Vehicle Model Creation Guide

Version 2009.11.22

Graviteam ®

Table of Contents

1 GENERAL INFORMATION ABOUT VEHICLE MODELS

Each vehicle model which is inserted into the game must contain: 3 visual levels of detail (which are displayed), 1 physical level which is used to check a visibility, armor penetration and damage designs as well as different aspects of physical modeling.

Additionally, 5 texture maps need to be created: 24-bits/color, 1-bit transparency, 8-bit roughness (bump-map), shininess, and bumpiness, each per 8-bit. Overlay of these maps (mapping) is performed using the same texture coordinates. It is also necessary for the physical modeling to create an armor map.

The received levels of detail and physical level must be converted into X-format. See Section 2.1 for more details on Export.

The created textures must be combined into two 32-bit texture maps and converted into DDS format (See Section 2.2). The armor map needs to be saved in TGA format (See Section 1.3).

1.1 Requirements to Models

Each vehicle model must consist of 1 physical and 3 visual levels of detail. A number of triangles in the visual levels must not exceed 10, 5, 1.5 thousand. The physical level must have no more than 2-2.5 thousand of triangles.

Each level must contain the texture coordinates and normals. The number of fragments (chunks) in the visual levels must not exceed 62, all the chunks must be combined into hierarchy which begins with a single chunk to which the others must be attached directly or through the hierarchy.

A failure of any chunk or chunks at the younger levels (with less number of triangulars) is allowed, but an appearance of new ones is not allowed. The physical level is considered as a base (initial) level of detail. All the chunks presented in the model must be at this level. The chunks not having geometry (dummy chunks) are not allowed to be presented at the visual levels.

The name of each chunk that is displayed must begin with a prefix "d_" and contain only Latin letters, figures or underline character. The name length must not exceed 30 characters with account of the prefix.

The dummy chunks must be at the physical level only, begin with the prefix "s_", a requirement to the name is like a previous type of the chunks. This type of chunks are used to assign anchor points of different objects on the vehicle, to assign observation and illumination devices, points of shell trajectory, pivots, exhaust pipes, seats for crew and assault, etc.

The chunks that assign the vehicle’s internal aggregates used for the physical modeling must have a simple geometry (they can have no correct texture coordinates) that presents a form and position of an aggregate in simplified form. The name begins with the prefix "p_". The requirements to the name are is like a previous type of the chunks. These chunks exist at the physical level only.

The recommended names of the chunks for different parts of the vehicle are listed in the Table 1.1. If there are several parts of the same type, a suffix "_xx" is added to the name, where xx is a part number. For instance, for wheels: d_wheel_01, d_wheel_02, d_whell_03. The wheels need to be numbered sequentially starting with the first front one attached to the port side of the vehicle (if one sees in direction of the vehicle motion).

Table 1.1

Recommended Names of the Vehicle Parts

Name	Position in Hierarchy	Part Type
d_hull	Main part, with which the hierarchy begins	hull

d_head	d_hull	turret
d_gunmask	d_head	gunmask
d_gun	d_gunmask	main gun
d_mgun	d_gunmask	machinegun
d_hatch	d_hull, d_head	hatch
d_light	d_hull	light
d_wheel	d_hull	wheel
d_caret	d_hull	caret
d_shield	d_rama	gun shield
d_rama	d_hull	gun frame
d_stanina	d_hull	gun mount
s_light	d_light	light source position
d_sight	d_head, d_hull	sight
s_sight	d_sight, d_head, d_hull	point of sight and observation device
s_gun, s_mgun	d_gun, d_mgun	points of shell and bullet trajectory
s_trace	d_hull	point of trace
s_mark	d_head, d_hull	point of marking signs and captions
s_smoke	d_hull	smoke point
s_waterline	d_hull	waterline
d_comhead	d_head	commander’s turret
s_fire	d_head, d_hull	fire point
s_gunner, s_driver, ….	d_hull, d_head	crew position
s_pass	d_hull, d_head	position of assault and passengers of vehicle
s_driver_out, s_gunner_out, …	d_hull, d_head	travelling position of crew
s_driver_exit, s_gunner_exit…	d_hull	points of crew and passenger exits

The most of the chunks are usually connected to the hull (wheels, carets, stanins, and turret) and turret (gun mantlet, hatches).

