T-637-GEDE Game Engine Architecture
Problem Set 2 – Due Monday March 14th, 2016

Problem 1 – Engine Design (25%)

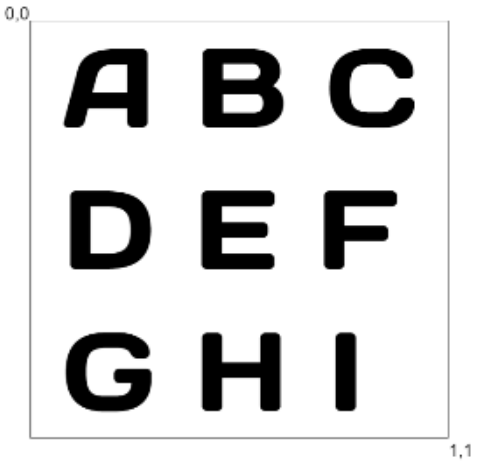
Human interface devices (HIDs) bring game players into the heat of action; they are therefore an incredibly important piece in the whole game experience. But how do you know whether a certain HID is good for a certain game? In an attempt to understand the connection between game engine design and the wide variety of HIDs available, discuss the following two questions:

A. If you are designing a game engine for a certain genre (you can pick one you like, e.g. RPG, Sports, RTS, etc.), what range of HIDs should you support? Do certain HIDs fit that type of game better than others? Why or why not?

B. If the game world and game play is complex, how can the game engine help developers make use of a very simple HID (think of one of the simplest possible HIDs: one button)? Would this work for every game genre?

Problem 2 – Textures as Sprite Maps (25%)

Sprite Maps are a useful way to reduce the number of textures the GPU has to have in memory, thus also reducing the number of loading in and out of memory. Sprite maps work by including many different sprites/images in a single texture and then addressing individual elements by offsetting UV coordinates.



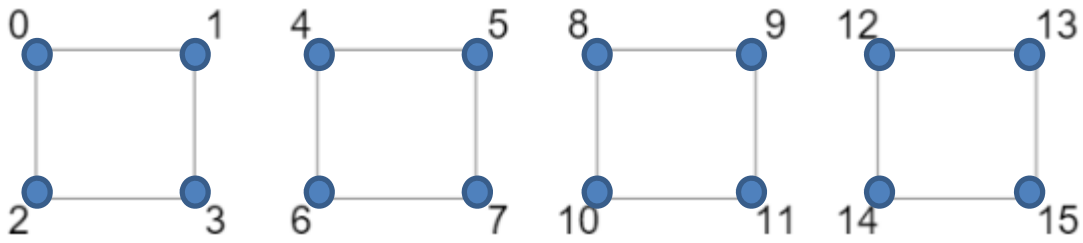
You are given a set of four quads. Vertices are stored in a buffer in the order shown below. A single texture, shown on the right, is provided.

Assuming a counter-clockwise winding order and that the texture coordinate system originates at the upper-left (a Direct3D convention):

A. Provide the **Indexed Triangle List** required by the GPU to draw the quads, and

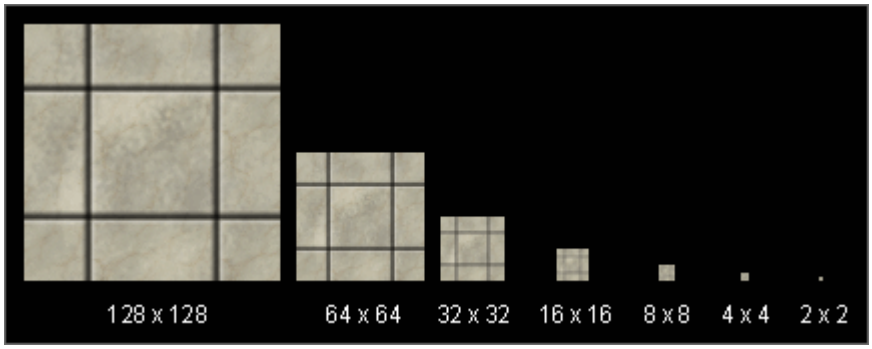
B. Provide the **Texture Coordinates** for each vertex, so that the quads spell the word “IDEA” using the texture shown.

Vertices of the four quads:



Problem 3 – Mip Mapping (25%)

After rasterizing a single quad (made up of two triangles), it is determined that the quad surface occupies a screen area of 20x20 pixels. The texture coordinates in the lower left corner of the quad are (0, 0) and (0.5, 0.5) in the upper right corner. The texture to be applied provides 7 mip levels (see below).



0 1 2 3 4 5 6

A. Without any particular texture filtering, which mip level should get picked to preserve optimum **Texel Density**?

B. How would this texture get sampled if **Trilinear Filtering** was used?

Problem 4 – Shaders (25%)

A. There are various kinds of shader programs that can be written for the rendering pipeline, the two most common are the **Vertex Shaders** and **Fragment Shaders**. Briefly explain the bare minimum expected behaviour of these two kinds of shaders. What are their essential inputs and outputs?

B. Is the shader program shown below a Vertex shader or a Fragment shader?

C. What is the effect that this shader program below achieves beyond the minimum expected behavior?

```
void main(
        float4 vtx_position         : POSITION,
        float2 vtx_texcoord0        : TEXCOORD0,
        float3 vtx_normal           : NORMAL,
        uniform float4x4  mat_modelproj,
        uniform float t,
        out float4 l_position       : POSITION,
        out float2 l_texcoord0      : TEXCOORD0)
{
     float4 temp = vtx_position;
     temp.xyz = temp.xyz + vtx_normal*0.2f;
     l_position = mul(mat_modelproj, temp);
     l_texcoord0 = vtx_texcoord0;
}
```