TNT PARTICLES ENGINE

FREE. EASY. FAST.

TNT Particles Engine v1.11

By Gianluca D'Angelo

TNT Particles Engine it's free and you can use in free and commercial games, but if you like it and want to see updates, new features, speed optimizations and more please support me and make a free donation. thanks.

Gianluca D'Angelo (GregBUG)
you can contact me at
gregbug@gmail.com or
support@tntparticlesengine.com (preferred)

for bug, report, suggestion, optimization or simple questions please ask in the forum.
www.tntparticlesengine.com

special thanks to Scouser and Paolo Manna for their help and bug fix.

L'Aquila (Italy), 11 Marzo 2011 - Gianluca D'Angelo.
TNT Particles Engine is dedicated
to my Wife (Cinzia) and my son (Mattia). i Love you.

What is TNT Particles Engine ?

a particles engine allows developers to add rich and realistic visual effect to their games, like explosions, smoke, fog, rain, snow and many many more effects (the limit is only your imagination) TNT Particles Engine is a fast, easy and lightweight engine ready to be added to your games.

Oh no !!! another particles Engine and what are the features of TNT Particles Engine ?

- *Developed with and for Gideros Studio 2012.2.1 (the best mobile SDK out there!) (visit www.giderosmobile.com)*
- *It's free, easy and fast!*
- *Unlimited user definable emitters*
- *Unlimited user definable particles*
- *every emitter can have unlimited particles assigned*
- *every emitter can have their position, direction can be hidden, paused, stopped, restarted at any time*
- *every particles can have they gravity, direction, speed, rotations, color, size, transparency and many more properties defined by user.*
- *every particles definitions can have their Color, Speed, Size, Alpha and Direction "In and Out" Morphing (values variations defined in time)*
- *every Morphing effect can be activated or deactivated at any time by the user*
- *and many more to come**...***

How to use this engine ?

The philosophy of TNT Particles Engine (TNTPE) is that at the base of every particles effect there is an emitter. The emitter define the point from where the particle comes out, for this propose the engine define a class called CEmitter.

Every emitter can have unlimited particles definitions associated, but a particles definition can be associated only at one emitter.

Ps: all functions parameters inside [ ] are optionals.

CEmitter Class

- **function CEmitter:init(xPos, yPos, direction, parentGroup)**

Create new particle emitter at position "xPos, yPos" with angle "direction" as child of "parentGroup".

Example:

local newEmitter = CEmitter.new(160, 320, 90, stage)

- **function CEmitter:free()**

free all memory allocated by emitter and all particles associated.

Emample:

newEmitter:free()

newEmitter = nil -- remember to nil all reference variables !!!

- **function CEmitter:getVisibleParticles()**

return current visible particles

Example:

local countParticles = newEmitter:getVisibleParticles()

- **function CEmitter:setPosition(xPos, yPos)**

set emitter position at "xPos, yPos"

Example:

newEmitter:setPosition(160, 240)

- **function CEmitter:getPosition()**

return current xPos, yPos position.

Example:

local x, y = newEmitter:getPosition()

- **function CEmitter:setDirection(direction)**

set current emitter direction in degree (0, 360)

Example:

newEmitter:setDirection()

- **function CEmitter:getDirection()**

return current emitter direction in degree (0, 360)

Example:

Local emitterDirection = newEmitter:getDirection()

- **function CEmitter:setParent(parentGroup)**

set new parent group of emitter

Example:

newEmitter:setParent(spriteGroup)

- **function CEmitter:getPartent()**

get parent group of emitter

Example:

Local parent = newEmitter:getParent()

- ## function CEmitter:start([startHidden], [forceStart])

start emitter and all related particles.

If "startHidden" is true then emitter start but start hidden.

If "forceStart" is true emitters "restart" even if emitter is already started an running.

Raise event "EMITTER_START". (for how to use event see demo 5)

Example:

newEmitter:start()

- ## function CEmitter:stop()

stop emitter and all related particles.

