

## T-637-GEDE Game Engine Architecture

### Problem Set 3 – Due Sunday April 14th, 2013

#### Problem 1 – Internationalized Software Design (50%)

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 – Hashing Game Object Names (50%)

Consider a game engine where all game objects are identified at design time using a short string, maximally 10 characters in length. We want to avoid as much as possible to use string comparisons in our code to identify objects and therefore decide to hash each string into a unique 32 or 64 bit integer (your choice) for faster comparisons.

- a) Create OR find 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?