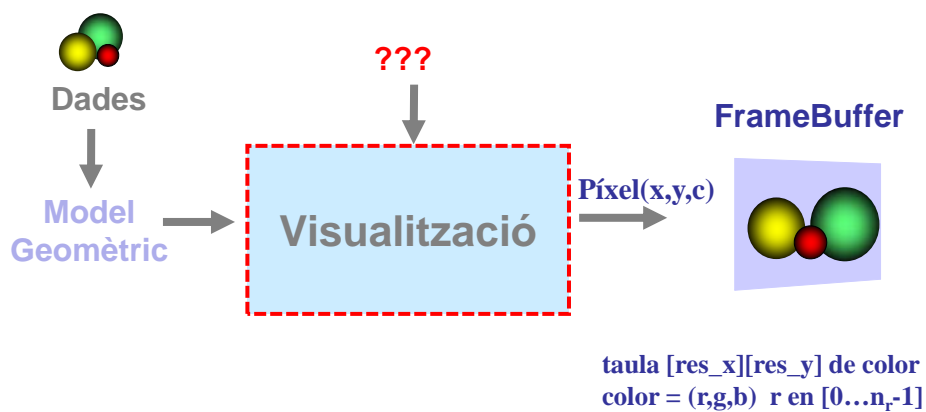
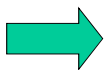


Elements d'un sistema gràfic: Visualització (1)



