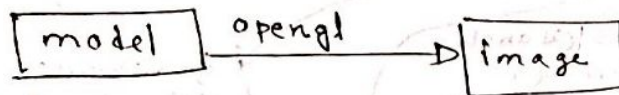


Using OpenGL

OpenGL: একটি API



init() → সব global variable এতে initialise হবে

DisplayFunc (display);

↓
code এর শুরু এখানে display function

KeyboardFunc (Keyboard Listener)

specialFunc (special key) → special key press করলে কী হবে

mainLoop(); একবার নিজে থেকে display function call করবে, call করে event (mouse event, keyboard event) এর জন্য wait করবে, যখন কোন event পাঠবে তখন Idle funct. কে call করবে.

↓
এটি দিয়ে animation করা যায়

glut Post Redisplay () ; (change করার পর ছবিটা যাতে আবার draw হয়)

animate () ; [যা change করতে চাই, তা এটিতে লিখবে]

display () ; [স্ক্রিন আঁকতে চাই, এখানে লিখবে]

gluPerspective (80, 1, 1, 1000.0);

eye
camera-ka parameters set krta

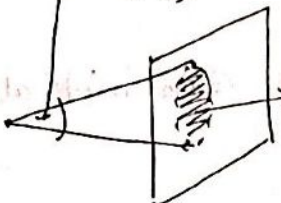
object ke y axis angle ke create krta

x axis (x angle) increment krta

y axis (y angle) increment krta

far value (is value se distance dikha deta)

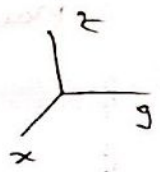
near value (is value se distance dikha deta)



gluLookAt (100, 1000, 1000, 0, 0, 0, 0, 0, 1)

camera (x, y, z) coordinate

camera se dekhne wala direction (x, y, z) coordinate



drawAxes()

3 principal axis (x, y, z)

color set krta, warna color set krta

glBegin (GL_LINES);

glVertex3f (x1, y1, z1);

glVertex3f (x2, y2, z2);

glEnd

drawAxes = 1 (isliye use krta)

is a global variable, init() se initialize krta hai

draw Grid () Grid আঁকবে

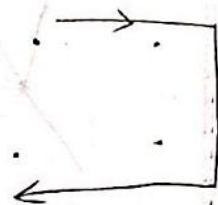
[

drawSquare () {

glBegin (GL_QUADS) {

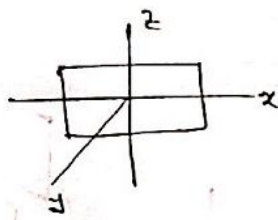
}

}

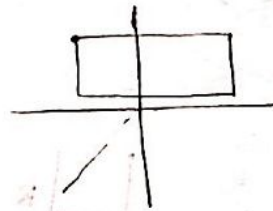


Point
গুলো
serially
দিবে

Translate (orientation change না করে সরানো)



y axis
সরাব
translate
করা হবে



glTranslate (20, 0, 0)

x axis
সরাব 20 unit সরাবে
y axis " সরাবে না

scaling:

glScalef (2, 1, 1)

x axis
বড় হবে
দ্বিগুণ হবে

y axis
বড় হবে
same amount

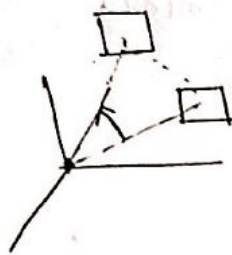
z axis বড় হবে same amount

Rotation: (হবে মূলবিন্দু বা সেন্টার)

glRotates (30, 0, 0, 1) (axis এর সেন্টার notation)

↓ ↓

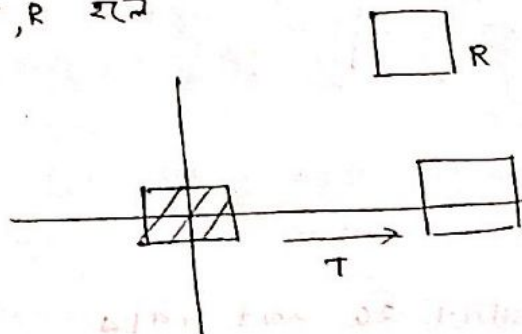
কত ডিগ্রি z axis বরাবর notation



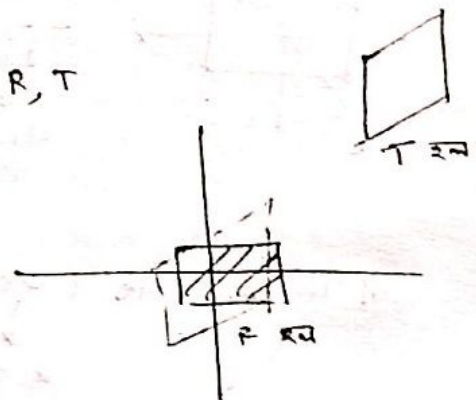
যদি আমি একতরফে translation, scale or Rotation দিই:

