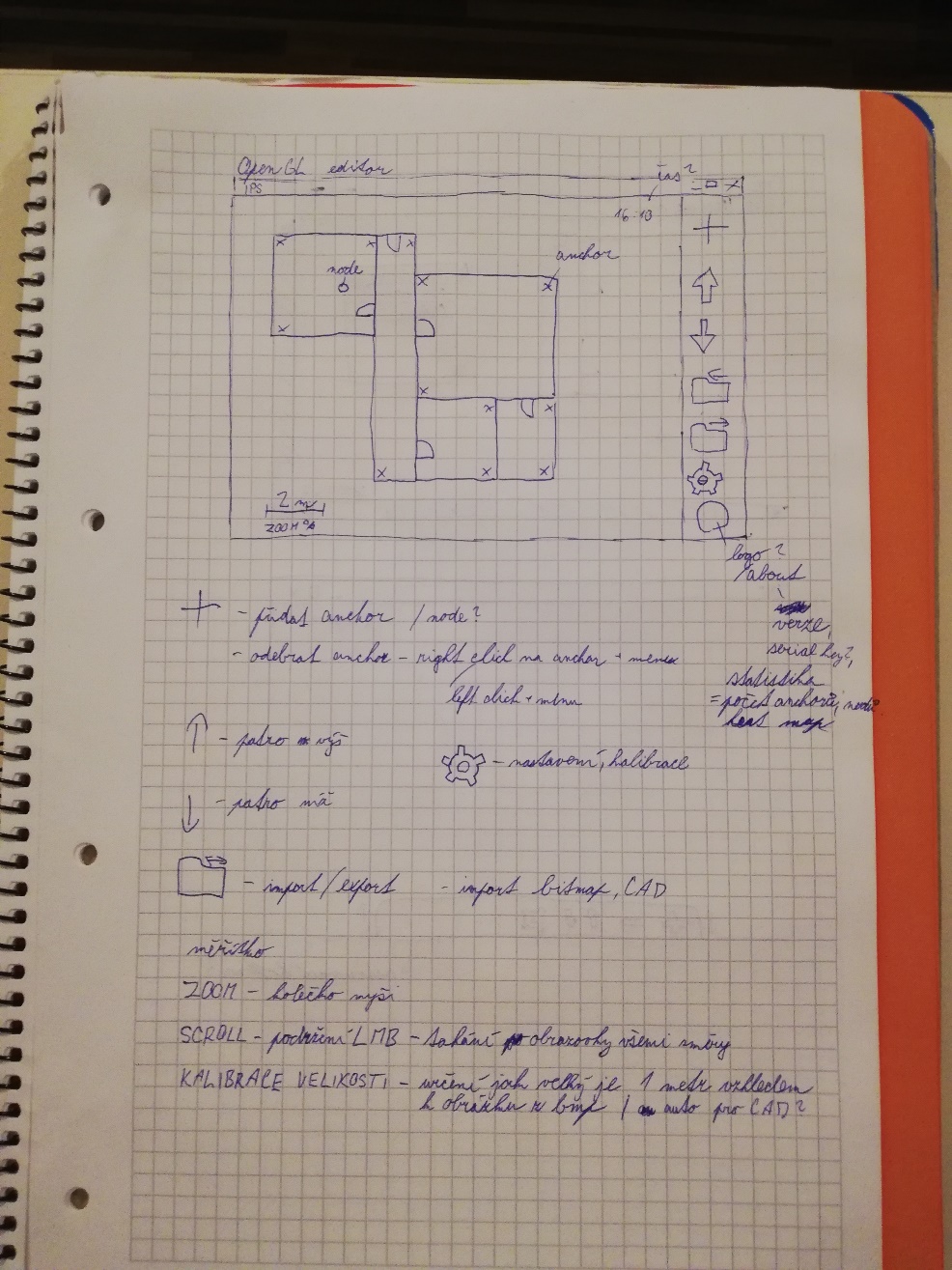
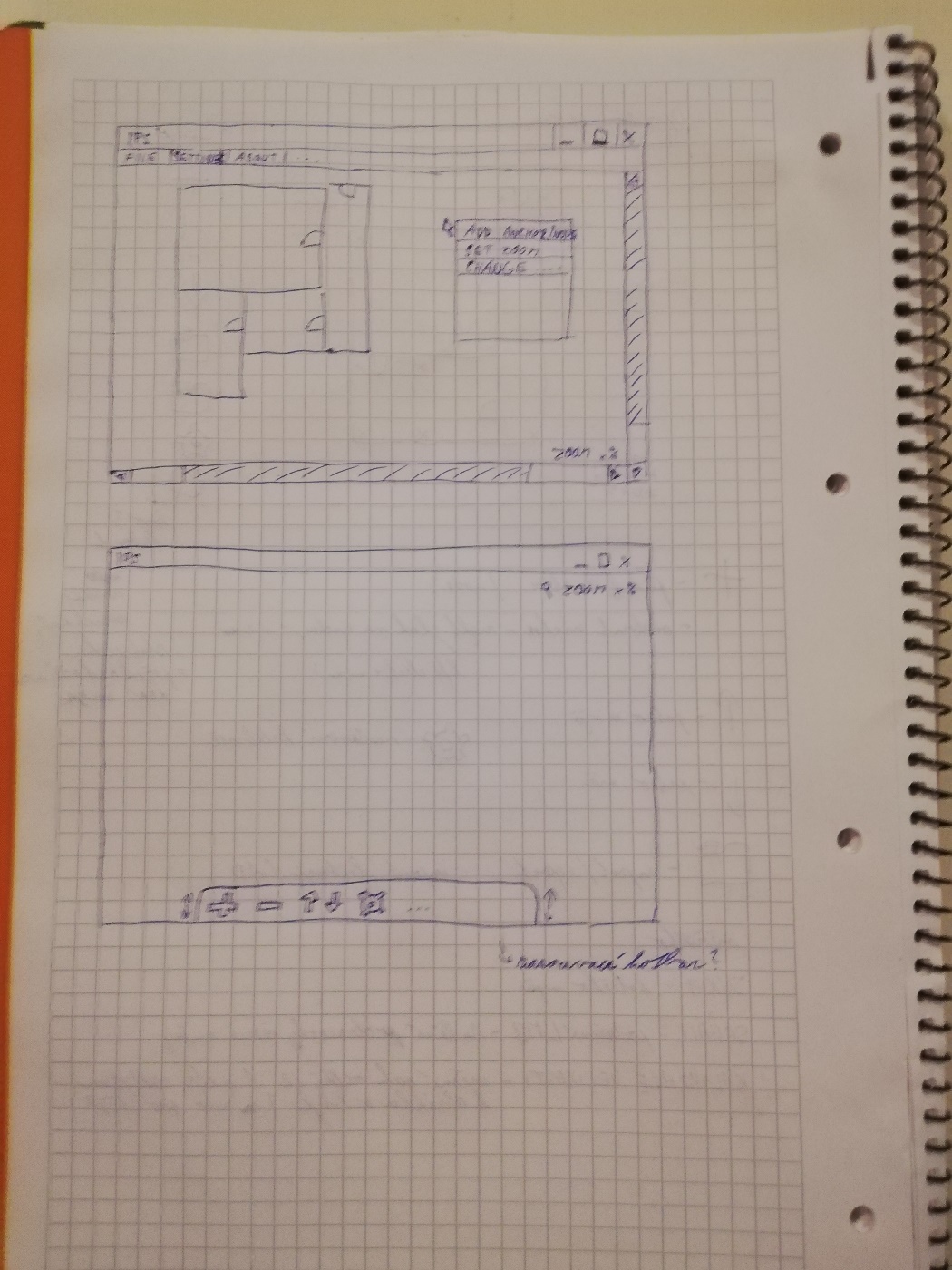
# IPS Demo Diary

-indoor positioning system? (jako gps?)

