

Assignment 3

CS 381: The Game Development Pipeline

Spring 2015

Max Score: 100

Objectives

1. Demonstrate an ability to apply knowledge of computing, mathematics, science, and engineering by learning and applying knowledge of Python to solve a problem (1)
2. Demonstrate an ability to analyze a problem, and identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution (5)
3. Demonstrate an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice
4. Demonstrate an ability to apply design and development principles in the construction of software systems or computer systems of varying complexity

Assignment

Go through the first four python-ogre tutorials on the python-ogre website. Make sure that you can make the camera move around the world.

Non-oriented physics

First, make a large (10000×10000) planar textured surface located at the origin on the xz plane at a height specified by a `surfaceHeight` variable (5 points).

Load a cube into python-ogre at the origin. Make the cube move in 3D with a **velocity** determined by a set of keys such as the numeric pad's arrow keys and PGUP/PGDOWN. Key presses increase or decrease the x , y , and z components of the cube's velocity (60 points).

You will control the camera with the WASD keys for camera motion in the xz plane and the E/F keys for camera motion along the y -axis (10 points).

Note carefully, that the control keys (numeric pad's arrow, PGUP, PGDOWN keys) change

the cube's velocity (**not** position). For example, if you hit the uparrow key (numeric 8), just once, and let it go, your cube should start moving forward in the xz plane and **continue** moving. Contrast this with the camera which stops moving as soon as you release the key.

A physics aspect attached to your cube entity will compute the position of the cube from the cube's velocity every tick. That is, your entity's tick should call the attached physics aspect's tick (15 points).

Hitting the space bar should set the cube's velocity to $(0, 0, 0)$ and thus stop it from moving (5 points).

0.1 Extra Credit

- Texture the cube and make it look like the Borg Cube (+1)
- Implement tab selection where the tab key selects the next entity modulo the number of entities. Create multiple cubes, and apply user input, as in the assignment, to the currently indicated entity indicated by a framebox (+5).
- Use a ship model from <http://www.cse.unr.edu/~sushil/models/> and ensure that the ship is oriented correctly as it moves (+5)
- I am open to other ideas, so talk to me (+ x)

1 Turning in your assignment

Assume that this format will be used for all your laboratory assignments throughout the semester unless otherwise specified.

1. Demonstrate your working program in the lab on the due date. Demonstrate each of the above features
2. In lab, submit your code using a submission program (5 points).
 - (a) Make a folder in your home directory named `as3`.
 - (b) Place all your project files in this folder (I expect that you will demo from this folder in lab).
 - (c) Type `~sushil/bin/subAs3` while you are in this folder
 - (d) submission will open on the due date
 - (e) Your FULL name and email address
 - (f) Source code listing
 - (g) Screenshots (if any)

Ask me (sushil@cse.unr.edu) if you have questions.