How To:
A tutorial on how to get vertigo in Farcry

Written by Clivey
20th May 2005

Links:
www.crymod.com

Introduction

EDIT: THE CODE WORKS, HOWEVER, COPYING THE CODE FROM THE PDF DOESN'T SEEM TO WORK. PLEASE DOWNLOAD THE MOD AS WELL

Firstly I will still release the HATE mod eventually, which will enable you to fly properly. The code discussed in the document has nothing to do with what I have done and this script will only give you the basic functions of flight. Having looked at this Ovni code this week, I have no doubt it could be configured eventually be more controllable.

What we cover here is to configure the scripts of the so called Ovni script. All we are going to do is to get you airborne. The rest is up to you to learn. Personally I would like to set some ground rules and create a standard format for this code, to enable mappers to release maps, knowing the code will work with any flymod, but alas I don't think that will ever happen.

History:
From what I have researched extensively, this code to fly was not from the Ovni script, but from the University of Birmingham. A group of people were formed to create a Microsurgery Simulator on a PC in 3D. After much debate, the core engine was confirmed as the CryEngine. Main reasons were the ability to configure and bespoke the engine to their requirements. After going into the Uni file archive, the time stamps were before the FC release, hence why I would have the assumption they wrote the code.
(If the ovni coder wishes to dispute this then he/she should take it up with the University, not me).

If you Google search VE_Final.pdf, you will see that they created the code and spent 6 months writing up and created a whole mod around navigating the Human Heart in a Virtual Sub. I only wish they would release it! The documentation itself is actually an interesting read (well, if your sad like me)

Tools You Will Need

NONE. Yes that right. No Max, No LuaEdit, Nothing.

We don't need any tools to get your baby flying. What we will do is get a buggy flying. Yes, strange but true.

I would expect that should you want to look into this seriously, that you would know how to import a plane into Sandbox from Max with all the relevant helpers. For the benefit of non Max experts, we will use a model that is already pre-built.

I will be assuming that you know how to create a mod, Edit script files with Notepad. If not, there any many tutorials in how to do so.

I would really suggest doing this as a mod and not to edit your original scripts files if you go online as a multiplayer. That's my first, last and only warning.

So, normal disclaimer. I take no responsibility in your ability to edit your script files, the aftermath left behind should things not work out for you.

Structure

There will be certain references in this documentation where I will be assuming that you will be calling your mod “FLYMOD”. Should your current MOD name be different, then assume to change this name.

Once you have unpacked your script files, objects and created you MOD folder, the first thing we need to do is to register the class. Those using DCL, won’t need to do this and again, I’ll assume that you will know how to do it yourself.

So beginners:

So, to start with, you need to create a folder in you MOD directory called FLYMOD. When following the below, it is important not to alter your .PAK files.

Go into your FarCry/FCData folder and unzip your scripts.pak into your MOD/FLYMOD folder.

Now do the same with objects.pak, objects1.pak and objects2.pak.

You should now have:

FarCry/FLYMOD/Scripts and FarCry/FLYMOD/Objects folders.

Normally Modders would only unpack what they require from the data files, but to make it easy, it is best to unpack the whole thing.

ClassRegistry.lua

Within your scripts folder, you will find a file named ClassRegistry.lua

Open this up with Notepad and scroll down until you get to a section called

-- vehicles ----------

{"Vehicle", "Boat", 90,"Vehicles/boat.lua"},
{"Vehicle", "FWDVehicle", 91,"Vehicles/fwdvehicle.lua"},
{"Vehicle", "Buggy", 92,"Vehicles/Buggy.lua"},

You will now add a line directly beneath the buggy line as follows:

{"Vehicle", "FlyBuggy", 232,"Vehicles/FlyBuggy.lua"},

(if you are already a modder and added your own vehicles, you hopefully will know not to duplicate classes)

Now close and save that file.

Now go into the Scripts/Default/Entities/Vehicles folder and copy the buggy.lua file. Rename your “Copy of buggy.lua” to FlyBuggy.lua

Open up the FlyBuggy.lua with Notepad. From the Edit Menu, chose Select All. Once all the text is highlighted, press your delete key, so the file is completely empty.

Now copy all of the following text into your FlyBuggy.lua file and then save it.

