Getting sprites to face the camera properly in OpenGLES

Asked 15 years, 10 months ago   Modified 14 years, 6 months ago

Viewed 5k times

6

I have a game that is more or less 2D, but rendered in 3D. The camera hovers above the 2D game field tilted about 20 degrees from perfectly perpendicular to give a little 3D perspective to it.

I have some sprites that need to be rendered perfectly square, because they represent spherical objects. My first approach rendered quads flat on the 2D field which positioned them properly, but the sprite was subject to perspective distortion and it didn't always look round.

My current approach is that when rendering the sprites I move the camera to be perpendicular to the game field (and the sprites) which guarantees that the quads are never distorted by perspective. The drawback being that the position of the sprites is slightly different from where they would be if the camera was tilted.

So, how can I render sprites that are:

1. Perfectly square and camera facing but does not require me to move my camera to achieve this.

2. Have a size that is affected by distance from the camera (unlike point sprites)

`opengl-es`

Share

Improve this question

Follow

3 Answers

Sorted by: Highest score (default) ↕

▲

10

▼

You can achieve what you want using a polygon which faces the camera at all times. This is called a "billboard". [Here is](#) a nice tutorial which seems to explain the math involved and how to implement it using OpenGL

Share Improve this answer

Follow

▲

Draw the quads perpendicular to the camera as you suggest, but position them a little above the field. The

0

distance above should probably be the radius of the sphere.

Share Improve this answer

Follow

answered Feb 4, 2009 at 17:49

Liam
20.9k ● 24 ● 90 ● 129

I'm not sure how moving them will help. And part of my problem is I can't figure out what math I need to make the quad face the moving camera when not perpendicular to the play field. Sorry if I am not explaining this well. It's tricky to put into words. – Alex Wayne Feb 4, 2009 at 17:59

0

you can still use point sprites...using glPointParameter() you can give it depth attenuation. i'm currently using 0.00025 for my depth attenuation for my particle system i have in development. the down-side is that on the 1st gen machines (i assume you're doing this for the iphone or ipod touch) you'll be stuck with a max sprite size of 64x64 pixels

Share Improve this answer

Follow

answered Oct 4, 2009 at 4:01

roy

I know this comment is old but for potential readers I just have to stress that there is another downside to this method. Your sprite will disappear if the center if off screen due to clipping. – Rene Jun 27, 2016 at 19:41