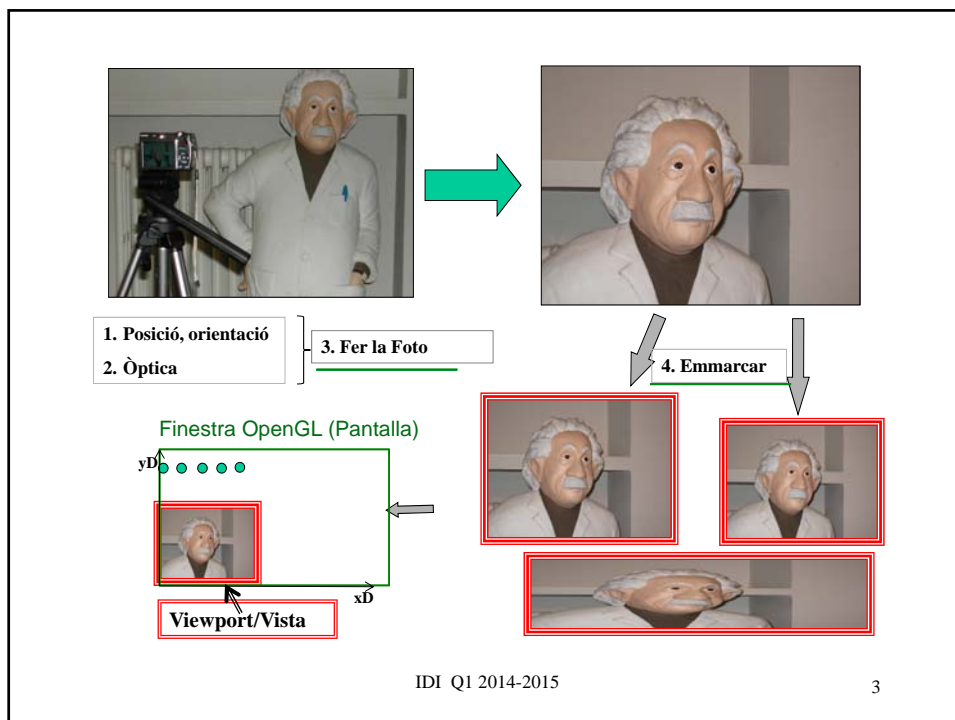
IDI Q1 2014-2015

1

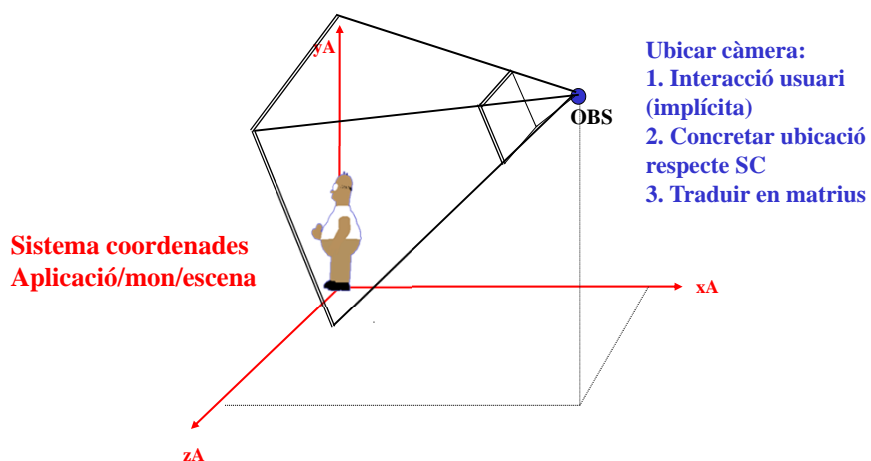


IDI Q1 2014-2015

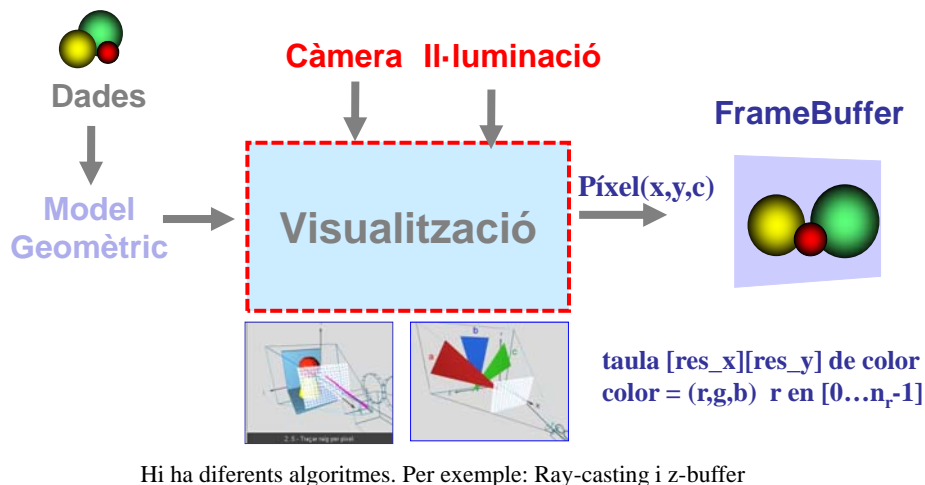
2



Com indicar posicionament?



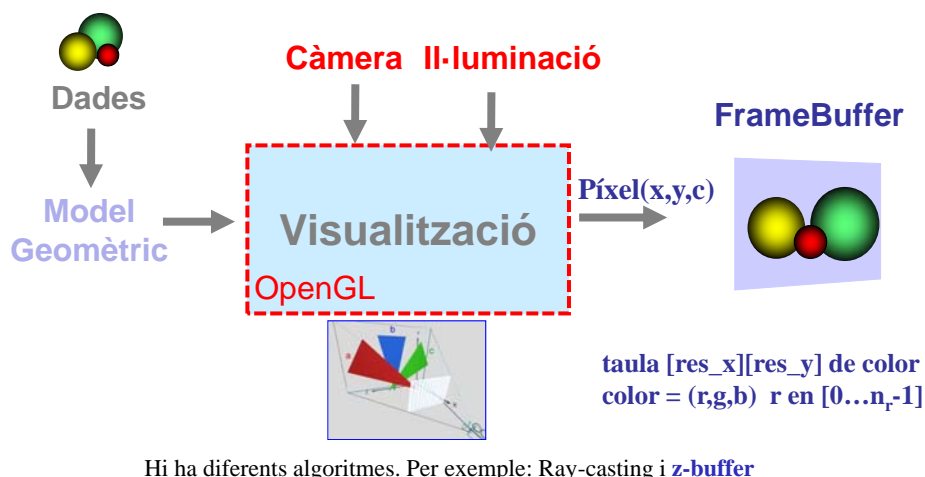
Elements d'un sistema gràfic: Visualització (1)