-- START COPY BELOW

```
		FlyBuggy = {
--		type = "Vehicle",


		ammoMG = 500,
		ammoRL = 30,

		-- vehicle gets different damage depending on who's shooter
		-- defines the intensity of the damage caused by the weapons to
		-- the vehicle

		--
		DamageParams = {
			fDmgScaleAIBullet = 0.1,
			fDmgScaleAIExplosion = 0.335,--patch2:vehicles must explode with 1 missile --before was 0.4,
			fDmgScaleBullet = 0.25,
			fDmgScaleExplosion = 0.25,--patch2:vehicles must explode with 1 missile --before was 1,

			dmgBulletMP = 1.0,--
		},

		--model to be used for destroyed vehicle
		fileModelDead = "objects/Vehicles/buggy/buggy_wreck.cgf",
		fileGunModel = "Objects/Vehicles/Mounted_gun/mounted_gun_Buggy.cga",

		fPartUpdateTime = 0,
		fGroundtime = 0,
		dWeapon = 0,
		tpvStateD = 0,

		--entering fake_jump/blending staff
		entVel = 9,

		userCounter = 0,
		driverWaiting = 1,
		driverDelay = 0,
		passengerLimit = 1,
		curPathStep = 0,
		-- previous state on the client before entering the vehicle
		bDriverInTheVehicle = 0,
		-- previous driver on the client before leaving the vehicle
		pPreviousDriver=nil,
		-- previous passenger state on the client before entering the vehicle
		bPassengerInTheVehicle = 0,
		-- previous passenger on the client before leaving the vehicle
		pPreviousPassenger=nil,

		IsPhisicalized = 0,
		--//////////////////////////////////////////////////////////////////////////
		-- damage stuff
		-- particle system to display when the vehicle is damaged stage 1
		Damage1Effect = "smoke.vehicle_damage1.a",
		-- particle system to display when the vehicle is damaged stage 2
		Damage2Effect = "smoke.vehicle_damage2.a",
		-- particle system to display when the vehicle explodes
		ExplosionEffect = "explosions.4WD_explosion.a",
		-- particle system to display when the vehicle is destroyed
		DeadEffect = "fire.burning_after_explosion.a",
		-- material to be used when vehicle is destroyed
		DeadMaterial = "Vehicles.Gunboat_Screwed",
		szNormalModel="Objects/vehicles/buggy/buggy.cgf",
		-- szNormalModel="objects/Vehicles/patrolboat/patrolboat_hull.cgf",
		PropertiesInstance = {
```

```
		sightrange = 220,
		soundrange = 10,	-- rememeber that sound ranges intersect and sound range for AI doubles when in alert
		aibehavior_behaviour = "Boat_idle",
		groupid = 154,
		bUsable = 1,			-- if this boat can be used by _localplayer
	},


	Properties = {

		bHasRockets = 1,-- to be able to have boats with no rockets - empty RL ammo
		bActive = 1,		-- if vehicle is initially active or needs to be activated
		bUseRL = 0,		-- weapon for gunner is RocketLauncher
		bUseRLguided = 0,	-- weapon for gunner is COVERRL
		damage_players = 1,
		sDriverName = "",
--		bHasMWeapon = 1,

		bLightsOn = 0,

		fAISoundRadius = 30,

		AttackParams = {
			sightrange = 220,
			horizontal_fov = -1,
		},

		GunnerParams = {
			responsiveness = 50,
			sightrange = 150,
			attackrange = 350,
			horizontal_fov = -1,
--			aggression = 1,
--			accuracy = 1,
		},

		bSetInvestigate = 0,
		bSameGroupId = 1,		-- send signals withing same group

		bLandOnPlayerLand=0,
		bTrackable=1,
		fileName = "objects/Vehicles/gunboat/gunboat.cgf",
		--fileName = "objects/Vehicles/gunboat/gunboat.cgf",

		fStartDelay = 2,

		bDrawDriver = 0,
		damping = 0.1,
		fLimitLRAngles = 150,
		fLimitUDMinAngles = -45,
		fLimitUDMaxAngles = 40,

		ExplosionParams = {
			nDamage = 900,
			fRadiusMin = 8.0, -- default 12
			fRadiusMax = 10, -- default 35.5
			fRadius = 10, -- default 17
			fImpulsivePressure = 200, -- default 200
		},
	-- those are AI related properties
		pointReinforce = "Drop",
		pointBackOff = "Base",
		aggression = 1.0,
		commrange = 100.0,
		cohesion = 5,
		attackrange = 70,
		horizontal_fov = -1,
		vertical_fov =90,
		eye_height = 2.1,
		max_health = 70,
		accuracy = 0.6,
		responsiveness = 7,
		species = 1,
		fSpeciesHostility = 2,
		fGroupHostility = 0,
```

```
                fPersistence = 0,
                aicharacter_character = "BoatGun",
                bodypos = 0,
                pathname = "none",
                pathsteps = 0,
                pathstart = 0,
                forward_speed = 1,  -- don't scale down fwd impuls - max speed
                bUsePathfind = 0,    -- pathfind when outdoors
