Helicopter Script

Written by Adalace Jewell - Last Updated Sunday, 08 March 2009 19:38

```
// Script to fly a helicopter in Second Life
// author: unknown// Retrieved from Free SL Scripts on www.gendersquare.org/sl
 key avatar;
float ROTATION_RATE = 2.0;     // Rate of turning  float FWD_THRUST = 20;      // Forward
thrust motor force                                                   float BACK_THRUST =
7;      // Backup thrust                                                                fl
oat VERTICAL_THRUST = 7;
// Keep a running linear motor value for better response
vector linear_motor = <0,0,0>;
default {    state_entry() {      llSitTarget(<.4, 0.0, 0.4>, <0,0,0,1>);
llSetVehicleType(VEHICLE_TYPE_AIRPLANE);
    llSetVehicleFloatParam(VEHICLE_ANGULAR_DEFLECTION_EFFICIENCY, 0.1);
llSetVehicleFloatParam(VEHICLE_LINEAR_DEFLECTION_EFFICIENCY, 0.1);
    llSetVehicleFloatParam(VEHICLE_ANGULAR_DEFLECTION_TIMESCALE, 100);
    llSetVehicleFloatParam(VEHICLE_LINEAR_DEFLECTION_TIMESCALE, 100);
    llSetVehicleFloatParam(VEHICLE_LINEAR_MOTOR_TIMESCALE, 0.2);
llSetVehicleFloatParam(VEHICLE_LINEAR_MOTOR_DECAY_TIMESCALE, 10);
    llSetVehicleFloatParam(VEHICLE_ANGULAR_MOTOR_TIMESCALE, 0.2);
    llSetVehicleFloatParam(VEHICLE_ANGULAR_MOTOR_DECAY_TIMESCALE, 0.1);
    llSetVehicleVectorParam(VEHICLE_LINEAR_FRICTION_TIMESCALE, <5,5,5>);
llSetVehicleVectorParam(VEHICLE_ANGULAR_FRICTION_TIMESCALE, <10,10,10>);
    llSetVehicleFloatParam(VEHICLE_BUOYANCY, 1.0);
    llSetVehicleFloatParam(VEHICLE_VERTICAL_ATTRACTION_EFFICIENCY, 0.2);
llSetVehicleFloatParam(VEHICLE_VERTICAL_ATTRACTION_TIMESCALE, 3.0);
    llSetVehicleFloatParam(VEHICLE_BANKING_EFFICIENCY, 1.0);
llSetVehicleFloatParam(VEHICLE_BANKING_MIX, 0.75);
    llSetVehicleFloatParam(VEHICLE_BANKING_TIMESCALE, 0.05);
    llSetCameraEyeOffset(<-7.0, 0.0, 3.0>);     llSetSitText("Fly");     llSetCameraAtOffset(<0,
0, 1>);                                                              }
   touch_start(integer num) {      llWhisper(0, "Buy me! Right click and choose 'Buy' then take
me out of your inventory to fly!");                               }
   changed(integer change) {      if (change & CHANGED_LINK) {        key agent =
llAvatarOnSitTarget();
 if (agent) {
          if (agent != llGetOwner()) {
             // only the owner can use this vehicle
             llSay(0, "You aren't the owner of this vehicle. Please visit Zoe Airfield or contact
Ker Dawn to purchase one.");
             llUnSit(agent);
             llPushObject(agent, <0,0,10>, ZERO_VECTOR, FALSE);
             llGiveInventory(agent, "Zoe Airfield");
          } else {
             // driver is entering the vehicle
             llSetStatus(STATUS_PHYSICS, TRUE);
             llRequestPermissions(agent, PERMISSION_TRIGGER_ANIMATION |
PERMISSION_TAKE_CONTROLS);
```

```
            }
        } else {
            // driver is getting up
            llSetStatus(STATUS_PHYSICS, FALSE);
            llReleaseControls();
        }
    }
}

    run_time_permissions(integer perm) {
        if (perm) {
            llTakeControls(CONTROL_FWD | CONTROL_BACK | CONTROL_RIGHT |
CONTROL_LEFT | CONTROL_ROT_RIGHT | CONTROL_ROT_LEFT | CONTROL_UP |
CONTROL_DOWN, TRUE, FALSE);
        }
    }
    control(key id, integer level, integer edge) {       if(level & (CONTROL_LEFT |
CONTROL_ROT_LEFT)) {
 llSetVehicleVectorParam(VEHICLE_ANGULAR_MOTOR_DIRECTION,
<-ROTATION_RATE,0,0>);
        } else if (edge & (CONTROL_LEFT | CONTROL_ROT_LEFT)) {
            llSetVehicleVectorParam(VEHICLE_ANGULAR_MOTOR_DIRECTION, <0,0,0>);
        }
        if(level & (CONTROL_RIGHT | CONTROL_ROT_RIGHT)) {
            llSetVehicleVectorParam(VEHICLE_ANGULAR_MOTOR_DIRECTION,
<ROTATION_RATE,0,0>);
        } else if (edge & (CONTROL_RIGHT | CONTROL_ROT_RIGHT)) {
            llSetVehicleVectorParam(VEHICLE_ANGULAR_MOTOR_DIRECTION, <0,0,0>);
        }
        if(level & CONTROL_FWD) {        linear_motor.x = FWD_THRUST;        } else if (edge &
CONTROL_FWD) {
         linear_motor.x = 0;
        }
        if(level & CONTROL_BACK) {
            linear_motor.x = -BACK_THRUST;
        } else if (edge & CONTROL_BACK) {
            linear_motor.x = 0;
        }

        if(level & CONTROL_UP) {
            linear_motor.z = VERTICAL_THRUST;
        } else if (edge & CONTROL_UP) {
            linear_motor.z = 0;
        }
        if(level & CONTROL_DOWN) {
            linear_motor.z = -VERTICAL_THRUST;
```

```
        } else if (edge & CONTROL_DOWN) {
            linear_motor.z = 0;
        }
        llSetVehicleVectorParam(VEHICLE_LINEAR_MOTOR_DIRECTION, linear_motor);

    }
}
```