
```



```
}
```

```
conditions
```

```
{
```

```
    Is Using Ability 2(Event Player) == True;
```

```
}
```

```
actions
```

```
{
```

```
    Set Damage Dealt(Event Player, 1);
```

```
}
```

```
}
```

```
rule("not deflecting")
```

```
{
```

```
    event
```

```
    {
```

```
        Ongoing - Each Player;
```

```
        All;
```

```
        Genji;
```

```
    }
```

```
conditions
```

```
{
```

```
    Is Using Ability 2(Event Player) != True;
```

```
}
```

```
actions
```

```
{
```

```
Set Damage Dealt(Event Player, 100);
```

```
}
```

```
}
```