

.CPP

```
void MyGLWidget::createBuffersModelPatricio()
{
    // Carreguem el model de l'OBJ - Atenció! Abans de crear els buffers!
    patricio.load("./models/Patricio.obj");

    // Calculem la capsula contenidora del model
    calculaCapsulaModel (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 shininess
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;