The Teleporter Walkaround

04 March 2000

Let me begin by saying that this is not a complete tutorial for creating and using teleporters. That's why I called it a 'Walkaround' and not a complete tutorial. My knowledge is limited to building them and making them work. The actual scripting portion of the teleporters is above and beyond me. I'm a mapper and not a scripter. If I had more time to invest in figuring out what everything meant, I would. Unfortunately, I don't. So, this will hopefully teach you how to build functional teleporters and that is it. This will be as clear as mud until you figure it out, so be patient.

Goldberg

I'm going to try and make this as simplistic as possible. For explanation's sake, I'm going to treat this somewhat as a recipe. So, tonight, we will learn how to make teleporters!

Ingredients:

1 CS file
1 MIS file

Each teleporter will need:

1 trigger
1 drop point
Various objects to build the visual part of the teleporter.

Despite what you think, a teleporter is nothing more than a trigger and a point on the map where the player respawns. Let's start with the easiest part of the teleporter. The visual part the players will come in contact. These pieces are merely cosmetic to tell the player where the teleporter is located. The most popular objects are the big electro beam and the flag stand. For a visual example of this, grab a copy of "Towering" off of our site (www.planetstarsiege.com/extreme) You can find it on my map page. Inside the base is two teleporters that spawn in two places. One spawn is on the roof of the base and the other is on a floating platform. You will see that the teleporter consists of the electro beam and a flag crash. Another different example is on "Three O'clock High". There are multiple examples of teleporters in this map. Inside the base, in the arena and in the tube. The tube, for example, gives the player the impression there is nothing in front of him but an open hallway, but when loaded up in the editor, you will see the triggers at the midpoint that spawns the players into the arena. Here is an example that shows you need nothing "to inform" the player that there is a teleporter. He will find out soon enough. So, my point here is that the visual pieces of the teleporter are up to you. Use what you would like.

Now, down to the actual teleporter build. To make a teleporter work, you will need to create a group in the editor (the MIS file) called Teleporters (or whatever you want to call it). Teleporters are not team oriented. They work for anyone, even if you assign them to a team. So, what I usually

do is create a separate folder under the teams folder and put the teleporters in there. Each teleporter will need its own folder in the "Teleporters " folder. Each of those folders (each teleporter) will contain a trigger and a spawn point. First off, once you've created the Teleporter folder and the number of folders for each inside that folder, we can start with the first teleporter. Decide where you want to place the "entrance" to the teleporter. This is the trigger. Triggers are found in the Mission folder (1st on the list) in the editor. Triggers can be changed in size according to how big you want the "entrance" to be. This can be done by editing (F2) the dimensions. Play with the size to make it fit the way you want. Ok, the entrance is built. Now, you will need a spawn point. This is where you want the player to be respawned when they step into the trigger. The trigger just tells the game that the player should now be where the spawn point is. So, go to where you want the player to spawn and place either a drop point or a marker. Despite what people say, you don't actually need the drop point or marker, you need the exact spot on the map (coordinates) they sit on. So, in "Towering", team 1's teleports can be found in the Power section (yes, I put them in the team, but it isn't necessary). If you look at the first marker, you will see it's Position is '175.114 –22.979 281.881'. This is the exact location on the map the player will respawn after walking into the trigger. I got that exact location from placing the trigger where I wanted it and cutting/pasting the coordinates. The editor will give you the coordinates automatically when you place the marker. You with me so far?

Ok, so, you've built the visual part of each teleporter and you've placed the trigger and respawn points for each. Why doesn't it work yet? Now comes the fun part. To make them actually work, you will need a CS file and you will also need the MIS file. I've included a CS file in the zip that you can use for every map you include teleporters. If you look at the CS file, this is what you will see:

```
function GroupTrigger::onEnter(%this, %object)
{
%client = Player::getClient(%object);
if(%this.num == "Main1"){
     %positionIn = "-452.957 807.625 85.5675";
     %positionOut = "-9.77895 1694.47 1471.63";
   }
     else if(%this.num == "Main2"){
     %positionIn = "646.831 233.683 76.1779";
     %positionOut = "-5.61334 1694.48 1471.42";
   }
   else if(%this.num == "Main3"){
     %positionIn = "-17.8541 1819.83 1483.64";
     %positionOut = "-22.3762 1819.84 1483.47";
   }

          if(%this.in){
       GameBase::setPosition(%client, %positionIn);
       //messageAll(0, "~wshieldhit.wav");
```

```
            Client::SendMessage(%client,0,"~wshieldhit.wav");
    }
        else if(%this.out){
      GameBase::setPosition(%client, %positionOut);
      //messageAll(0, "~wshieldhit.wav");
      Client::SendMessage(%client,0,"~wshieldhit.wav");
        }

}
```

This is where the scripting confuses me a bit, but here's what it kinda looks like. Each separate group will stand for a spawn point (or two if you chose).

```
else if(%this.num == "Main2"){
    %positionIn = "646.831 233.683 76.1779";
    %positionOut = "-5.61334 1694.48 1471.42";
```

What is important here is the "Main" # and the positionin/out #'s. The Main # is just the number for each separate teleporter. So, if you have a total of 4 teleporters in your map, I suggest you have 4 "Main" groups like the one right above. The positionin/out is where the player respawns or where you place the trigger/drop point (you will get the positionin/out # from the MIS file). The CS file has nothing to do with the triggers, just the spawn points. Here is where it gets a bit foggy for me. Despite what 'In/Out' means to you, it only means out (or spawn) to the game. So, the two numbers above in the 'In' and 'Out' are both exits or spawns. I personally only use the 'In' option and leave the 'Out option empty like so:

```
%positionOut = "";
```

I think it has to be there, but doesn't have to have a numeric output.

