

Motorcycle script

Written by Adalace Jewell -

```
// example motorcycle script. Originally written by Cory Linden.
// Then modified and tweaked by Andrew Linden for the forum script library.

// Retrieved from Free SL Scripts on http://www.freeSLscripts.com or www.gendersquare.org/sl
// Root prim should be oriented such that its local X-, Y- and Z-axes are
// parallel to forward, left, and up respectively.
//
// Sound triggers are commented out but not removed, so if you
// want sounds, just add the sounds to the cycle's contents and uncomment
// the triggers.
//
// Be careful when changing parameters. Some of them can be very
// sensitive to change, such that a change of less than 5% can have a
// noticeable effect. You can tell some (but not necessarily all) of the
// more sensitive settings in this example by looking for the ones that
// have been set to double precision. Changing only one at a time is a
// good idea.
//
// The geometry of the motorcycle itself can have significant impact on
// whether it is in a straight line when not trying to turn. For best results
// use asymmetric design with as wide of a base as you can tolerate.

// These are globals only for convenience (so when you need to modify
// them you need only make a single change). There are other magic numbers
// below that have not yet been pulled into globals.
float gMaxTurnSpeed = 12;
float gMaxWheelieSpeed = 5;
float gMaxFwdSpeed = 30;
float gMaxBackSpeed = -10;
float gAngularRamp = 0.17;
float gLinearRamp = 0.2;

// These are true globals whose values are "accumulated" over
// multiple control() callbacks.
float gBank = 0.0;
vector gLinearMotor = <0, 0, 0>;
vector gAngularMotor = <0, 0, 0>;
```

Motorcycle script

Written by Adalace Jewell -

```
default
{
    state_entry()
    {
        // init stuff that never changes
        llSetSitText("Ride");
        llCollisionSound("", 0.0);
        llSitTarget(<0.6, 0.05, 0.20>, ZERO_ROTATION);
        llSetCameraEyeOffset(<-6.0, 0.0, 1.0>);
        llSetCameraAtOffset(<3.0, 0.0, 1.0>);


        // create the vehicle
        llSetVehicleFlags(-1);
        llSetVehicleType(VEHICLE_TYPE_CAR);
        llSetVehicleFlags(VEHICLE_FLAG_LIMIT_MOTOR_UP
            | VEHICLE_FLAG_LIMIT_ROLL_ONLY);
        llSetVehicleFloatParam(VEHICLE_ANGULAR_DEFLECTION_EFFICIENCY, 0.2);
        llSetVehicleFloatParam(VEHICLE_LINEAR_DEFLECTION_EFFICIENCY, 0.8);
        llSetVehicleFloatParam(VEHICLE_ANGULAR_DEFLECTION_TIMESCALE, 0.8);
        llSetVehicleFloatParam(VEHICLE_LINEAR_DEFLECTION_TIMESCALE, 0.3);


        llSetVehicleFloatParam(VEHICLE_LINEAR_MOTOR_TIMESCALE, 0.8);
        llSetVehicleFloatParam(VEHICLE_LINEAR_MOTOR_DECAY_TIMESCALE, 0.4);
        llSetVehicleFloatParam(VEHICLE_ANGULAR_MOTOR_TIMESCALE, 0.01);
        llSetVehicleFloatParam(VEHICLE_ANGULAR_MOTOR_DECAY_TIMESCALE, 0.35);


        llSetVehicleVectorParam(VEHICLE_LINEAR_FRICTION_TIMESCALE, <1000, 100, 1000>);
        llSetVehicleVectorParam(VEHICLE_ANGULAR_FRICTION_TIMESCALE, <100, 10, 100>);


        llSetVehicleFloatParam(VEHICLE_VERTICAL_ATTRACTION_EFFICIENCY, 0.49);
        llSetVehicleFloatParam(VEHICLE_VERTICAL_ATTRACTION_TIMESCALE, 0.44);


        llSetVehicleFloatParam(VEHICLE_BANKING_EFFICIENCY, 3.0);
        llSetVehicleFloatParam(VEHICLE_BANKING_MIX, 0.7);
        llSetVehicleFloatParam(VEHICLE_BANKING_TIMESCALE, 0.01);
```