It is not required to model tracks for track vehicles; they are stretched automatically. Do this requires to create 1 track and set up the whole track in configuration file "common_res_mod" in the section tracks.

The vehicle model must not have animations and pivots (for turret, gun mantlet, hatches, etc.) must be correctly placed.

The models must be in the metric system; a unit of measure is a meter. A center of coordinates must comply with the model's centre of gravity (approximately). The axis x of the pivot of the chunks meant for marking captions and signs must be set on the normal from the supposed point of the sign marked and attached to that part of the vehicle, on which this sign is marked.
The names of internal aggregates of the vehicle are given in the table 1.2.

Table 1.2

Recommended Names of the Internal Aggregates of the Vehicle

Name	Position in the Hierarchy	Type of Part
p_accum	d_hull	accumulator
p_driver	d_hull	driver (only for closed area)
p_gunner	d_head	gunner (only for closed area)
p_loader	d_head	loader (only for closed area)
p_engine		engine
p_transm		transmission
p_rothead		turret rotation mechanisms
p_rotgun		gun lifting mechanisms
p_fueltank		fueltanks
p_ammo		ammo
p_radio		radio

1.2 Requirements to Textures

To display the vehicle correctly two 32-bit texture maps are required that have a suffix "_dift" and "_norsp", each channel of which contain different information on the vehicle cover material. The required textures and their layout under channels are given in the table 1.3. The texture name is formed as follows: reg_<name>_dift and bump_<name>_norsp.

Table 1.3

Textures and their layout under channels

Map	Channel	Bit	Purpose
dift	RGB	24	color
dift	A	1	transparency contains only points of two colors: black for transparent parts of texture and white for nontransparent ones.
norsp	RG	8	roughnesses – light color means convexes, dark color – concaves; it takes 2 channels after transformation to normal map.
norsp	B	8	patch of light is a surface reflectance; the more light the point is, the lighter the point is, the stronger the reflection is.
norsp	A	8	surface roughness assigns a form and size of a patch of light

The textures must be square; their sizes must be divisible by 2. The size 1024x1024 pixels is recommended for outsizal equipment and 512x512 – for small-sized equipment (mortar launchers, machinegun).

A transformation from roughness map to normal map can be made through:

- nVidia® plugin for Adobe® PhotoShop® http://developer.nvidia.com/object/photoshop_dds_plugins.html.
- Blender http://wiki.blender.org/index.php/Doc:Manual/Textures/Maps/Bump_and_Normal_Maps
- ATI/AMD ® plugin http://ati.amd.com/developer/sdk/radeonsdk/html/tools/toolsplugins.html

And copy red and green channels of the texture into the appropriate map.

All the textures must contain a complete set of MIP levels.

1.3 Armor Map

The armor map represents a grey graduated texture (but actually 32-bit texture, RGB channels of which are complied), its size is 256x256 or less, in each point of which the armor thickness is indicated. The texture coordinates of the armor map are equivalent to the texture coordinates of the main textures of the vehicle.

To determine a color equivalent of the armor x (brightness) it is necessary to use the following relation:

$$\begin{array}{ccc} 100 & - & M \\ x & - & N \end{array},$$

where M – a maximal level of the vehicle armor (level 100 corresponds to it);

N – an armor level, for which the brightness is searched.

Thus:

$$x = \frac{100 \cdot N}{M}.$$

It is necessary to “paint” the entire model using appropriate colors (by means of creation of materials having a necessary color) and render them to a texture, which must be used as an armor map.

The black points in the alpha channel of the armor map mean fully transparent places and white points – armor places. The transformation of the received map saved in TGA format to the format using by the game can be made by a command tga2am by selecting an appropriate file.

