**Understanding Vis and Hint Brushes**

|  |
| --- |
| 1. [Understanding Vis and Hint Brushes](http://tremmapping.pbworks.com/w/page/22453205/Understanding%20Vis%20and%20Hint%20Brushes#UnderstandingVisandHintBrushes)    1. [Basic Vis Theory](http://tremmapping.pbworks.com/w/page/22453205/Understanding%20Vis%20and%20Hint%20Brushes#BasicVisTheory)    2. [Visual Vis Theory](http://tremmapping.pbworks.com/w/page/22453205/Understanding%20Vis%20and%20Hint%20Brushes#VisualVisTheory)    3. [Optimising Vis with Hint Brushes](http://tremmapping.pbworks.com/w/page/22453205/Understanding%20Vis%20and%20Hint%20Brushes#OptimisingViswithHintBrushes)    4. [Improper Hint Brush use](http://tremmapping.pbworks.com/w/page/22453205/Understanding%20Vis%20and%20Hint%20Brushes#ImproperHintBrushuse)    5. [Irregular Hint Brush use](http://tremmapping.pbworks.com/w/page/22453205/Understanding%20Vis%20and%20Hint%20Brushes#IrregularHintBrushuse)    6. [Creating Hint Brushes](http://tremmapping.pbworks.com/w/page/22453205/Understanding%20Vis%20and%20Hint%20Brushes#CreatingHintBrushes) |

**Understanding Vis and Hint Brushes**

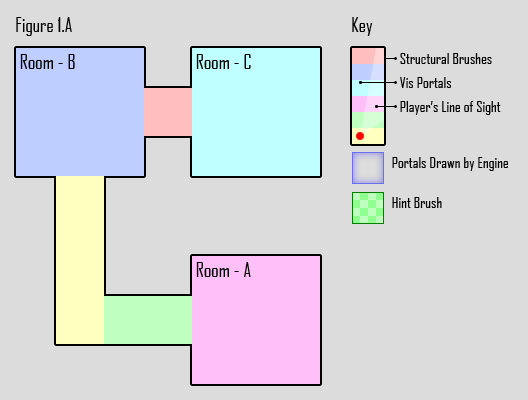
*by Nick "jex" Jansens*

**Basic Vis Theory**

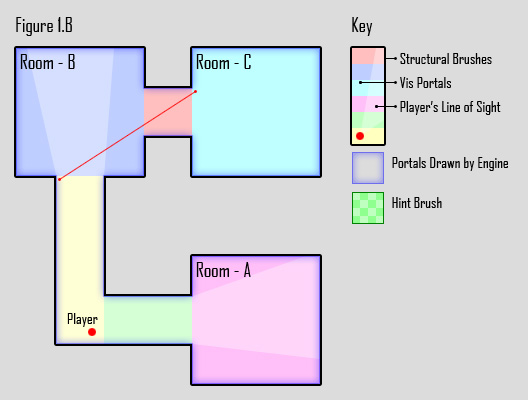
When *q3map2* compiles a map, it cuts it up into vis "portals". These portals may be as big as a room, or as small as only a few units big, depending on how you build your map. The vis process is entirely dependant on these portals. The compiler will look at each individual portal, and attempt to draw a straight line from any point within that portal to any point with every other portal in the map. If a line can be drawn from Portal A to even the furthest edge of Portal B, everything within Portal B will be rendered by the engine while the player is anywhere within Portal A.

Hint brushes can be used to divide the default vis portals into smaller more efficient shapes to help reduce the amount of polygons the engine is rendering that can not actually be seen by the player.

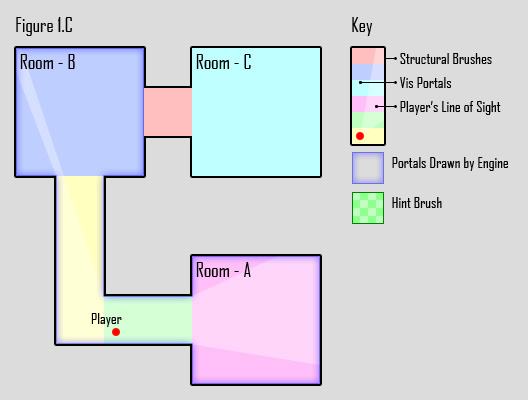
**Visual Vis Theory**



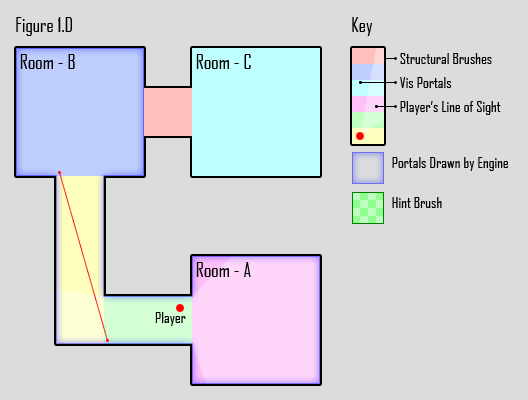
Here is a typical example of how *q3map2* creates vis portals.



In this example Player's line of sight is limited to Rooms A and B but the engine is still rendering Room C. This is because Room C can be seen from a small part of the vis portal Player is in.

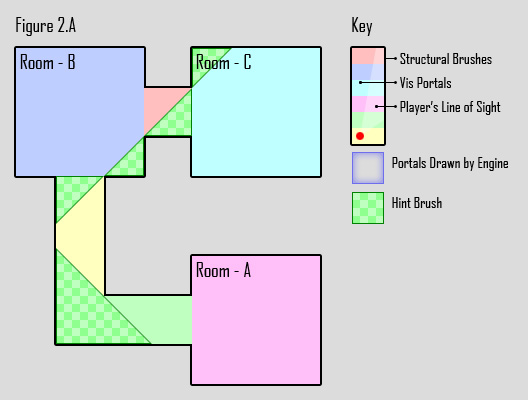


Here Player's line of sight still includes both Room A and B but Room C is no longer being drawn by the engine. This is because a straight line cannot be made from Player's current vis portal to that of Room C without intersecting with Structural Brushes.

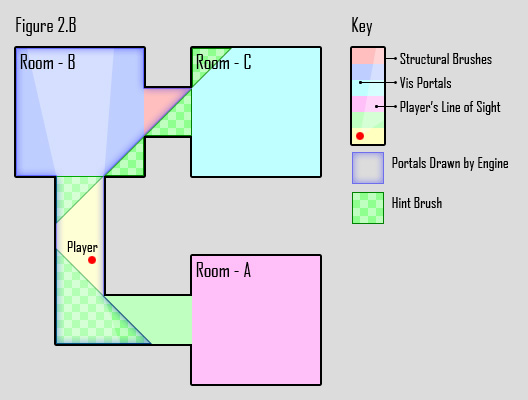


Now the Player's line of sight no longer includes Room B but because Player is still within the same vis portal to which Room B is visible the engine continues to do the work of rendering it.

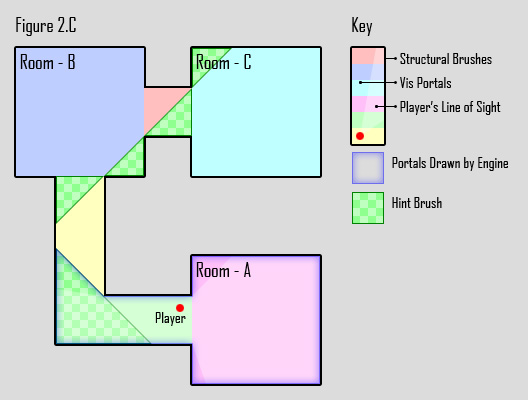
**Optimising Vis with Hint Brushes**



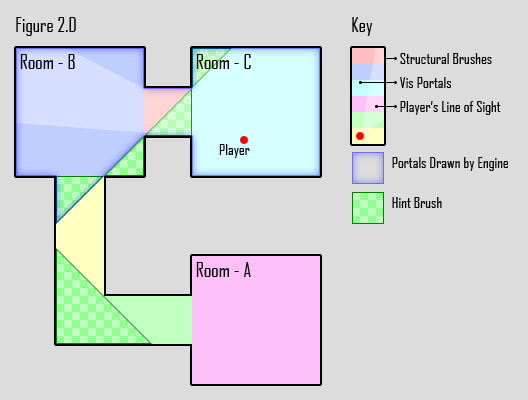
Here is the same example with hint brushes added to optimise the Vis.



