

Programming Language Theory – Summer 2016

deepwaterooo

June 11, 2016

Contents

1	Introduction	1
2	References	2
2.1	Haskell	2
2.2	opengl sgl	3
2.3	Animation	3
2.4	Texture	3
2.5	OOP	4
2.6	robot dance	4
2.7	other	4

1 Introduction

- hw2: haskell practice exercise code for any practice due on 6/15/2016 11:55pm.
-
-
-
-
- Considering that I understand the way how to traverse and draw different parts of the zombie, I know I could but don't have to take all the effort to draw all the fingers in order to make a "fighting" pose (a hand cube stands for another "zombie"-like rotation for 2 or 3 different finger parts).
- I don't like the latter part of the project during to rotating simply around the center, though I want to make the zombie walk (but I don't have enough time to work on it). While I will try to work on the walking for a few (3-4) hours tonight to be able to at least have some taste.
- My team buddy's spiderman Dancing stage could simply be rendered by defining the four glVertex3d vertex ([-25, -9.4, 25], [25, -9.4, 25], [25, -9.4, -25], [-25, -9.4, -25]), tex coordinate four points (00 10 11 01), and a Texture. By including norm data, it supposed to have some lights.
- Todos:
 - Speed a few hours (3-4) tonight to work on the zombie walking.
 - Need try hard to figure out animation, how to dance with time changes, or need to move according to keytype inputs triggers.
 - Sub obj% for cubes, and spheres if I need and want to implement any sphere for head, or eyes. But it seems I won't have enough time for this one now. Will try these ones through other projects later on then.
- A current rotatable zombie and my team buddy's spiderman are looking like:



2 References

2.1 Haskell

- Haskell <http://fleurer-lee.com/lyah/ready-begin.htm>
- <http://wiki.jikexueyuan.com/project/haskell-guide/ready-go.html>
- real world <http://rwh.readthedocs.io/en/latest/index.html>
- <http://wiki.bitbegin.com/read/docs/9-haskell/1-haskell-brief-introduction>
- <http://www.cnblogs.com/youxin/category/511831.html>
- <https://wiki.haskell.org/OpenGL>
- opengl <https://github.com/madjestic/Haskell-OpenGL-Tutorial>
- <http://www.arcadianvisions.com/blog/2011/modern-opengl-with-haskell.html>
- cube <https://github.com/haskell-opengl/GLUT/blob/master/examples/RedBook4/Cube.hs>
- <https://wiki.haskell.org/OpenGLTutorial2>
- 3d 旋转魔方 <http://wenku.baidu.com/view/6e7b0d22915f804d2b16c1c1.html>
-
-

2.2 opengl sgl

- rect hello world <https://lists.racket-lang.org/users/archive/2010-October/042474.html>
- cube base: <https://gist.github.com/tonyg/5425736>
- Texture Atlases <http://jeapostrophe.github.io/2013-05-06-texture--post.html>
- Planet Cute http://docs.racket-lang.org/teachpack/2htdpPlanet_Cute_Images.html
- Texture <https://www.mail-archive.com/racket-users@googlegroups.com/msg03203.html>
- <http://lists.racket-lang.org/users/archive/2010-November/043118.html>
- sgl <https://github.com/racket/sgl>
- cube https://rosettacode.org/wiki/Draw_a_cuboid#Racket
- pict3d <https://github.com/ntoronto/pict3d>
- pict3d <https://docs.racket-lang.org/pict3d/index.html>
- buffering <https://lists.racket-lang.org/users/archive/2015-March/066355.html>
- c++ racket ex <http://home.adelphi.edu/sbloch/class/archive/333/fall2013/examples/pentagon/>
- <https://rosettacode.org/wiki/OpenGL#Racket>
- 原理: <http://cuiqingcai.com/1867.html>
- <http://cuiqingcai.com/1867.html>
- 2d <http://cuiqingcai.com/1597.html>
- tech cube <http://wiki.jikexueyuan.com/project/opengl-es-basics/3d-images.html>
- colorful <http://cs317y982s961535.blogspot.com/2010/04/2-3d.html>
- <http://www.d3dweb.com/Documents/201202/15-15182458704.html>
- define-struct <http://lists.racket-lang.org/users/archive/2008-July/026133.html>
- class ex <https://learnxinyminutes.com/docs/racket/>
- gui <https://docs.racket-lang.org/pict3d/rendering.html>

2.3 Animation

- 3d programming: <http://cs317y982s950831.blogspot.com/>
- ruby <https://www.youtube.com/watch?v=Iq5YbRDYVE4>
- ex https://www.ntu.edu.sg/home/ehchua/programming/opengl/CG_Examples.html
- sphere Texture <http://www.angelfire.com/linux/nexusone/projects.html>
- sphere https://www.opengl.org/discussion_boards/showthread.php/137753-Texture-map-on-a-gluS
- s trs https://www.opengl.org/discussion_boards/showthread.php/163561-How-to-posistion-a-gluS
- emacs lambda http://ergoemacs.org/emacs/emacs_pretty_lambda.html
- ani example https://groups.google.com/forum/#!topic/racket-users/ZQ_6_cIirDk

2.4 Texture

- <https://gist.github.com/tonyg/5425736>
- <http://stackoverflow.com/questions/30709454/racket-opengl-glv viewport-not-correctly-mapping>
- <http://lists.racket-lang.org/users/archive/2010-November/043118.html>
- main <https://gist.github.com/tonyg/5425736>

2.5 OOP

- oop <https://docs.racket-lang.org/guide/classes.html>
- creating classes <https://docs.racket-lang.org/reference/createclass.html>
- struct-copy <http://yuyang0.github.io/notes/scheme.html>

2.6 robot dance

- <https://www.youtube.com/watch?v=lacAgc7rv1o>
- <https://www.youtube.com/watch?v=AoCXPicEa8o>
- <https://www.youtube.com/watch?v=wQ4KXoFHwL4>

2.7 other

- framework <https://github.com/NetEase/lively-logic>
- <https://www.youtube.com/watch?v=Sch0zmP6R5A>
- <https://www.youtube.com/watch?v=ayqhX9UA6FY>
- <http://racket.tchen.me/practical-racket.html>
- 图形: <https://www.zhihu.com/question/20789155>
- threads <http://www.ithao123.cn/content-4141200.html>
- <http://docs.racket-lang.org/guide/classes.html>
- <https://docs.racket-lang.org/quick/>
- <http://docs.racket-lang.org/draw/index.html>