2 EXPORT

2.1 File Export to X-Format

The received levels of detail and the physical level must be converted into X format by means of built-in capabilities of DDC tool (for example, Blender) or by means of exterior plugins, for instance **Panda DirectX Exporter** that can be downloaded under the following link: http://www.andytather.co.uk/Panda/directxmax_downloads.aspx.

File names must be as follows:

1) <name>l0.X for the physical level;

2) <name>l1.X for the main level of detail;

3) <name>l2.X for the second level of detail that is displayed at the distance of more than 50 meters;

4) <name>l3.X for the last, simplest level of detail that is displayed at the distance of more than 150 meters;

To transform models into GO2 format used by the game a command x2go needs to be used that checks the models and collect all the levels of detail into a single file. To transform any level of detail or physical level needs to be selected.

If there are errors during the transformation, a report on the transformation containing their detailed description will be drawn up.

2.2 Texture Export

A command dds2atf is designed to convert textures; it allows to transform the textures from DDS format into ATF.

An instance of texture transformation:

```
starter.exe dds2atf, users\modwork\reg_tex_dift.loc_def.dds, users\modwork\ reg_tex_dift.loc_def.texture
```

transforms a texture reg_tex_dift.loc_def.dds into reg_text_dift.loc_def.texture. If the second parameter is not set, the received file will be located in that folder as the unpacked one but it will have an extension texture.

To create textures in dds format a number of programs can be used:

- Paint.NET, under the link: http://www.getpaint.net/index.html;
- GIMP, under the link: http://gimp-win.sourceforge.net/stable.html;
  DDS plugin http://nifelheim.dyndns.org/~cocidius/dds/;
- nVidia® plugin for Adobe® PhotoShop®
  http://developer.nvidia.com/object/photoshop_dds_plugins.html.

3 VEHICLE CONFIGURATION

To configure new properties of new vehicle a config file with a description of the vehicle needs to be created where the blocks listed in the table 3.1 must be presented. Instances of complete config files with description are located in the file "techn_base" (archieve tabs.flatdata), they can be used as a sample by choosing the description of the vehicle that is similar by type than other existing ones. All the blocks of the description must be collected into one block that has a name of vehicle.

Table 3.1

Vehicle Properties Blocks

Name	Purpose
props	main properties of vehicle: mass, geometry, type, sounds, armor, engine description, etc.
gears	gear box: gear transmission ratio, efficiency
wheels	wheels of vehicle: unique number, chunk, properties
cpillars	tracks: track name, properties, number of wheels, and their unique numbers
col_pts	impact points on the ground: chunk and point displacement relative to its center of rotation
col_bounds	chunks for collisions calculation
weapons	weapon description: weapon name from table, shell flyoff chunk, recoil chunk, turret, loader position
shells	default ammo: shell, amount
heads	description of turrets: name, vertical rotation chunk, horizontal rotation chunk, horizontal and vertical angles of rotation, rotation speeds with or without electric drive
work_places	description of crew work places: name, properties, turret, sights, hatches, work places for internal and external position of crew member, equipment escaping points
sights	sighting equipments: name, observing point, sight type as table shows, group of sights (indicated in working place)
damages	equipment units for damage calculation: unit type, chunk, armor thickness, shielding value, burning probability, firing probability for this damage (for AI)
mslots	titles and signs description: type, location, location for extended titles, rotation and distance parameters from the

	surface, distance between the letters
flames	burning points: position, group of hatches, engine compartment (1) or fighting compartment (0)
traces	leaving points of traces of wheels and pillars: track type, point position
smk_gen	exhaust pipes: exhaust chunk
carets	carets:
lights	light sources: name, chunk of source position, source type, glow distance, glow radius, color
stanins	mounts: mount chunk, mounting angle, rotation axis
propellers	propellers: chunk of propeller, rotation speed
hatchs	hatches: group, hatch chunk, opening angle, a number of positions, initial position, opening axis

Basic properties of the vehicle are listed in the Table 3.2.