Confused yet?

Ok, so far, we've built the actual teleporters in the editor and we have created a CS folder that for each 'Main' group has a spawn point. Now, the CS file tells the game where to spawn the player, but we need one more location in the game or there is no link between the editor and the CS file. Like I said, the CS file is only the output or spawn. How does it know where to get the player from so it can spawn him on this spot? In the MIS file. This is where it's gonna get fuzzier.

Open up the MIS file in notepad or wordpad and do a search on teleporters (or whatever you named the teleporter group). Once found, you will see each teleport group consisting of the trigger and a spawn point. What we need to do is tell the trigger to look for the exit locations that have been defined in the CS folder. Here is an example of what you will see in the MIS file:

```
instant SimGroup "Teleports" {
            instant SimGroup "TP1" {
                  instant Trigger "GroupTrigger1" {
                  dataBlock = "GroupTrigger";
                  name = "";
                  position = "-0.0189608 -197.908 172.178";
                  rotation = "0 -0 0.399994";
                  boundingBox = "-0.5 -0.5 -3.5 0.5 0.5 3.5";

            };
            instant Marker "Marker1" {
                  dataBlock = "PathMarker";
                  name = "";
                  position = "-2.71768 -221.581 230.836";
                  rotation = "0 0 0";
```

This is the teleporter group you created (you should have one of the above for each teleporter). It includes the trigger and the spawn point. Looking at the spawn point, you will see the 'Position' #. This is the number you need for the CS file. This is the exit location on the map. You can just cut and paste this number into the CS file. If the number has a '-' negative sign in front of it, be sure to include that. Just like the one above. To make the trigger work right, we need to tack on something at the end of it. This is what you will need:

```
            locked = "0";
            in = "True";
            num = "Main1";
```

These three lines make the trigger work. Don't ask me how, cause I just don't know. But, they have to be there or the teleporter is worthless. The third line should look familiar. This tells the CS file which 'Main' group it should look for. Below is how the group should look once you've tacked the three lines on:

```
instant SimGroup "Teleports" {
            instant SimGroup "TP1" {
                  instant Trigger "GroupTrigger1" {
                  dataBlock = "GroupTrigger";
                  name = "";
                  position = "-0.0189608 -197.908 172.178";
                  rotation = "0 -0 0.399994";
                  boundingBox = "-0.5 -0.5 -3.5 0.5 0.5 3.5";
                  isSphere = "True";
                  locked = "0";
                  in = "True";
                  num = "Main1";
```

This trigger will look for the 'Main1' group in the CS file and spawn the player in the coordinates that are attached to the 'Main1' drop. Is it starting to make sense? I hope so. Look at the MIS file on Towering and find the teleporters group. You will see the four groups containing the above with

a corresponding marker. Each of the groups have an increasing or different 'Main' # that correspond to its drop point in the CS file.
So, after tacking on the three lines to the end of each trigger, we assign which 'Main' group that trigger spawns the player (defined in the CS file). This is the difficult part. Once you get this down, there is only one more thing you need to do. That is to execute the CS file. You do this at the bottom of the MIS file. At the very bottom, is a small group that defines Score Limit, CD track, game type, etc. This is where you execute the CS file so the spawn points work. This is what you will put:

Exec(CS fle name);

The CS file name should not be complicated. I usually use the initials of the game name. Looking at the Towering MIS file at the bottom, you will see I called the file 'TT'. You will see this is the name of the CS file. This is where the CS file is executed. This line has to be put in right after the execution of the game type.

Ok, save everything. The CS file has to be put in the Base directory (Dynamix/Tribes/BASE) for the whole thing to work. Load up the map and see if the triggers work. Triggers are testy little objects. You may need to mess with them a bit, because they don't always work. Also, the bigger the trigger, the less chance you have of them working. There is a limit to the size of a functional trigger.

To see some examples of triggers in other maps, pull back these maps from our site:

Viper Pit
Space CTF
The Highdive (the Multipass Remix)

The Viper Pit is a good example of teleporters that the player doesn't see. They are there without all the front-end cosmetics. Space CTF shows a map with multiple teleporters (16 I believe). This map full of teleporters was a huge undertaking. Look at the MIS and CS file to see. The Highdive (the Multipass Remix) shows teleporters in large size. This was about as big as we could get the triggers and have them still work. If you would like some descriptions of the CS file and its purpose, grab a copy of "Golden-X" off of my map page. This was a coop map I did quite a while back with XTERM of the XFAB Map Station. The file includes some descriptions of how things work as well. It's worth the once-over.

Well, I'm sure you are probably a bit confused or have some questions. If you have questions, please feel free to ask. EEPROM and myself can answer (hopefully) any questions you have about teleporters. If you are having problems with some that you have built, send us the map and we'll see if we can figure out the problem. My suggestion is to be patient and play around with them and see what you can do. Good luck!

Goldberg