

BUNNY TRAILS MAP MAKING TUTORIAL

Intro

Bunny Trails gametype is meant for mastering movement skills that can be used in other UT3 game types. We are not trying to copy exactly the BunnyTrack's look and feel.

Always make gradual increase of difficulty in the maps, no matter the overall difficulty level of the map. Let players of all levels practice while masters pass by. This game type is all about helping each other and learning practical things in a fun way. So, do not underestimate the importance of learning curve in every map.

In Bunny Trails TWO TEAMS race against each other to reach the goal 3 times before the other team does. We call reaching the goal a "Cap" (Capture) from capture-the-flag terminology, used in original Bunny Tracks game. We don't use flags anymore; instead we use a special Goal Destination Teleporter. See section 4 for details.

Also, note that we publish the maps **uncooked**. This is mainly due to two reasons:

1. Because there does not seem to be a way to cook a map so it does not include the gametype into the map. If you put such map on the server and your server runs different version of gametype, clients won't be able to log in and vote for another map, so that basically stalls the server.
2. Also, uncooked maps tend to be up to 10 times smaller in size, and that is a considerable advantage for everybody and especially, online players with not-so-fast network connection.

You may cook it once, if you want, just to get the .ini file, but then remove the cooked version, and only publish the uncooked version.

1.1 Naming Convention

It is important to use naming conventions, because once there are more than 20-or-so maps on the server, sorting them out will be difficult and who knows, maybe some new voting mod will be developed and then maps will probably be sorted alphabetically using ASCII, where depending on the

case the maps will be in different places and probably hard to find. And it also looks nice when all map names are properly formed in any voting list.

Each map has two names: **file name**, and the human readable **map name** (Title in the WorldInfo properties, see image below, as well as in the .ini file). These two are different! They both need to be well-formed and clearly specified.

File name looks like this:

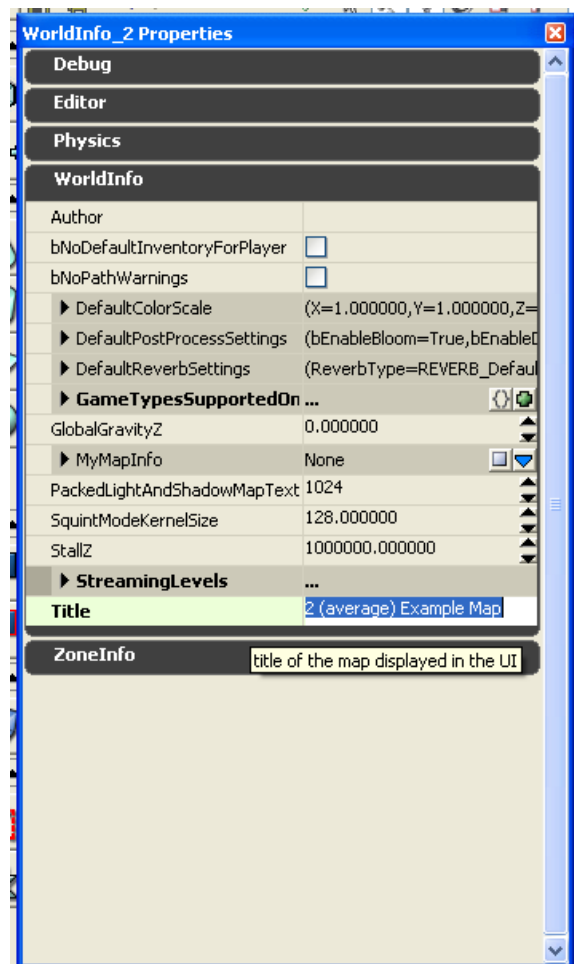
B T - 3 - E x a m p l e M a p . u t 3

And its corresponding **map** name looks exactly like this (please don't use underscores _):

3 (e x p e r i e n c e d) E x a m p l e M a p

Note the **spaces** in the map name instead of dashes, and also note the difficulty level number: in BT, difficulty of the game depends on the map itself, so it should be marked in the file name, for people to know what they are voting for. Difficulty level number indicates the hardest bottle-neck part of the map. They are same as bot levels in UT:

- 1 (novice)
- 2 (average)
- 3 (experienced)
- 4 (skilled)
- 5 (adept)
- 6 (masterful)
- 7 (inhuman)
- 8 (godlike)



For duple/triple/quad maps a special symbol ~ (tilde) is required. This is for sorting convenience, so later on, when new voting interfaces is developed, these maps can be easily put aside, in a separate list. These maps require at least # number of players in a team (map with subteams). Example:

```
B T - 3 ~ 2 - E x a m p l e M a p . u t 3
```

Map name:

```
3 ~ 2 ( e x p e r i e n c e d ) ( d u p l e ) E x a m p l e M a p
```

2 (duple)

3 (triple)

4 (quad)

5 (pent), 6 (hex), 7 (sept), 8 (oct), 9 (non), A (dec), B (11-tuple) – very unlikely, but that’s the idea.