Translate ()
rotate ()

T, R হবে

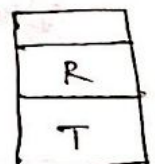


R, T



translate, rotate, scale এর ক্ষেত্রে

প্রথমে rotate হবে, কারণ OpenGL stack maintain করে, stack এ তখন T, R, draw এক সময়! pop হবার সময় প্রথমে R আসবে, পরে T



যদি একটি জিনিষ কে animate করে ঘুরাতে চাই, তাহলে angle এর value increase করে animate function দিচ্ছি,
 ↓
 global variable থাকবে

কার্যিক গতি:

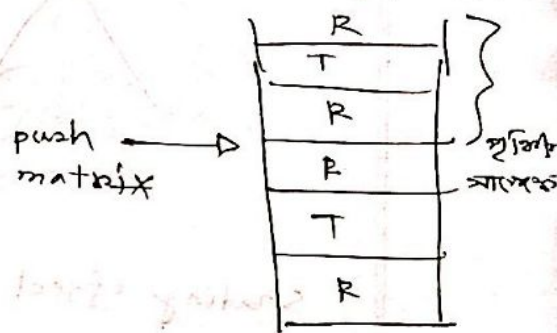
Rotate → কার্যিক গতি
 Translate
 Rotate → কার্যিক গতি

চাঁদ আঁকতে চাইলে, সূর্যের সাপেক্ষে পৃথিবী আঁকার ফলে যে stationary co-ordinate সৃষ্টি হয়েছে তার সাপেক্ষে চাঁদটা আঁক

২ টি চাঁদ আঁকতে চাইলে- যদি আমি আগের মত করে আঁকতে চাই, তাহলে ২য় চাঁদ টা ১ম চাঁদের সাপেক্ষে ছুঁড়বে, but আমরা এটা চাই না, তাই stack থেকে কিছু operation বাদ দিতে হবে।

push matrix call কাজ:

stack এর current location টা save করে রাখে,



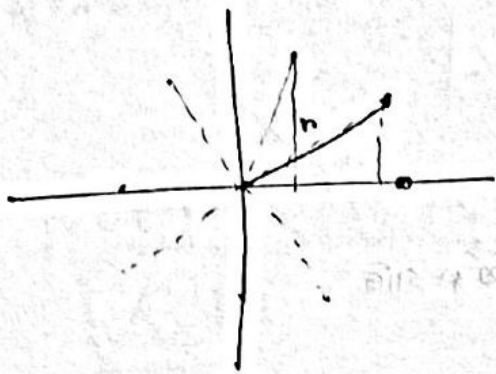
পৃথিবীর stack এর ধরে,

push matrix () {

R
 T
 R

}
 GL PopMatrix()

বৃত্ত আঁকতে গিয়ে:



$$x = r \cos t$$

$$y = r \sin t$$

t change
 করে point
 generate
 করা যায়

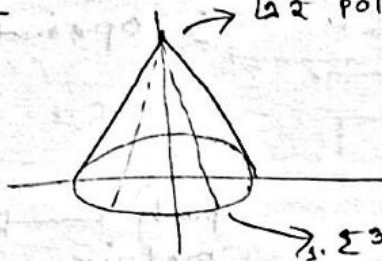
উপস্থাপনা:

$$x = a \cos t$$

$$y = b \sin t$$

বৃত্ত ঘেঁষে একটি axis বসাবার scale করে দিবে।

কৌশল: → এই point থেকে বৃত্তের point গুলিকে line দিয়ে connect করবে



→ বৃত্ত আঁকবে

shading effect লাগানোর জন্য একেকটা line একেকটা color দিবে।

কিভাবে point বা camera যাবার change করতে পারবেন?

