

# Implementing a Software Rasterizer in C

Thomas Dizon

September 24, 2025

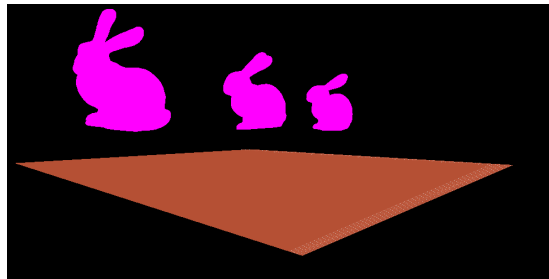


Figure 1:

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	What is a <i>Software Rasterizer</i> ? . . . .	3
1.3	Why <i>C</i> ? . . . .	3
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Algorithms . . . . .	3
2.1.1	Rasterisation . . . . .	3
2.1.2	Clipping . . . . .	3
2.1.3	Bresenham Line Drawing . . . . .	3
2.1.4	Anti-Aliasing . . . . .	3
2.2	Math Concepts . . . . .	3
2.2.1	Vectors . . . . .	3
2.2.2	Matrices . . . . .	3
2.2.3	Transformations . . . . .	3
<b>3</b>	<b>System Architecture</b>	<b>3</b>
3.1	Overview . . . . .	3
3.1.1	System Diagram . . . . .	3
3.1.2	Project Structure . . . . .	3
3.1.3	Conventions . . . . .	3
3.2	Data Design . . . . .	3
3.2.1	Vectors & Matrices . . . . .	3
3.2.2	Primitives: Triangles and Vertices . . . . .	3
3.2.3	Textures and Materials . . . . .	3
3.2.4	Framebuffer and Zbuffer . . . . .	3
3.2.5	Scene, GameObject, Camera, LightObject . . . . .	3
3.3	Software Design . . . . .	3
3.3.1	Procedural Programming vs. Object Oriented Programming . . . . .	3
3.3.2	Graphics Pipeline . . . . .	3
3.3.3	Scene Management . . . . .	3
3.3.4	Rendering . . . . .	3
3.3.5	Shading . . . . .	3
<b>4</b>	<b>Conclusion &amp; Future Work</b>	<b>3</b>
4.1	What I learnt . . . . .	3
4.2	Limitations . . . . .	3
4.3	A future implementation in C++ . . . . .	3
<b>5</b>	<b>References</b>	<b>3</b>

# 1 Introduction

## 1.1 Motivation

## 1.2 What is a *Software Rasterizer*?

## 1.3 Why *C*?

# 2 Background

## 2.1 Algorithms

### 2.1.1 Rasterisation

### 2.1.2 Clipping

### 2.1.3 Bresenham Line Drawing

### 2.1.4 Anti-Aliasing

## 2.2 Math Concepts

### 2.2.1 Vectors

### 2.2.2 Matrices

### 2.2.3 Transformations

# 3 System Architecture

## 3.1 Overview

### 3.1.1 System Diagram

### 3.1.2 Project Structure

### 3.1.3 Conventions

## 3.2 Data Design

### 3.2.1 Vectors & Matrices

### 3.2.2 Primitives: Triangles and Vertices

### 3.2.3 Textures and Materials

### 3.2.4 Framebuffer and Zbuffer

### 3.2.5 Scene, GameObject, Camera, LightObject

## 3.3 Software Design

### 3.3.1 Procedural Programming vs. Object Oriented Programming

### 3.3.2 Graphics Pipeline

### 3.3.3 Scene Management

### 3.3.4 Rendering

3

### 3.3.5 Shading

# 4 Conclusion & Future Work

## 4.1 What I learnt

## 4.2 Limitations