IDI Q1 2014-2015

5

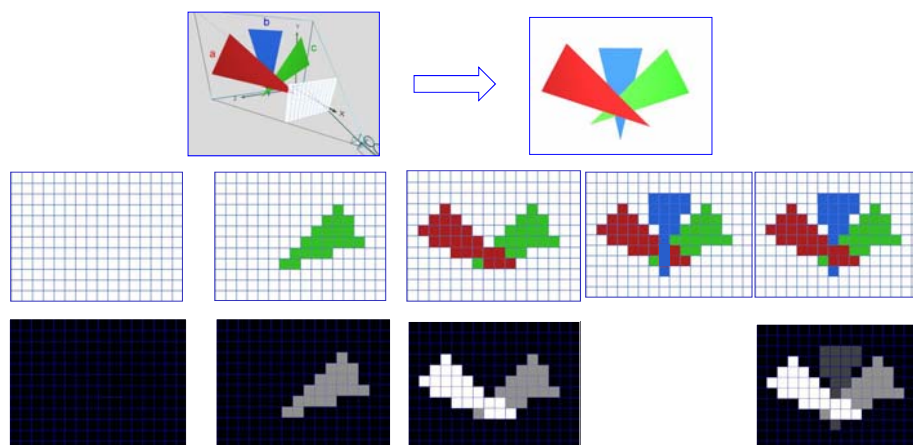
Elements d'un sistema gràfic: Visualització (1)



IDI Q1 2014-2015

6

Depth Buffer (z-buffer)



IDI Q1 2014-2015

7

Visualització OpenGL

1. Netejar finestra → `glClear(GL_COLOR_BUFFER_BIT | ...);`
2. Definir càmera i viewport
 - 2a Definir Vista/Viewport → `glViewport(...)`
 - 2b Posició i orientació
Model-view Transformation → `gluLookAt (...)` o “movent” escena → ModelView Matrix
 - 2c Optica
Projection Transformation → `gluPerspective(...)` o `glOrtho(...)` → Projection Matrix
3. Recorregut de l'escena i pintar cares →


```
glBegin (GL_POLYGON);
for (i=id; i<id+nV; i++)
    glVertex3d (v[i].x, v[i].y,v[i].z);
glEnd();
```



