Andrew Patton

COMPUTER SCIENCE WITH GAMES DEVELOPMENT

andrewpatton.dev | atpatton97@gmail.com

[On Request] | 28 The Meadows, Fawdon NE3 3NA

UBISOFT	ONGOING

User Research Lab
Evaluating not only the quality of a game, but reporting back what areas are lacking, and how they might be improved.

ACCENTURE	JULY, 2015

Work Shadowing
Working in AGILE/scrum teams. Executing & documenting tests, following a predefined plan. Exploring different work aspects of a team and how they interact to co-develop a software product

UNIVERSITY GRADES

Computer Science with Games Development

First Year	**74% (1:1)**
Second Year	**77% (1:1)**
Third Year	**72% (1:1)**
Final Grade	**75% (1:1)**

HOBBIES & INTERESTS

- Video Games
- Podcasts about technology and physics (StarTalk, Vergecast)
- Comics
- