Game Development BCN - Revaluation Exam - February 2nd 2017

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.**(3 points)** Elaborate on the concept of the Virtual File System that PHYSFS provides. Detail how "mounting" works and create a simple example of folder structure that shows its utility in video game development.

A Virtual File System hides our OS file system from the code and acts as a middleman or interface to the real virtual file system.

First of all, it hides the differences between different OS like windows and unix-style. It also allows us to work transparently with compressed files. It can also "mount" OS file system folders in different places of the virtual file system, allowing overlap in a way that enables ignoring files and using new ones transparently.

Example of use would be for patching. In our original game we have a texture in our folder "Textures"

Assets/Textures/texture.png

We can add a patch.zip and mount it in our root folder with this content:

Assets/Textures/texture.png

Transparently when the game executes again it will use the new texture from the patch instead of the original texture without any changes in the code.

2. **(3 points)** Explain the concept of a Navigation Mesh in video games: how many types exist ? what are the benefits and drawbacks of each type ? how are they generated ? how they behave in real-time when pathfinding algorithms use them ?

The navigation mesh is an abstraction of the game map in a form of a graph. We have two types, a regular graph and irregular one:

Regular Graphs: it spreads its nodes at the same distance from each other forming a regular grid. They are normally generated procedurally or used in 2D games with tilesets, having one node per tileset for simplification. They tend to be easy to maintain. Still, regular graphs more nodes than what is needed thus adding more calculation time during pathfinding.

Irregular Graphs: the nodes are spread manually on the map, minimizing its amount and adapting to gameplay and map needs. They are more cumbersome to maintain since they always require a human to create them. On the other hand, irregular graphs produce smaller amount of node so they perform better in real-time when pathfinding is applied.

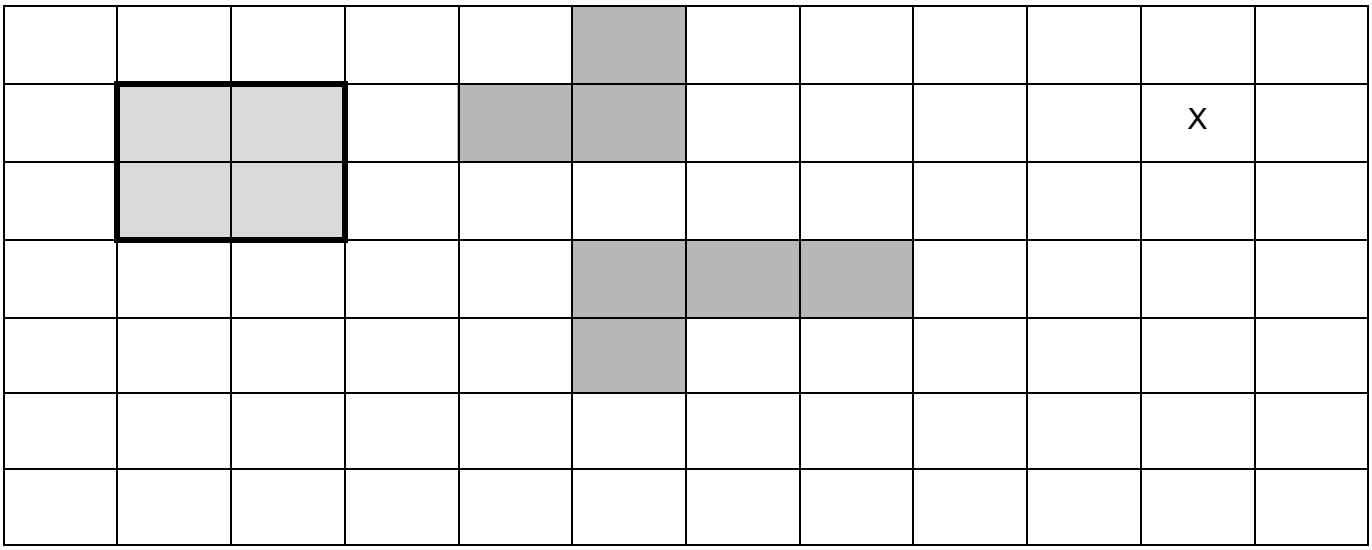
3. **(2 points)** During a specific frame our game where the logic uses 30 ms. We have vsync turned on (monitor refresh rate of 70 Hz) and our main character moves at 350 pixels per second. How much, is the character going to move this frame taking in account that we have variable time step ? How much vsync will make the application wait ?

With a monitor at 70 Hz we and vsync turned on we can only operate at 35,23,17 … fps

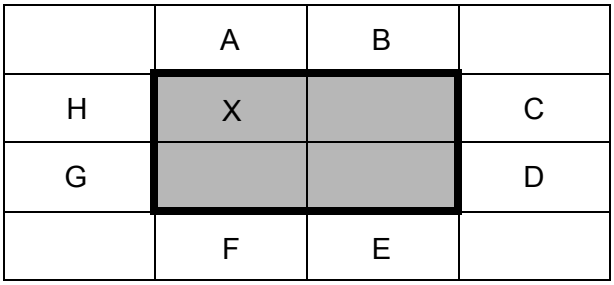
If the game logic is consuming 30 ms it means is running at 33 fps, but vsync will have us wait until we go down to 23 fps, which totals 43 ms. ***This means that vsync will wait for ~13 ms.***

*Our main character will be moving with a dt of 1/23 = 0.04 This means that it will move at 350 * 0.04 =* ***~15 pixels per second.***

4. **(3 points)** Adapt the A* algorithm to accept creatures with size of 2x2. Paths that including tight spaces of 1 square should be rejected. Explain your reasoning to solve the problem. Assume you cannot move in diagonals. *E. g.: Our Ogre is 2x2 and needs to reach the destination X.*



The only change that we would need to make the A* would be the generation of adjacent and walkable squares around us:



- We use a reference position X
- We iterate in pairs of AB(north), CD(east), FE(south), HG(west):
    - To be considered walkable **both** have to be walkable
    - The F = G + H is calculated from both pairs F added up

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