ATILIM UNIVERSITY

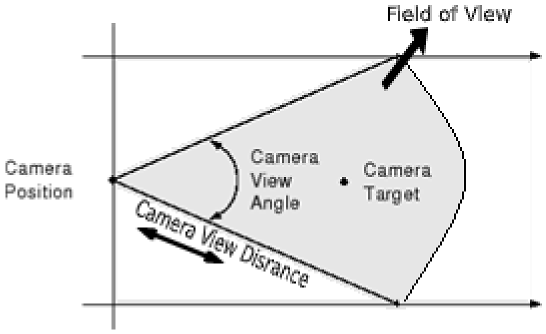
COMPE 225 – OBJECT-ORIENTED PROGRAMMING
HOMEWORK I

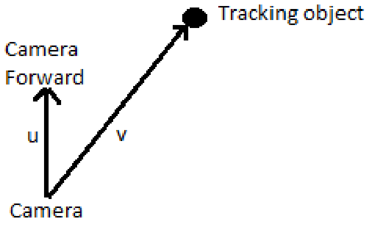
Instructors: Çiğdem Turhan - Güler Kalem

Assistants: Buğra Yener Şahinoğlu, Gizem Altınay

Due Date: 23.03.2020, 23:55

A company wants you to create a field of view system for the simulation test of their surveillance cameras.





u=> a vector from the Camera object to camera's forward position

v=>a vector from the Camera object to Tracking object

if the magnitude of v vector less than or equal to view distance and the angle between u and v vectors less than or equal to half of the view angle, tracking object is in the field of view of the Camera object

In your program,

- You will create two classes called Tracking and Camera.
- Tracking class has two private double data members, positionX and positionY. Its constructor takes two parameters and assigns their values to the data members. Also write the get functions of those data members.
- Camera class has four private double data members to hold values for forward position of the camera on x axis, forward position of the camera on y axis, view angle, view distance. Its constructor takes four parameter and assigns their values to the data members. It has also three member functions.
  - First one is a private function called isItInTheViewingDistance. It takes Tracking object as a parameter. In this function assuming that the camera is in origin position (0,0), you will calculate the distance between Camera object and Tracking object using the following formula:

    $$distance = \sqrt{(y_{to})^2 + (x_{to})^2}$$

    If the distance calculated is greater than the view distance member of the Camera class, the function will return false, otherwise it will return true.

    - The second one is a private function that is called isItInTheViewingAngle. It takes Tracking object as a parameter. In this function you will calculate the angle between Tracking object and Camera objects forward positions using the following formula.

      $$angle = arccos\left(\frac{x_c * x_{to} + y_c * y_{to}}{\sqrt{{x_c}^2 + {y_c}^2} * \sqrt{{x_{to}}^2 + {y_{to}}^2}}\right) * 180/\pi$$

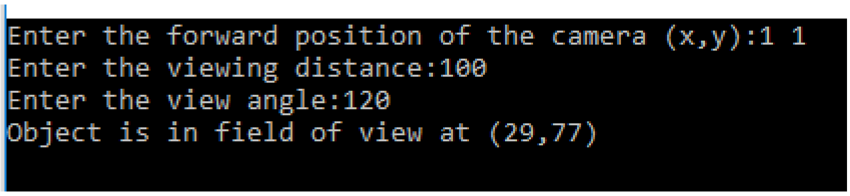
      If the calculated angle is greater than half of the view angle member of the Camera class, the function will return false, otherwise it will return true.
    - Last one is a public function called viewFieldControl. It takes Tracking object as parameter. isItInTheViewingDistance and isItInTheViewingAngle functions are called in this function. If the result of both function is true, display a message "Object is in field of view", otherwise display a message "Object is not in field of view" along with the x,y coordinates of the Tracking object.
  - In the main function, create a Tracking object with randomly generated x and y values between -100 and 100. Then, create a Camera object using user inputs for position of the camera on x axis, position of the camera on y axis, view distance and view angle. Call viewFieldControl function sending Tracking object as parameter.

x_c => forward position of the camera on x axis x_{to}=> positionX variable of Tracking
y_x => forward position of the camera on y axis y_{to}=> positionY variable of Tracking

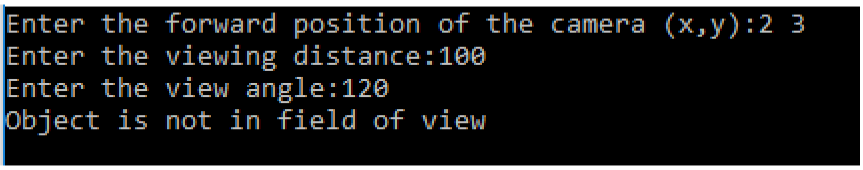
!! Include math.h library to use acos(n) to calculate arccosinus and sqrt(n) to calculate squareroot. Also use the M_PI constant in this library for the Pi value.

!!You can use any random function to generate number values. But everytime your program generates those values, they should be generated differently. Otherwise your code will be evaluated over 90 points.

Sample Run:



```
Enter the forward position of the camera (x,y):1 1
Enter the viewing distance:100
Enter the view angle:120
Object is in field of view at (29,77)
```



```
Enter the forward position of the camera (x,y):2 3
Enter the viewing distance:100
Enter the view angle:120
Object is not in field of view
```

IMPORTANT NOTES:

- Cheating will NOT be tolerated. Special software will be used to verify if the submitted homework is your original work or copied from someone/somewhere else. If any case of cheating is detected, at any time, **you will get ZERO from your homework.**
- You should upload your homework file to the Moodle system.
- The name of your homework file should be **"StudentID_HwNumber.cpp",** for example: "11122233_hw1.cpp".
- You should use indentation and comments in your code.
- Late submissions will be evaluated by using formula 100-20*d^2 where d is the number of late submission days.
- If there is any type of compiling error or runtime error seen in your code, your code evaluated by the half of the maximum evaluation score. (Late submissions will change the maximum evaluation score.)