        },



--UBI changes - new boat settings


        boat_paramsAI={
                Damprot         = 50000, --8000,
                Dampv           = 10500, --3500,
                Dampvs          = 33500, --3500,
                Dampvh          = 100000,
--              Dampw           = 30000,  --waves dump
                Dampw           = .3,       --waves damp
                Turn            = 200000,--200000
                Speedv          = 375000,--115000, --275000
                Speedturnmin    = .5,
                Stand           = 100000,-- forsing to normal vertical position impuls
                WaveM           = 100,      --fake waves momentum
                TiltTurn    = 1570,     --tilt momentum when turning
                TiltSpd         = 100,      --tilt momentum when speeding up
                TiltSpdA   = 0.06,      --tilt momentum when speeding up (acceleration thrhld)
                TiltSpdMinV     = 10.0,     --tilt momentum when speeding up (min speed to tilt when not accelerating)
                TiltSpdMinVTilt = 0.37,     --tilt momentum when speeding up (how much to tilt when not accelerating)
                fMass           = 15000,
                Flying          = 0,

                StandInAir      = 10000,  -- forsing to normal vertical position impuls, when in air.
                gravity         = -9.81,--gravity , used whe the boat is jumping
        },

        boat_params={
                Damprot         = 50,       -- turning dampening
                Dampv           = 90,       -- vertical movement dampening
                Dampvs          = 50,       -- accelerating turn dampening
                Dampvh          = 10,       -- accelerating vertical turn dampening
                Dampw           = 40,       -- stationary turn dampening
                Turn            = 100,      -- turn speed
                Speedv          = 15000,-- maximum vehicle speed
                Speedturnmin    = 0,        -- minimum turning speed
                WaveM           = 90,       -- simulated fluid effects of water
                Stand           = 100,      -- forces vertical alignment when stationary
                TiltTurn    = 1,        -- tilt momentum when turning
                TiltSpd         = 0.5,      -- tilt momentum when speeding up
                TiltSpdA   = 0.01,      -- tilt momentum when speeding up (acceleration thrhld)
                TiltSpdMinV     = 10.0,     -- min speed to tilt when not accelerating
                TiltSpdMinVTilt = 100,      -- tilt momentum when speeding up (how much to tilt when not accelerating)
                DownRate        = 0,        -- Rate of return to downward motion
                BackUpRate      = 450,      -- Rate of return to upward motion
                BackUpSlowDwnRate=0,
                UpSpeedThrhld   = 1,
                fMass           = 10,       -- Mass of object.
                Flying          = 3,

        },


--UBI changes - over

--      b_speedv = 1200,  -- controls boat speed (movement forward/backward impulse)
--      b_turn = 100,  -- controls how fast boat turns (turning left/right impulse)
--      fMass = 2500,
        sound_time = 0,
        particles_updatefreq = 0.24, --initial frequency of updating particles
        partDmg_time = 0,
--// particles definitions
```

```
--////////////////////////////////////////////////////////////////////////////////////
		bExploded=false,
		-- engine health, status of the vehicle
		-- default is set to maximum (1.0f)
		fEngineHealth = 100.0,
		-- damage inflicted to the vehicle when it collides
		fOnCollideDamage=0.90,
		-- damage inflicted to the vehicle when it collides with terrain (falls off)
		fOnCollideGroundDamage=0.90,
		--damage when colliding with another vehicle, this value is multiplied by the hit.impact value.
		fOnCollideVehicleDamage=3.5,
		bGroundVehicle=0,
		-- to smooth speed changes
		timecount=0,
		previousTimes={},
		driverT = {
			type = PVS_DRIVER,

			helper = "driver",
			in_helper = "driver_sit_pos",
			sit_anim = "buggy_driver_sit",
			anchor = AIAnchor.AIANCHOR_BOATENTER_SPOT,
			out_ang = -90,
			message = "Press USE to fly the Pod",
			timePast=0,
