

NCTU 3D Game Programming

Saving Private Penguin

Zhang Zhexian; Student ID: 0545080; Assignment: 1; Email: zhangzhexian@outlook.com

[Yes, this is my own work 😊]

Introduction

This project is inspired by the American war film *Saving Private Ryan*.

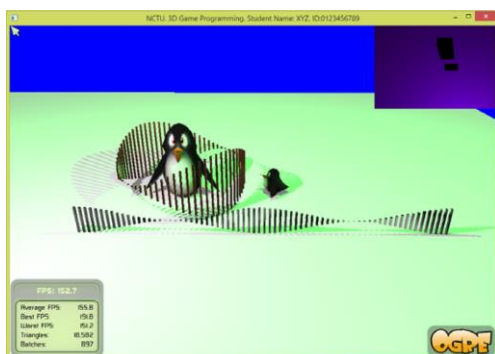


(By Source, Fair use, <https://en.wikipedia.org/w/index.php?curid=897557>)

Instead of real soldiers, this project aims for a greater audience appeal and uses cute penguins instead.

The storyline: master penguin (apparently bigger in size) teaches private penguin (smaller) survival tricks -- surrounding itself with a set of wavy cubes that form a fence. The small penguin is still learning, and it currently can only make a straight fence in front of it instead of forming a surrounding circle. On the same screen but different viewport we may see a cube submerged in purple light – this is what the fences are for—because the light is detrimental to the penguin's health.

The story is illustrated as below:



The task is to render this significant moment in OGRE.

We choose OGRE, because simply put, Ogre is the most powerful open source real time 3D rendering library currently available, and we would love to leverage the power of modern real-time hardware-accelerated 3D graphics with the best-in-class 3D graphics library.

What need to be rendered include but not limited to:

- 2 view ports, with one full screen and one on the upper right hand corner
- Main viewport with two penguins with different orientation, one wavy fence in a circle and one in a row, special lighting and shadow effects
- Secondary viewport with a cube on a plane with shadow and lighting

(word count: 238)

System Architecture



Root Object

The main point of access to an Ogre application is through the Root object

Scene Manager

The SceneManager is the object that organizes everything. It keeps track of the entities and nodes in your scene and determines how to display them.

Scene Node

A SceneNode is what attaches an object to your scene.

An Entity is not rendered in your scene until it is attached to a SceneNode.

In addition, a SceneNode is not a visible object in your scene. It only holds abstract information like location and orientation.

Entity

An Entity is anything represented by a mesh.

Lights, Billboards, Particles, and Cameras are examples of scene elements that are not entities.

An Entity is not rendered in your scene until it is **attached** to a SceneNode.

(word count: 129)

Method

Task	Implementation
Create a scene manager. Use it to create its own camera and viewport.	<code>mSceneMgr->createCamera();</code> <code>mWindow->addViewport(mCamera, z-index);</code>
The viewport occupies the entire screen space	<code>mWindow->addViewport(mCamera, z-index, 1, 1, 1, 1);</code> or leave the last four parameters empty (default is 1)
Background color is blue	<code>viewport->setBackgroundColour(Ogre::ColourValue(0,0,1));</code>
Ambient light and shadow	<code>mSceneMgr->setAmbientLight();</code> <code>mSceneMgr->setShadowTechnique(Ogre::SHADOWTYPE_STENCIL_ADDITIVE);</code>
Create a penguin to look at the user	<code>Ogre::Entity* entPenguinFacingUser =</code> <code>mSceneMgr->createEntity("PenguinFacingUser", "penguin.mesh");</code>
Create a second penguin to look at the first penguin	<code>Ogre::Entity* entPenguinFacingPenguin =</code> <code>mSceneMgr->createEntity("PenguinFacingPenguin", "penguin.mesh");</code> <code>ogreNodePenguinFacingPenguin->yaw(Ogre::Degree(-90));</code>
Create a plane	Custom function <code>createDefaultPlane();</code>
Create two set of objects by scaling cube objects. One set of objects form a circle around a large penguin. Another set of objects form a row	Custom function <code>createCircleOfObjects("cube.mesh", numCubes, 100, mSceneMgr);</code> and <code>createRowOfObjects("cube.mesh", numCubes, mSceneMgr);</code>
Create two lights	<code>createPointLight();</code>
Enable shadows	<code>setCastShadows(true);</code>

Show the mouse cursor	<code>showCursor();</code>
Show your name and ID on the top bar of the window	<code>initialise(true, "NCTU 3D Game Programming Assignment 1; Author: Zhang Zhexian (ID: 0545080)");</code>
Set the camera position to (0,350,0.0). Set the camera to look at (0.0001, 0,0.0)	<code>setPosition(Ogre::Vector3(0,350,0));</code> <code>lookAt(Ogre::Vector3(0.0001,0,0));</code>
The viewport occupies the upper right corner	<code>addViewport(mCameraSmall,1, 0.75, 0, 0.25, 0.25);</code>
Edit your own material (pure green color). Apply the green material to the object	Edit .media\materials\scripts\Examples.material file Add entry material Examples/Green <code>setMaterialName("Examples/Green");</code>

(word count: 210)

Discussion

1. What do we see on the screen:

On the primary screen, the background colour is blue, while the plane is mainly lighted up in green. There are two penguins standing on the plane, a bigger one facing in the positive z-direction, the other smaller one facing the bigger penguin. There are rows of objects forming wavy fences, one round one straight. There are two main shadow, one in purple/grey and one in green. The two shadow do not completely overlap, but mainly shine in different directions.

On the secondary screen, the background is mainly blue, with black cube and near black shadow casted to the left. The cube object and its shadow are both black and indistinguishable.

2. Reasons why the object in the secondary viewport is black:

It is black because the blue light shines on the green object (material is purely green) and no light is reflected, thus it appears black.

3. What happens when I change the camera position for the secondary viewport:

When the camera position is changed, with z-axis set to 0 and camera looking straight down, the cube is not present in the viewport anymore. This is possibly due to an internal bug in OGRE.

4. Other interesting findings:

- a. Debugging: initially I set the cube size too large, so nothing can be seen on the secondary viewport, because everything is blocked by the cube (solution: do not scale the cube to 20 times);
- b. Debugging: cannot render the secondary view port, because I set the top and left position to whole number such as 20, while it should be decimal like 0.25, as the whole window has default length of 1.
- c. Adjusting: when I change the light colour of the secondary view port to white (1.0, 1.0, 1.0), the green object with black shadow can be seen clearly.
- d. Adjusting: when I change the number of cubes to 1 for the circle of objects and row of objects (i.e. the "fence"), there is an error message "R6010 Abort() has been called"; when I change the number to 2, the cubes are huge (bigger than the penguins) and has metallic reflection (2 are grey and 1 is orange); when I change the number to 1000, the program becomes very slow (FPS is only 2.7), and the objects form interesting fingerprint-like patterns.

(word count: 402)

Conclusion

This project not only allows me to recreate a Saving Private Ryan in the penguin theme, but also consolidated my understanding of basic OGRE project, concepts such as the static entity, light, shadow, material and geometry through a hands-on practice.

In future I would love to try more animation and interaction design instead of just a static 3D image. Games will need animation so let's do it!

Thanks for the marking my long report and hope you've enjoyed it. Have a great day ☺

(word count: 109)