

In the PAIN Engine you can have huge, complex levels that still render very fast. This is because you can tell the engine to render only that which can actually be seen on screen and not the geometry behind the wall or behind a corner. To do that you have to create a set of Zones, Portals and Antiportals in your Maya level geometry. Think of zones as of virtual rooms and think of portals as virtual doors connecting those rooms. When you're standing in one room you can see only as much of another room as the open door lets you. So all you have to do is place in Maya cubes that represent the rooms (they must have 'Zone' string in their name, like FirstZone_shape, or Zone32_shape etc.) and place portals (with 'Portal' string in the name), which can be of any convex shape as long as it lies exactly on a plane where one room ends and another one begins. In some cases Antiportals come in handy - these are convex models that block the view of geometry. They don't need to be placed between zones (that would be impossible since they are 3d models and not 2d shapes like portals), For example if you have a thick column in the middle of the room you can simply place a cylinder shape inside that column, name it with 'Antyp' string in the name and you're set. Yes, thats 'Antyp' with y, not 'Antip', sorry - that's Polish spelling ;-).

There are a few things to remember. First, you can't rotate or distort the cubes representing zones. Second, one portal can connect only two zones, but you can have as many portals connecting same two zones as you ant. Third, if some object lies in two zones then it will render all the time, regardless of portals, so it's good to avoid heavy pieces of geometry sharing more than one zone. Fourth, you have to keep in mind that in order to render geometry fast the PAIN Engine considers an object visible only if it's bounding box is visible to the renderer. So even if you can't see some object through the door, you might still be able to see its bounding box, therefore the engine will still render such an object.

It seems then that it might be best to break down all your geometry into small pieces that have small bounding boxes that will work great with the portal system. That is only partially true, because the more objects you have the longer it takes to analyze the whole scene and the longer it takes to render. From a performance point of view, you should find a perfect balance between the number of objects, the number of faces in each objects and their size with respect to visibility. That's a really tough balance to reach, it will take some practice before you do, but the better you get, the faster your levels will render.

It is our general rule at PCF to keep no more than couple hundred objects in one map, we try to have our objects with no less than 1000 triangles (modern graphics cards render 1k triangle objects almost as fast as 10 triangle objects) and to have a rather small number of zones and portals and definitely to have a small number of antiportals. We have a profiler built in the editor to help you reach a balance. Just hit the 'H' key once to display basic information with FPS, and hit the 'H' key another time to display detailed information on what percent of processing power is spent on visibility calculation, rendering and other aspects of the game.

Thanks to the profiler you will be able to see if it is rendering that slows down your map or if it is too complex a set of zones, portals and antiportals. Third, if you make some mistakes placing your zones, portals and antiportals you may have parts of geometry disappearing or your FPS may seriously drop so its always good to check the log after loading your map - there is a ton of information in the bottom Output window of PainEd. Load your map, then scroll the Output window up until you see information about zones, portals and antiportals. If something is wrong (for example if you have one portal that connects more than two zones) you will see a warning somewhere in the Output log.