Motorcycle script

Written by Adalace Jewell -

```
//IISetVehicleFloatParam(VEHICLE_HOVER_HEIGHT, 2.0);
//IISetVehicleFloatParam(VEHICLE_HOVER_TIMESCALE, 1.0);
//IISetVehicleFloatParam(VEHICLE_HOVER_EFFICIENCY, 0.5);
}

changed(integer change)
{
  if (change & CHANGED_LINK)
  {
    key agent = IIAvatarOnSitTarget();
    if (agent)
    {
      if (agent != IIGetOwner())
      {
        // owner has mounted
        IISay(0, "You aren't the owner");
        IISit(agent);
        IIPushObject(agent, <0,0,100>, ZERO_VECTOR, FALSE);
      }
      else
      {
        // not the owner ==> boot off
        IISetStatus(STATUS_PHYSICS, TRUE);
        IIRestPermissions(agent, PERMISSION_TRIGGER_ANIMATION |
PERMISSION_TAKE_CONTROLS);
        //IISound("start", 0.40);

        // reset the global accumulators
        gAngularMotor = <0, 0, 0>;
        gLinearMotor = <0, 0, 0>;
        gBank = 0.0;
      }
    }
  }
  else
  {
    // dismount
    IISetStatus(STATUS_PHYSICS, FALSE);
    IIRestControls();
    IISetAnimation("motorcycle_sit");
    //IISound("off", 0.4);
  }
}
```

Motorcycle script

Written by Adalace Jewell -

```
}
```

```
run_time_permissions(integer perm)
{
    if (perm)
    {
        IISStartAnimation("motorcycle_sit");
        IITakeControls(CONTROL_FWD | CONTROL_BACK | CONTROL_RIGHT |
CONTROL_LEFT
        | CONTROL_ROT_RIGHT | CONTROL_ROT_LEFT | CONTROL_UP, TRUE, FALSE);
        //IILoopSound("on", 1.0);
    }
}
```

```
control(key id, integer level, integer edge)
{
    // The idea here is to ramp up the motors when the keys are held down for a long
    // time and to let the motors decay after they are let go. This allows fine-
    // tuning of the motion of the vehicle by throttling the key controls.
    //
    // Note that this probably doesn't work well when the client FPS and/or the server
    // FPS is lagging. So for best results you'll want to turn off as much visual
    // effects as you can tolerate, and drive in the more empty areas.
```

```
// linear
integer key_control = FALSE;
if(level & CONTROL_FWD)
{
    gLinearMotor.x = gLinearMotor.x + gLinearRamp * (gMaxFwdSpeed - gLinearMotor.x);
    key_control = TRUE;
}
if(level & CONTROL_BACK)
{
    gLinearMotor.x = gLinearMotor.x + gLinearRamp * (gMaxBackSpeed - gLinearMotor.x);
    key_control = TRUE;
}
if (key_control)
{
    IISetVehicleVectorParam(VEHICLE_LINEAR_MOTOR_DIRECTION, gLinearMotor);
    key_control = FALSE;
}
```

Motorcycle script

Written by Adalace Jewell -

```
}
else
{
    if (gLinearMotor.x > 15 || gLinearMotor.x < -5)
    {
        // Automatically reduce the motor if keys are let up when moving fast.
        gLinearMotor.x *= 0.8;
        IISetVehicleVectorParam(VEHICLE_LINEAR_MOTOR_DIRECTION, gLinearMotor);
    }
    else
    {
        // reduce the linear motor accumulator for the next control() event
        gLinearMotor.x *= 0.8;
    }
}

// angular
if(level & (CONTROL_RIGHT|CONTROL_ROT_RIGHT))
{
    gAngularMotor.x = gAngularMotor.x + gAngularRamp * (gMaxTurnSpeed -
gAngularMotor.x);
    key_control = TRUE;
}
if(level & (CONTROL_LEFT | CONTROL_ROT_LEFT))
{
    gAngularMotor.x = gAngularMotor.x - gAngularRamp * (gMaxTurnSpeed +
gAngularMotor.x);
    key_control = TRUE;
}
if(level & CONTROL_UP)
{
    gAngularMotor.y = gAngularMotor.y - gAngularRamp * (gMaxWheelieSpeed +
gAngularMotor.y);
    key_control = TRUE;
}
if (key_control)
{
    // turn on banking and apply angular motor
    gBank = 3.0;
    IISetVehicleFloatParam(VEHICLE_BANKING_EFFICIENCY, gBank);
    IISetVehicleVectorParam(VEHICLE_ANGULAR_MOTOR_DIRECTION,gAngularMotor);
    gAngularMotor *= 0.95; // light attenuation
}
else
```

Motorcycle script

Written by Adalace Jewell -

```
{
  if (gAngularMotor.x > 4
      || gAngularMotor.x < -4)
  {
    // We were turning hard, but no longer ==> reduce banking to help
    // the motorcycle travel straight when bouncing on rough terrain.
    // Also, turn off the angular motor ==> faster upright recovery.
    gAngularMotor *= 0.4;
    gBank *= 0.5;
    llSetVehicleFloatParam(VEHICLE_BANKING_EFFICIENCY, gBank);
    llSetVehicleVectorParam(VEHICLE_ANGULAR_MOTOR_DIRECTION,gAngularMotor);
  }
  else
  {
    // reduce banking for straighter travel when not actively turning
    gAngularMotor *= 0.5;
    gBank *= 0.8;
    llSetVehicleFloatParam(VEHICLE_BANKING_EFFICIENCY, gBank);
  }
}
}
```