

EPL 426

Lab 2 - Miro Ray Tracer

Andreas Andreou

Miro Ray Tracer

- ▶ Όπως είδαμε στο προηγούμενο εργαστήριο, ο Miro είναι ένας Ray Tracer με τον οποίο μπορούμε να παράξουμε πολύ ρεαλιστικά γραφικά
- ▶ Ουσιαστικά είναι χτισμένος σε C++
- ▶ Χρησιμοποιούμε το Visual Studio για να το τρέξουμε
- ▶ Ο Miro παρέχει τη δυνατότητα θέασης ενός OpenGL Wireframe προτού γίνει το Ray Tracing και παραχθεί η τελική εικόνα
- ▶ Στο Wireframe μπορούμε ουσιαστικά να αλλάξουμε τη θέση της camera
- ▶ Η Camera μετακινείται με τα πλήκτρα W,A,S,D,Q,Z
- ▶ Με τη χρήση του mouse περιστρέφεται

Miro Ray Tracer

- ▶ Πατώντας το κουμπί “r” ξεκινάειτο ray tracing και παράγει την τελική εικόνα
- ▶ Πατώντας το κουμπί “g” επιστρέφουμε και πάλι στο wireframe
- ▶ Πατώντας το κουμπί “l” αποθηκεύεται η παραγόμενη εικόνα στον τρέχον φάκελο με όνομα miro_xxxx.rpm όπου “xxxx” ο χρόνος που χρειάστηκε για να γίνει το Ray Tracing
- ▶ Εμείς θα ασχοληθούμε με το αρχείο main.cpp που βρίσκεται στο φάκελο Source files και αποτελεί το αρχείο με το οποίο θα στήσουμε τη σκηνή μας και έπειτα θα δημιουργήσουμε την εικόνα που επιθυμούμε
- ▶ Παρατηρώντας το αρχείο, βλέπουμε ότι είναι ένα κοινό αρχείο cpp με τις δηλώσεις στην αρχή, κάποιες συναρτήσεις και τη συνάρτηση main μέσα στην οποία και καλείται και εκτελείται ο miro

Miro Ray Tracer

Καλούμε τη συνάρτηση που στήνει τη σκηνή μας

Καλούμε το miro ο οποίος ξεκινά το ray tracing με βάση τη σκηνή που δημιουργήσαμε προηγουμένως

```
int main(int argc, char*argv[])  
{  
    // create a scene  
    makeSpiralScene();  
  
    MiroWindow miro(&argc, argv);  
    miro.mainLoop();  
  
    return 0; // never executed  
}
```

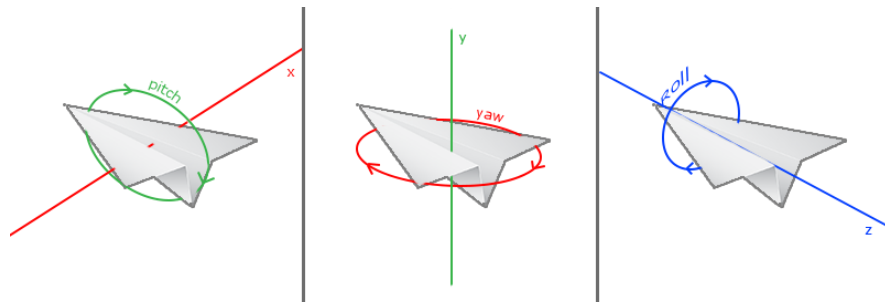
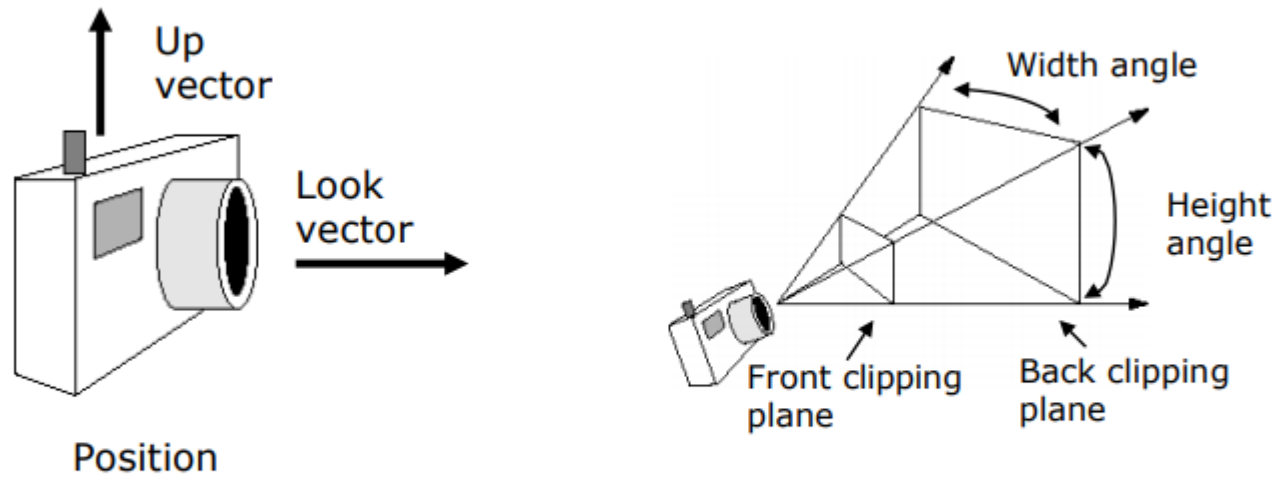
Miro Ray Tracer

