.CPP

void MyGLWidget::createBuffersModelPatricio()

{

// Carreguem el model de l'OBJ - Atenció! Abans de crear els buffers!

patricio.load("./models/Patricio.obj");

// Calculem la capsa contenidora del model

calculaCapsaModel (patricio, escalaModelPatricio, centreBaseModelPatricio);

// Creació del Vertex Array Object del Model

glGenVertexArrays(1, &VAO\_ModelPatricio);

glBindVertexArray(VAO\_ModelPatricio);

// Creació dels buffers del model patr

// Buffer de posicions

glGenBuffers(1, &VBO\_ModelPosPatricio);

glBindBuffer(GL\_ARRAY\_BUFFER, VBO\_ModelPosPatricio);

glBufferData(GL\_ARRAY\_BUFFER, sizeof(GLfloat)\*patricio.faces().size()\*3\*3, patricio.VBO\_vertices(), GL\_STATIC\_DRAW);

// Activem l'atribut vertexLoc

glVertexAttribPointer(vertexLoc, 3, GL\_FLOAT, GL\_FALSE, 0, 0);

glEnableVertexAttribArray(vertexLoc);

// Buffer de normals

glGenBuffers(1, &VBO\_ModelNormPatricio);

glBindBuffer(GL\_ARRAY\_BUFFER, VBO\_ModelNormPatricio);

glBufferData(GL\_ARRAY\_BUFFER, sizeof(GLfloat)\*patricio.faces().size()\*3\*3, patricio.VBO\_normals(), GL\_STATIC\_DRAW);

glVertexAttribPointer(normalLoc, 3, GL\_FLOAT, GL\_FALSE, 0, 0);

glEnableVertexAttribArray(normalLoc);

// En lloc del color, ara passem tots els paràmetres dels materials

// Buffer de component ambient

glGenBuffers(1, &VBO\_ModelMatambPatricio);

glBindBuffer(GL\_ARRAY\_BUFFER, VBO\_ModelMatambPatricio);

glBufferData(GL\_ARRAY\_BUFFER, sizeof(GLfloat)\*patricio.faces().size()\*3\*3, patricio.VBO\_matamb(), GL\_STATIC\_DRAW);

glVertexAttribPointer(matambLoc, 3, GL\_FLOAT, GL\_FALSE, 0, 0);

glEnableVertexAttribArray(matambLoc);

// Buffer de component difusa

glGenBuffers(1, &VBO\_ModelMatdiffPatricio);

glBindBuffer(GL\_ARRAY\_BUFFER, VBO\_ModelMatdiffPatricio);

glBufferData(GL\_ARRAY\_BUFFER, sizeof(GLfloat)\*patricio.faces().size()\*3\*3, patricio.VBO\_matdiff(), GL\_STATIC\_DRAW);

glVertexAttribPointer(matdiffLoc, 3, GL\_FLOAT, GL\_FALSE, 0, 0);

glEnableVertexAttribArray(matdiffLoc);

// Buffer de component especular

glGenBuffers(1, &VBO\_ModelMatspecPatricio);

glBindBuffer(GL\_ARRAY\_BUFFER, VBO\_ModelMatspecPatricio);

glBufferData(GL\_ARRAY\_BUFFER, sizeof(GLfloat)\*patricio.faces().size()\*3\*3, patricio.VBO\_matspec(), GL\_STATIC\_DRAW);

glVertexAttribPointer(matspecLoc, 3, GL\_FLOAT, GL\_FALSE, 0, 0);

glEnableVertexAttribArray(matspecLoc);

// Buffer de component shininness

glGenBuffers(1, &VBO\_ModelMatshinPatricio);

glBindBuffer(GL\_ARRAY\_BUFFER, VBO\_ModelMatshinPatricio);

glBufferData(GL\_ARRAY\_BUFFER, sizeof(GLfloat)\*patricio.faces().size()\*3, patricio.VBO\_matshin(), GL\_STATIC\_DRAW);

glVertexAttribPointer(matshinLoc, 1, GL\_FLOAT, GL\_FALSE, 0, 0);

glEnableVertexAttribArray(matshinLoc);

glBindVertexArray(0);

}

.H

GLuint VAO\_ModelPatricio, VBO\_ModelPosPatricio, VBO\_ModelNormPatricio, VBO\_ModelMatambPatricio, VBO\_ModelMatdiffPatricio, VBO\_ModelMatspecPatricio, VBO\_ModelMatshinPatricio;

Model patricio;