* Vybral jsem OpenGL kvůli tomu, že je multiplatformní a jednodušší než vulcan nebo directX, knihovna GLFW a GLEW podle tutorialu: <https://www.youtube.com/watch?v=W3gAzLwfIP0&list=PLlrATfBNZ98foTJPJ_Ev03o2oq3-GGOS2> tutorial je od programátora v EA, kde pracuje na Frostbite enginu
* Pro openGL matematiku používám GLM a pro ui ImGui
* Na kod pro zoom kamery jsem využil nápad z <https://stackoverflow.com/questions/5854244/opengl-2d-zooming-using-scale-and-translate-instead-of-glortho> , nakonec jsem ale použil vlastní algoritmus, který je zatím zaměřený pouze na zoom na střed a od středu, protože scrolling ještě nebyl implementován
* Na scrolling plánuji využít <https://stackoverflow.com/questions/19884182/moving-a-drawing-around-in-opengl-with-mouse> , kde je scrolling pomocí držení LMB a posouvání plochy místo scroll barů -> jednoduché pro touch použití
* Pro dokumentaci OpenGL používám [docs.gl](http://www.docs.gl/)
* Import zatím funguje pro běžné bitmapy (testován png), skrze hardcoding, později windows explorer/linux ekvivalent pro vybrání souboru. Na CAD import se dá využít dxf formát, pomocí knihovny <https://github.com/codelibs/libdxfrw> se dá číst i psát dxf.
* Scrollování zatím pomocí debug gui – slidery
* Nynější engine není moc userfriendly ani nepodporuje moc možností, rozhodl jsem se ho předělat – tutoriál dělá stejný člověk, a navazuje na serii OpenGL tutorialů <https://www.youtube.com/watch?v=JxIZbV_XjAs&list=PLlrATfBNZ98dC-V-N3m0Go4deliWHPFwT&index=1> , některé svoje funkce mohu použít jako ortogonální kamera, takže by vývoj neměl trvat dlouho
* Engine1 by mohl podporovat i více než jen OpenGL kvůli optimalizaci pro specifické OS, jako directX 11/12 pro windows, vulcan pro linux, metal pro OSX
* Engine1 nový engine – zatím logger a event systém
* Engine1 – premake, precompiled headers, vrstvy, ImGui
* Engine1 – ImGui docking a viewport, linking errory které se musí vyřešit -> engine teď používá dynamické linkování -> převod na statické, protože errory pochází z dll, nejspíše je statický engine i lepší design než dynamický, dynamicky se potom může linkovat aplikace ke statickému engine než obráceně
* Engine1 – statický link vyřešil errory i spoustu warningů
* GitHub – progress vývoje je na mém githubu s historií pro dokumentaci atd, každý commit je většinou přidání pár nových funkcí
* Engine1 – vyrenderován první trojúhelník, potřeba abstrakce rendereru
* Engine1 – abstrakce bufferů
* Engine1 – userfriendly buffer layouty
* Engine1 – vertex arrays -> už je možné renderovat objekty, zbývá ještě abstrakce a bude možné renderovat velmi jednoduše, poté kamera
* Engine1 – abstrakce rendereru hotová
* Engine1 – přidány transformace objektů – nyní je možné renderovat více objektů na různých místech, zbývají textury a možná trochu zlepšit kameru(zoom, rotace ovlivňuje směr pohybu)
* Poté už samotná funkcionalita aplikace -> přidávání anchorů, nodů a renderovat je (přidávání pomocí popup menu nebo ikona v hotbaru), umisťování anchorů přesně podle vzdáleností od zdí, potom hardware implementace?
* Engine1 – textury, pomohl <https://learnopengl.com/Getting-started/Textures>
* Engine1 – přidány základní anchory, a jejich přidávání relativně ke scéně a ne ke kameře
* Engine1 – anchory jdou přidávat pomocí right clicku, je potřeba saving
* Engine1 – anchory dokážou detekovat vzdálenost od zdí podle barvy pixelu – půdorysy mají bílé pozadí a na ní černé/šedé zdi -> detekce těchto barev, počet pixelů se poté přepočítá podle poměru metr/pixely, který určí uživatel graficky podle měřítka, které zvětší/zmenší na relativní velikost 1 metru v půdorysu - <https://community.khronos.org/t/is-it-possible-to-get-the-pixel-color/60642>
* Pro debugging barev se hodí <https://www.w3schools.com/colors/colors_rgb.asp>
* Koncept/design