For decuple (10 players per sub team) use capital letters A, B, C, D, etc. until Z, then small letters a, b, c, until z, and corresponding description (n-tuple). This might never be needed but this is the naming convention that will sort well in ASCII.

2.1 Installing Bunny Trails so that the Editor can see it

When you install Bunny Trails for just playing, you copy BunnyTrailsGame.u to your

```
\My Documents\My Games\Unreal Tournament 3\UTGame\Published\CookedPC\Script
```

And of course, the corresponding .ini file, UTBunnyTrailsGame.ini goes into your

```
\My Documents\My Games\Unreal Tournament 3\UTGame\Config
```

For the UnrealEditor to see that game, though, two more steps are required:

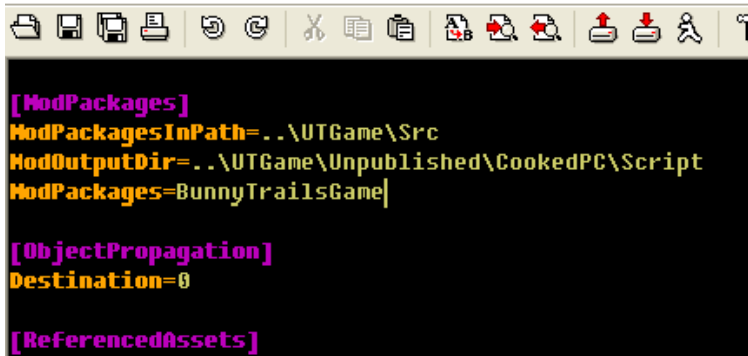
1. copy BunnyTrailsGame.u to

```
\My Documents\My Games\Unreal Tournamnt 3\UTGame\Unpublished\CookedPC\Script
```

2. In the aforementioned Config directory, edit UTEditor.ini with your favorite text editor. Use the editor's **Search** function to find this text: **[ModPackages]** – It is a section, to the end of which you must add this line:

ModPackages=BunnyTrailsGame

After editing the section should look like this (of course, text highlighting depends on particular editor and its settings):



```

[ModPackages]
ModPackagesInPath=..\UTGame\Src
ModOutputDir=..\UTGame\Unpublished\CookedPC\Script
ModPackages=BunnyTrailsGame]

[ObjectPropagation]
Destination=0

[ReferencedAssets]

```

Save this file and restart your Unreal Editor.

3. Bunny Trails' TriggerVolume

If you properly installed BunnyTrailsGame, next time you start the editor, two new volumes appear:

HelpMessageTriggerVolume and **EquipTriggerVolume**, as seen on the image, in the menu that appears when you right-click the volumes button.

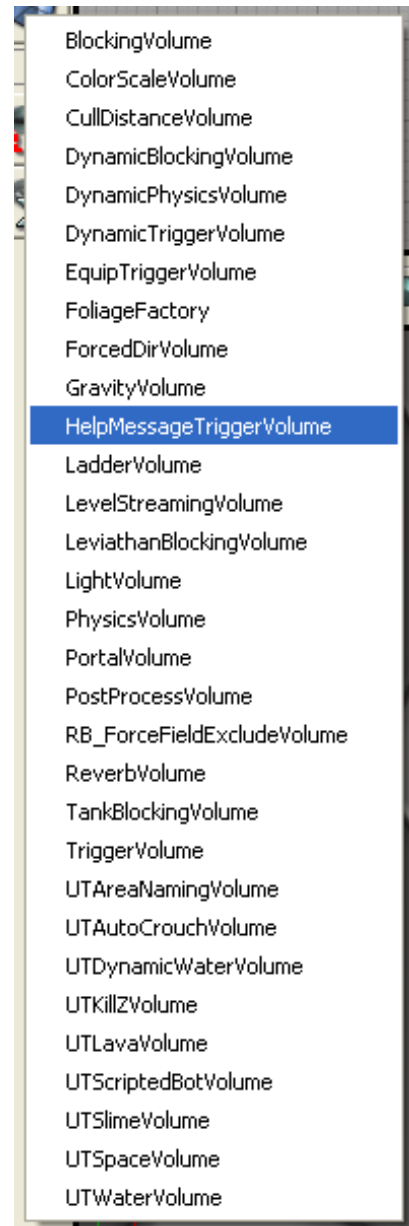
These two will only work in Bunny Trails game!

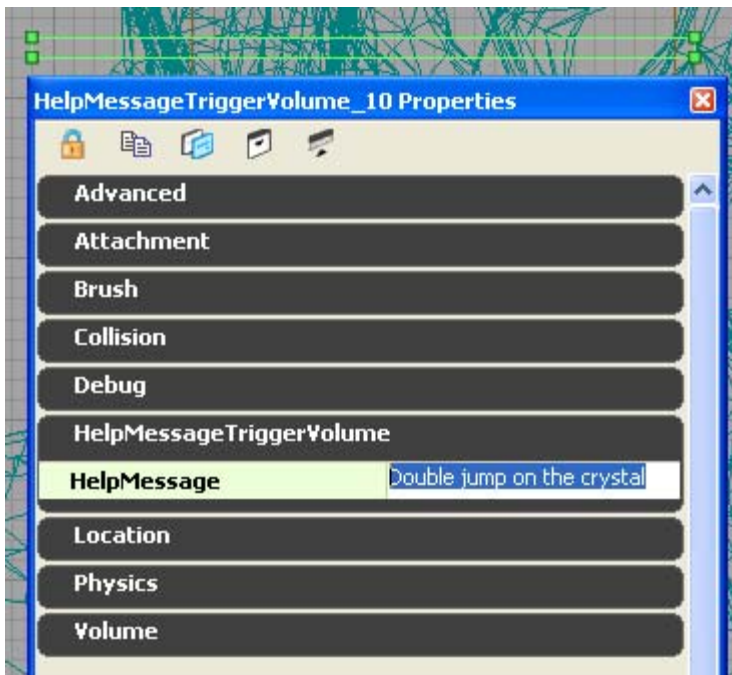
3.1 HelpMessageTriggerVolume

I made this help trigger volume because it is used so often. Note that same effect can be achieved by using kismet action

BunnyTrails->BT Help Message. But for everyday use it's

much more handy to use this special trigger volume. Just place it wherever and edit its HelpMessage property:

