HARRY HOLLANDS

CAMBRIDGE, UK • HARRYSJH98@GMAIL.COM

Technical Skills

- **Languages:** C++, C, Lua, GLSL, Java, C#, Python
- **Tools:** Git, SVN, Perforce, JIRA, Hansoft, Confluence, CMake, Doxygen
- **Platforms:** Windows, Linux, PlayStation 4, PlayStation 5, Xbox Series X
- **Standards/Formats:** Vulkan, OpenGL, GLTF, JSON, TTF
- **Frameworks/APIs:** WinAPI, x11, Dear ImGui, Unity

Work Experience

Frontier Developments (Cambridge, UK)
Full Game Engine Programmer, February 2021-Present

- Extended Cobra, the engine's asset compression to work for Xbox Series X, Playstation 5 and Nintendo Switch, aswell as maintaining the compression for existing past-gen platforms (including Windows)..
- Served as a first contact point between the Jurassic World Evolution 2 team and the engine team. Also worked on integrating smoother, quarterly engine releases for the game as opposed to continuous monthly updates.
- Took over maintenance of the engine's new user interface module, allowing the old module to be wholly replaced, and adding new features requested by game teams, co-operating with the engine rendering and game UI teams in the process. Warhammer Age of Sigmar: Realms of Ruin will be the first game to ship with the new features.
- Joined a small feature group of engine programmers to work on CMake support, so the older, outdated proprietary build system could be replaced.

Frontier Developments (Cambridge, UK)
Graduate Game Engine Programmer, June 2019-February 2021

- Added debug tooling to improve the productivity of the localisation QA team for Planet Coaster: Console Edition, mainly by adding support for changing the language of the game on-the-fly within the engine.
- Maintenance and bug fixing work for Planet Zoo.
- Helped to maintain and improve the underlying custom build system that the engine uses, including helping to support Xbox Series X and Playstation 5.
- Maintained a bespoke integration branch of Jurassic World Evolution 2 throughout its early development. My responsibility was to fast-forward this branch's game code on a monthly basis against the bleeding edge engine, ensuring that other members of the engine team could check that their changes didn't break this particular game.

Simplay Studio (Nottingham, UK)
Game Development Intern, June-July 2018

- Created a proof-of-concept demo game from the ground-up using the Unity engine. This demo was intended to gamify construction planning, to engage civil engineering students with the monetary and logistical costs of construction and its processes.

Website: https://harrand.github.io/
GitHub: https://github.com/Harrand
LinkedIn: https://www.linkedin.com/in/harrand/

HARRY HOLLANDS

CAMBRIDGE, UK • HARRYSJH98@GMAIL.COM

Education

University of Nottingham (Computer Science BSC Hons, 2016-2019)
First Class (Equivalent to 4.0 GPA using Duke conversion)

Some of the key modules as part of this course are as follows:

- C++ Programming - consolidated knowledge of advanced C++11 programming and later (93%)
- Software Engineering Group Project - Unity C# 3D Game Development (87%)
- Individual Dissertation - Contrasting occlusion culling techniques on the CPU (75%)

Portsmouth Grammar School (2009-2016)

A-Levels Received: Maths (A), Electronics (A), Physics (B)

GCSEs Received: 1 A*, 1 A, 5 B's, 2 C's. Including Maths: A*

Additional Experience

Red Nightmare | C++20, Lua, Topaz
Game, 2021-2023

- Working alone on a high-fantasy action roguelite called Red Nightmare using my game engine, Topaz, during my free-time.
- Requirements and game design document produced early-on in development.
- Lua scripts drive game logic, heavy-duty work such as animation and rendering is done within C++.
- Development of the initial prototype was partially live-streamed on YouTube. Gameplay inspired by Diablo 2 and Risk of Rain 2.

Topaz | C++20, OpenGL, Vulkan
Game Engine, July 2015-Present

- Wrote a 3D game engine largely from scratch.
- Bespoke graphics API implemented over OpenGL 4.6, or Vulkan 1.3, configurable at compile-time via CMake presets.
- Custom shader language, TZSL - a strict superset of GLSL. Unlike GLSL, TZSL shaders work across OpenGL and Vulkan without modification.
- Support for game-side Lua scripting.
- Support for embedding small text files directly within the executable, cutting down on initialisation time for games with many config/script files.
- Support for importing 3D animated models using a custom-built GLTF importer.
- Automated regression testing, unit testing and documentation generation via Github Actions. Documentation can be found at https://harrand.github.io/Topaz/.
- Rich profiling support using the Tracy profiler.

Software Engineering Group Leader | Unity, C#
Unity Game, September 2017-May 2018

- As part of a university course, I led a group of six other computer science students to create a mountain-climbing virtual board game within the Unity engine.
- Utilised scrum, pair-programming and unit tests to create a solid, robust project. It scored 87% of the available marks.

Website: https://harrand.github.io/
GitHub: https://github.com/Harrand
LinkedIn: https://www.linkedin.com/in/harrand/