Raise event "EMITTER_STOPPED". (for how to use event see demo 5)

Example:

newEmitter:stop()

- ## function CEmitter:pause(mode)

pause emitter and all related particles. If "mode" is true then emitter pause and raise event "EMITTER_PAUSED"; if mode is false emitter is unpaused and raise event "EMITTER_UNPAUSED"

Example:

newEmitter:start()

- ## function CEmitter:hide(mode)

hide emitter if "mode" is true and raise event "EMITTER_HIDDEN" else unhide and raise event "EMITTER_UNHIDDEN".

Example:

local emitterIsHidden = newEmitter:isHidden()

- ## function CEmitter:isHidden()

return current hide status (true if emitter is hidden, false if not)

Example:

local emitterIsHidden = newEmitter:isHidden()

- ## function CEmitter:paused()

return current pause state (true if emitter is paused, false if not)

Example:

local emitterInPause = newEmitter:paused()

- ## function CEmitter:started()

return if emitter is sarted (and running) or not (true if emitter is started, false if not)

Example:

local isStarted = newEmitter:started()

- ## function CEmitter:assignParticles(particlesObject)

assign a user defined particle to emitter.

ATTENTION: you can't assign the same particles definition to multiple emitters.

Example:

newEmitter:assignParticles(particle_example_1)

CParticles Class

An emitter to work (emit some particles) need that the user define particles and associate to it.

TNT Engine use CParticles Class to manage user defined particles.

Every particles definition can be associated only at one emitter (but one emitter can have multiple particles associated).

function CParticles:init(particlesSprite, particlesMax, particlesMaxLife, particlesMaxStartupDelay, blendingMode)

Create new particle definition
"particlesSprite" sprite texture used for the particles
"particlesMax" max particles alloewed (do not exagerate!)
"particlesMaxLife" max duration of particle
"particlesSprite" max startup delay for particles to start
"particlesSprite" gideros blending mode
(for 2012.2.1 valid mode are: "alpha", "noAlpha", "add", "multiply", "screen")

Example:

```
local sparklesGFX = (Texture.new("exp1.png"))
sparkles = CParticles.new(sparklesGFX, 20, 1, 0.5, "add")
```

function CParticles:free()

Free particles definition. If you call CEmitter.free() do not use this function.

function CParticles:setMaxLife(maxLife)

set max life (in seconds) of particles;

function CParticles:getMaxLife()

get current max life (in seconds) of particles;

function CParticles:setMaxDelay(maxDelay)

set max born delay of particles

function CParticles:getMaxDelay()

get current max born delay of particles

function CParticles:setSpeed(speed, [speedMax])

define speed of particles;

"speed" speed of particles

"speedMax" max speed of particles (optional)

if defined particles speed are random from speed to speedMax

Example:

particles:setSpeed(100, 550) – set speed of particles random from 100 to 550

function CParticles:setParticlesOffset(x,y)

set particles offset from emitter position

Example:

particles:setParticlesOffset(100, 550) – set x, y offset of particles

function CParticles:setMoveXY(x,y)

move position of particles at coordinate x,y

Example:

particles:setMoveXY(100, 550) – Move particles at x, y coordinates

function CParticles:setDisplacement(width, height)

define particles displacement

Example:

particles:setDisplacement(320, 480) – set displacement for all screen size (if screen size is 320x480)

function CParticles:setSize(size, [sizeMax])

set size of particles

"size" 1 is original size (2 is double size and so on)

"sizeMax" is max size of particles. If definited size is random form "size" to "sizeMax"

Example:

particles:setSize(100, 550) – set size of particles from 1 to 4

function CParticles:setDirection(direction, [directionMax])

set direction of particles (angle defined in degree 0-360)

"direction" is direction (angle) of particles

"directionMax" is max direction of particles. If definited direction is random from "direction" to "directionMax"

Example:

particles:setDirection(0, 360) – set direction of particles random from 0 to 360 (angle)

function CParticles:setRotation(rotation, velocity, [rotationMax], [velocityMax])

set particles sprite rotation and it's rotation speed (velocity)

"rotation" set initial rotation of particles (angle)

"velocity" set initial velocity (rotation speed) of particles (angle)

"rotationMax" set max rotation of particles (angle). If defined rotation is random from "rotation" to "rotationMax"

"velocityMax" set max velocity (rotation speed) of particles (angle). If defined velocity is random form "velocity" to "velocityMax"

function CParticles:setAlpha(alpha, [alphaMAX])

set particles alpha value form 0 (transparent) to 255 full visible.

