CENG 242

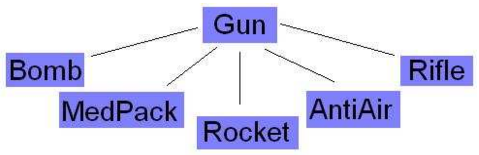
Hw #5

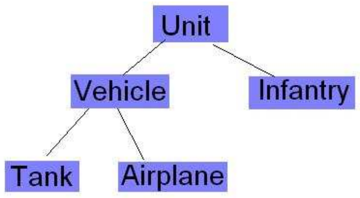
Spring 2007/2008

(Due: May 11th, 2008 Sunday 23:59)

In this homework you will implement a two layered (land and sky) 10*10 board game with different kind of units like tanks, soldiers, airplanes … etc. Each of them will move and fire with specific properties.

You are provided two abstract classes, "unit" and "gun" to represent all the types of units. The units are derived from the "unit" class. Here's the class inheritance diagram.





The units used in the game are made by combination of a base unit and a gun. Properties of units and guns change according to the derived classes.

Derived Unit Classes:

<u>Infantries</u>:move one square to its orientation.

<u>Vehicles</u>(abstract class): move two squares to its orientation

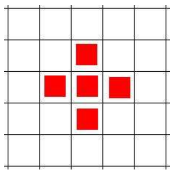
<u>Tanks</u>: are on the land (z=0)

<u>Airplanes</u>: move on the sky (z=1)

Derived Gun Classes:

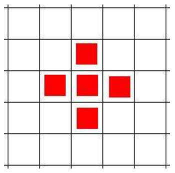
Bomb (damage land units)

Damage Type	
toInfantry	90
toVehicle	50



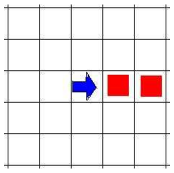
MedPack (damage (heal) land units):

Damage Type	
toInfantry	-30
toVehicle	0



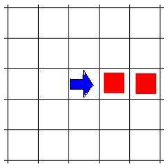
Rocket (damage land units – blue is the guns orientation):

Damage Type	
toInfantry	30
toVehicle	50



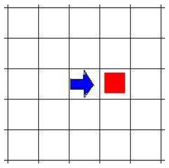
Anti-Air (damage air units):

Damage Type	
toInfantry	0
toVehicle	50



Rifle (damage land units):

Damage Type	
toInfantry	50
toVehicle	0



Unit types used in the game:

- Fighter: Airplane + Anti-air gun
- Bomber: Airplane + Bomb
- Tank: Tank + Rocket
- AntiAir: Tank+ Anti-air gun
- Medic: Infantry + Medpack
- Rocketeer: Infantry + Rocket
- Marine: Infantry + Rifle

General Rules:

- The units are inserted to the game one by one.(When inserted each unit and its gun will have their orientation SOUTH as default) . The units are called from the game using getUnit function. They are controlled via move,fire,setOrientation and setGunOrientation functions.
- The units give damage to all the units (friendly or not)
- When moving if two units collide(on the same layer), they are destroyed. If one unit passes over a unit (on the same layer) it is accepted as a collision.
- Each unit will have 100 health at start, if one units health is equal to or below 0, it is destroyed. (its gun will also be destroyed)
- Each units gun will have 10 ammo at start, if it fires more than 10 times, you will ignore when it fires. (but he can do kamikaze)
- The board is composed of two layers (sky and land), each having 10*10 squares. You will ignore when a unit moves or fires outside of the borders. (a vehicle will move one unit if its distance to borders is 1)

- The games capacity is 100 units. You can ignore the cases which will have more than 100 units.

Internal Structure:

- Abstract classes' (gun and unit) and the game class' definitions are given to you. You need to implement derived classes and the required methods for given and derived classes.
- When a unit fires, it calls its guns fire method with his position. The guns fire method calls the games fired method for each square it damages (if ammo. When a unit moves it calls the games moved method to set its position and to control collisions.
- The game object is pointed by a global variable "theGame" (declared at hw5main.cpp mainly, in header as external) . You can reach it via this pointer when you need it.

Specifications:

- You will have two files. One is hw5.h for your class declarations, structures, enumerated types; the other is hw5.cpp for your definitions. In addition to these files "hw5given.h" file is given initially, and you will not change anything on it. You should not write main function in these files. I will write a *main* function in **hw5main.cpp** to test your codes (a simple one is provided with the homework). If you want to test your codes before submitting, write a *main* function in *hw5main.cpp* and use **Makefile** to compile your codes on inek machines.
- Follow the newsgroup for any questions and possible updates.
- All the work should be done **individually**.
- You will submit a single tar file **hw5.tar** including *hw5.h* and *hw5.cpp*. You can tar your files with the command "*tar cvf hw5.tar hw5.cpp hw5.h*". Do not send me in other formats like ".tar.gz", ".rar", ".zip" etc.
- You will submit your codes through cow system. Specifications (file name, method names, class name, types etc.) are strict. Breaking any of them will cost you getting a 0 since black box method is used.