James Huxtable

Experienced and passionate C++ Games Developer with a love for Games and Technology. And a damn good Lecturer

Soft Skills

- Diligient
- Empathetic
- Self-Motivated
- Determined
- Mentor
- Detail-Focused
- Adaptable
- Problem Solver

Programming Proficiency

C++

C#

HTML/CSS

C.I.

CMake

Python

OpenGL

Version Control

Biography

An experienced C++ games developer with a passion for programming and undying love for Computer Games. Games have become the formative part of my career which has driven me to learn to program in the first place. I'm extremely passionate about choice and a strong advocate of cross-platform development with a huge admiration for those who write opensource libraries and support communities off their own backs. Whilst I have experience with other languages such as Python, my current preferred choice is still C++, though I've a keen eye on others such as Rust.

One of my most prominent traits is to never stop learning. I love to embrace new technologies and ideas, to stop learning is to me the dictionary definition of boredom. I love to teach and to support others. It's extremely satisfying when you can help grow others around you and at the same time learn and improve from them. I believe we must never stop growing as individuals.

Employment History

Senior Lecturer at *University of the West of England, Bristol*

Sep 2018 - Present

Promoted due to my success as a lecturer and my revival of the 2nd year C++ module. In this new role, I was given additional responsibilities and input in to how the programme is delivered.

- Driving force behind a greater focus on the types of technology used in game development, leading to the creation of a new module named Games Tech 101.
- Proactively involved in the software consortium group and provided direct communications between IT and the Creative Technology programmes.
- Provided mentorship to both BSc and MSc students leading to their successful employment within the Games Industry.

Lecturer at *University of the West of England, Bristol*

Sep 2015 - Sep 2018

Jumped at the opportunity to inspire and educate students with whom I shared a common passion. My initial role was to refresh the poorly performing C++ module taught to the second years.

- Revitalised and introduced modern C++ language features to the second year module. This led to a more relevant and tightly focused module which proved popular amongst students.
- Worked closely with the first year C++ module and helped to reshape it, resulting in a module that students found more enjoyable as they spent more time making games.
- Covered maternity leave for additional C++ modules which included the writing of new exams and coursework assignments.
- Introduced best practices into C++ modules, such as the use of version control, continuous integration systems and portfolios.

Software Stack

Clion
PyCharm
git
VS Code
OBS
Visual Studio
Shotcut
Linux
Windows
Audacity
11ty
Tiled

Education

MSc with Distinction [1st] in Computer Games and Entertainment (Goldsmiths University, London)

BSc with Distinction [1st] in Games Technology (University of the West of England, Bristol)

PGCert HE in Teaching and Learning in Higher Education (University of the West of England, Bristol)

Games Published

Football Manager 2011
Football Manager 2012
Football Manager 2013
Football Manager 2014
Football Manager 2015
Football Manager 2016

Links

http://huxyuk.github.io
/HuxyUK
/jameshuxtable

Gameplay Programmer at *Sports Interactive, London*

Feb 2012 - Sep 2015

Moved to a more senior role, with a greater influence on non-match related gameplay elements. The primary focus was on improving interactions between players, non-players and the media reporting within the game.

- Successfully undertook an extremely large refactoring project that decoupled a monolithic data class that was being shared throughout the large codebase. Changes to the source code had to be tested in a sandboxed environment due to the critical areas being involved.
- Improved on relationship forming algorithms and how non-player characters should react to the manager's comments depending on a set of relevant criteria.
- Wrote a narrative system based on Neural Networks from the ground up. The idea was to make inter-manager communications and their responses to feel more realistic. This required a number of networks for different types of game event and the use of offline training. To help train the networks a training application was written using Qt.

Junior 3D Programmer at *Sports Interactive, London*

May 2010 - Feb 2012

My first job within the industry entailed working closely within the match team to help improve various aspects of the games' 3D match experience. Many of these features ended up being "back of the box features"

- Added several new cameras to the game that improved on the player's experience when watching the matches.
- Embedded a video encoder that could capture the gameplay, encode it and upload the resultant video to YouTube. The codec used was Theora's and offscreen render targets were used to grab the 3D output, before then being compressed. Chromo sub-sampling (4:2:0) was used due to limitations at the time.
- Wrote an AI system that could correctly frame cameras to follow the game's action. Directly inspired by the coverage you would see on TV, it was combined with an FSM to control which camera's should be used and when. This was affectionately nicknamed "The Director"

Interests

Games
Arsenal FC
Linux
Homelabs
Family
Gymming it
DevOps

Projects

ASGE

URL: https://github.com/HuxyUK/ASGE

C++ OpenGL Linux MacOS Windows MIT License

ASGE is an easy to use C++ OpenGL games framework and was designed from the ground up to allow 2D games to be quickly developed and prototyped. Its core focus was to provide the building blocks required for students to develop their own games. It specifically aims to reduce the complexity of larger game engine and frameworks. Over the years its functionality has drastically morphed and many more features have been added, whilst still remaining easy to use.

At present, there are almost **1000** games written using it. The framework is opensource and makes use of several dependent opensource libraries as well. The heavily integrated batching system ensures that draw counts are kept to a minimum, whilst still supporting OpenGL 3.3 and the more modern OpenGL 4.5 API. Features available include: custom fragment shaders, MSDF font rendering, texture caching, off-screen rendering and instancing. For batching, OpenGL 3.3 makes use of traditional UBOs and the modern version uses a single large SSBO.

PyASGE

URL: https://github.com/HuxyUK/PyASGE

Python PyBind11 Linux MacOS Windows MIT License

PyASGE are python bindings for the game framework ASGE. Whilst these are considerably newer than the game engine itself, they provide the same functionality as the C++ framework and are documented using Sphinx. The bindings themselves are constructed using PyBind11 and aim to embrace Pythonic constructs such as properties where appropriate. These bindings are currently in their second year of use and are working well, with positive feedback from students who have used them.

Simple Software Installer

URL: https://github.com/HuxyUK/simple-software-installer

C++ Elements MVC Windows YAML

SSI (Simple Software Installer) is a tool designed as a response to the pandemic and is a software installer for a Windows applications. It supports categories, dependencies as well as the typical operations i.e. install. Applications can easily be added or even modified by dropping in a new YAML file, which themselves can be configured in a matter of minutes. The installer was hugely successful and became the students preferred choice over the University's central deployment application.

XMLTV-SD Unraid Plugin

URL: https://github.com/HuxyUK/docker-xmltv-sd-json

Docker Bash Unraid 98k+ Downloads

A docker image with a JSON Schedules Direct enabled script. With almost 100k downloads, this docker image was designed to be deployed on an Unraid server. Its job is to scrape Schedules Direct listings which can then be parsed by third-party apps such as TVHeadend. To prevent the constant need to poll for changes, a UNIX socket can be used by the script which can be monitored by supported TV software stacks.

References & Testimonals

Happy to provide references and testimonals on request