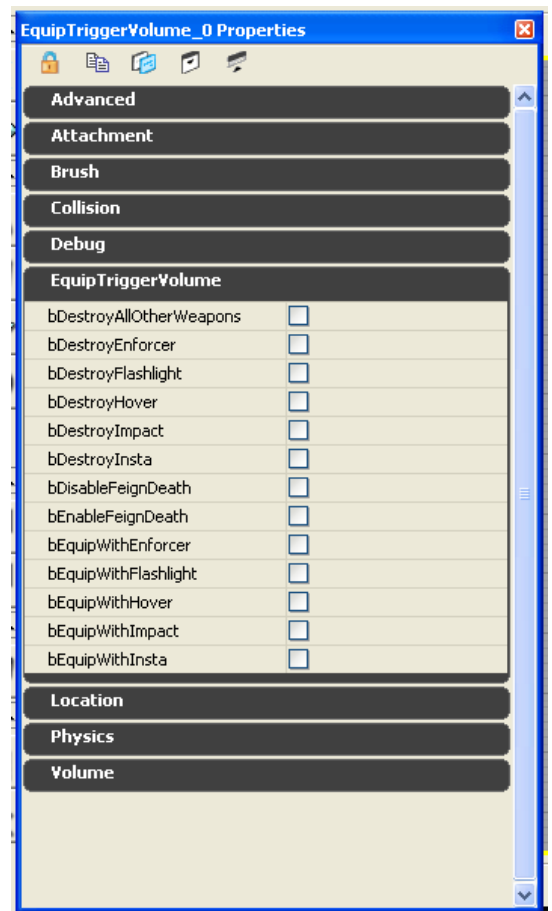




3.2 EquipTriggerVolume

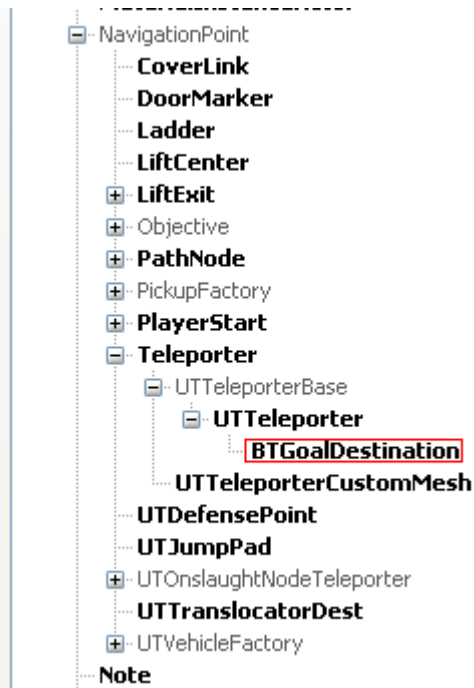
You can use any weapons or vehicles in your map: use weapon lockers or kismet for that. EquipTriggerVolume allows you to give or take from the player those weapons that are not available by normal means such as weapon lockers. Also, you can give the hoverboard and a special Bunny Trails flashlight. In the properties menu you should find these options (see the image):

Here is how it works. If you want to give a certain item, check the box next to corresponding bEquipWith variable. If you want to take an item from a player, check the box next to corresponding bDestroy variable. You can give and take several items at the same time. You can also empty player's inventory of all regular weapons by checking bDestroyAllOtherWeapons. For UT2004 Trial-style maps I also added an option to disable feign death, for it is considered cheating by purists of that particular genre.



4. The Goal Destination

In Bunny Trails we don't use flags like in Bunny Track. There is now a special destination teleporter to score goals. Find it in ActorBrowser:



When a player gets **TELEPORTED TO** this **destination teleporter**, he scores a goal for his team. It is placed at the BEGINNING of the map. This teleporter can be used by either team which makes it possible to make a single map. However, it is discouraged, especially if the map has movers, lifts and other things that can be exploited by one of the teams to slow down the other team in an unfair way.

This information should be enough to start making amazing BunnyTrails maps. Next section is for those who want to use a UT99 physics exploit which allows jumping very high by dodging against a slope. This was easily done in UT99 and UT2004, but to make that work in UT3, I made a dirty hack, which, with some limitations, can do the trick (sort of). It took me a month of research, and I studied the motion patterns by logging the positions of the player versus time, just to get the right formula. I took a scientific approach and the result is as close to UT99 as humanly possible.

HAPPY MAPPING!

— arDru

5. BTRamp Making Tutorial

This is a step-by-step explanation how to use the BTRamp.u script in your map to get an effect resembling those famous UT99 and UT2004 slope-dodge jumps. And so on.

BTRamp is a script that allows map makers creating special ramps (slopes), that give the player vertical boost when dodging against, similar to UT99 and UT2004, so if you never played those games – you probably can just happily ignore the rest. It's been a year since this script is around and so far it's been used only in one BunnyTrails map, as far as I know. Unless this technique ever becomes popular again, but at that point you will know what this is about.

Whatever...



On this screenshot you can see Drummer.unr going up along the slope. He did that by wall-dodging at the bottom of it.

BTRamp.u comes with BunnyTrailsGame.zip, but you can probably use it with other gametypes.

1. Close UnrealEd and all instances of UT3 game if any is running.
2. Extract the .zip file to the desktop (or anywhere else outside UT3 folder structure). Ignore the Source folder, it is there just for reference. Copy BTRamp.u into your

Documents\My Games\Unreal Tournament 3\UTGame\Published\CookedPC\Script

And also to this folder:

Documents\My Games\Unreal Tournament 3\UTGame\Unpublished\CookedPC\Script

Important: This file BTRamp.u should also come with your map's distribution package. In the installation instruction, please advise the user to copy this file to folder

Documents\My Games\Unreal Tournament 3\UTGame\Published\CookedPC\Script

In case the file with same name already exist, the user should keep newer version or overwrite older version with new one depending on the case.

If you use source code and modify it, you must also change the name of the class.

3. Now, go to

Documents\My Games\Unreal Tournament 3\UTGame\Config

4. Backup UTEditor.ini by copying it to your Desktop.
5. Open

Documents\My Games\Unreal Tournament 3\UTGame\Config\UTEditor.ini

in your favorite text editor.

