Stephen Badrick – Software Developer and Games Programmer
Framwellgate Moor, Durham
Email: stephenbadrick@gmail.com Phone: 07599800952, no private or withheld numbers
https://stephenbadrick.co.uk

Skills

- Object-oriented programming using C#, C++, and Java
- Games development using XNA/MonoGame, Unreal Engine 3 and 4 with Blueprint and C++, and Unity 4 and 5
- Git, SVN, Kanban, Visual Studio

Experience

Frontwire Studios, October 2016 – Present

Volunteer Lead Programmer, December 2016 – April 2017

- Define task priorities and task allocation
- Ensure tasks are being completed and deadlines are being met
- Aid other developers
- All previous tasks

Volunteer Game Programmer, October 2016 – Present

- Design, implement and test game mechanics for Galaxy in Turmoil, using Blueprints in Unreal Engine 4
- Provide regular updates to the build manager
- Meet deadlines and achieve objectives
- Extend Unreal functionality with plugins written in C++

JobCentre Plus, Chester-le-Street, August 2016 – September 2016

Volunteer Digital Support and Customer Services

- Provide aid with using the Department of Work and Pensions systems to people who are unfamiliar with the systems, or computers
  - Improved interpersonal and communication skills

Education

BSc (Hons) Games Software Development 2:1, University of Sunderland, 2013 – 2016

- Create a Weapon for UDK
  - Used UnrealScript to develop a mod that introduced a custom weapon and HUD. Developed skills in UnrealScript, and modelling skills using 3ds Max
- Develop a game using Allegro 5 and C++
  - Developed a small 2D platformer engine in Allegro 5. It is capable of reading data from storage and parsing it as a level, including collision data, player spawn point, enemy spawn points and enemy types
- Game Design Documentation
  - Produced a Game Design Document for a game, detailing high-concept, story, mechanics, and assets
- Object-Oriented WinForms Application
  - Tasked to create an application which appropriately used object-oriented principles. Performed research into the MVC pattern and applied it to the program to better adhere to the SOLID principles of OO design
- Advanced Android Mobile Application

- - Used Java and Android Studio to develop a mobile application, using persistent data storage and social network connectivity
- Artificial Intelligence Pathfinding
    - Implemented the A* pathfinding algorithm using C#. The application can read level data of arbitrary size from a file and determine the best path from a start point to a target
- Project Proposal and Proof-of-Concept Prototype
    - Produced a Game Design Document for a game and used Unreal Engine 4 and Blueprints to produce a proof-of-concept
- Multiplatform Unreal Engine 4 Game
    - Developed a game using Unreal Engine 4 that targeted PC and Android. The game was written in C++ and features a simple chase AI
- Dissertation: Colour-blind Suitable Video Game
    - Took a project from initial idea and concept and produced a proof-of-concept prototype, by performing research into UX and human-centric design, with a focus on colour-vision disorders. Presented prototype to several people as a viva

Personal Projects – GitHub.com/Zero-One101

UQuake – C++, Unreal Engine 4
UQuake is a remake of elements from the original Quake, remade in Unreal Engine 4. It currently has a weapon system with 3 weapons, using shared ammo pools. I aim to have enough of Quake recreated to accurately remake E1M1.

MonoTroid – C#, MonoGame
MonoTroid is a Metroid engine written in C#, using the MonoGame library. It is a 2D tile-based platformer. It is currently capable of reading in data from a file using a custom format and parsing it to produce a playable level. The player class is designed to use a hierarchical finite state machine to reduce redundancy and complexity, and allows for quickly extending to add in more functionality to the player. This also serves to follow the Single Responsibility Principle.

ZrnBot – C#, IronPython
ZrnBot is an IRC bot, designed to join chatrooms and accept and respond to user commands. For example, a user can say the command "%log" in the chat, and ZrnBot will retrieve the log of everything said for that day, upload it to a remote server, and reply with the link to the log in the chat. Previous versions of ZrnBot were written entirely in C#. However, this meant that every time a change in functionality needed to be made, ZrnBot would have to be closed. The current revision will utilise IronPython as a scripting language, while using C# only for core processes, such as connecting to chat rooms. This way, functionality can be added or extended, without the bot suffering any downtime.