Mahdi Golmohammadi

MahdiGolmohamadii | MahdiGolmohamadi | mhdglmd@gmail.com

Summary

Creative technical artist/pipeline developer with more than two years of experience in the industry. Specialized in developing plugins and pipelines for DCC software (such as Blender, Maya, etc.), using **Python**, **C++**, and **C#**. My distinct blend of experience in programming and creative insights from my academic background, have positioned me to be a unique candidate.

Work Experience

Game QA Tester - PTW (Montreal, Canada) May 2024 - Present

- Executed test procedures to identify and report game bugs, and collaborated with development teams to replicate issues and verify fixes, ensuring a seamless gaming experience.

Technical Artist - William Smart Trading (London, UK) Feb 2023 - Dec 2023

- Collaborated with artists to create streamlined workflows and **pipelines** for asset importation and data visualization with **Python** for **Blender** (DCC software) resulting in a 35% reduction in production time.
- Conducted rigorous testing and prototyping, continuously adding features for artists to enhance the pipeline efficiency.

Software Engineer - Telina (Tehran, Iran) Jan 2019 - Nov 2019

- Enhanced and maintained a suite of desktop software applications using **Python** and **C++** resulting in at least a 10% increase in the team's efficiency and productivity.
- Implemented automated testing processes such as unit tests, and reported defects to ensure high-quality software.

Education

MA (Computer Animation) at **Sheffield Hallam University** - UK Sep 2021 - Dec 2022
BSc (Software Engineering) at **Tehran Polytechnique** - Iran Sep 2015 - Feb 2020

Skills

Programming Languages:	Python, C++, JavaScript, C#
API / Libraries:	PySide/PyQt, OpenCv, .Net
Databases:	SQL, MySQL, MongoDB
Software:	Maya, Blender, Unity3D, Unreal Engine, Adobe Suite
Misc:	OpenUSD, FFmpeg, Pipeline development, Game engine & DCC plugin development, Shader development, Version control systems(Git)

Projects

Procedural Building Generator - (Python, Blender)

- Designed and implemented an add-on for Blender that generates distinct buildings using Python and Geometry Nodes.
- Allows artist to automatically create building structures by putting dimensions and number of levels.

EZLattice - (Python, Blender) GitHub Page

- Implemented the "EZLattice" plugin for Blender with Python, streamlining the process of applying lattice (non-destructive mesh deformations) to 3D meshes.

Asset Library - (Python, PySide/PyQt) GitHub

- Designed and implemented a centralized asset management solution utilizing Python and PySide/PyQt frameworks.
- The system reads asset information from JSON files and presents it in an intuitive interface. streamline asset management processes, improving workflow efficiency.

AR ball-tossing game - (Unity3D, C#)

- Developed an **Augmented Reality (AR)** game utilizing **Unity**, **C#**, and Vuforia for my Bachelor's thesis project. The game seamlessly detects flat surfaces and dynamically integrates targets into the environment, fostering player engagement and interaction.

Repeat (short animation)- (Blender) Watch here

- Produced the concept and executed a trailer for a short animation as part of the master's thesis, showcasing proficiency in **character modeling**, **rigging**, and **animation techniques and principles**.
- Developed **custom shaders** tailored to the project's aesthetic requirements, underscoring a deep understanding of rendering principles and leveraging advanced rendering techniques to elevate visual fidelity and narrative impact within the animation.
- Developed **custom Python scripts** to streamline production time, automate repetitive tasks, and establish a pipeline for the project.

AI-enabled Pac-Man - (Java)

- Developed an original version of the classic game "Pac-Man" using Java. It features three distinct AI-controlled ghost characters with varying behaviors.

Music identification program - (C++, multi-threaded programming)

- Developed an application that can identify a given part of music, using **multi-thread programming** techniques and **C++**.