"alpha" set alpha value of particles

"alphaMax" set max alpha value of particles. If set alpha is random from "alpha" to "alphaMax"

Example:

particles:setAlpha(0, 255) – set alpha of particles random from 0 to 255

function CParticles:setColor(r,g,b, [rMax], [gMax], [bMax])

set particles color

"r" set red color value of particles (0-255)

"g" set green color value of particles (0-255)

"b" set blue color value of particles (0-255)

"rMax" (optional) set max red color value of particles (0-255).
If set red color is random from "r" to "rMax"

"gMax" (optional) set max green color value of particles (0-255)
If set green color is random from "g" to "gMax"

"bMax" (optional) set max blue color value of particles (0-255)
If set blue color is random from "b" to "bMax"

Example:

particles:setColor(200, 100, 100) – set particles color (the same for all particles)
particles:setColor(50, 50, 50, 255, 255, 255) – set random particles colors

function CParticles:setGravity(gravityX, gravityY)

set x and y gravity force applied to particles

"x" x gravity force

"y" y gravity force

function CParticles:setGravityForce(gravityX, gravityY)

set x and y initial gravity forces for the particles (thanks to Scouser)

"x" x force

"y" y force

function CParticles:setLoopMode(loopMode)

set particles loop mode (default is infinite)

"loopMode" = 0 infinite loop

"loopMode" > 0 loop n time the particles. N = loopMode

Example:

particles:setLoopMode (0) – loop mode is infinite
particles:setLoopMode (1) – loop 1 time only then stop particles emission
particles:setLoopMode (3) – loop 3 times then stop particles emission

Particles Morphing Functions

Particles morphing functions are special modifier functions used to manipulate and modify particles at run time.

Morphing function can modify particles properties from a value to another in a specified fixed or variabile time; thanks to this functions we can create a very wide range of particles effects.

In this version of TNT Particles engine there are five special morphing functions supported (hope more in future) and they are:

- *Alpha Morphing In/Out*
- *Size Morphing In/Out*
- *Color Morphing In/Out*
- *Speed Morphing In/Out*
- *Direction Morphing In/Out*

Every morphing function can be enabled or disabled at any time by user and the general parameters syntax is:

Morphing function (targetValue, targetime, [targetValueMax], [targetTimeMax])
(remember that parameters inside [ ] are optional)

*when optionals **"tagetValueMax"** or **"targetTimeMax"** parameters are specified the the values are random generated from targetValue to targetValueMax;*

for Example:

MorphingFunction(2,3, 8,10)

targetValue is random generated from 2 to 8
targetTimeMax is random generated from 3 to 10

Before analyze every morphing function in details we must say that every morphing function have In and Out definition.

*When we define **"In" morphing functions**, time values are referred from the start of particles life*

Example:
function setAlphaMorphIn(100, 3)
(where 100 is targetValue and 3 is targetTime)

Every particles ″morph″ (change) from it′s current alpha value to 100 in 3 seconds from when the particle is ″born″ (drawn into the screen).

function setAlphaMorphIn(100, 3, 200, 6)
(where 100 is targetValue and 3 is targetTime, 200 is targetValueMax and 6 is targetTimeMax)

Every particles ″morph″ (change) from it′s current alpha value to a random alpha from 100 to 200 in random time from 3 to 6 seconds from when the particle is ″born″ (drawn into the screen).

When we define ***″Out″ morphing functions****, time values are referred from the end of particles life:*

Example:
function setAlphaMorphOut(0, 2)
(where 0 is targetValue and 2 is targetTime)

Every particles ″morph″ (change) from it′s current alpha value to 0 in 2 seconds before the particle is ″killed″ by the engine (erased from the screen).

function setAlphaMorphIn(20, 1, 60, 2)
(where 20 is targetValue and 1 is targetTime, 60 is targetValueMax and 2 is targetTimeMax)

Every particles ″morph″ (change) from it′s current alpha value to a random alpha from 20 to 60 in random time from 1 to 2 seconds before the particle is ″killed″ by the engine (erased from the screen).

