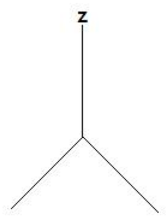
Projection- Transforming 3d to 2d

I have problem or well, I do not know how to transform 3d point with x,y,z values to 2d point, I have to draw projection, where I do have x,y,z values for points but I don't know how to transform them into 2d so I can move them onto my axis.



I have been looking around wiki and google, howevever I'm not quite sure which matrix transformations should I use to get wanted result.

math graph matrix projection

asked Apr 5 '13 at 21:42
user2141889
300 2 7 18

do you know what plane you want to project them onto? how would this plane be given (in what form)? – Bitwise Apr 5 '13 at 21:46

Well, tbh I don't know about the plane, however I do have function so I calculate z point taking x y from min max, and then I have to set those points on the axis, that I have painted above and I have no idea how to even start doing this, I've been watching several movies about plotting points however I do have to plot function, so I have to somehow connect the points, I have found that it has to do something with matrix, but well, not sure what matrix transformation should I apply – user2141889 Apr 5 '13 at 22:08

You may find useful: en.wikipedia.org/wiki/3D_projection and anything related to it – Alexandr Apr 9 '13 at 14:40

5 Answers

Let's first assume the camera looking at your scene is centered at the origin, and looking at the `-z` direction. Then:

- a perspective projection is given by:
  ```
  x' = x/z
  y' = y/z
  ```
- an orthographic projection is given by:
  ```
  x' = x
  y' = y
  ```
  (i.e., just discard the z component)

Now that you have applied the step above, you might obtain a point that is at `(x',y') = (-28.4, +134.5)`. You now need to scale and center them based on your screen resolution and camera "zoom factor" and aspect ratio : for instance, you might want to multiply by `Zoom` and add `screen_center` to both your `x` and `y` components (beware: most graphics rendering systems have the `y` direction pointing down, so you might need to swap signs for the `y` component). You might still end up with pixels with negative coordinates or whose coordinates are greater than your canvas size. Just discard them : it means that they are outside of your view frustum.

Finally, you might wonder what to do if your camera is **not** pointing toward `-z` or not centered at the origin. For the later, it is simple: just subtract the camera coordinates to the component