- ▶ Για να στήσουμε μια σκηνή στο Miro αλλά και σε οποιοδήποτε άλλο πρόγραμμα είτε ray tracing είτε άλλων μεθόδων, χρειαζόμαστε τα ακόλουθα:
 - ▶ Camera
 - ▶ Lights
 - ▶ Objects

Camera



Camera



Miro Camera set

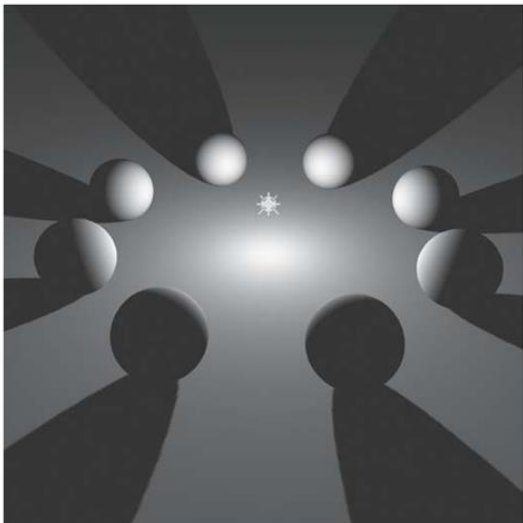
- ▶ Τα πιο πάνω στο Miro αποτυπώνονται από τις πιο κάτω εντολές:
 - ▶ setEye
 - ▶ setLookAt
 - ▶ setup
 - ▶ setFOV

```
void makeSpiralScene()
{
    g_camera = new Camera;
    g_scene = new Scene;
    g_image = new Image;

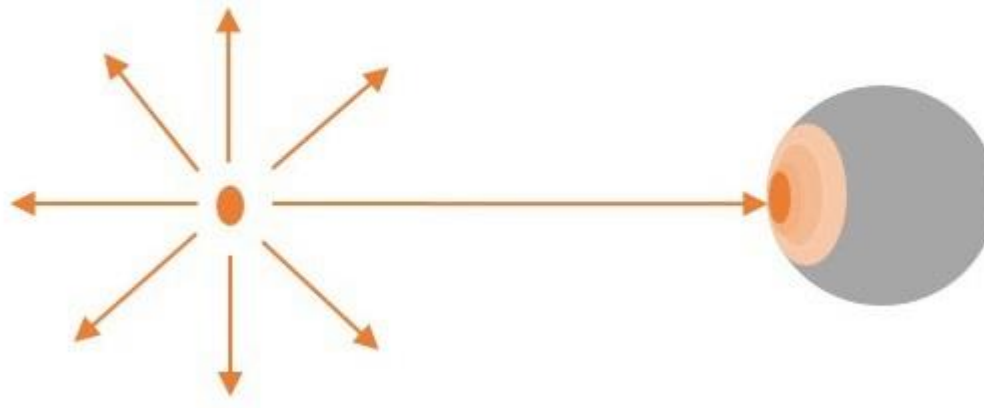
    g_image->resize(512, 512);

    // set up the camera
    g_camera->setBGColor(Vector3(1.0f, 1.0f, 1.0f));
    g_camera->setEye(Vector3(-5, 2, 3));
    g_camera->setLookAt(Vector3(0, 0, 0));
    g_camera->setUp(Vector3(0, 1, 0));
    g_camera->setFOV(45);
}
```


Lights



Point light Source

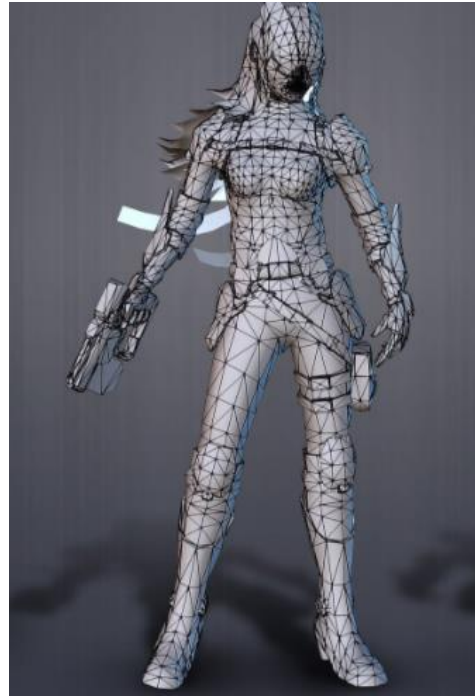
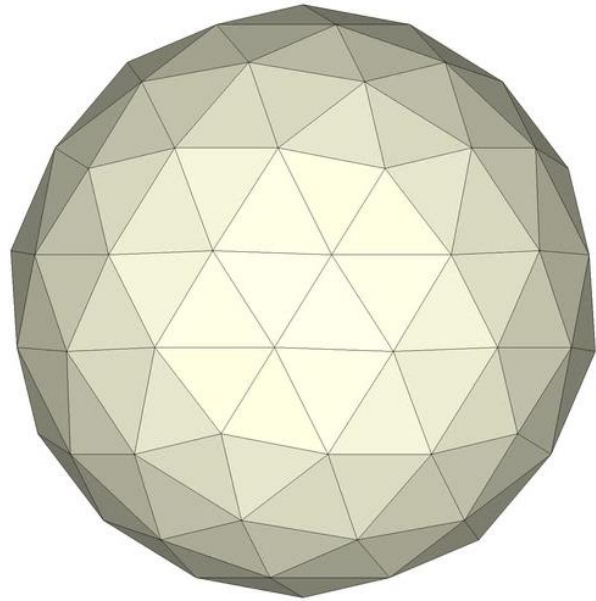


Miro Lights set

- ▶ Το επόμενο βήμα είναι η προσθήκη φωτισμού (Point light)
- ▶ Για το φωτισμό χρειαζόμαστε τη θέση του φωτός, το χρώμα και την έντασή του
- ▶ Τα πιο πάνω στο Miro αποτυπώνονται από τις πιο κάτω εντολές:
 - ▶ setPosition
 - ▶ setColor
 - ▶ setWattage

```
// create and place a point light source
PointLight * light = new PointLight;
light->setPosition(Vector3(-3, 15, 3));
light->setColor(Vector3(1, 1, 1));
light->setWattage(1000);
g_scene->addLight(light);
```