Table 3.2

Basic Properties of the Vehicle

Name	Purpose
name	name of vehicle
mesh	geometry of vehicle
ostov_type	frame of vehicle
mesh_lods	a number of visible levels of detail (always 3)
type	vehicle type: TANK, GUN…
select_as	mask for vehicle selection (for interface)
vis_factor	demask factor (1 – standing soldier in a field)
grass_hide	hiding value while in the grass
effect	equipment efficiency (1 – soldier with rifle)
trace_map	trace map, used for pathfinding
manual_rot	equipment can rotate manually by crew (towed cannon)
immobile	immobile equipment (machinegun, mortar)
deform	deforming equipment (guns, machineguns)
forw_techn	equipment can move ahead of the formation
snd_pillars	pillars sound
snd_engine	engine sound
snd_horn	horn sound
snd_starter	starter sound
snd_move	move sound
snd_gear	gear sound
mass	mass in kg

j	moments of inertia
engine_m	moment of inertia of engine
engine_pow	engine power (hp)
eng_tau	engine spin-up time constant
min_w, max_w	engine speed range, rpm
water_line	point (chunk) which when submerged under water - equipment is considered to be sunken
diesel	diesel engine
fuel_rash	fuel rash l/100 km
fuel_tank	fuel tank, l
acc_capacity	accumulator capacity
eng_start_cur	engine starter current
suo_cur	current consumed by command systems
eng_cur	engine current
gen_power	generator power
arm, arm_week	additional armor for AI: front, side, back, top
armor_thick	armor thickness (map armor scale in mm for 100 level)
armor_map	armor map
armor_qual	armor quality: 0.99 – 0.7
armor_frail	armor fragile
armor_str	armor strength: 1500 – 2800
flame_dam	flamethrowing weapon damage ability
pike??, fly??	dive and flight parameters for aircraft
crew_move	number of crew for equipment moving
crew_rotate	number of crew for equipment rotation
move_speed	equipment moving speed by crew
rot_speed	calculation of velocity of rotation of vehicle
move_dist	equipment moving distance before stopping by crew

To add a new type of tracks it is necessary to create 1 track (model) and texture, add a description into the file common_res_mod (tabs.flatdata) into the block tracks.

New weapon and shells are added into the file common_res_mod (tabs.flatdata) into the block of weapons and shells correspondingly.

Do not duplicate records that already exist in file common_res_mod! Create weapons and shells after a fashion of patterns of similar type that already exist.

3.1 Adding New Vehicles

To add a new vehicle into the game it is necessary to create its model, textures, armor map, and if necessary to add tracks, new gun, and shell types into common_res_mod (tabs.flatdata).

The vehicle description is filed in div_units.loc_def.config, which is located in the archive tabs.flatpack. To extract it, it is necessary to do as follows:

1) Unpack the archive tabs.flatpack (from patch) using the command unflat.

starter.exe root\programs\unflat.progpack,
data\k43t\dev_updates\shared\packed_data\tabs.flatdata, users\modwork\tabs_uf

2) Convert the file div_units.loc_def.config using the command cfgp2pd.

starter.exe root\programs\cfgp2pd.progpack, users\modwork\tabs_uf\div_units.loc_def.config,

To copy unpacked config file "div_units.loc_def.engcfg2" to another folder in order to work further. The vehicle files work according to storage system, i.e. each installed patch or addon, which has a file with such a name, adds a description of units or subdivision into the common list. If subdivisions are duplicated, the first one is used that has been found during installation of patches and addons.

In the section **units()** a description of separate units of vehicles and soldiers, with prefix "rkkau_" Soviet vehicles and soldiers, and with prefix "weru_" German vehicles and soldiers, is located.

A line with description of new vehicle needs to be added into the block "units" (other lines may be deleted).

The line must contain a vehicle name (with prefix), class " i_techn", sprite and name (not used), base name with vehicle description (a name of the description created by you needs to be written here), see Section 3), block name of the description created by you, number of places for crew, number of places for passengers/assault.

An instance of line with description of T-34 Tank:

rkkau_t34_42_uztm, i_techn, un_ussr_t34, , tabs\techn_base.cfgpack, t34_utz_m42, 4, 12

See Section 4.2 of the Addons Creation Guide for details on subdivision creation process by using new vehicles.