Now we can analyze the morphing functions in details.

Alpha Morphing ″In″ and ″Out″

Alpha morphing effect is used for modify the ″transparency″ value of particles
0 is full transparent 255 full color..

function CParticles:setAlphaMorphIn(targetAlpha, targetTime, [targetAlphaMax], [targetTimeMax])
define "In" alpha morphing
function CParticles:setAlphaMorphOut(targetAlpha, targetTime, [targetAlphaMax], [targetTimeMax])
define "Out" alpha morphing

RunTime Enabler/Disabler Alpha Morphing Effect

function CParticles:setEnbaleAlphaMorph(EnableMorphIn, EnableMorphOut)
"EnableMorphIn"   if true enable "in" alpha morphing
"EnableMorphOut" if true enable "out" alpha morphing
function CParticles:setEnbaleAlphaMorphIn(EnableMorphIn)
"EnableMorphIn"   if true enable "in" alpha morphing
function CParticles:setEnbaleAlphaMorphOut(EnableMorphOut)
"EnableMorphOut" if true enable "out" alpha morphing

Alpha Morphing Getters

function CParticles:getAlphaMorphEnabled()
return (true or false) if alpha morphing is enabled (in and out)
example: local in, out = particles:getAlphaMorphEnabled()
function CParticles:getAlphaMorphInEnabled()
return (true or false) if alpha morphing "in" is enabled
function CParticles:getAlphaMorphOutEnabled()
return (true or false) if alpha morphing "out" is enabled

Size Morphing "In" and "Out"

Size morphing effect is used for modify the "Size" value of particles
1 is original size, 2 is double size and so on. (ex: 0.5 is half size).

function CParticles:setSizeMorphIn(targetSize, targetTime, [targetSizeMax], [targetTimeMax])
define "In" size morphing
***function CParticles:setSizeMorphOut(targetSize, targetTime*, [targetSizeMax], [targetTimeMax])**
define "Out" size morphing

RunTime Enabler/Disabler Size Morphing Effect

function CParticles:setEnableSizeMorph(EnableMorphIn, EnableMorphOut)
"EnableMorphIn"   if true enable "in" size morphing
"EnableMorphOut" if true enable "out" size morphing
function CParticles:setEnableSizeMorphIn(EnableMorphIn)
"EnableMorphIn"   if true enable "in" size morphing
function CParticles:setEnableSizeMorphOut(EnableMorphOut)
"EnableMorphOut" if true enable "out" size morphing

Size Morphing Getters

function CParticles:getSizeMorphEnabled()
return (true or false) if size morphing is enabled (in and out)
example: local in, out = particles:getSizeMorphEnabled()
function CParticles:getSizeMorphInEnabled()
return (true or false) if size morphing "in" is enabled
function CParticles:getSizeMorphOutEnabled()
return (true or false) if size morphing "out" is enabled

Speed Morphing "In" and "Out"

Speed morphing effect is used for modify the "Speed" values of particles.

function CParticles:setSpeedMorphIn(targetSpeed, targetTime, [targetSpeedMax], [targetTime])
 define "In" speed morphing
function CParticles:setSpeedMorphOut (targetSpeed, targetTime, [targetSpeedMax], [targetTime])
 define "Out" speed morphing

RunTime Enabler/Disabler Speed Morphing Effect

function CParticles:setEnableSpeedMorph(EnableMorphIn, EnableMorphOut)
 "EnableMorphIn" if true enable "in" speed morphing
 "EnableMorphOut" if true enable "out" Speed morphing
function CParticles:setEnableSpeedMorphIn(EnableMorphIn)
 "EnableMorphIn" if true enable "in" speed morphing
function CParticles:setEnableSpeedMorphOut(EnableMorphOut)
 "EnableMorphOut" if true enable "in" speed morphing

Speed Morphing Getters

function CParticles:getSpeedMorphEnabled()
 return (true or false) if speed morphing is enabled (in and out)
 example: local in, out = particles:getSpeedMorphEnabled()
function CParticles:getSpeedMorphInEnabled()
 return (true or false) if Speed morphing "in" is enabled
function CParticles:getSpeedMorphOutEnabled()
 return (true or false) if Speed morphing "out" is enabled

