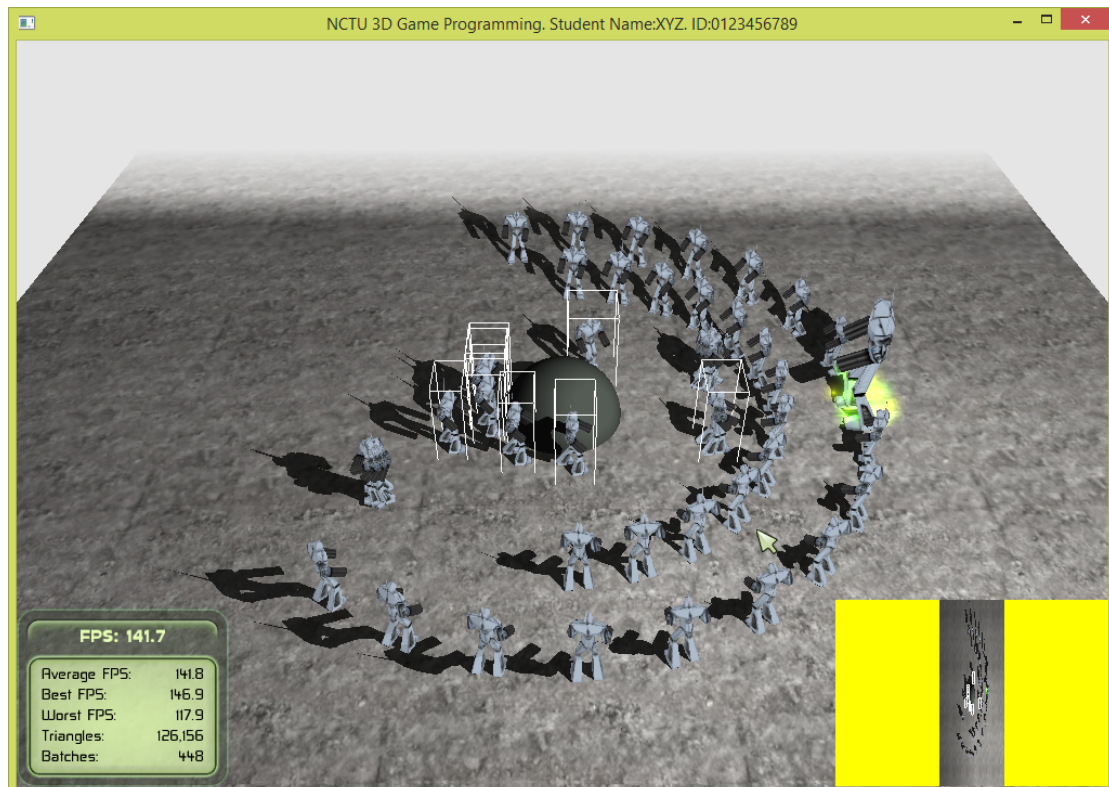


### 3D Game Programming: Programming Assignment Two

**YOU WILL RECEIVE ZERO POINTS IF YOU COPY and PASTE THE MATERIALS OF OTHER! LATE SUBMISSION WILL BE DEDUCTED BY 30% a day.**

In this assignment, you must implement the program individually. You should use the source code contained in the **Template** folder to do this assignment. You can use all the source code of our tutorials to implement this program.

You **MUST** implement something similar to the demo.



**Deduction: File organization is wrong [-20%]; program crashes [-20%], No student ID [-30%]. The program cannot be compiled on .NET2010 [-30%].**

**There are three tasks. You must finish all of them.** You must show two viewports with different contents as follows:

#### **Task One [80%]:**

1. [1%] Show your name and student ID at the top bar of the window.
2. [1%] Create a scene manager. Use it to create a camera and a viewport.
3. [2%] The viewport occupies the entire screen space. Set the background color of the viewport to black.
4. [2%] Setup an ambient light and enable linear white fog with parameters (0, linearStart=1400, linearEnd= 1600). Set the fog color to white.

5. [1%] Create a skybox.
6. [1%] Enable shadows. Use stencil additive shadow type.
7. [2%] Create a plane, set a material to it. Make sure that the plane does not cover the entire viewport. The background of the viewport can be seen.
8. [4%] Create two circles of robots in a space of dimension 600x600.
9. [1%] Set one robot to be the largest, e.g., scale(2, 2, 2).
10. [1%] Create a sphere with radius 70.0 and set it the center.
11. [2%] The sphere cannot be selected.
12. [5%] Create one light and rotate it around the scene periodically. While the light is rotating, the shadows casted by the robots are changing accordingly.
13. [10%] A rectangle region is defined by dragging the mouse while the LEFT MOUSE KEY is pressed. It must work for the following five cases: 1) Lower left to upper right; 2) upper left to lower right; 3) lower right to upper left; 4) upper right to lower left; and 5) click and release at the same position.
14. [10%] The robots are selected if they overlap with the rectangle region when the LEFT MOUSE KEY is released.
15. [5%] Show the bounding volume of a robot if the robot is selected.
16. [5%] Press RIGHT MOUSE KEY to define a target position and make the selected robot(s) walk to the destination and then stop there.
17. [3%] While the selected robot(s) move to the target, they must face to the target.
18. [7%] While the selected robot(s) move, they must avoid collision with the sphere. Set the radius of a robot as 20.
19. [2%] The footsteps of the moving robots should be correct. There should not be obvious slipping effect.
20. [3%] If the robot(s) is/are not walking, set it/them to idle.
21. [2%] Do not show the bounding box of a selected robot if it reaches the target.
22. [5%] After all the selected robots move to the target point, play a sound.
23. [5%] Attach a particle system to the largest robot.