We can see here that while everything in Player's line of sight is still being rendered, because of the addition of Hint Brushes, now Rooms A and C are no longer being needlessly rendered.

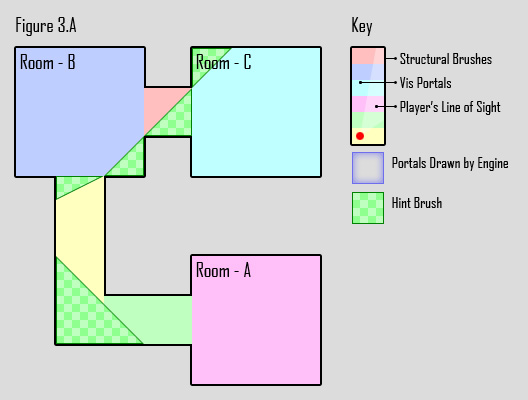


In this example Hint Brushes are preventing the engine from rendering Room B.

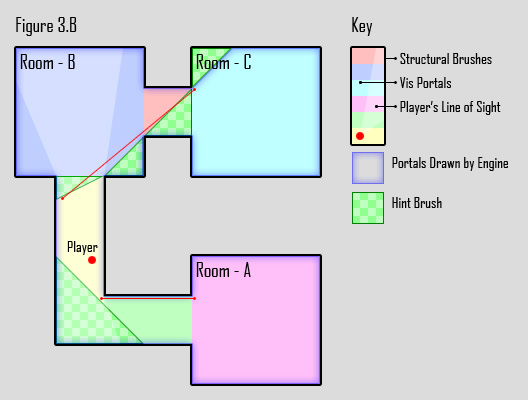


With Player in Room C the Hint Brushes limit the amount of the hallway between Rooms A and B that the engine has to render.

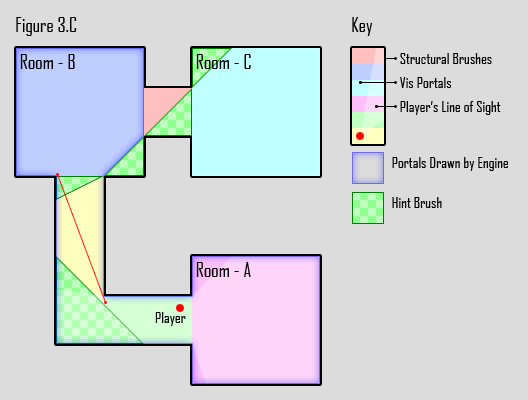
**Improper Hint Brush use**



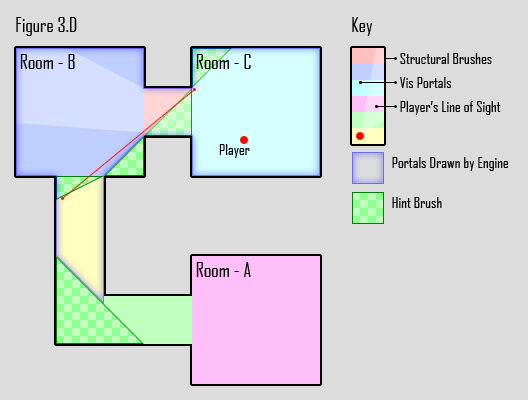
Here is the same example with improper hint brushes added.



Where with proper hinting Rooms A and C can be completely cut out of the Vis, here only a small corner of Room C is not being rendered. The concave angles of the hint brushes between Player and Room C has allowed a straight path to exists between Player's vis portal and that of Room C resulting in almost the entire room being rendered. Between Player and Room A the hint brush doesn't touch the inside corner of the hallway and Room A is also rendered by the engine.

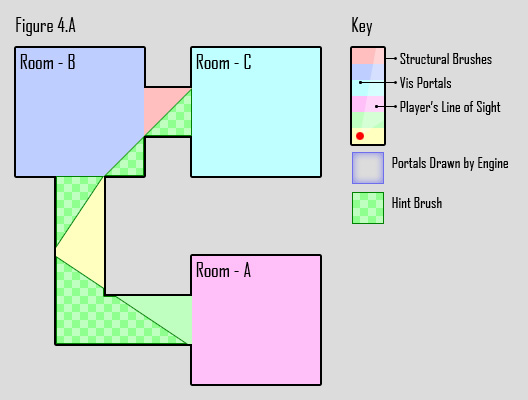


As in the previous example, the same problem exists from around the corner where the hint brush allows a straight line to exist between Player's vis portal and that of Room B.