6. Find the string **[ModPackages]**

It should look something like this:

[ModPackages]

ModPackagesInPath=..\UTGame\Src

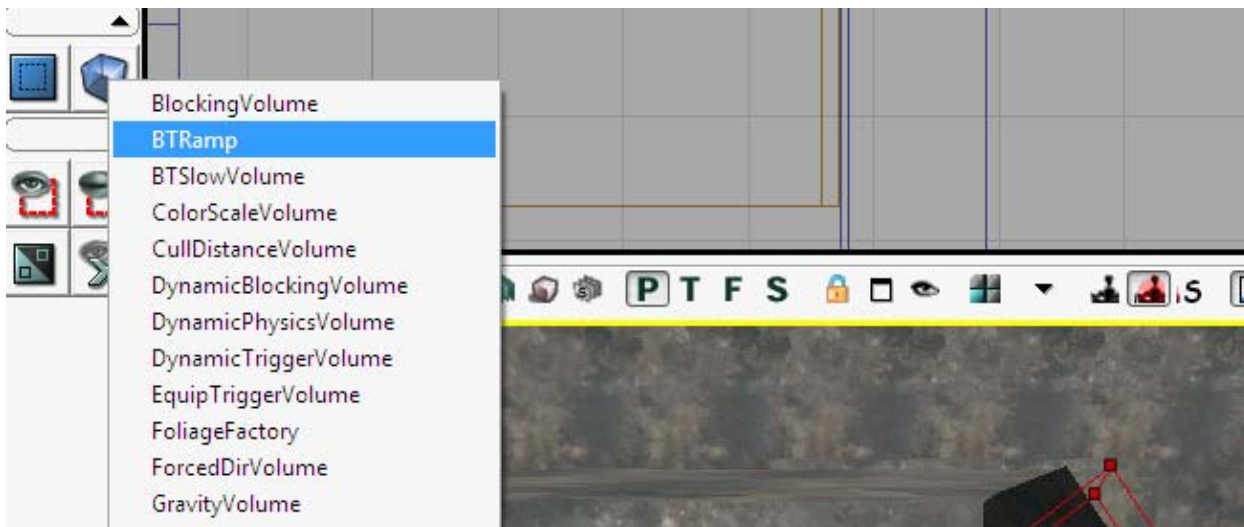
ModOutputDir=..\UTGame\Unpublished\CookedPC\Script

etc... At the end of this section add this line:

ModPackages=BTRamp

7. Save this file.

Preparation is done! Now when you open UnrealEd and right-click the Volumes buttons you should see BTRamp in the list.



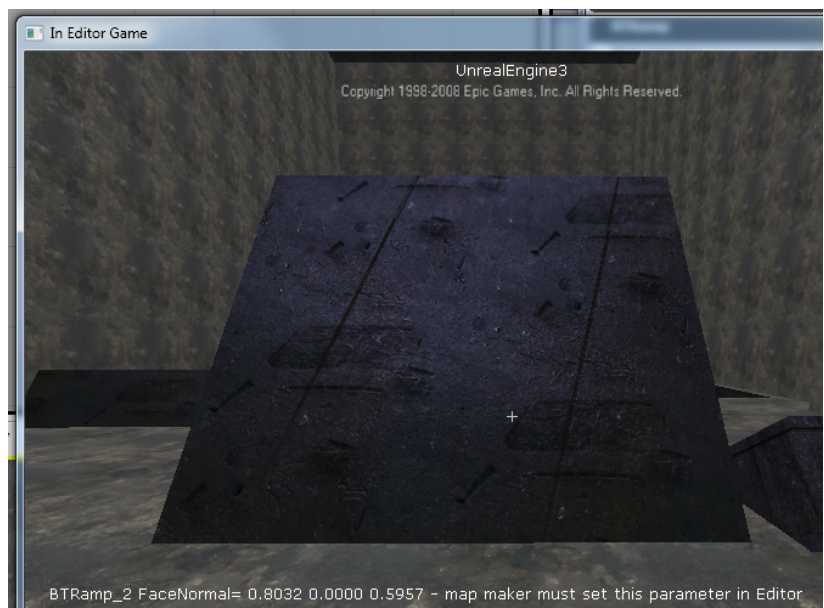
You need some solid object to act as a ramp. Make sure you rotate it slightly, otherwise the ramp will be sticky. Now you need to make volume that covers its face. The face has to be exactly at the same 3d angle. The easiest way is to select the purple brush and create red brush from it.

