April 25th 2016

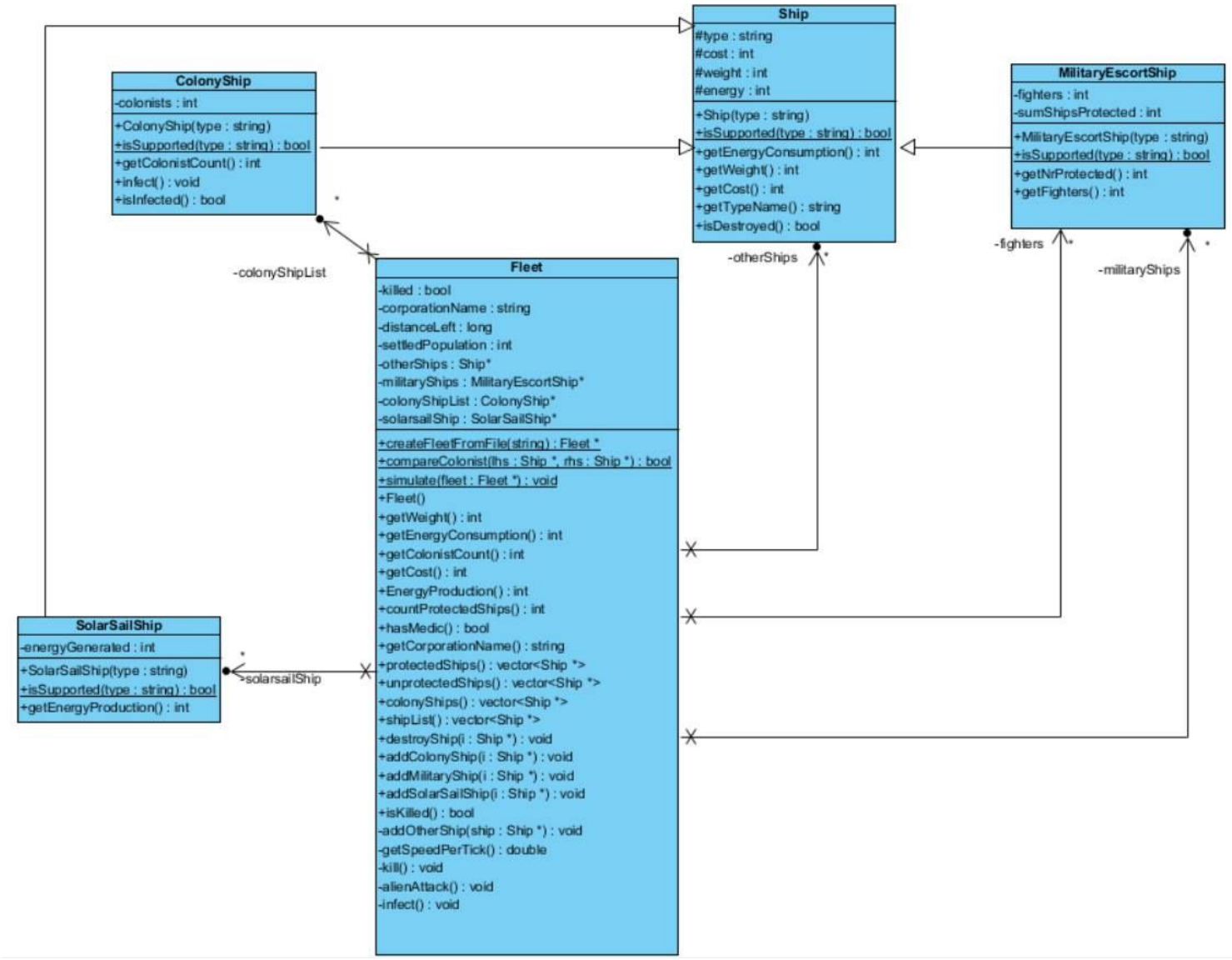
G52CPP Coursework Report - Race to Gaia

Farid Nouri Neshat(015606)
Ahmad Kamal Bin Kamariza(012960)
Siew Qing Want(013051)

How to run

Basically compile with `g++ -pthread -stl=c++11 Fleet.cpp -o out` and then run the out file. It'll then ask about file names which then after entering them all it'll run the simulation.

Class Diagram



Basically in this program, in this program, the main program will have one fleet for each player and each one is simulated in one thread. One fleet contains many ships and each ship will be stored in their own respecteable containers(colonyShipList, soloarSailShips, militaryShips and otherShips). otherShips will be used for medic and destroyed ships. This approach of using multiple vectors gave us advantage of less bugs since other approach of having single vector to contain them all would need dynamic casting in many places which is error prune.

Class Specifications

Class name: Fleet Component of: Main program Component classes: Ship, ColonyShip, SolarSailShip, MilitaryEscrortShip	
Function: To simulate a fleet that contains all the ships and the information needed for the game, including attack simulation on them	
Variables	Description
vector<Ship* > otherShips	a vector of ships other than military, colony, and solarsail ship
vector<MilitaryEscortShip* > militaryShips	a vector of military ships
vector<ColonyShip* > colonyShipList	a vector of colony ships
vector<SolarSailShip* > solarSailShips	a vector of solarsail ships
bool killed	a boolean whether the fleet is killed or not. When a ship is killed it's not in the game anymore.
string corporationName	the name of the corporation given to the fleet. In our case name of the file.
long distanceLeft	a total distance left to reach planet Gaia
int settledPopulation	amount of population that is settled in gaia from this fleet given this fleet is currently settled there.
Operations	Description
fleet *createFleetFromFile()	read and create a fleet from file
bool compareColonist()	return a boolean to compare which ship among two ships that has more colonist
void simulate	to simulate the game using the fleet created
int getFighters()	return the cumulative fighters in a fleet
int getWeight()	return the cumulative weight of a fleet
int getEnergyConsumption()	return cumulative energy consumption of fleet
int getColonistCount()	return cumulative colonist count of fleet
int getCost()	return cumulative cost of fleet
int EnergyProduction	return cumulative energy production of fleet
int countProtectedShips()	return number of ships get protected
double getSpeedPerTick()	return the speed of fleet
bool hasMedic()	return a boolean whether the fleet has Medic ship

bool isKilled()	return a boolean whether the fleet is killed or not
string getCorporationName()	return the corporation name
void destroyShip()	to remove a ship from a list
void addColonyShip()	to add a colony ship into the list
void addMilitaryShip()	to add a military ship into a list
void addSolarSailShip()	to add a solarsail ship into a list
void addOtherShip()	to add other ship into a list
void kill()	to set the boolean isKilled() to true. When a ship is killed it's not in the game anymore.
void alienAttack()	to implement alien attack on unprotected ships and apply the result
void infect()	to infect the colony ships in a fleet if there is no Medic ship
vector<Ship*> protectedShips()	return a vector of all protected ships
vector<Ship*> unprotectedShips()	return a vector of all unprotected ships
vector<Ship*> colonyShips()	return a vector of all colony ships
vector<Ship*> shipList()	return a vector of all ships
fleet *createFleetFromFile()	read and create a fleet from file
bool compareColonist()	return a boolean to compare which ship among two ships that has more colonist
void simulate	to simulate the game using the fleet created
int getFighters()	return the cumulative fighters in a fleet
int getWeight()	return the cumulative weight of a fleet

Class name: Ship Derived class: ColonyShip, SolarSail Ship, MilitaryEscortShip Component of: Fleet	
Function: To create the many different types of ship	
Variables	Description
String type; Int cost; Int weight; Int energy;	Type of ship Cost of ship Weight of ship Energy consumption of a ship
Operations	Description
int getEnergyConsumption() const;	Returns energy consumption of this ship
int getWeight() const;	Returns weight of this ship
int getCost() const;	Returns cost of this ship
string getTypeName() const;	Returns the ship type, e.g. Ferry, Cruiser, or Ebulient
bool isDestroyed() const;	Returns true if the ship is destroyed, false otherwise
static bool isSupported(string type);	Returns if this class supports the type or not.
Ship(const string type);	Constructor of the class. The `type` argument is the type of the ship.

Class name: ColonyShip Base class: Ship Component of: Fleet	
Function: For creating ship of type colony	
Variables	Description
int colonists;	Number of colonists in this colony ship.
Operations	Description
int getColonistCount() const;	Returns number of colonists of this ship
void infect();	Infects a colony ship. Will also set the colonists to 0.
bool isInfected() const;	Returns True if the ship is infected with a disease, False otherwise
static bool isSupported(string type);	Returns if this class supports the type or not.
ColonyShip(const string type);	Constructor of the class. The `type` argument is the type of the ship.

Class name: SolarSailShip Base class: Ship Component of: Fleet	
Function: For creating ship of type colony	
Variables	Description
int energyGenerated	Number of energy generated in this energy ship
Operations	Description
int getEnergyProduction() const;	Returns energy production of Solar Sail Ship
static bool isSupported(string type);	Returns if this class supports the type or not.
SolarSailShip(const string type);	Constructor of the class. The `type` argument is the type of the ship.

Class name: MilitaryEscortShip Base class: Ship Component of: Fleet	
Function: For creating ship of type military	
Variables	Description
int fighters;	Number of fighters in this military ship
Operations	Description

int getNrProtected();
 	
int getFighters() const;
static bool isSupported(string type);
 	
SolarSailShip(const string type);
	Returns number of colony ships protected by this ship
	Returns number of fighters in a military ship
	
	Returns if this class supports the type or not.
	Constructor of the class. The `type` argument is the type of the ship.