			HS=0,	-- used for fake jump arch calculatio - arch scale
			HK=0,	-- used for fake jump arch calculatio
			HO=0,	-- used for fake jump arch calculatio
			HT=0,	-- used for fake jump arch calculatio

			animations = {
				"buggy_driver_sit",		-- idle in animation
				"buggy_driver_moving",			-- driving firward
				"buggy_driver_forward_hit",	-- impact / break
				"buggy_driver_leftturn",	-- turning left
				"buggy_driver_rightturn",	-- turning right
				"buggy_driver_reverse",	-- reversing
				"buggy_driver_reverse_hit",	-- reversing impact / break
			},
		},

		--seats
			passengersTT = {
			{
				type = PVS_PASSENGER,
				helper = "passenger",
				in_helper = "passenger_sit_pos",--"passenger",
				in_anim = "buggy_passenger_in",
				out_anim = "buggy_passenger_out",
				sit_anim = "buggy_passenger_sit",
				anchor = AIAnchor.z_CARENTER_PASSENGER1,
				out_ang = 90,
				message = "@passengerbuggy",
				-- invehicle animations
				},
		},

		--canbepushed, if player is in contact with some entity and he is pressing use, its checked this function.
		--can return 3 kind of values:
		--		- nil if the entity cant be pushed
		--		- -1 or 0 if the push force to be used is the player standard
		--		- a different value means a custom push force
		CanBePushed = VC.CanBePushed,
		pushpower = 7000,
}

VC.CreateVehicle(FlyBuggy);

--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:OnReset()
		VC.OnResetCommon(self);
		self.fEngineHealth = 100.0;
		self:NetPresent(1);
		VC.EveryoneOutForce(self);
		self.bExploded=false;
```

```
		self.cnt:SetVehicleEngineHealth(self.fEngineHealth);
		--AI stuff
--		self:RegisterWithAI(AIOBJECT_BOAT,self.Properties);
		AI:RegisterWithAI(self.id, AIOBJECT_BOAT,self.Properties,self.PropertiesInstance);
--		AI_HandlersDefault:InitCharacter( self );
		VC.AIDriver( self, 0 );

		self.curPathStep = 0;
		self.fPartUpdateTime = 0;
		self.fGroundtime = 0;
-- Put physics asleep.
		self:AwakePhysics(0);

--front_light
--fron_light_left front_light_right
--back_light_left back_light_right
--textures/lights/front_light
		self.cnt:InitLights( "front_light","textures/lights/front_light",
					"front_light_left","front_light_right","humvee_frontlight",
					"back_light_left", "back_light_right","" );

end


--////////////////////////////////////////////////////////////////////////////////////
--////////////////////////////////////////////////////////////////////////////////////
--// CLIENT functions definitions
--////////////////////////////////////////////////////////////////////////////////////
--////////////////////////////////////////////////////////////////////////////////////
FlyBuggy.Client = {
		OnInit = function(self)
			self:InitClient();
		end,
		OnShutDown = function(self)
			self:OnShutDown();
		end,
		Alive = {
			OnBeginState = function( self )
				VC.InitBoatCommon(self);
			end,
			OnContact = function(self,player)
				self:OnContactClient(player);
			end,
			OnUpdate = function(self,dt)
				self:UpdateClientAlive(dt);
			end,
			OnCollide = VC.OnCollideClient,
			OnBind = VC.OnBind,
			OnUnBind = VC.OnUnBind,
		},
		Inactive = {
			OnBeginState = function( self )
				self:Hide(1);
			end,
			OnEndState = function( self )
				self.IsPhisicalized = 0;
			end,
		},
		Dead = {
			OnBeginState = function( self )
				VC.BlowUpClient(self);
			end,
			OnContact = function(self,player)
				VC.OnContactClientDead(self,player);
			end,
			OnUpdate = VC.UpdateClientDead,
			OnCollide = VC.OnCollideClient,
			OnUnBind = VC.OnUnBind,
		},
}
--- server stuff
FlyBuggy.Server = {
		OnInit = function(self)
			self:InitServer();
		end,
```

```
	OnShutDown = function(self)
		self:OnShutDown();
	end,
	Alive = {
		OnBeginState = function( self )
			VC.InitBoatCommon(self);
		end,
		OnContact = function(self,player)
			self:OnContactServer(player);
		end,
		OnDamage = VC.OnDamageServer,
		OnCollide = VC.OnCollideServer,
		OnUpdate = function(self,dt)
			self:UpdateServer(dt);
		end,
		OnEvent = function (self, id, params)
			self:OnEventServer( id, params);
		end,
	},
	Inactive = {
	},
	Dead = {
		OnBeginState = function( self )
			VC.BlowUpServer(self);
		end,
		OnContact = function(self,player)
			VC.OnContactServerDead(self,player);
		end,
	},
