C++ Code

```cpp
#include "MediaManager.h"

// Create Photo
std::shared_ptr<Photo> MediaManager::createPhoto(const std::string &name, const std::string &filename, double lat, double lon)
{
    std::shared_ptr<Photo> p(new Photo(name, filename, lat, lon));
    objects[name] = std::static_pointer_cast<MultimediaObject>(p);
    return p;
}

// Create Video
std::shared_ptr<Video> MediaManager::createVideo(const std::string &name, const std::string &filename, int duree)
{
    std::shared_ptr<Video> v(new Video(name, filename, duree));
    objects[name] = std::static_pointer_cast<MultimediaObject>(v);
    return v;
}

// Create Film
std::shared_ptr<Film> MediaManager::createFilm(const std::string &name, const std::string &filename, int duree)
{
    std::shared_ptr<Film> f(new Film(name, filename, duree));
    objects[name] = std::static_pointer_cast<MultimediaObject>(f);
    return f;
}

// Create Groupe
GroupePtr MediaManager::createGroupe(const std::string &name)
{
    GroupePtr g(new Groupe(name));
    groups[name] = g;
    return g;
}

// Display object
void MediaManager::displayObject(const std::string &name, std::ostream &out) const
{
    auto it = objects.find(name);
    if (it == objects.end())
    {
        std::cout << "Objet '" << name << "' introuvable." << std::endl;
        return;
    }
    it->second->affiche(out);
    out << std::endl;
}

```

```cpp
// Display groupe
void MediaManager::displayGroupe(const std::string &name, std::ostream &out) const
{
    auto it = groups.find(name);
    if (it == groups.end())
    {
        std::cout << "Groupe '" << name << "' introuvable." << std::endl;
        return;
    }
    it->second->affiche(std::cout);
    std::cout << std::endl;
}

// Play object
void MediaManager::playObject(const std::string &name, std::ostream &out) const
{
    auto it = objects.find(name);
    if (it == objects.end())
    {
        out << "Objet '" << name << "' introuvable." << std::endl;
        return;
    }
    it->second->jouer(out);
}

// Remove object
bool MediaManager::removeObject(const std::string &name)
{
    auto it = objects.find(name);
    if (it == objects.end())
        return false;
    objects.erase(it);
    return true;
}

// Remove groupe
bool MediaManager::removeGroupe(const std::string &name)
{
    auto it = groups.find(name);
    if (it == groups.end())
        return false;
    groups.erase(it);
    return true;
}

```