How to make shaped charges/Directionnal gibbing in CC

By Mehman

First of all: you don't need any lua if you use this technique.

I will base this tutorial on a mod I made some time ago: "Weapons of WWII" because some of its weapons use shaped charges/directional gibbing. This should work for Bombs, mines, rockets, bullets, missiles, eggs, or whatever you want.

To make directional gibbing you will need a weapon that fires an MOSRotating(Or AEmitter if you want it to be a rocket).

This projectile will then gib into an AEmitter that will emit the "gibs".

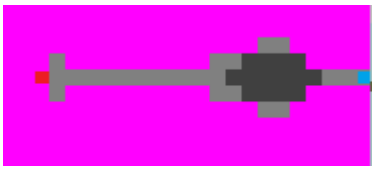
The projectile (An AEmitter here)'s ini-code should look like this:

```
AddAmmo = AEmitter //It can be an MOSrotating
        PresetName = Panzerschreck Rocket HEAT  //Or whatever you want
        Mass = 4
        GlobalAccScalar = 0.1
        RestThreshold = -500
        HitsMOs = 1
        GetsHitByMOs = 0
        ScriptPath = Ballistics.rte/Devices/Weapons/lua/HEAT.lua
        SpriteOffset = Vector //I will explain this later
                X = -3
                Y = -4
// I deleted some lines here, if you want a complete template for a rocket, go read the vanilla rocket
//launcher's ini file.
        GibSound = Sound
                AddSample = ContentFile
                        Path = Ballistics.rte/Devices/Sounds/Rocket-ex.wav
```

```
GibImpulseLimit = 25 //should be very low to gib on impact
AddGib = Gib
        GibParticle = AEmitter //This will make the actual gibbing
                CopyOf = Panzerschreck Emitter HEAT
        Count = 1
        Spread = 3.14
        MaxVelocity = 0 // Otherwise the direction of your gibs will be wrong
        MinVelocity = 0 // Otherwise the direction of your gibs will be wrong
        InheritsVel = 1 // Very important to orient the emitter correctly
```

The sprite of this projectile is very important: the emitter will appear at its offset, but it will gib when any of its pixels are touched:

```
SpriteOffset = Vector
        X = -3
        Y = -4
```



The red pixel is the SpriteOffset, and the red one is the pixel that will make the projectile gib. It is very important that they are sufficiently separated from one another because the emitter will need some time to activate, and we don't want it to hit something and bounce off. That time won't be seen by the player (a few ms).

Now I will explain the AEmitter part:

To work the AEmitter code must be placed before the projectile's code in the ini file.

```
AddEffect = AEmitter
        PresetName = Panzerschreck Emitter HEAT
        SpriteOffset = Vector
                X = -2
                Y = -2
        Mass = 0.4
        HitsMOs = 0
        LifeTime = 1 //1 ms, the duration of the emission, here we want an explosion, so the shorter
        //the better
        GetsHitByMOs = 0
        OrientToVel = 0 //That way it will keep the projectile's orientation
        GlobalAccScalar = 0.0 //We don't want gravity to interfere with our plans do we?
        SpriteFile = ContentFile
                FilePath = Ballistics.rte/Devices/Sprites/Px.bmp //Should be invisible
        AddEmission = Emission
                EmittedParticle = MOPixel  //Here you add your "gibs"
                        CopyOf = Thermite Fragment4
                ParticlesPerMinute = 1
                BurstSize = 125 //The number of "gibs" that will be emitted
                Spread = 0.01 //The spread, in rad(Pi~=3.14~=180°right and left from the
orientation=every direction)
                MaxVelocity = 125 //The maximum velocity of this "gib"
                MinVelocity = 75 //The minimum velocity of this "gib"
```

//The velocity of every “gib” will be randomly chosen between MinVelocity and MaxVelocity

PushesEmitter = 0 //0 because we don’t want our emitter to move from its trajectory
AddEmission = Emission //Another “gib”
EmittedParticle = MOPixel
CopyOf = Thermite Fragment2
ParticlesPerMinute = 1
BurstSize = 25
Spread = 1.2 //With a larger spread this time
MaxVelocity = 50
MinVelocity = 30
PushesEmitter = 0

//They can be as many emitted objects (“Gibs”) as you like.

EmissionEnabled = 1 //Do we want it to emit or not (1=yes, 0=no)
EmissionsIgnoreThis = 1 //That way the emitted “gibs” won’t destroy the emitter.
BurstScale = 1 //By how many do we want to multiply the BurstSize and Velocity, 1 is simpler: //you can choose whatever you want above so don’ bother multiplying it by something other than 1
BurstDamage = 0 //How much damage will each burst cause to the emitter’s parent(only //used for wounds)
BurstTriggered = 1 //It needs to be set to 1 because we want all the “gibs” to be emitted at the same time, 0 would be used for a rocket engine (1 could be too)
EmissionDamage = 0 //How much damage will each emission cause to the emitter’s //parent (only used for wounds)

Flash = None
FlashOnlyOnBurst = 0

The most important part of this is the spread: it is what will actually make the directional gibbing.

Now we’re almost done: all we need to do is make a round a magazine and a gun. The first part (the projectile) is what you will need to put in you round.

Then have fun!

The mod this tutorial was based on can be downloaded here:
http://forums.datarealms.com/viewtopic.php?f=61&t=19433