

# Grid

```
// GridGame
gridGame:public CApp
{
private:
    Grid * m_pGrid; // Este representa el grid completo
    vector <CGameobject*> m_pGameObjects; // Lista master de objetos
    vector <C3DObject*> m_pMeshes;
    initialize(); // m_inGrid = new Grid(...)
}

// CGameobject
CGameobject
{
    C3DObject * m_pMesh;
    float m_scale;
}

// CGrid
CGrid
{
private:
    GridCell * m_pGridCell; // Este apuntador, apunta a todas las celdas
    int m_size; // 3
    int m_cellsize;
    void buildWorld(); // Coordenadas de nuestros objetos
    void Load3DObjects(); // Cargar diferentes modelos de meshes
    // Para cada celda calcular los 6 vertices
    // en este for se define el tipo de grid, pointy o flat
}

// GridCell - Representa cada celda
CGridCell
{
    CGameobject *m_pGameobject; // Se enlaza con m_gameObjects
    CVector3D m_pVertices[6]; // En base a estos vertices , realizar triangulos para
render.
    CVector3D m_CenterPoint;
    int x, y , z; // No se esta seguro si sera necesario para calcular la coordenada
}
```