

```
settings
{
    main
    {
        Description: "Be on Team 1, all instructions are in game"
    }

    modes
    {
        Assault
        {
            Limit Roles: 2 Of Each Role Per Team

            enabled maps
            {
                Volskaya Industries
            }
        }

        Control
        {
            Limit Roles: 2 Of Each Role Per Team

            enabled maps
            {
            }
        }

        Escort
```

```
{  
    Limit Roles: 2 Of Each Role Per Team  
  
    enabled maps  
    {  
    }  
}
```

Hybrid

```
{  
    Limit Roles: 2 Of Each Role Per Team  
  
    enabled maps  
    {  
    }  
}
```

```
}
```

heroes

```
{  
    Team 1  
    {  
        enabled heroes  
        {  
            D.Va  
            Genji  
            Sigma  
        }  
    }  
}
```

```
General
{
    Genji
    {
        Deflect Cooldown Time: 0%
    }

    Sigma
    {
        Kinetic Grasp Cooldown Time: 0%
    }
}
}
```

variables

```
{
    global:
        0: Center
        1: Botspawn
        2: Isbot
        3: botid
        4: start
        5: Spawnrate
        25: Test
}
```

```
rule("----Spawnrate Set----")
```

```
{  
    event  
    {  
        Ongoing - Global;  
    }  
  
    actions  
    {  
        Set Global Variable(Spawnrate, 3);  
    }  
}  
  
rule("Init")  
{  
    event  
    {  
        Ongoing - Global;  
    }  
  
    actions  
    {  
        Set Global Variable(Center, Vector(-14, -4, 90));  
        Set Respawn Max Time(All Players(Team 1), 0.500);  
        Set Global Variable(start, -1);  
    }  
}  
  
rule("HUD")  
{
```

```

event
{
    Ongoing - Global;
}

actions
{
    Create HUD Text(All Players(Team 1), String("{0}: {1}", Custom String("Bot", Null, Null,
Null), Hero Icon String(Global Variable(
        Isbot)), Null), Null, Null, Left, 0, White, White, White, Visible To and String,
Visible Always);

    Create HUD Text(All Players(Team 1), Custom String("Press Thanks to die and change
hero", Null, Null, Null), Null, Null, Right, 0,
        White, White, White, Visible To and String, Visible Always);

    Create HUD Text(All Players(Team 1), Custom String("Press Interact and Crouch to cycle
bots", Null, Null, Null), Null, Null, Right,
        0, White, White, White, Visible To and String, Visible Always);

    Create HUD Text(All Players(Team 1), Custom String("Press Interact to Toggle bot
spawns", Null, Null, Null), Null, Null, Right, 0,
        White, White, White, Visible To and String, Visible Always);
}
}

rule("Center ring")
{
    event
    {
        Ongoing - Global;
    }
}

```

```
actions
{
    Create Effect(All Players(All Teams), Sphere, White, Global Variable(Center), 20, Visible
To Position and Radius);
}
}
```

```
rule("D.va Teleport")
```

```
{
    event
    {
        Ongoing - Each Player;
        Team 1;
        All;
    }

    conditions
    {
        Is In Spawn Room(Event Player) == True;
    }

    actions
    {
        Teleport(Event Player, Global Variable(Center));
    }
}
```

```
rule("---bot spawn---")
```

```
{
```

```
event
{
    Ongoing - Global;
}
}
```

```
rule("Bot spawn locations")
```

```
{
    event
    {
        Ongoing - Global;
    }

    actions
    {
        Modify Global Variable(Botspawn, Append To Array, Vector(-21, -4, 86));
        Modify Global Variable(Botspawn, Append To Array, Vector(-14, -4, 85));
        Modify Global Variable(Botspawn, Append To Array, Vector(-7, -4, 87));
        Modify Global Variable(Botspawn, Append To Array, Vector(-8, -4, 94));
        Modify Global Variable(Botspawn, Append To Array, Vector(-14, -4, 97));
        Modify Global Variable(Botspawn, Append To Array, Vector(-19, -4, 95));
        Modify Global Variable(Botspawn, Append To Array, Vector(-13, 2, 80));
        Modify Global Variable(Botspawn, Append To Array, Vector(-2, 2, 93));
        Modify Global Variable(Botspawn, Append To Array, Vector(-26, 2, 98));
        Modify Global Variable(Botspawn, Remove From Array By Value, 0);
    }
}
```

```
rule("start spawn")
```

```
{  
    event  
    {  
        Ongoing - Each Player;  
        Team 1;  
        All;  
    }  
  
    conditions  
    {  
        Is Button Held(Event Player, Interact) == True;  
        Is Button Held(Event Player, Crouch) != True;  
    }  
  
    actions  
    {  
        Modify Global Variable(start, Multiply, -1);  
    }  
}
```

```
rule("stop spawn")
```

```
{  
    event  
    {  
        Ongoing - Global;  
    }  
  
    conditions  
    {
```



```

        Global Variable(start) != 1;
    }

    actions
    {
        Destroy All Dummy Bots;
    }
}

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

    conditions
    {
        Global Variable(start) == 1;
        Number of Players(Team 2) == 0;
    }

    actions
    {
        Create Dummy Bot(Global Variable(Isbot), Opposite Team Of(Team Of(Event Player)), -1,
Random Value In Array(Global Variable(
        Botspawn)), Vector(0, 0, 0));
    }
}