}

--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:InitClient()


	if( self.Properties.bHasRockets == 0) then
		self.ammoRL = 0;
	end

	VC.InitSeats(self, FlyBuggy);

	--// load sounds on the client
	--////////////////////////////////////////////////////////////////////////////////////

	self.ExplosionSound=Sound:Load3DSound("sounds\\explosions\\explosion2.wav",0,255,150,300);

	--self.drive_sound = Sound:Load3DSound("sounds\\vehicle\\FlyBuggy\\PodRest.wav",0,185,50,300);
	self.drive_sound = Sound:Load3DSound("sounds\\vehicle\\FlyBuggy\\PodRest.wav",0,185,30,150);
	self.drive_sound_move = Sound:Load3DSound("sounds\\vehicle\\FlyBuggy\\PodRest.wav",0,35,10,150);
	self.maxvolume_speed = 25;

	self.accelerate_sound = {
		Sound:Load3DSound("sounds\\vehicle\\FlyBuggy\\PodRest.wav",0,0,7,100),
		Sound:Load3DSound("sounds\\vehicle\\FlyBuggy\\PodRest.wav",0,0,7,100),
		Sound:Load3DSound("sounds\\vehicle\\FlyBuggy\\PodRest.wav",0,0,7,100),
		Sound:Load3DSound("sounds\\vehicle\\FlyBuggy\\PodRest.wav",0,0,7,100),
	};

	self.engine_start = Sound:Load3DSound("sounds\\vehicle\\FlyBuggy\\PodIgnition.wav",0,185,30,150);
	self.engine_off = Sound:Load3DSound("sounds\\vehicle\\FlyBuggy\\PodStop.wav",0,185,30,150);
	self.crash_sound = Sound:Load3DSound("sounds\\vehicle\\boat\\gunboathit.wav",0,70,7,100);
	self.land_sound = Sound:Load3DSound("SOUNDS\\Vehicle\\boat\\boatsplash.wav",0,200,7,100);
	self.light_sound = Sound:Load3DSound("SOUNDS/items/flight.wav",SOUND_UNSCALABLE,160,3,30);
	--self.break_sound = Sound:Load3DSound("sounds\\vehicle\\break1.wav",0,0,7,100);
	--self.sliding_sound = Sound:Load3DSound("sounds\\vehicle\\break2.wav",0,0,7,100);

	-- init common stuff for client and server
	VC.InitBoatCommon(self);

	self:InitWeapon();

end

--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:UpdateClientAlive(dt)
```

```
	if(self.lifeCounter < 100) then
		self.lifeCounter = self.lifeCounter + 1;
	end


	VC.PlayEngineOnOffSounds(self);

	-- create particles and all that
	VC.ExecuteDamageModel(self, dt);

	-- plays the sounds, using a timestep of 0.04

	-- get vehicle's velocity
	local fCarSpeed = self.cnt:GetVehicleVelocity();

	self.sound_time = self.sound_time + dt;
	if ( self.sound_time > self.particles_updatefreq ) then

		if(self.fEngineHealth<1)then
			self.particles_updatefreq=0.4;
		end

		-- reset timer
		self.sound_time = 0;

		-- average the last 4 speed frames together to smooth changes out a bit

		self.previousTimes[band(self.timecount,3)]=fCarSpeed;
		self.timecount=self.timecount+1;

		local total=0;
		for idx, element in self.previousTimes do
			total=total+element;
		end

		if (total<=0.1) then
			total=1;
		end

		fCarSpeed=total/4.0;


		VC.PlayDrivingSounds(self,fCarSpeed);
	end

	if( self.gunnerT and self.gunnerT.entity ) then

		local offsDir=self.gunnerT.entity:GetDirectionVector();
		local handlerPos = self:GetHelperPos("gun",0);

		handlerPos.x = handlerPos.x + offsDir.x*.83;
		handlerPos.y = handlerPos.y + offsDir.y*.83;
		handlerPos.z = handlerPos.z + offsDir.z*.83;
		self.gunnerT.entity:SetHandsIKTarget( handlerPos );
	end

	VC.UpdateUsersAnimations(self,dt);

	VC.PlayMiscSounds(self,fCarSpeed,dt);
end


--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:OnContactClient( player )

	if( player==_localplayer and self.PropertiesInstance.bUsable==0 ) then return end
	VC.OnContactClientT(self,player);
end


--////////////////////////////////////////////////////////////////////////////////////
--////////////////////////////////////////////////////////////////////////////////////
--// SERVER functions definitions
--////////////////////////////////////////////////////////////////////////////////////
```

```
--////////////////////////////////////////////////////////////////////////////////////

--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:InitServer()


	if( self.Properties.bHasRockets == 0) then
		self.ammoRL = 0;
--		self.Ammo["VehicleRocket"] = nil;
	end

	VC.InitSeats(self, FlyBuggy);
	-- init common stuff for client and server
	VC.InitFlyBuggyCommon(self);

	self:OnReset();

	self:InitWeapon();
end

-- called on the server when the player collides with the FlyBuggy
--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:OnContactServer( player )