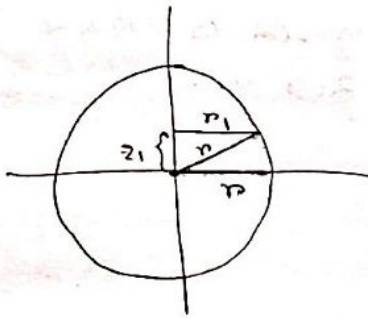
code la যুগিয়ে camera angle change করতে পারবে, ওটা use করবে।

Cylinder কীভাবে আঁকব:

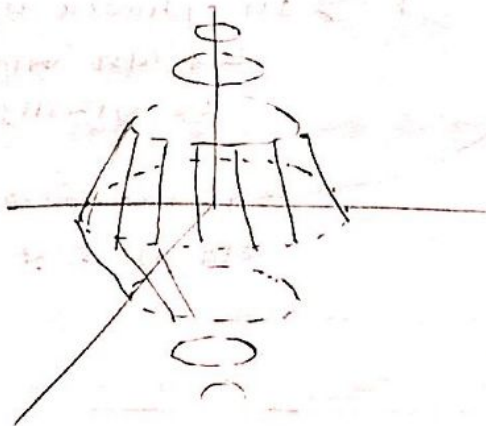
circle এর জন্য যে point গুলো আছে, সেগুলোকে একটা axis
যারপর same পরিমাণ বাড়িয়ে আর 3 কিছু point দাব।
corresponding point গুলো চতুর্ভুজ দিয়ে মোটা করব।

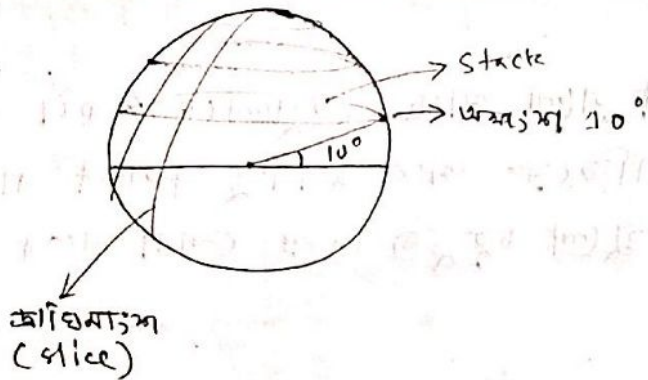


sphere কীভাবে আঁকব:

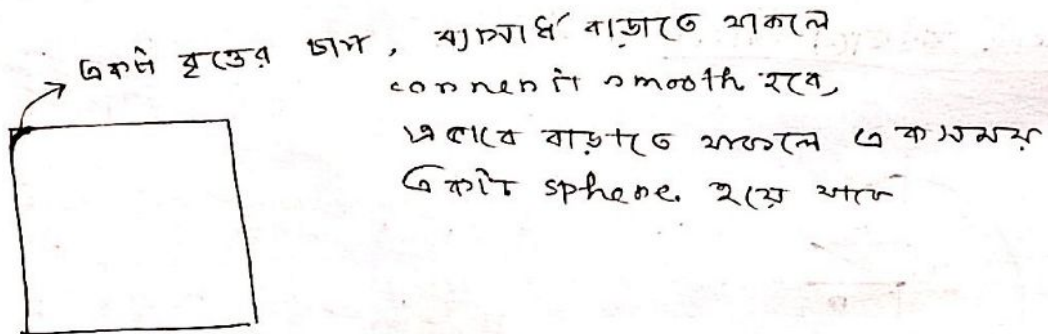


সমস্ত z axis এর থাকবে, ^{corresponding} point গুলো হল circle এর point
 z axis বরাবর যত উল্লেখ করি, যুগ্মভাবে করতে হবে



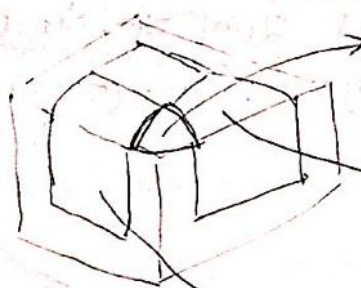


drawsphere (30, 24, 5) → stack কম্পো
→ slice কম্পো



একটি বৃত্তের চাপ, ব্যাসার্ধ বাড়তে থাকলে
connection smooth হবে,

এভাবে বাড়তে থাকলে এক সমকক্ষ
একটি sphere হয়ে যাবে



একটি গোলককে ৪ ভাগের

এক ভাগ,

এটাকে ৪ ভাগে ভাগ করা যায়,
(translate, rotate করে)

