

Realizētā funkcionalitāte:

- Framebuffer inicializācija
 - Framebuffer tiek inicializēts pirmajā FrameBufferGetAddress izsaukšanas brīdī
 - Izmērs tiek iegūts no #define frameWidth un #define frameHeight, pirmais elements framebufferī nosaka pašlaik lietojamo krāsu
 - Framebuffer inicializācijas brīdī tā saturs tiek piepildīts ar melniem pikseļiem (r=0, g=0, b=0, op=0)
- setPixColor - Nomaina framebuffer pirmā elementa saturu uz padoto
- pixel - Atrod attiecīgo offset iekš framebuffer padotajām x,y koordinātām, izlaižot pirmo elementu (tekošo krāsu) un nomaina atrastā elementa vērtību uz padoto
- line - Aprēķina dx un dy no abiem dotajiem punktiem. Soļu skaitu izvēlas kā max no abām deltām, katrā solī izrēķina atbilstošā soļa pikseļa koordinātas lietojot implementēto dalīšanas algoritmu, kas noapaļo rezultātus. Netiek lietotas peldošā punkta operācijas, jo daļskaitļus glabājam atmiņā kā skaitītāju un saucēju.
- triangleFill
 - Atrodam taisnstūri kas satur trijstūri (tā virsotnes sastāv no min x, min y, max x, max y no padotajiem punktiem).
 - Izejam cauri katram no šiem punktiem, pārbaudot vai tas atrodas trijstūra iekšpusē lietojot half-plane metodi (<https://stackoverflow.com/a/2049593>)
- circle - Implementēts Bresenhama riņķa zīmēšanas algoritms, kas pielieto riņķa 8-virzienu simetriju lai izvairītos no liekiem aprēķiniem (<https://www.geeksforgeeks.org/bresenhams-circle-drawing-algorithm/>)

Nerealizētā funkcionalitāte:

- Krāsu operācijas (COPY, AND, OR, XOR) – netika implementētas jo risinājumam nepietika laiks.

Lielākais izaicinājums projektā bija dalīšanas implementācija, kad sapratu ka nevarēšu lietot SDIV komandu. Problēma radās no tā ka mēģināju internetā atrast konkrētus risinājumus, bet risinājumi priekš ARM ASM nebija plaši pieejami, tos kurus atradu, īsti nevarēju izprast. Beigās nolēmu izveidot savu algoritmu neskatoties uz citiem avotiem (kas gan jau kopē kādu esošu, ņemot vērā ka dalīšana nav nekas jauns :D), kas izrādījās daudz vieglāk nekā tas šķita sākumā.

Nākamreiz mēģinātu tik daudz nemeklēt jau specifiskus gatavus risinājumus manām problēmām, bet vairāk skatīties uz vispārīgiem risinājumiem līdzīgām problēmām, kas manuprāt gan paātrinātu manu problēmas izprašanas ātrumu, gan tās atrisināšanas ātrumu.

Projekts bija interesants, jo pirms projekta realizēšanas nekad nebiju aizdomājies par to kā uz monitora pikseļu grida tiek zīmētas bezgalīgi precīzas figūras (piem. līnija, trijstūris), tagad esmu iepazinies ar vairākām pieejām šīm problēmām.