	if( self.PropertiesInstance.bUsable==0 ) then return end
	VC.OnContactServerT(self,player);

end


--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:UpdateServer(dt)

	VC.UpdateEnteringLeaving( self, dt );
	VC.UpdateServerCommonT( self, dt );

	if(self.cnt.inwater==1) then
		self.fGroundtime = 0;
	else
		-- LUC eject the pilot and/or the gunner if the FlyBuggy is flipping upside down
		local angles = new(self:GetAngles());
		if(angles.x > 90 and angles.x <270) then
			-- release driver
			if ( self.driverT.entity ) then
				VC.ReleaseUser( self, self.driverT );
--				VC.AIDriver( self, 1);
			end
			-- release gunner
			if ( self.gunnerT and self.gunnerT.entity ) then
				VC.ReleaseUser( self, self.gunnerT );
			end
			-- release the passengers (..who wanna get on! deny the passengers, who wanna get on! wanna get
on! they wanna get on!...)
			local idx;
			for idx=1, self.passengerLimit do
				if( self.passengersTT[idx].entity ) then
					VC.ReleaseUser( self, self.passengersTT[idx] );
				end
			end
		end

		self.fGroundtime = self.fGroundtime + dt;
		if(self.fGroundtime > 5.5) then
			AI:Signal(0, 1, "ON_GROUND", self.id);
		end
	end



end


--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:OnEventServer( id, params)
```

```
	if (id == ScriptEvent_PhysicalizeOnDemand) then
		self:SetPhysicParams( PHYSICPARAM_FLAGS, {flags_mask=pef_pushable_by_players, flags=0} );
	end

end



--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:OnShutDown()

	VC.EveryoneOutForce(self);
	VC.RemovePieces(self);
end
--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:OnSave(stm)
	VC.SaveAmmo( self, stm );
	stm:WriteInt(self.fEngineHealth);
end
--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:OnLoad(stm)
	VC.LoadAmmo( self, stm );
	self.fEngineHealth = stm:ReadInt();
end
--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:OnWrite( stm )

end
--////////////////////////////////////////////////////////////////////////////////////
function FlyBuggy:OnRead( stm )

end
--////////////////////////////////////////////////////////////////////////////////////

-----------------------------------------------------------------------------------------------------------------------
--
--	to test-call reinf
function FlyBuggy:Event_Reinforcment( params )

	if(VC.FreeToUse( self )==0) then return end		-- can't use it - player is in
--printf( "signaling BRING_REINFORCMENT " );

	VC.AIDriver( self, 1 );
	AI:Signal(0, 1, "BRING_REINFORCMENT", self.id);
end
-----------------------------------------------------------------------------------------------------------------------
--
function FlyBuggy:Event_GoPath( params )

	if(VC.FreeToUse( self )==0) then return end		-- can't use it - player is in
--System:Log("\001  Humvee GoPath  ");

	self.curPathStep = self.Properties.pathstart-1;
	VC.AIDriver( self, 1 );
	AI:Signal(0, 1, "GO_PATH", self.id);
end
-----------------------------------------------------------------------------------------------------------------------
--
-- to test
function FlyBuggy:Event_StopAttack( params )

--System:Log("\001  Event_StopAttack  ");
	self.curPathStep = self.Properties.pathstart-1;
	AI:Signal(0, 1, "PLAYER_LEFT_VEHICLE", self.id);
end
-----------------------------------------------------------------------------------------------------------------------
--
function FlyBuggy:Event_GoPatrol( params )

	if(VC.FreeToUse( self )==0) then return end		-- can't use it - player is in
	self.curPathStep = self.Properties.pathstart-1;
	VC.AIDriver( self, 1 );
	AI:Signal(0, 1, "GO_PATROL", self.id);
end
```

```
--------------------------------------------------------------------------------------------------------------------------------
--
function FlyBuggy:Event_GoAttack( params )

--System:Log("\001  Humvee GoPath  ");

	if(VC.FreeToUse( self )==0) then return end		-- can't use it - player is in

	VC.AIDriver( self, 1 );
	AI:Signal(0, 1, "GO_ATTACK", self.id);

end
--------------------------------------------------------------------------------------------------------------------------------
--
-- to test
function FlyBuggy:Event_Load( params )

--	if(VC.FreeToUse( self )==0) then return end		-- can't use it - player is in

