Programming Session Assignment 11

2017/12/19 by TA 陳姿玲

REQUIRED FILES

Please **compress a folder** named **PSA11_b06901XXX** (student ID) that contains the following files:

- ✓ b06901XXX _p1 project
- ✓ b06901XXX _p2 project

Do not submit executable files (.exe). Files with names in wrong format will not be graded. In your .cpp files, we suggest you write comments in details as much as you can. It will be good for TAs to read your code and for your future reference and maintenance. (Due date: 12/20 06:00)

PROBLEM DESCRIPTION

1. **[Required file: b06901XXX _p1 project]**

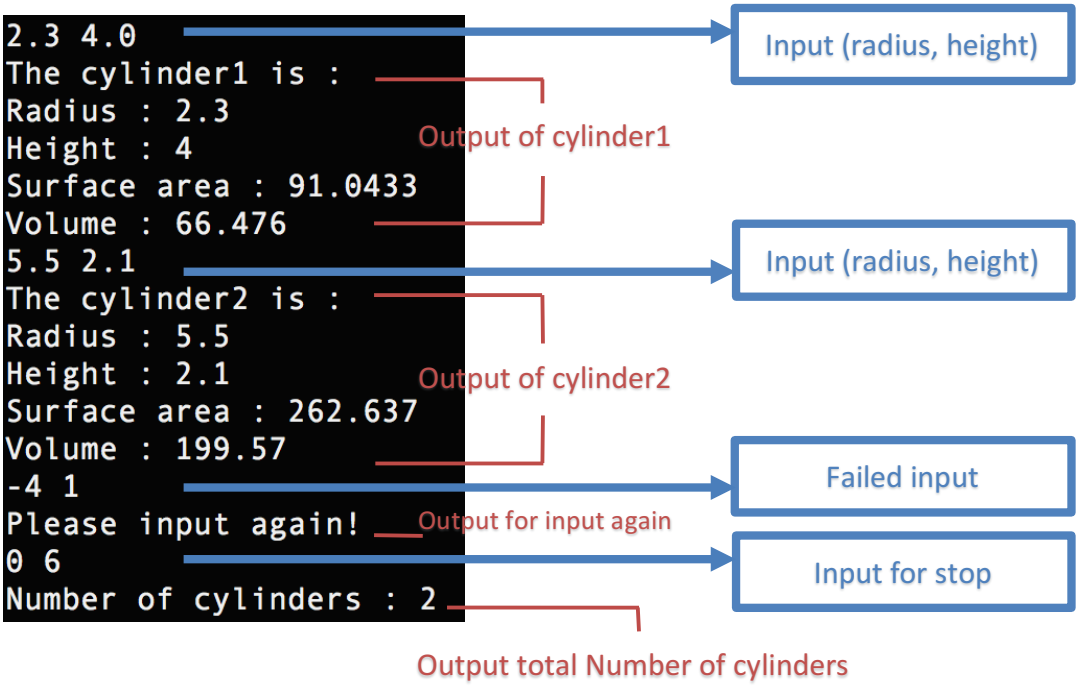
   Finish [TODO] in the files cylinder.h & cylinder.cpp provided for PSA11_P1 project that reads the radius and the height of the cylinder from the user, outputs the cylinder's information, including the radius, the height, the surface area, and the volume, and then outputs the total number of cylinders calculated. Based on the following format, the program will not stop until you input "0" as either the radius or the height. Note that the program will ask you to input again when any input value is negative.

   Note:

   - ✓ The header file cylinder.h should contain definition of the Cylinder class data members and declaration of the function members. The source-code file cylinder.cpp should define class member functions.
   - ✓ Use variable "count" with type "int" as "static data member" to record how many cylinders the project has calculated.

- ✓ Use variable "PI" with type "double" as the "constant data member" which should be initialized in the constructor. (PI=3.14159)
- ✓ In this problem you only need to submit cylinder.h and cylinder.cpp in b06901XXX_p1 folder.

Format:



2. [Required file: b06901XXX _p2 project]

Design a program based on the following rule:

Using SDL to create a 640x480 screen. There are two dots in the screen, one can move by press down direction keys and another just stay at the coordinate (160,120). There are also a 120x400 wall and a timer at the center of the screen. The timer works as follows:

(1) Timing begins when the program executes.

(2) Timing pauses if collision occurs between the moved dot and the wall

(3) Timing pauses if collision occurs between the moved dot and another dot.

(4) Timing continues if no collision occurs

When collision occurs, if you hold down the direction keys, the timer will pause; otherwise, the timer will continue.

Note:

- ✓ You can use two sample codes as follows:

  http://lazyfoo.net/tutorials/SDL/23_advanced_timers/index.php

  http://lazyfoo.net/tutorials/SDL/29_circular_collision_detection/index.php

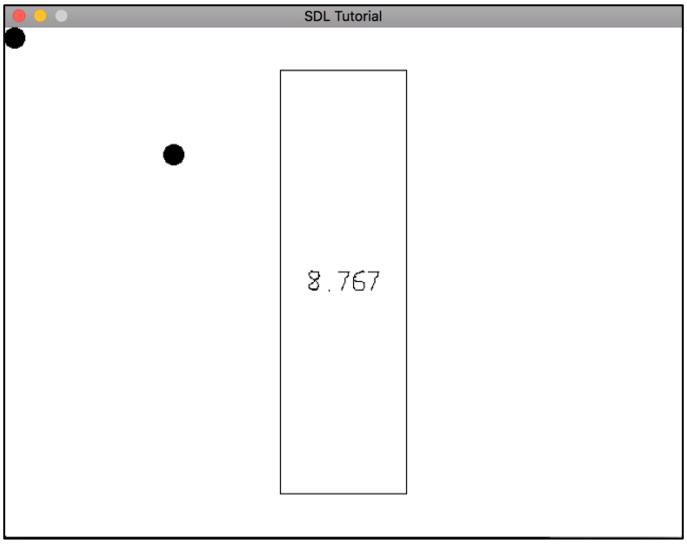
- ✓ The header file game.h should contain all class definitions and declare all functions you need. The source-code file game.cpp defines all class member functions and other functions which you use in main.cpp. The main.cpp only contains the main function.

- ✓ The path of dot.bmp and lazy.ttf is “Resource/xxx.xxx”.

- ✓ In this problem you need to submit main.cpp, game.h and game.cpp in b06901XXX_p2 folder.

Format:

Game start:



Collisions happen: