

设计目标：

1. 优化性能：确保摄像机移动时，只有摄像机视野范围内的 gameobject 被渲染，减少计算和渲染负担。
2. 动态加载：实现 gameobject 的动态加载和卸载，以适应不同区域的需要。
3. 高效查询：快速查询和访问摄像机视野范围内的 gameobject。
4. 数据一致性：保证 gameobject 的状态在摄像机移动时能够及时更新。

管理机制设计：

1. 空间划分

使用四叉树或八叉树对 2D 场景进行空间划分。这样可以快速确定哪些 gameobject 位于摄像机视野内。以下是伪代码

```
class Quadtree
    constructor(bounds)
        this.bounds = bounds
        this.objects = []
        this.nodes = []

    function insert(object)
        if this.bounds.contains(object)
            if this.nodes.length == 0
                this.objects.push(object)
            else
                for each node in this.nodes
                    node.insert(object)
            else
                this.objects.push(object)

    function query(cameraView)
        result = []
        if this.bounds.intersects(cameraView)
            if this.objects.length > 0
                for each object in this.objects
                    if cameraView.contains(object)
                        result.push(object)
            for each node in this.nodes
                node.query(cameraView)
        return result
```

2. 动态加载与卸载

根据摄像机的位置动态加载和卸载 gameobject。
以下是伪代码

```
function update(cameraPosition)
    visibleObjects = quadtree.query(cameraPosition.getView())
    for each object in visibleObjects
        if not object.isLoaded
            loadObject(object)
    for each object in allObjects
        if not cameraPosition.getView().contains(object) and object.isLoaded
            unloadObject(object)
```

3. 摄像机视野更新

摄像机移动时，更新其视野范围内的 gameobject。
以下是伪代码

```
function onCameraMove(newPosition)
    currentView = calculateViewFromPosition(newPosition)
    objectsInView = quadtree.query(currentView)
    render(objectsInView)
```

4. 渲染优化

只渲染摄像机视野内的 gameobject。
以下是伪代码

```
function render(objects)
    for each object in objects
        drawObject(object)
```

5. 状态同步

确保 gameobject 的状态与摄像机视野同步。
以下是伪代码

```
function synchronizeObjectStates(objects)
    for each object in objects
        updateObjectState(object)
```