Objects

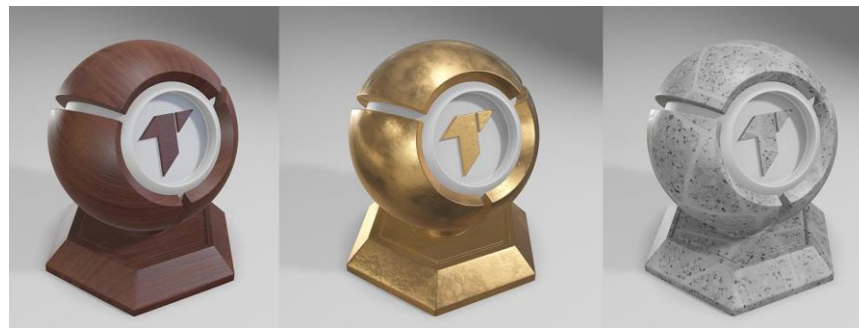


Miro Lights set

- ▶ Το επόμενο και τελευταίο βήμα είναι να προσθέσουμε τα αντικείμενά μας στη σκηνή
- ▶ Το Miro προσφέρει τη δυνατότητα προσθήκης σφαιρών και τριγώνων
- ▶ Αν θα χρησιμοποιήσουμε σφαίρες, τότε το μόνο που χρειαζόμαστε είναι το κέντρο της σφαίρας και η ακτίνα της
- ▶ Αντίστοιχα ισχύουν και για τα τρίγωνα

```
Sphere * sphere = new Sphere;  
sphere->setCenter(Vector3(x,y,z));  
sphere->setRadius(r/10);  
sphere->setMaterial(mat);  
g_scene->addObject(sphere);
```

Materials



Miro Materials set

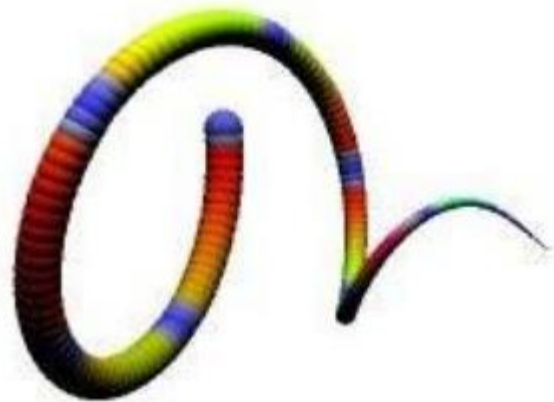
- ▶ Τέλος, μας δίνεται η δυνατότητα, σε κάθε αντικείμενο (πχ σε κάθε σφαίρα) να προσθέσουμε ένα υλικό το οποίο περιέχει κάποιες ιδιότητες αντανάκλασης και χρώματος
- ▶ Αυτό γίνεται χρησιμοποιώντας την κλάση Material

```
// create a sphere of spheres  
Material* mat = new Lambert(Vector3(1.0f, 0.0f, 0.0f));
```

```
sphere->setRadius(1/10);  
sphere->setMaterial(mat);
```

Miro practice 1

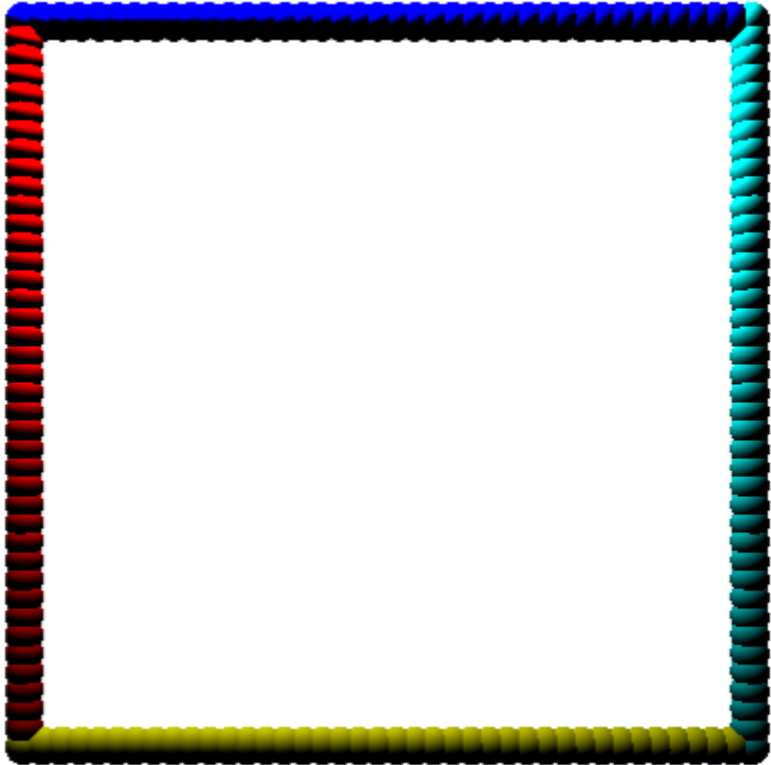
- ▶ Παίξτε λίγο με τον κώδικα που σας έχει δοθεί, που είναι για την παραγωγή μιας σπείρας
- ▶ Δοκιμάστε να αλλάξετε τα χρώματα της σπείρας και να πάρετε το πιο κάτω αποτέλεσμα



