T-637-GEDE Game Engine Architecture

Problem Set 2

Problem 1 – Engine Design (25%)

Internationalization refers to designing a piece of software such that it will be possible to adapt it to different regions of the world without much re-engineering. That is, the process of localization should become relatively straight-forward and completely data-driven. Imagine that you are developing an engine for classic third person action/adventure games, in the vein of Just Cause or Uncharted. Describe at least five different ways in which your engine could be designed for internationalization. Describe both technically what it is that you could provide and also how that feature could be used during localization with an example.

Problem 2 – Ogre Objects and Materials (25%)

Using the information provided in "Lab 2" (http://bit.ly/AuC5w0), program an Ogre scene that contains at least five visually distinct 3D objects. All five objects should be instances of a new class that you create and based on the same mesh, but at construction time you should be able to pass in separate material references to ensure that they look different. Define the different materials inside material file(s).

Submit a zipped copy of your application directory (that includes your solution, source files, configuration files, assets **AND built application executable**).

Problem 3 – Aligned Memory Allocation (25%)

a) Define a new object type `Projectile` using the `struct` keyword. Make sure to **pack it efficiently**. This type should contain the following variables: id (of type 16 bit short), loc_x, loc_y, loc_z (each of type 32 bit integer), is_moving (8 bit boolean), vel_x, vel_y, vel_z (each of type 32 bit integer), payload (32 bit pointer) and is_active (8 bit Boolean).
b) An aligned stack-based memory allocator currently points at empty memory address `0x4A364B51`. It is asked to allocate memory for a variable of type `Projectile`. What address will the allocator return and what is the address of the stack's empty memory pointer after the operation (assuming a typical implementation as described in 5.2.1.3)?

Problem 4 – Hashing Game Object Names (25%)

Consider a game engine where all game objects are identified using a short string, maximally 10 characters in length.

a) Create or pick any **two different** hash functions and calculate the hash value for the IDs "BUILDING1", "BUILDING2","P1SPAWN" and "M2SPAWN".
b) Now imagine you have a fixed hash table of 1024 entries where you wish to store pointers to the actual objects along with their string IDs. What indices in the table do the above IDs map onto? If probing is necessary, explain what probing method you are using.
c) What are the advantages and disadvantages of using each of the two has functions you used?