**Task Two [10%]:**

1. [3%] Create another camera and another viewport. The background color of the viewport is YELLOW.
2. [2%] Set the viewport aspect ratio as  $(4*W/H)$ , where W is the width of the viewport and H is the height of the viewport.
3. [1%] Set the camera position as (0, 1400, 0). Let it look at point (0, 0, 0). Be careful about the bug!
4. [2%] The viewport occupies the upper right corner.
5. [1%] Disable sky.
6. [1%] Disable overlays.

[10%] **Task Three:** Use doxygen to generate an on-line documentation browser for the program. You **MUST** document each function and each class that you implement. You **MUST** write down your name, student ID and email address in the TutorialApplication.h. Your name, student ID and email address must be shown in the documentation. You should document at least EIGHT functions.

### BasicTutorial\_00 Class Reference

3D Game Programming  
My Name: AA BB CC  
My ID: 0123456789  
My Email: aaa@cs.nctu.edu.tw More...

#include <TutorialApplication.h>

Inheritance diagram for BasicTutorial\_00:

```

graph BT
    BaseApplication --> BasicTutorial_00
    
```

List of all members.

#### Public Member Functions

virtual void	<b>createViewports</b> (void)
virtual void	<b>createScene</b> (void)
virtual void	<b>createCamera</b> (void)
virtual void	<b>chooseSceneManager</b> (void)

---

#### Detailed Description

3D Game Programming  
My Name: AA BB CC  
My ID: 0123456789  
My Email: aaa@cs.nctu.edu.tw

This is an assignment of 3D Game Programming

---

The documentation for this class was generated from the following files:

Please see the demo for details. You must implement the similar items in the demo.

### File organization:

Make sure that your folder STUDENT\_ID\_NAME must be organized as follows:

\lib : contains the precompiled libraries

\programs, inside \programs it contains the \assign\_02.

Inside assign\_02, we have the following items:

\bin : contains the executable, materials, dll, etc.

\media :contain media files

\docs : documentation, such as \html generated by doxygen

\source : contains all the .cpp and .h

\release

.sln : the project file for .NET2010

### Submission:

1. Zip and upload your source code to E3.
2. Submit a hardcopy report to describe the way how you implement the program. Submit it in class.

## Report format

Name: \_\_\_\_\_ Student ID: \_\_\_\_\_ Assignment: \_\_\_\_\_

Email: \_\_\_\_\_

THIS MUST BE YOUR OWN WORK!      YES.      Please tick "YES".

**[10%] Introduction:** //At least 100 words.

**WORD COUNT:** \_\_\_\_\_ //must fill this blank.

// describe the purpose of this assignment

// describe the tasks that you have to finish

**[10%] System architecture**

-[5%] Draw a diagram of the system. At least FIVE components.

-[5%] Describe in words about the system. At least 50 words.

**WORD COUNT:** \_\_\_\_\_ //must fill this blank.

**[30%] Methods:** //At least 300 words.

**WORD COUNT:** \_\_\_\_\_ //must fill this blank.

//describe how you finish the tasks one by one.

//You must state clearly how to implement each item!

**[40%] Discussion:** //At least 500 words.

**WORD COUNT:** \_\_\_\_\_ //must fill this blank.

//You should discuss the observation here.

//You must answer the following questions:

**[2%] What do you draw the selection rectangle?**

**[2%] How do you compute the intersection point between a ray and the plane?**

**[2%] How do you disable skies for the second viewport?**

**[2%] How do you do so that the robots walk to the target position?**

**[2%] The background color of the first viewport is set to black. However, why does the background color of the first viewport appear to be white?**

**[5%] How do you do collision handling for the moving robots and the sphere?**

**Draw a figure for illustration.**

**[10%] Conclusion:** //At least 100 words.

**WORD COUNT:** \_\_\_\_\_ //must fill this blank.

// what you have learnt, any problems, difficulties, the assignment tough or easy for you? What do you suggest for the next assignment?

**\*\*\*BONUS: The best report(s) will be received at most 10% extra points.**

## Guideline

### Disable skies in a viewport?

```
vp->setSkiesEnabled(false)
```

### Disable logo?

#### In BaseApplication.cpp

```
//mTrayMgr->showLogo(OgreBites::TL_BOTTOMRIGHT);
```

**The material assigned to Selection Rectangle is as follows:  
(defined in Examples.material)**

```
material Examples/Hilite/Yellow
{
    technique
    {
        pass
        {
            scene_blend colour_blend
            cull_hardware none

            texture_unit
            {
                texture dkyellow.png
            }
        }
    }
}
```

### Collision detection between a robot and the sphere.

Apply circle-circle or sphere-sphere collision detection. Move the robot out of the sphere if the robot hits the sphere.

### How to make a robot look at the target?

Make the robot look at the (ghost) target!