Now offset the red brush by about 32 unreal units like so:

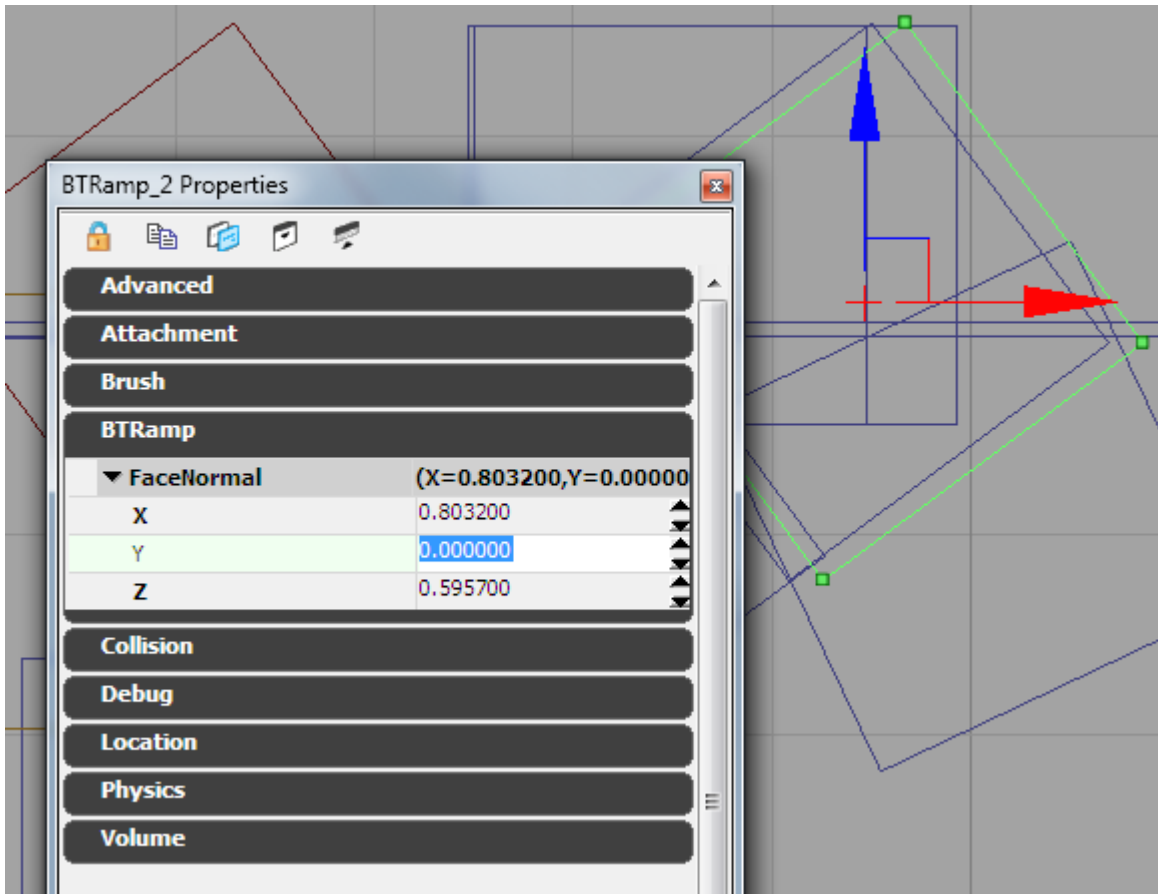


Now, add BTRamp volume by right-clicking that volume button and selecting BTRamp as seen on one of the previous images.

Run the in-editor game. When you hit the face it will tell you the three parameters you should take note of:



Enter these three numbers into FaceNormal variable of the BTRamp volume:



Note: These values specify this particular face's normal vector in world coordinates. So, if you copy/paste your volume and apply it to other ramp, and they are not facing exactly same world 3d direction, you must set Z value to 0 in order to get new values in in-editor game (to trigger a refresh of the values).

And that's it!

Very important (in practice things are more complicated):

Remember that there are critical limitations due to this simplified implementation of slope dodging volume:

- while touching the BTRamp volume, make sure that player cannot possibly touch any rigid surface other than the ramp face itself and any walkable floor (and, theoretically, a surface that is parallel to the slope face's normal vector).
- don't put two BTRamp volumes close to each other so that a player can be touching both at the same time.
- only one flat face can be assigned to one BTRamp volume.
- static meshes can be used too, but make sure its collision volume (if any) has a flat face that you will be using as a ramp. If geometry of the face is complex you must wrap it into a nice and flat-faced collision volume.

HAPPY MAPPING!

— arDru

Recent version of this tutorial can be freely downloaded here:

<http://tamtemple.com/BunnyTrailsMapMakingTutorial.pdf>