Major Enhancement

M2:
add a new ship_ability that has all the command function. Then ship and group inherit from that base class. Group has a vector<shared_ptr<ship_ability>>. For each command, just use a for loop. Group name has to be different from ship name and island name.

Commandable:
public:
    void update() = 0
    void describe() = 0;
    void set_destination_position_and_speed(Point destination_position, double speed) = 0
    void set_destination_island_and_speed();
    void set_course_and_speed(double course, double speed);
    void stop();
    void dock();
    void set_load_destination(std::shared_ptr<Island>)
    void set_unload_destination(std::shared_ptr<Island>);
    void attack(std::shared_ptr<Ship> in_target_ptr);
    void stop_attack();

Ship: public Movable, public Sim_object {
    void update() override {if(state == …) …}
}

Group: public Movable {
public:
    void update() override (for(member : members) { member->update(); })
    void set_load_destination() { for(member : members) { try{ member->set_load_destination() }}}}
    add_member(Movable*);
    delete_member(Movable*);
    is_member_present(Movable*);
private:
    vector<Movable*> members;
}

Controller:
create_group(group_name). Group name can not be the same as the existing name.
add_member(string group_name, string object_name).

Model:
is_group_name_valid(string name)
is_group_present(string name)
get_group_ptr(string name)
attach_group(Group* group_ptr);
detach_group(string name)
vector<Group*>

Notes:
Add a different exception to Utility.

Changes to original spec:

M4: Save/store
Controller
public:
    save_cmd() {
        save Controller views. call Model save
    }
    restore_cmd() {
        restore views. call Model attach_view(). call Model save
    }

Ship Factory
public:
    restore_ship() {
        if (type == “Tanker”) {
            return Tanker(fin);
        }

Save function:

Format:
number of views
Map/Sailing/Bridge/GPS
…
….
map data
…
…