IDI Q1 2014-2015

8

glBegin (GL_Triangles)
glVertex (v1)
glVertex (v2)
glVertex (v3)
glEnd

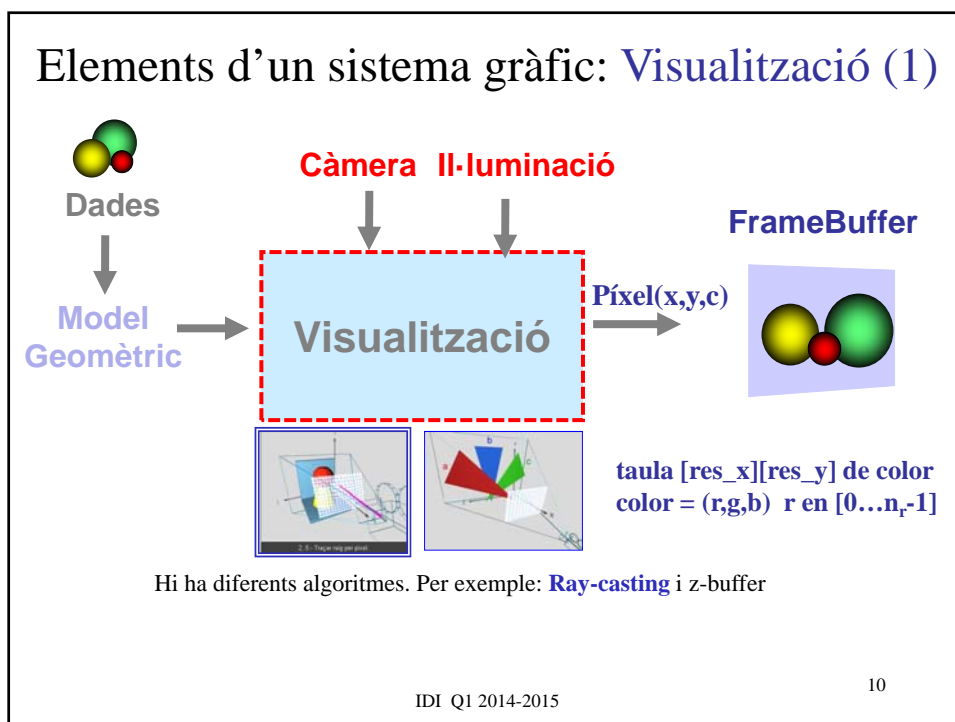
Repasar Àlgebra i TG

Per ubicar objectes/càmera

versus càmera/objectes

Window i viewport han de
tenir la mateixa relació
d'aspecte: amplada/alçada

IDI Q1 2014-2015
9



Ray Casting

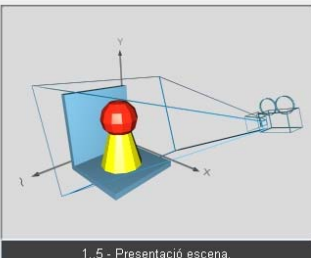
```

accio ray_casting
per y en [vyM...vym] fer
  per x en [vxm...vxM] fer
    r:=calcul_raig(x,y,Obs)
    per cada cara fer
      trobar_interseccions_raig_cara
    fper
  ordenar_interseccions
fper
fper
faccio

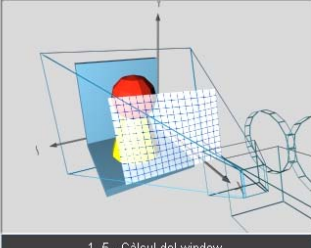
```

Anterior

Següent



1..5 - Presentació escena.



1..5 - Càlcul del window.

11

IDI Q1 2014-2015

Ray Casting

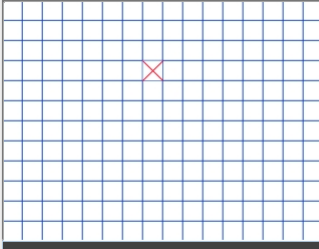
```

accio ray_casting
per y en [vyM...vym] fer
  per x en [vxm...vxM] fer
    r:=calcul_raig(x,y,Obs)
    per cada cara fer
      trobar_interseccions_raig_cara
    fper
  ordenar_interseccions
fper
fper
faccio

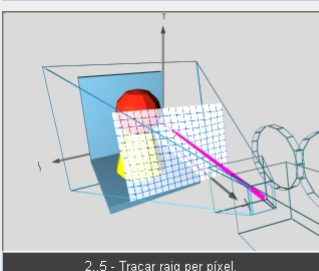
```

Anterior

Següent



2..5 - Donat un pixel.



2..5 - Traçar raig per pixel.

12

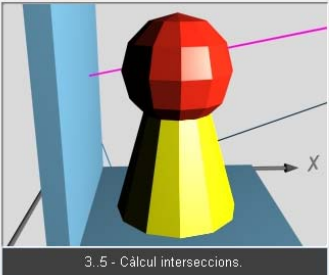
IDI Q1 2014-2015

Ray Casting

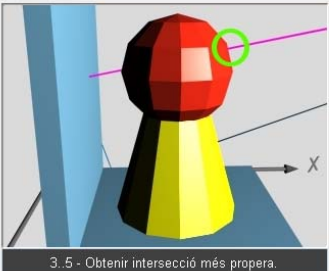
```

accio ray_casting
per y en [vyM...vym] fer
per x en [vxm...vxM] fer
r:=calcul_raig(x,y,Obs)
per cada cara fer
  trobar_interseccions_raig_cara
fper
ordenar_interseccions
fper
fper
faccio
        
```

Anterior
Següent



3.5 - Càlcul interseccions.



3.5 - Obtenir intersecció més propera.