Hints:

- Color is a vector3 between (0,0,0) and (1,1,1)
 - <https://www.tug.org/pracjourn/2007-4/walden/color.pdf>
 - https://www.google.com/search?q=colorpicker&rlz=1C1GCEA_enCY907CY907&oq=colorpicker&aqs=chrome..69i57.4135j0j7&sourceid=chrome&ie=UTF-8
- Lerp = Interpolates between the points a and b by the interpolant t. The parameter t is clamped to the range [0, 1]

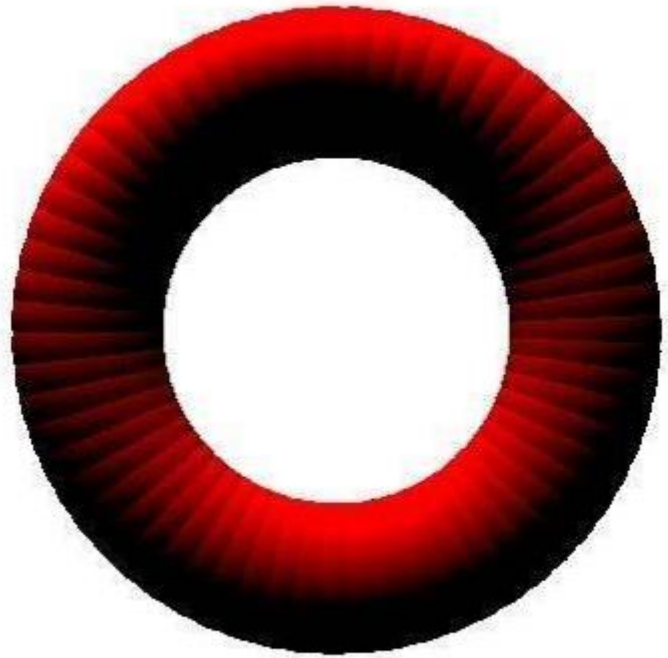
Miro practice 2



Hints:

- Colors
 - $(1.0f, 0.0f, 0.0f)$
 - $(0.0f, 0.0f, 1.0f)$
 - $(1.0f, 1.0f, 0.0f)$
 - $(0.0f, 1.0f, 1.0f)$

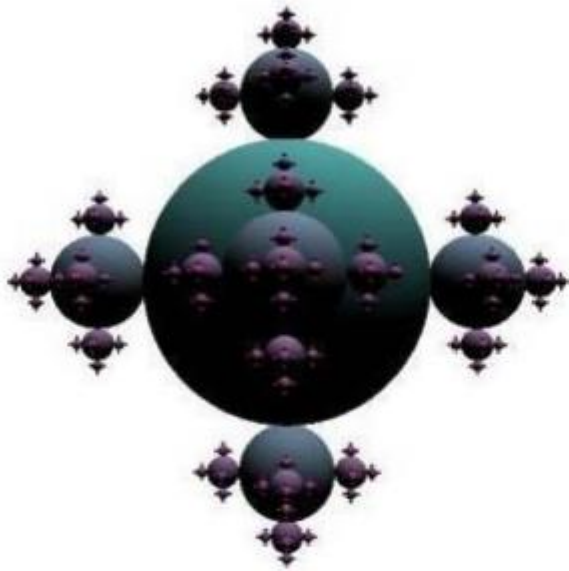
Miro practice 3



Hints:

- Παραμετρικές εξισώσεις του κύκλου
 - $xx=rr*\text{costt}$
 - $yy= rr*\text{sintt}$
 - $0 \leq tt \leq 2\pi$

Miro practice 4



Hints:

- 1st sphere to $(0,0,0)$ with r
- Create 6 new spheres with $r/3$
 - Up
 - Down
 - Left
 - Right
 - Front
 - Back
- Distance from the 1 sphere is $r+r/3$