SAGIT BOLAT

University of California, Berkeley

@ sagitbolat@gmail.com Portfolio Linkedin GitHub

SUMMARY

A passionate student of the computer and data sciences who is looking for experience and an opportunity to learn and develop his skills. A team player who is not afraid of challenges, but knows when to ask for help.

TECHNICAL SKILLS

Languages:

- C and C++
- Python
- C#
- Java
- HTML/CSS/Javascript/React
- SQL
- Lisp and Scheme

Frameworks and APIs:

- [C/C++] **Backends:** Win32, SDL2, OpenGL, DirectX, OpenAL
- [C/C++] **UI:** ImGui, nuklear, nCurses
- [Python] **Data processing:** Numpy, Pandas, matplotlib (and related), sqlite3 etc.
- [Python] **Automation and testing:** Selenium, PyTest, BeautifulSoup, OpenCV, etc.
- [C#] .NET/.NET Core
- [C#] Unity

Other:

- Linux OS
- Raspberry Pi and Arduino

EDUCATION

Data Science

University of California, Berkeley

Higher Secondary

Quality Schools International of Bratislava

PERSONAL SKILLS

- Enjoys challenging problems.
- Ability to work under pressure.
- Comfortable working independently.
- Eager to learn new skills and technologies.

PROJECT SUMMARIES

[C/C++] Custom game engine (WIP)

- Written from scratch with an SDL2 windowing back-end
- Hardware accelerated rendering API for 2D graphics with an OpenGL backend
- Audio playback API with an OpenAL backend
- User Interface API with an ImGui backend
- Vector and matrix math library
- Random number generation library, including sequential generators, and 1D and 2D noise
- Memory allocations: bump, pool, and free-list allocation
- 2D collision engine.
- Tilemap system, and a full tilemap editor. Uses a custom file format that is loosely based on the BMP file specification
- Scene system and scene dispatch system.
- Custom build system that allows incremental building. Written in batch/bash script.
- Built for simplicity and customizability. Architecturally agnostic. Provides multiple execution entry points to the user.

[C#] Sudoku Solver

- 3 different levels of difficulty for generated puzzles.
- A way to guarantee unique solutions to the puzzle, making the quality comparable to handmade puzzles.
- A speed slider that determines how fast the algorithm runs (slower speeds are meant to show how the algorithms execute in real time).
- An input system that allows the user to attempt to solve the puzzles by hand.
- An iteration counter to benchmark relative speed of the algorithms involved.

[Python] Esoteric Programming Language Interpreter

- A parser for the graphical programming language Piet. Uses OpenCV for image processing.
- A full program Stack and Stack frame manager.
- Command-line interface with argument parsing for the interpreter.