13

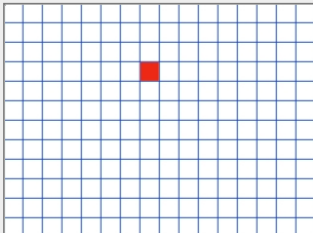
IDI Q1 2014-2015

Ray Casting

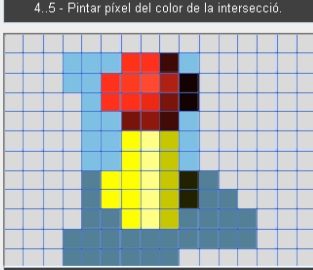
```

accio ray_casting
per y en [vyM...vym] fer
per x en [vxm...vxM] fer
r:=calcul_raig(x,y,Obs)
per cada cara fer
  trobar_interseccions_raig_cara
fper
ordenar_interseccions
fper
fper
faccio
        
```

Anterior
Següent



4.5 - Pintar píxel del color de la intersecció.



4.5 - Resultat en baixa resolució.

14

IDI Q1 2014-2015

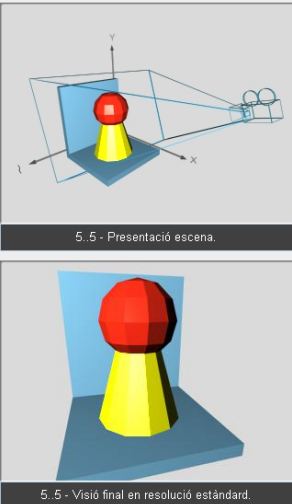
Ray Casting

```

accio ray_casting
per y en [vyM...vym] fer
  per x en [vxm...vxdM] fer
    r:=calcul_raig(x,y,Obs)
    per cada cara fer
      trobar_interseccions_raig_cara
      fper
        ordenar_interseccions
      fper
        fper
        facció
          
```

Anterior

Següent



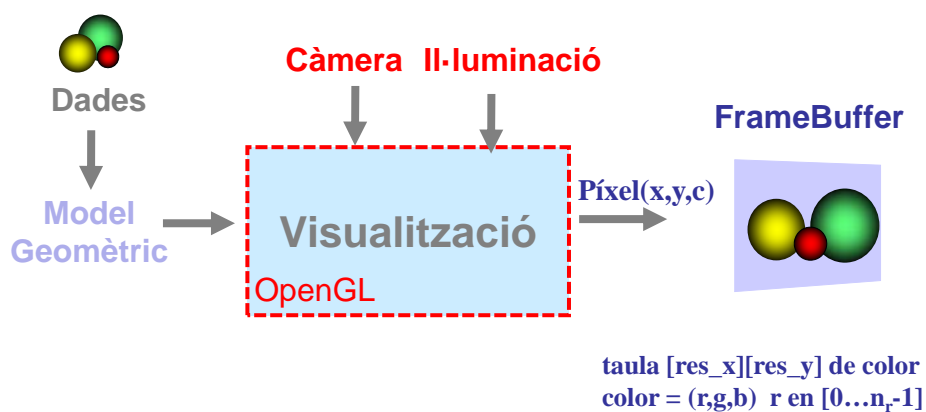
5.5 - Presentació escena.

5.5 - Visió final en resolució estàndard.

IDI Q1 2014-2015

15

Elements d'un sistema gràfic: Visualització (1)



IDI Q1 2014-2015

16

Primer exemple

```
#include <GL/gl.h>
#include <GL/freeglut.h>
void refresh (void)
{ glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glBegin(GL_TRIANGLES);
    glColor (0.5,0.5,0.5)
    glVertex3f(-0.5,-0.5,0.0);
    glVertex3f(0.5,0.0,0.0);
    glVertex3f(0.0,0.5,0.0);
  glEnd();
  glutSwapBuffers();
}
int main(int argc, const char *argv[])
{ glutInit(&argc, (char **) argv);
  glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
  glutInitWindowSize(600,600);
  glutCreateWindow("IDI: Practiques OpenGL");
  InitGL(); //inicialitzacions OpenGL –si calen-
  glutDisplayFunc (refresh);
  ...
  glutMainLoop();
  return 0;
}
```

1. Inicialitzacions
glut

2. Registre *callbacks*

3. Bucle processament events

IDI Q1 2014-2015

17