Game Developer Expertise Test

1. Space Avenger – Unity game

Fort the expertise test I had to make a simple 2D or 3D Space Shooter game. The result is exactly that, a simple 2D Space Shooter game. In it you take control of a space ship, with your only goal to get as many points as possible before you die. Deciding what I wanted to make was simple, 3 different player ships, many different enemy ships and formations, different weapons for enemies and the player and different kinds of space obstacles. I knew I could make this very simply in the time frame, but because of an exam that I had to study up for, I was left with little time to truly realize everything I wanted. In the end I was able to make only the one player ship, one weapon for the player and only a handful of enemies and formations. My focus was more on the feel of the game and to make it as fun as possible with the elements I had built. With the help of hybrid ECS(witch I am still learning to fully use) I was able to build a very robust system, that makes it possible to easily add new elements to the game. It also sped up debugging because I could easily identify where the problem was, because everything was separated into its own system. The main challenge was balancing everything out so the game wasn’t incredibly easy or impossible. I still don’t think its balanced, but I know that with more time and playtesting I would have fount the perfect balance. My friends thought it was pretty fun.

1. Question answers

Mipmapping is when we use a high-quality image with the size of it being a power of two(1024x1024) and then we calculate a progressively lower resolution(also power of two) of that same image. It is used for optimizing game(Level of detail) and everywhere where 3D graphics are needed. The images don’t have to be square, they can also be triangles.

Transformation matrices are 4x4 because its then easier to deal with multiple transformations(translation and rotation), so it is more effective to have it in a uniform way.

A stencil buffer is an extra data buffer and is primarily used to limit the area of rendering.it works in integer values(usually binary 1s and 0s)

An inverse matrix transform in basically if a matrix transforms a vector in a particular way, an inverse of that same matrix would transform it back to its original state.

Frustum culling, where we render only what the camera in the game sees, we can also use quad and oct trees to speed up collision detection or we push long processes to threads if possible, like start something in a frame then act on the results in a future frame.