--	VC.AIDriver( self, 1 );
	self:LoadPeople();
end
--------------------------------------------------------------------------------------------------------------------------------
--
-- to test
function FlyBuggy:Event_Release( params )

	VC.AIDriver( self, 1 );
	self:UnloadPeople();
end
--------------------------------------------------------------------------------------------------------------------------------
--
--
function FlyBuggy:RadioChatter()
end
--------------------------------------------------------------------------------------------------------------------------------
--
--
function FlyBuggy:Event_KillTriger( params )


	if (self.driverT.entity and _localplayer) then
		if (self.driverT.entity ~= _localplayer) then
			self.cnt:SetVehicleEngineHealth(0);
			self:GotoState( "Dead" );
		end
	else
		self.cnt:SetVehicleEngineHealth(0);
		self:GotoState( "Dead" );
	end

--	self.fEngineHealth = 0;
--	VC.OnDamageServer(self,hit);

end
--------------------------------------------------------------------------------------------------------------------------------
--
--
function FlyBuggy:LoadPeople()

	if(VC.FreeToUse( self )==0) then return end		-- can't use it - player is in

System:Log("boat LoadPeople  ");

	if(self.driverT.entity and self.driverT.entity.ai) then
System:Log("boat LoadPeople  +++++ DRIVER IS IN ");
		AI:Signal(0, 1, "DRIVER_IN", self.id);
	end

	if( self.Properties.bSameGroupId == 1 ) then
		AI:Signal(SIGNALFILTER_GROUPONLY, 1, "wakeup", self.id);
		AI:Signal(SIGNALFILTER_GROUPONLY, 1, "SHARED_ENTER_ME_VEHICLE", self.id);
	else
		AI:Signal(SIGNALFILTER_ANYONEINCOMM, 1, "wakeup", self.id);
		AI:Signal(SIGNALFILTER_ANYONEINCOMM, 1, "SHARED_ENTER_ME_VEHICLE", self.id);
```

```
	end


--	AI:Signal(SIGNALFILTER_GROUPONLY, 1, "wakeup", self.id);
--	AI:Signal(SIGNALFILTER_GROUPONLY, 1, "SHARED_ENTER_ME_VEHICLE", self.id);
	self.dropState = 1;

end
----------------------------------------------------------------------------------------------------------------------
--
--
function FlyBuggy:UnloadPeople()

	VC.DropPeople( self ,1);
--	VC.ReleaseGunner(self);		--, 1);
--	VC.ReleaseDriver(self);		--, 1);

--	self:RemoveAIGunner( );
--	self:DriverOut( );

end
----------------------------------------------------------------------------------------------------------
--
--
function FlyBuggy:InitEntering( puppet, enterTable, destPos )

	local dir = DifferenceVectors( destPos, puppet:GetPos() );
	local dist = sqrt(LengthSqVector(dir));
	local angl = self:GetAngles();

	ConvertVectorToCameraAngles(angl, dir);
	puppet:SetAngles( angl );

	enterTable.Time = dist/self.entVel;
	enterTable.HT = enterTable.Time*.5;
	enterTable.HK = dir.z/enterTable.Time;
	enterTable.HO = puppet:GetPos().z;
	enterTable.HS = 10/enterTable.Time;

	puppet:ActivatePhysics(0);

	puppet.cnt.AnimationSystemEnabled = 0;
	enterTable.TimePast = 0;
	puppet:StartAnimation(0,"jump_air");			-- to have this as previous animation
	puppet:StartAnimation(0,"jump_start");

end
----------------------------------------------------------------------------------------------------------
--
--
function FlyBuggy:DoEnter( puppet )

	if( puppet == self.driverT.entity ) then		-- driver
		VC.AddUserT( self, self.driverT );
		VC.InitEnteringJump( self, self.driverT );
	elseif( puppet == self.gunnerT.entity ) then			-- gunner
		VC.AddUserT( self, self.gunnerT );
		VC.InitEnteringJump( self, self.gunnerT );
	end
end
----------------------------------------------------------------------------------------------------------
--
--
function FlyBuggy:AddDriver( puppet )

	if (self.driverT.entity ~= nil)		then	-- already have a driver
		do return 0 end
	end

	if(self.Properties.sDriverName ~= "" and self.Properties.sDriverName ~= puppet:GetName()) then return 0 end

	self.driverT.entity = puppet;
	if( VC.InitApproach( self, self.driverT )==0 ) then
		self:DoEnter( puppet );
	end
```

```
		do return 1 end
end
--------------------------------------------------------------------------------------------------------
--
--
function FlyBuggy:AddGunner( puppet )


		if (self.gunnerT.entity ~= nil)		then	-- already have a driver
			do return 0 end
		end

		self.gunnerT.entity = puppet;
		if( VC.InitApproach( self, self.gunnerT )==0 ) then
