

---

Subject: Re: [SSGM 4.0 Plugn] Kill Harvester Console Commands

Posted by [Gen\\_Blacky](#) on Sat, 16 Aug 2014 03:10:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Someone make this man a plugin from these example commands please.

```
class CommandKILLVEHICLES :
public ConsoleFunctionClass
{
public:
const char* Get_Name()
{
return "killvehicles";
}
const char* Get_Help()
{
return "KILLVEHICLES - Destroys all vehicles.";
}
void Activate(const char* argumentsString)
{
int count = 0;

for (SLNode<VehicleGameObj> *x = GameObjManager::VehicleGameObjList.Head();x;x =
x->Next())
{
VehicleGameObj *veh = x->Data();
Vector3 pos = Commands->Get_Position(veh);
if ( Functions:::IsTank_Obj(veh) )
{
Commands->Destroy_Object(veh);
Commands->Create_Explosion("Explosion_with_Debris_small",pos,veh);
count++;
}
}

if ( count >= 1 )
{
Console_Output("[BB] All Vehicle's Destroyed.\n");
}
}
};

class CommandKILLVEHICLE :
public ConsoleFunctionClass
{
public:
const char* Get_Name()
{
```

```

return "killvehicle";
}
const char* Get_Help()
{
return "KILLVEHICLE <clientId> - Destroy a player's vehicle.";
}
void Activate(const char* argumentsString)
{
CommandLineParser arguments(argumentsString);
const char* clientIdentifier = arguments.getString();

if (!clientIdentifier || clientIdentifier[0] == '\0')
Console_Output("Please enter a client identifier.");
else
{
const int clientId = getClientIdByIdentifier(clientIdentifier);
if (!isClientId(clientId))
{
Console_Output("Please enter a valid client identifier.");
}
else
{
int killtype = arguments.getInt();
GameObject *obj = Get_GameObj(clientId);
GameObject *Veh = Get_Vehicle(obj);
Vector3 pos = Commands->Get_Position(obj);

if (Veh)
{
if (Get_Vehicle_Driver(Veh) == obj)
{
if (killtype == 1)
{
Commands->Apply_Damage(Veh,99999.0f,"BlamoKiller",0);
}
else
{
Commands->Destroy_Object(Veh);
Commands->Create_Explosion("Explosion_with_Debris_small",pos,obj);
}
Console_Output("[BB] Player's Vehicle Destroyed.\n");
}
}
}
}
};

```

```

class CommandKILL :
public ConsoleFunctionClass
{
public:
const char* Get_Name()
{
return "Kill";
}
const char* Get_Help()
{
return "KILL <clientId> <killtype> - Kills the player's character (if they're driving a vehicle it will kill
that too).";
}
void Activate(const char* argumentsString)
{
CommandLineParser arguments(argumentsString);
const char* clientIdentifier = arguments.getString();

if (!clientIdentifier || clientIdentifier[0] == '\0')
Console_Output("Please enter a client identifier.");
else
{
const int clientId = getClientIdByIdentifier(clientIdentifier);
if (!isClientId(clientId))
{
Console_Output("Please enter a valid client identifier.");
}
else
{
int killtype = arguments.getInt();
GameObject *obj = Get_GameObj(clientId);
GameObject *Veh = Get_Vehicle(obj);
Vector3 pos = Commands->Get_Position(obj);

if (Veh)
{
if (Get_Vehicle_Driver(Veh) == obj)
{
Commands->Destroy_Object(obj);
Commands->Destroy_Object(Veh);
Commands->Create_Explosion("Explosion_with_Debris_small",pos,obj);
}
}

if (PData[clientId]->Spectating == true)
{
Console_Output("[BB] Player %ls cannot be killed while in spectate mode. Use !spectate
command.\n",Get_Wide_Player_Name(Get_GameObj(clientId)));
}
}
}
}
}

```

```

    return;
}

if ( obj )
{
    if(!killtype)
    {
        Commands->Apply_Damage(obj,99999.0f,"BlamoKiller",obj);
        Console_Output("Player has killed theirself.");
        Commands->Create_2D_WAV_Sound("humiliation.wav");
    }
    else if(killtype == 1)
    {
        Commands->Destroy_Object(obj);
        Console_Output("[BB] Player has been killed silently.");
    }
    else if(killtype == 2)
    {
        Commands->Destroy_Object(obj);
        Set_Deaths(clientId, Get_Deaths(clientId) - 1);
        Console_Output("[BB] Player has been killed silently no death.");
    }
    else
    {
        Commands->Apply_Damage(obj,99999.0f,"BlamoKiller",0);
        Console_Output("Player has been killed.");
    }
}
};

```

```

class CommandKILLHARV :
public ConsoleFunctionClass
{
public:
const char* Get_Name()
{
    return "killharv";
}
const char* Get_Help()
{
    return "KILLHARV <team> - Destroy a team harvester.";
}
void Activate(const char* argumentsString)
{
    StringClass OriginalString = argumentsString;

```

```

CommandLineParser arguments(argumentsString);
int team = arguments.getInt();
if ( !team )
{
    Console_Output("Please enter a valid team ID.\n");
}
else
{
    Functions::Kill_Harvester(team);
}

}
};

void Functions::Kill_Harvester(int Team)
{
    SLNode<SmartGameObj> *x = GameObjManager::SmartGameObjList.Head();
    while (x)
    {
        GameObject *o = x->Data();
        if (o)
        {
            if (o->As_VehicleGameObj())
            {
                int TeamType = Commands->Get_Player_Type(o);
                if (TeamType == Team)
                {
                    if (Is_Harvester(o) || Is_Harvester_Preset(o))
                    {
                        Commands->Destroy_Object(o);
                        //Commands->Apply_Damage(o,99999,"Death",0);
                    }
                }
            }
        }
        x = x->Next();
    }
}

```

---