Game Development - Final Examination - January 16th 2019

YOUR FULL NAME: **Solution Provided by the teacher**

- You have 2 hours to complete the assignment.
- Only valid text will be the one inside each box, everything else will be ignored by the teacher

1. **(2 points)** Describe your approach to **draw** an isometric map. Use pseudocode to describe its core loop. Be specific about the math formula needed to find final screen coordinates.

```
We will have to take in account the following
elements:
  - Sprite sheet used
  - Visible layers to draw, from back to front
  - Size of the map (maximum rows and columns)

for each layer
{
  for each row in layer
  {
    for each column in layer
    {
      find the sprite id for row,column on layer
      find the sprite_coordinates in the sprite
sheet

      screen_x = (row-column) * (tile_width / 2)
      screen_y = (row+column) * (tile_height / 2)

      draw sprite_coordinates at screen_x, screen_y
    }
  }
}
```

2. **(2 points)** Make a step by step description on how to move from A to C using a BFS algorithm. Write down the content of all data structures at each step. Take in account that B and H are blocked.

A	**B**	C
D	E	F
G	**H**	I

We assume that the we are not using diagonals and opening neighbors in north, east, south, west order.
BFS will explore all childs of a tree in all directions. We will use a **frontier** stack (queue would be ok too) and a **visited** array. To get the path back we will create a **breadcrumbs** array parallel to visited to store where you are coming from. At each step we will pop one element from frontier and add all non-visited neighbors, to frontier and visited. That tile will also be tracked in breadcrumbs array as the tile we came from:

Step 1: Add A into all structures
Frontier: A
Visited: A
Breadcrumbs: A
Step 2: Pop A
Frontier: D
Visited: A, D
Breadcrumbs: A, A
Step 3: Pop D
Frontier: E, G
Visited: A, D, E, G
Breadcrumbs: A, A, D, D
Step 4: Pop E
Frontier: G, F
Visited: A, D, E, G, F
Breadcrumbs: A, A, D, D, E
Step 5: Pop G
Frontier: F
Visited: A, D, E, G, F
Breadcrumbs: A, A, D, D, E
Step 6: Pop F
Frontier: C, I
Visited: A,D,E,G,F, C, I
Breadcrumbs: A, A, D, D, E, F, F

We stop as destination entered Frontier stack. Using breadcrumbs we go back as from it tracking the tile we came from: C, F, E, D, A. This list flipped is our path.

3. **(2 points)** Explain the concept of Profiling in the context of video game programming. Why we need to do it, who should do it and what kind of areas of the code we apply it.

Profiling means measuring how long takes to execute different parts of the code. It is used for optimization and required since the code complexity is way beyond what any developer can grasp.

On top of that, the variability of the development process makes it unpredictable to understand the ramifications of each decision taken to improve the game experience.

On top of that, we want to be able to deliver a smooth experience in the lowest possible hardware to attract more players.

Pretty much all developers, including artist and designers, need to be aware of the profiling tools available and use them to understand the consequences on the performance from the content they want to put into the game.

A Profiler mainly concerns about the cpu time spend in the code, but there are other factors like memory usage, gpu load and thread overhead.

4. **(2 points)** Explain the concept of **double buffering** applied to computer graphics.

In computer graphics we use double buffering mainly to remove graphics artifacts while drawing to the screen. In order to do that, we duplicate the frame buffer (memory area that stores the pixels that needs to be rendered on the screen).

Doing this, while we use one buffer to render, the other is used to construct the final frame step by step, making layers of rendering or any other technique used. This process is hidden as this frame is not shown on the screen.

Once the frame is finally rendered we “flip” or swap the buffers immediately (or with delay, read below), reducing considerably the amount of time that the frame will produce artifacts.

Still, if the refresh rate of the monitor is not in sync with the process of swapping the buffers, we could have tearing (we render more than one frame in the same screen). In order to fix this, we use vertical synchronization (vsync).

5. **(2 points)** Explain the key concepts behind a Quake-style console. Explain the main steps that need to be taken in order to implement it (assuming we have a functional GUI system in place).

A quake-style console is a tool were we can input commands into our game similar to a traditional command prompt of an operative system. It provides a tool for fast iteration on gameplay elements without closing the application. We can quickly check the current values of different gameplay elements, tweak them and save them for future uses.

The main elements are the commands and the cvars (console variables). The first are for executing specific functionality, like loading a new level, saving to a file or quitting the game. The cvars store values that are relevant to the game, like player HP, damage performed with different attacks and so on.

The console also outputs many debug information that can be scrolled up for fast tweaking/debugging of the game.

In order to implement it, the strategy would be (there are many valid ways, this is what was proposed in class):

- Creating a new module to handle all commands and cvars
- Using the GUI system we create a multiline label and an input box
- The GUI should appear/disappear on pressing a key (normally tilde, left of 1)
- We redirect all LOG entries to the multiline label
- We make sure there is a way to scroll up/down on the multiline label (normally with mouse scroll using the third mouse button)
- We create a small class to represent the concept of command and cvar
- We create method to add new commands and cvars with callbacks for each module to handle their specific implementation
- Optionally we can save the historic of commands and have autocomplete feature in the input box

Use this page for your own notes. You cannot use any other page. Do not remove this page from the set.