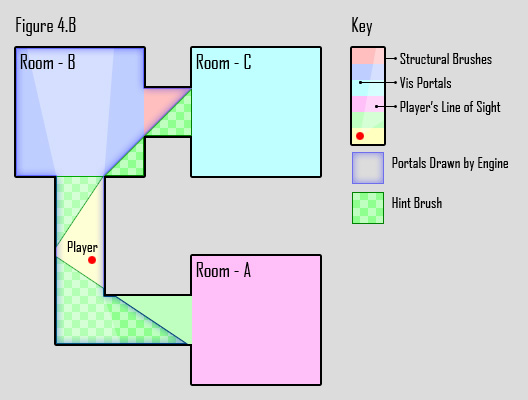


Here the results are less severe, but still unnecessary.

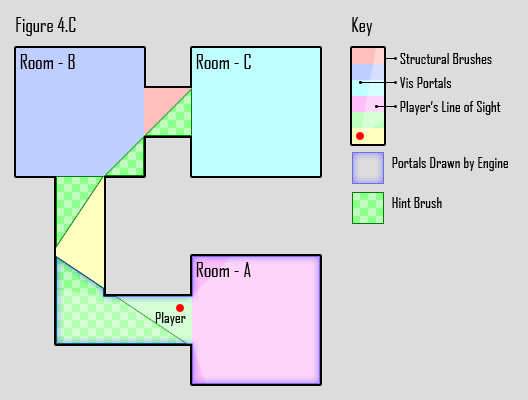
**Irregular Hint Brush use**



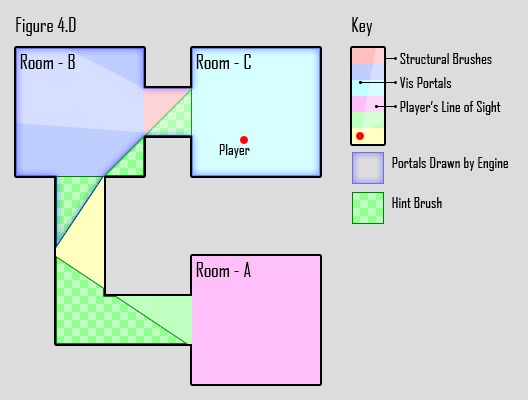
In this example the hinting is done somewhat irregularly but still with good results. Some cases will call for more irregular hinting.



As before Rooms A and C are not being rendered. It is okay for the Hint Brushes to overlap Structural Brushes or to create a convex line with other hint brushes. (Note: because the Quake 3 engine does not allow concave volumes to exist, the overlap of the Hint Brush and Structural Brushes between Player and Room A would cause at least 2 vis portals to be created within the Hint Brush. This will not reduce its effectiveness.)



Here again the Hint Brush between Player and Room B performs adequately if not optimally.



As before the Hint Brushes limit the amount of the hallway between Rooms A and B that the engine has to render but with slightly inferior results.

**Creating Hint Brushes**

Hint brushes are created by making a regular brush and simply applying the *textures/common/hint* shader to it.

textures/common/hint

{

qer\_nocarve

qer\_trans 0.30

surfaceparm nodraw

surfaceparm nonsolid

surfaceparm structural

surfaceparm trans

surfaceparm noimpact

surfaceparm hint

}

Here is the recipe for a hint shader; the active ingredient being *surfaceparm hint*. This shader is included in *common.shader* in [this](http://www.clanopener.com/catalyc/common_files.zip) [archive](http://tremulous.sf.net/users/jex/radical/common_files.zip).

When optimising the Vis horizontally, ensure that your Hint Brushes fully extended from the bottom of a room to the top to prevent any unnecessary straight paths between portals from existing either above or below your Hint Brushes. The same applies when optimising the Vis vertically. Ensure that your Hint Brushes fully extend from Structural wall to Structural wall.