→ ১টি cylinder এর ছয় ভাগের

এক ভাগ জোড় ব, এটাকে

১২ ভাগে ভাগ করা যায়

১

৩ axis বরাবর

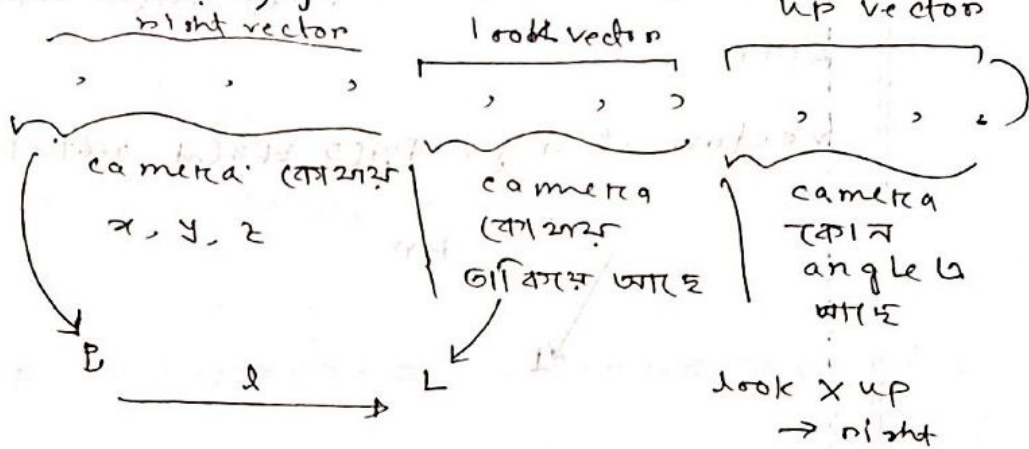
→ ১৭০০ এর জাঁক, ডোটকে

চার ভাগে ভাগ করা যায়

4.09.2016

camera control করার জন্য:

glutlookat (



task 1:

fully controllable camera implement করা

৬টি move এর জন্য

৬টি rotation -

→ look vector বরাবর ঘোরিয়ে আওয়া
look " " বিপরীত দিকে

→ camera নড়বে না, but আমি কোন angle এ ভাবনা-
জানিয়ে বাসে ডাকছি → up vector এর সাপেক্ষে (up change
হবে না)
look ও right change হবে

উলটে, নিচে → up vector rotate হবে

কীভাবে করব?

globally set করব → একটি position - pos

৩টি vector থাকবে, u, r, l (unit vector হিসেবে)

আসারো: $pos + l * 2$

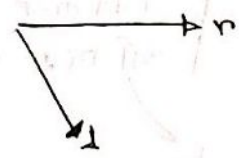
যাবে: $pos + u$ এর opposite * 2

move দিলে
vector এর
direction
change
হবে না

12/11/2017

move দিলে - সে point এক দিকে ডাকাট্টে গেলে ও change হবে।

vector change করতে চাইলে rotate করতে হবে।



task 2: (taurus)

Donought এক inner and outer radius change করতে হবে।



সাদা, কালো zigzag
এ maintain করতে
হবে।

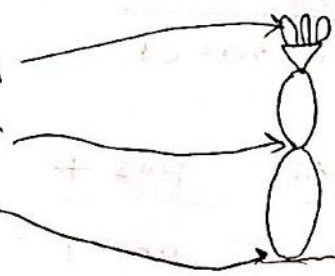
জিগজগৎ বৃত্তের হবে $r_{out} - r_{in}$

- একটা বৃত্তের point generate করা
- আরেকটু বড়িয়ে আরেকটা বৃত্তের point generate করা

task 3:

hand এক motion simulate করা

আঙুলগুলো 0-90° ঘুরবে
কন্ট্রি 0-90° "
main hand -45°-45° "



task 4:

હાકા left, right, up, down

કરો હાકા, ^{કુલ 360°} ફરો 2x ન સરિમાન-માર.

task 1 + ~~task 2~~ → camera-કે integrate કરે દિતે
રહે