Simple Battleship Design Document

Graphics:

The rendering is done using DirectX11 Renderer + DirectX Toolkit I use its functions for a fast and efficient 2d rendering of textures and text.

Class Renderer is initializing Directx11 pipeline and the is passed as a pointer during construction of game objects to be referenced in Rendering function.

Asset Manager:

Is a singleton that handles textures. It’s uses WICTextureLoaded and Microsoft functions to scan the asset folder for all the PNG files and adds them automatically to map. It uses the file name as a key to insert to a map.

At Its core is a map of unique pointers that stores custom TTexture structure that holds on to texture data needed for rendering. All game objects that use textures for rendering reference it from the Asset Manager.

Input Manager:

I grabbed this class from my previous project all it does is handles the input but I’ve only used it to get Mouse Position and check if the player pressed space bar to rotate.

Game Implementation:

Game uses state machine and two states one for main menu and another for versus player game. I used std::stack for the state machine and a few Functions managers the pushing and popping of states.

The way I implemented the game is probably not the best implementation I didn’t research how to implement battleship or read anything online and that was probably a mistake. I used the idea that came to me when I was drawing the game out.

I have 100 Tiles so I have created a special Sea Tile class that is designed to know what occupies it like is it empty, adjacent or has a ship in it. So basically at its core I have 100 Sea Tiles vector that acts a player field. If you click it when you are placing ship it will work out whether it can place a full ship at the tile where you started not hitting boundaries or adjacent tiles of other ships. It will also mark all the adjacent tiles automatically.

I like to pay attention to all the little details so I have coded Sea Tile class to have a lot of states so it can visually show maximum of information to the player.

Then I have the Ship class it acts as a sort of background manager each player has 10 ship objects and they are able to check if they are alive and if they are taking damage etc.

So for two players I have 4 vectors of 100 Sea Tiles. 2 for the placing and viewing fleets and the other 2 for targeting and feedback on shooting.

I Also have a Two UI Classes the Button and Text Label both are using DirectX Toolkit for rendering.