Direction Morphing "In" and "Out"

Direction morphing effect is used for modify the "direction" values of particles.

function CParticles:setDirectionMorphIn(targetAngle, targetTime, [targetAngleMax], [targetTimeMax])
 define "In" direction morphing
function CParticles:setDirectionMorphOut(targetAngle, targetTime, [targetAngleMax], [targetTimeMax])
 define "Out" direction morphing

RunTime Enabler/Disabler direction Morphing Effect

function CParticles:setEnableDirectionMorph(EnableMorphIn, EnableMorphOut)
 "EnableMorphIn" if true enable "in" direction morphing
 "EnableMorphOut" if true enable "out" direction morphing
function CParticles:setDirectionSpeedMorphIn(EnableMorphIn)
 "EnableMorphIn" if true enable "in" direction morphing
function CParticles:setDirectionSpeedMorphOut(EnableMorphOut)
 "EnableMorphOut" if true enable "in" direction morphing

Speed Morphing Getters

function CParticles:getDirectionMorphEnabled()
 return (true or false) if direction morphing is enabled (in and out)
 example: local in, out = particles:getDirectionMorphEnabled()
function CParticles:getDirectionMorphInEnabled()
 return (true or false) if direction morphing "in" is enabled
function CParticles:getDirectionMorphOutEnabled()
 return (true or false) if direction morphing "out" is enabled

Color Morphing “In” and “Out”

Color morphing effect is used for modify the “color” values of particles.

function CParticles:setColorMorphIn(rTarget, gTarget, bTarget, targetTime, [rTargetMax], [gTargetMax], [bTargetMax], [targetTimeMax])

define “In” color morphing
“rTarget” is red value of particle (0-255)
“gTarget” is green value of particle (0-255)
“bTarget” is blue value of particle (0-255)

function CParticles:setColorMorphOut(rTarget, gTarget, bTarget, targetTime, [rTargetMax], [gTargetMax], [bTargetMax], [targetTimeMax])

define “Out” color morphing
“rTarget” is red value of particle (0-255)
“gTarget” is green value of particle (0-255)
“bTarget” is blue value of particle (0-255)

RunTime Enabler/Disabler color Morphing Effect

function CParticles:setEnableColorMorph(EnableMorphIn, EnableMorphOut)

“EnableMorphIn” if true enable “in” color morphing
“EnableMorphOut” if true enable “out” color morphing

function CParticles:setColorSpeedMorphIn(EnableMorphIn)

“EnableMorphIn” if true enable “in” color morphing

function CParticles:setColorSpeedMorphOut(EnableMorphOut)

“EnableMorphOut” if true enable “in” color morphing

color Morphing Getters

function CParticles:getColorMorphEnabled()

return (true or false) if color morphing is enabled (in and out)
example: local in, out = particles:getColorMorphEnabled()

function CParticles:getColorMorphInEnabled()

return (true or false) if color morphing “in” is enabled

function CParticles:getColorMorphOutEnabled()

return (true or false) if color morphing “out” is enabled

Release HISTORY

v1.11 (April, 3 2012)

! fixed AlphaMorphIn Bug (thanks to Paolo Manna)
+ function CParticles:setGravityForce(gravityX, gravitY)
new function to define initial gravity force. (thanks to Scouser)

v1.10 (March, 24 2012)

! ***setLoopMode – bug correction. Now work as expected! ;)***
! setDirectionMorphIn and Out - bug correction. Now work as expected! ;)
! *Some other little bugfix and optimizations.*

+ ***function CEmitter:start([startHidden], [forceStart])***
Added optional parameter [forceStart] to force restart emitter even if emitter is playing.
+ ***added function CParticles:setMaxLife(maxLife)***
+ ***added function CParticles:getMaxLife()***
+ ***added function CParticles:SetMaxDelay(maxDelay)***
+ ***added function CParticles:GetmaxDelay()***
+ function CParticles:setColor and setRandomColors joined together to simplify use and better colors control.
Now exists only with optional parametres setColor(r,g,b, [rMax], [gMax], [bMax])
+ ***Now all morphing function have the same interface and offer major control over values and time.***

v1.00 (March, 11 2012) - First Public Release