Daniel **Martinez Amigo**

Computer Systems Engineer

Alcala de Henares, Madrid, Spain

Spain(+34) 611684999 / UK(+44) 7756691924 | danimtz@outlook.com | danimtz.github.io | danimtz | danimtz15

Education

University of Bath *Bath, UK*

M.Eng in Computer Systems Engineering *Oct. 2016 - Jun. 2020*

- First-Class Honours.

Engineering Experience

Unreal Engine Developer *Madrid, Spain*

EPAM Systems (OptivaMedia) *Jul. 2022 - Jul. 2023*

- Developed an in-house prototype for a multiplayer virtual reality cinema and multimedia application using Unreal Engine 5.
- Completed an 8 week professional power work shop for Unreal Engine 5 run by INCAS training.
- Co-led technical and game design aspects of a small team to create educational gameplay experiences using UEFN(Unreal Editor for Fortnite) to be used within EPAMS eKids learning programme.

Veist Engine : Vulkan renderer/game engine *Madrid, Spain*

Graphics programming personal project *2021*

- Vulkan based engine written in C++17 with the objective of learning the graphics API.
- Command buffer recording abstraction layer that allows for easy creation and recording of Vulkan commands.
- Physically Based Rendering of GLTF models using GLSL shaders. The shaders can be compiled at runtime using SPIRV-Reflect or loaded from a cache.
- Image Based Lighting based on ”Real Shading in Unreal Engine 4” from Siggraph 2013
- Entity component system handles scenes and provides a data driven approach to scene simulation that improves performance.
- Scene editor that allows adding/editing/removing entities and their components.
- Forward and Deferred Rendering
- Framegraph rendering system that enables custom rendering architectures to be easily created and altered, as well as automatically handling descriptor sets and synchronisation of vulkan GPU structures such as barriers.

SRGB : Software Renderer *Madrid, Spain*

Graphics programming personal project *2020*

- Software renderer written in C++ capable of physically based rendering of 3D models with multiple directional lights.
- The dependencies are SDL2 for window management and stbimage for texture loading. Everything else was written from scratch including math utilities such as vector and matrix operations and an .obj file parser. CMake can be used to build the project.
- Programmable vertex and fragment shader graphics pipeline using C++ virtual functions.
- Main shader achieves PBR rendering using Cook Torrance BRDF shader.
- Parallellism with OpenMP is used in multiple stages of the main render pipeline including vertex shader, primitive clipping, the rasterizer and fragment shaders to greatly increase performance.

BBS template matching algorithm and its application to cloud tracking *Bath, UK*

University final year project *Feb. 2020 - May. 2020*

- Digital image processing based project that compared the Best Buddies Similarity (BBS) algorithm to other template matching algorithms when matching templates from sequences of clouds 15 and 30 minutes apart taken from satellite images that had been corrupted by different types and amounts of noise. Overall the BBS algorithm was deemed to be a useful method for cloud tracking, however not enough to de-throne other alternatives in either speed (Cross Correlation Coefficient) or noise performance (Ordinal Measures)

Skills

Programming	***Experienced:*** C++, C, MATLAB ***Familiar:*** C#, Python, Java, JavaScript
Graphics	***Experienced:*** Vulkan, GLSL ***Familiar:*** OpenGL, HLSL
Other software/Game Engines	***Experienced:*** Unreal Engine 5 (Blueprints & C++) ***Familiar:*** Unity, GameMaker
Developer Tools	***Experienced:*** Visual Studio, RenderDoc, Git, Github, Jira
Languages	***Native:*** Spanish ***Fluent:*** English