			self:DoEnter( puppet );
		end
		do return 1 end
end
--------------------------------------------------------------------------------------------------------
--
--
function FlyBuggy:AddPassenger( player )
		return 0
end
--------------------------------------------------------------------------------------------------------
--

function FlyBuggy:Event_AddPlayer( params )

		if(_localplayer.theVehicle) then return end		-- this player is already in some (this) vehicle

		local theTable = VC.GetAvailablePosition(self);

		if(theTable == nil) then return end

		_localplayer.cnt.use_pressed = nil;
		theTable.entity = _localplayer;
		VC.AddUserT(self, theTable);



end
--////////////////////////////////////////////////////////////////////////////////
--------------------------------------------------------------------------------------------------------
--
-- to test
function FlyBuggy:Event_TArgetOnLand( params )

		AI:Signal(0, 1, "PLAYER_LEFT_VEHICLE", self.id);

end
--------------------------------------------------------------------------------------------------------
--
-- to test
--function FlyBuggy:Event_GunnerOut( params )
--
--		AI:Signal(0, 1, "GUNNER_OUT", self.id);
--
--end
--------------------------------------------------------------------------------------------------------
--
function FlyBuggy:Event_DriverIn( params )

		BroadcastEvent( self,"DriverIn" );

end
-----------------------------------------------------
function FlyBuggy:Event_Unhide()

		self:Hide(0);

end
-----------------------------------------------------
function FlyBuggy:Event_Hide()
```

```
	self:Hide(1);

end
---------------------------------------------------------------------------------------------
--
--
function FlyBuggy:Event_Activate( params )

	if(self.bExploded == 1) then return end

	self:GotoState( "Alive" );
end
---------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
-- empty function to get reed of script error - it's called from behavours
function FlyBuggy:MakeAlerted()
end
---------------------------------------------------------------------------------------------
--
function FlyBuggy:InitWeapon()

--do return end
	-- //m_MinVAngle, m_MaxVAngle, m_MinHAngle, m_MaxHAngle
	self.cnt:SetWeaponLimits(-60, 10,-120,120);

	if( self.Properties.bUseRL == 1 ) then
		if( self.Properties.bUseRLguided == 1 ) then
			self.cnt:SetWeaponName("VehicleMountedRocketMG", "COVERRL");
		else
			self.cnt:SetWeaponName("VehicleMountedRocketMG", "VehicleMountedRocket");
		end
	else
		self.cnt:SetWeaponName("VehicleMountedRocketMG", "VehicleMountedMG");
	end

	VC.InitAutoWeapon( self );

	self.driverShooting = 1;
end
---------------------------------------------------------------------------------------------------------------------
--
--this is called from vehicleProxy when all the saving is done
function FlyBuggy:OnSaveOverall(stm)

	VC.SaveCommon( self, stm );

end
---------------------------------------------------------------------------------------------------------------------
--
--this is called from vehicleProxy when all the loading is done and all the entities are spawn
function FlyBuggy:OnLoadOverall(stm)

	VC.LoadCommon( self, stm );

end
------------------------------------------------------------
```

-- END COPY HERE

Sandbox

Personally, I prefer this method of opening Sandbox with a MOD, although there are other ways.

Copy your Sandbox Icon and rename it to SandBox FLYMOD

Right Click it and put the following at the END of the target line D:\Games\FarcrySDK\Bin32\Editor.exe **-MOD:FLYMOD**

Now Click OK and open your new FLYMOD Sandbox icon.

Create a new landscape (1024x1024), texture it and create a flat piece of terrain within your map.

When you go to Entities now, under vehicles, you should have a new vehicle type called FlyBuggy.

Drag and drop this onto your map. Start the game (CTRL+G)

Jump in and fly.

The highlighted bits in the script are variables that you can change to get a different feel. To notice the change, you will have to Reload the script of the vehicle by highlighting the buggy and click on the Reload Script button.

If this doesn't work first time, save the map and then reload it into Sandbox.

That's it

Clivey.