

# 3D Rendering Engine

## Introducere

Proiectul va constitui un scanline rasterizer (tip de 3D Renderer) ce va randa o scena 3D simpla pe un ecran OLED.

Utilizatorul se va putea plimba prin aceasta scena folosind maneta de joystick si cele 3 butoane.

Scopul principal al proiectului este de a incorpora cat mai multe functionalitati ale unui 3D renderer in memoria de program de 32K al unui Arduino Uno, si randarea in timp real la un framerate acceptabil.

## Descriere generală

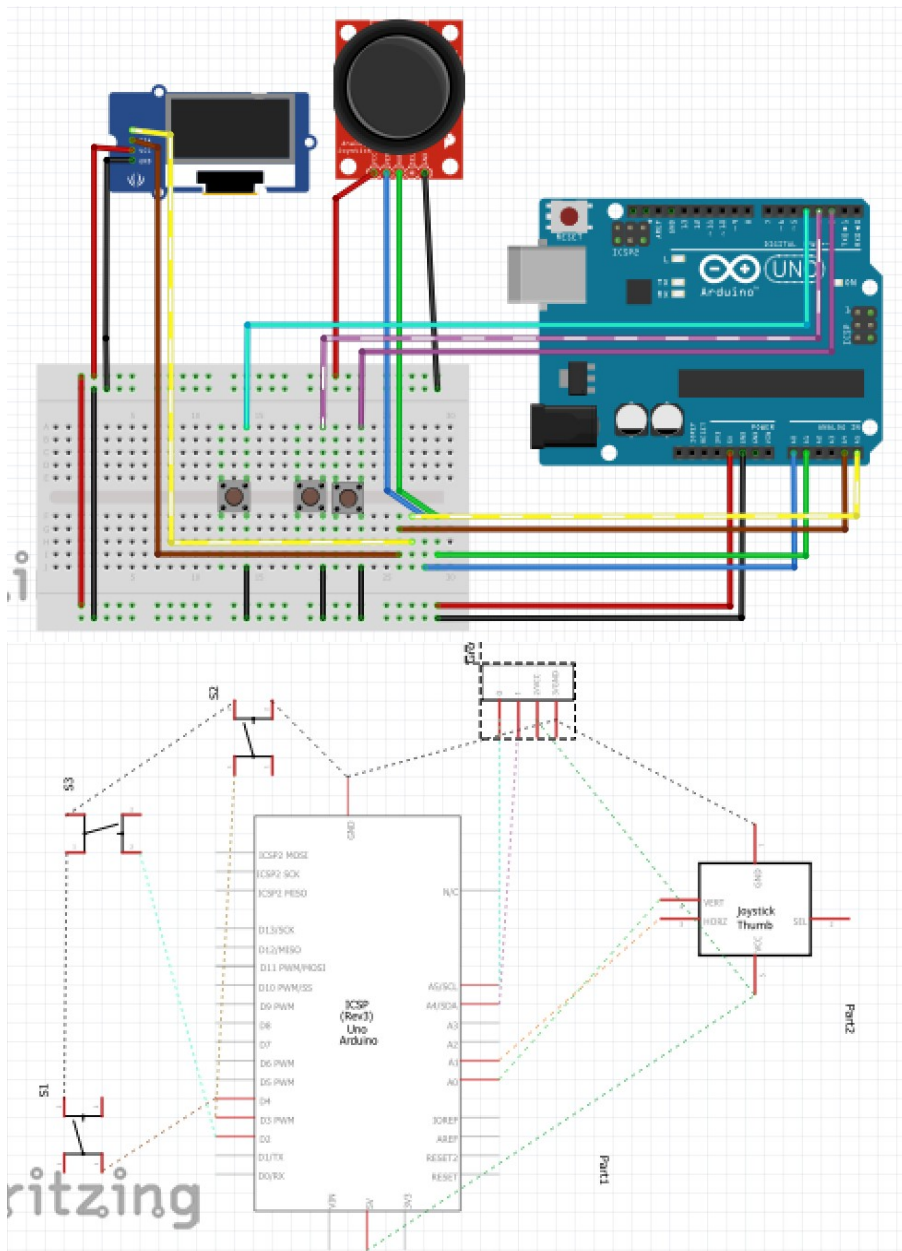
Folosind **manetele de joystick**, utilizatorul se va putea plimba prin **scena** in directiile sus, jos, stanga, dreapta, inainte si inapoi (in raport cu directia de privire si de orientarea vectorului ce descrie directia verticala a camerei), si va putea privi in jur (schimbarea directiei de privire). Cadrele vor fi randate pe **Ecranul LCD**



## Hardware Design

Lista de componente:

- Ecran OLED 128x64 pixeli SSD1306
- 1 maneta (knob) de joystick
- 1 placa dezvoltare Arduino Uno
- Fire de legatura
- 1 placa de prototipare (Breadboard)
- 3 push-buttons



## Software Design

Poligoanele ce alcatuiesc scena vor fi tinute in memoria programului. Calculele vor fi efectuate folosind numere cu virgula fixa, intrucat placa Arduino Uno nu dispune de un (co)procesor matematic dedicat, iar utilizarea calculelor cu virgula mobila ar fi extrem de lenta, ceea ce nu este de dorit intr-un sistem real-time.

Numerele ce reprezinta componentele vectorilor pentru pozitii si viteze vor fi pe 16 biti, virgula fixa, 8 biti dupa virgula. Numerele ce reprezinta componentele vectorilor unitari (asadar, orientarea observatorului: up, look, right) vor fi tot pe 16 biti in virgula fixa, dar cu 14 biti dupa virgula. Sinusurile si cosinusurile vor fi pastrate cu 15 biti dupa virgula.

Dupa initializarea driver-ului pentru afisarea de primitive (linii) pe display, va incepe randarea. In functia loop, au loc 3 procesari importante:

1. Citirea senzorilor (joystick-ul si butoanele) si actualizarea pozitiei si orientarii observatorului, in

- functia `update_observer()`.
2. Transformarea vertex-ilor tinuti in memoria de program din coordonate world-space in coordonate view-space, iar apoi in clip space (proiectarea acestora pe pozitiile pixelilor de pe ecran ce le corespund) si stocarea acestor coordonate in vectorul `processedVertices[]`, in functia `process_geometry()`.
  3. Randarea wireframe-urilor pe ecran, in functia `render_frame()`, ce preia liniile din memoria de program (vectorul `edges`), iar pentru fiecare preia coordonatele capetelor acestora, realizeaza clipping-ul necesar, si utilizeaza o functie de biblioteca (`drawLine()`) pentru a le afisa pe ecran

## Rezultate Obținute

## Concluzii

## Download

## Jurnal

## Bibliografie/Resurse

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/razvans/3d-engine>



Last update: **2023/05/30 12:16**