```

```
        Heal(Event Player, Null, 600);
    }
}

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

    conditions
    {
        Is Button Held(Event Player, Interact) == True;
        Is Button Held(Event Player, Crouch) == True;
    }

    actions
    {
        Modify Global Variable(botid, Add, 1);
    }
}
```

```
rule("is bot 1")
{
    event
    {
```

```
        Ongoing - Global;
    }

    conditions
    {
        Global Variable(botid) == 0;
    }

    actions
    {
        Set Global Variable(Isbot, Hero(Ana));
    }
}

rule("is bot 2")
{
    event
    {
        Ongoing - Global;
    }

    conditions
    {
        Global Variable(botid) == 1;
    }

    actions
    {
        Set Global Variable(Isbot, Hero(Hanzo));
    }
}
```

```
    }  
}  
  
rule("is bot 3")  
{  
    event  
    {  
        Ongoing - Global;  
    }  
  
    conditions  
    {  
        Global Variable(botid) == 2;  
    }  
  
    actions  
    {  
        Set Global Variable(Isbot, Hero(Zarya));  
    }  
}
```

```
rule("is bot 4")  
{  
    event  
    {  
        Ongoing - Global;  
    }  
  
    conditions
```

```
{  
    Global Variable(botid) == 3;  
}  
  
actions  
{  
    Set Global Variable(Isbot, Hero(Mei));  
}  
}
```

```
rule("toggle reset")
```

```
{  
    event  
    {  
        Ongoing - Global;  
    }  
  
    conditions  
    {  
        Global Variable(botid) > 3;  
    }  
  
    actions  
    {  
        Set Global Variable(botid, 0);  
    }  
}
```

```
rule("--bot control--")
```

```
{
    event
    {
        Ongoing - Global;
    }
}
```

```
rule("bot 1")
```

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

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

    actions
    {
        disabled Set Facing(Event Player, Direction Towards(Event Player, Position Of(Closest
        Player To(Event Player, Opposite Team Of(Team Of(
            Event Player))))), To World);
        Set Facing(Event Player, Direction Towards(Event Player, Add(Position Of(Closest Player
        To(Event Player, Opposite Team Of(Team Of(
            Event Player))))), Vector(0, 2, 0))), To World);
    }
}
```

```

        Communicate(Event Player, Hello);
        Wait(Random Integer(0.500, 1.500), Ignore Condition);
        Press Button(Event Player, Ability 2);
        Wait(Global Variable(Spawnrate), Ignore Condition);
        Destroy All Dummy Bots;
    }
}

rule("bot 2")
{
    event
    {
        Ongoing - Each Player;
        All;
        Hanzo;
    }

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

    actions
    {
        Set Facing(Event Player, Direction Towards(Event Player, Position Of(Closest Player
To(Event Player, Opposite Team Of(Team Of(
        Event Player))))), To World);
        Communicate(Event Player, Hello);
        Set Ultimate Charge(Event Player, 100);
    }
}

```

```

        Wait(Random Integer(0.250, 1), Ignore Condition);
        Press Button(Event Player, Ultimate);
        Wait(Global Variable(Spawnrate), Ignore Condition);
        Destroy All Dummy Bots;
    }
}

rule("bot 3")
{
    event
    {
        Ongoing - Each Player;
        All;
        Zarya;
    }

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

    actions
    {
        Set Facing(Event Player, Direction Towards(Event Player, Position Of(Closest Player
To(Event Player, Opposite Team Of(Team Of(
        Event Player))))), To World);
        Communicate(Event Player, Hello);
        Set Ultimate Charge(Event Player, 100);
        Wait(Random Integer(1, 2), Ignore Condition);
    }
}

```



```

        Press Button(Event Player, Ultimate);
        Wait(Global Variable(Spawnrate), Ignore Condition);
        Destroy All Dummy Bots;
    }
}

rule("bot 4")
{
    event
    {
        Ongoing - Each Player;
        All;
        Mei;
    }

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

    actions
    {
        Set Facing(Event Player, Direction Towards(Event Player, Position Of(Closest Player
To(Event Player, Opposite Team Of(Team Of(
        Event Player))))), To World);
        Communicate(Event Player, Hello);
        Set Ultimate Charge(Event Player, 100);
        Wait(Random Integer(0.250, 1), Ignore Condition);
        Press Button(Event Player, Ultimate);
    }
}

```

```
        Wait(Global Variable(Spawnrate), Ignore Condition);
        Destroy All Dummy Bots;
    }
}
```

```
rule("---Restart---")
```

```
{
    event
    {
        Ongoing - Global;
    }
}
```

```
disabled rule("---Dev tools---")
```

```
{
    event
    {
        Ongoing - Global;
    }
}
```

```
disabled rule("position Please delete")
```

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

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

actions
{
    Set Global Variable(Test, Position Of(Event Player));
}
}
```

```
rule("kill restart")
```

```
{
    event
    {
        Ongoing - Each Player;
        Team 1;
        All;
    }

    conditions
    {
        Is Dummy Bot(Event Player) != True;
        Is Communicating(Event Player, Thanks) == True;
    }

    actions
    {
```

```
Kill(Event Player, Null);
```

```
}
```

```
}
```