T-637-GEDE Game Engine Architecture

Problem Set 1- Due Friday January 31st, 2014

Problem 1 – Engine Design (25%)

Imagine that you are developing a game engine that you want to use with your own game but also license to other developers. At some point you have to implement an **in-game inventory system**, where the player can bring up a decorative window that contains a visual representation of all held objects, which he can then equip/use, combine, drop or examine. How much of this inventory system would you implement as part of your game engine and how much would be game specific? How does this depend on what genre of games your engine should support? What other game engine components would you be relying on? What possible platform specific things might you have to consider?

Problem 2 – Applying Math to 3D Game Problem (25%)

A spaceship is flying towards a desolate planet (or so is thought), due to disturbance in the space time continuum or (whatever could disrupts ones spaceship sensors), which results in that its sensors can only pick up objects on the sensor in a **100° cone** in-front of the spaceship (The cones fat end will be down the ships traveling direction). When the spaceship is located at point **P_{Ship} = (3.5,-1.2, 4.0)** heading towards a planet at coordinate **P_{Planet} = (8.0, 1.5, 3.5),** out of nowhere a pirate ship detects our sensor-crippled spaceship and reports it back to his base on the previously thought desolate planet. If the pirate ship was within range of the ships sensors, our Captain could intercept the transmission and evade a confrontation with the pirates. Would our Captain be able to intercept the transmission if the pirate ship is located at **P_{Pirate} = (4.0, 0.0, 1.0)?** Given that the pirate ship would be visible for only an instant when it transmits the warning and would only be visible if its given point is located within the sensor cone of our Captain's spaceship.

First visualize this on paper (and perhaps guessing the solution!) and then solve this using a dot product of two vectors. Make sure to show each step of your calculation.

Problem 3 – Game Loop and Object Update (25%)

Our trusty spaceship Captain is hungry, supposedly those at the Restaurant at the End of the Universe sell some mighty fine doughnuts. The ship flies at 2400 astronomical units per hour **(2400 AU/h)** in the direction **D= [3, 1, 7]** which leads straight to the restaurant. The flight will take some time and the Captain wants to visualize the ship flying in space graphically. He has everything ready except the vector to add to the ships location per rendered frame. The 3D visualization device refreshes at 120 frames per second **(120FPS).** Find the vector that has to be added to the ships location per update of the 3D environment. Given that one unit of the 3D environment is 1 AU and the location of the ship is updated once per rendered frame.

Problem 4 – High Resolution Timer (25%)

When considering whether a 32 bit register would be enough to hold a high resolution timer you must consider the clock rate of the CPU. How much time would it take a 32 bit register to wrap around (start again at 0) on a 2.4 GHz CPU. Show your calculations.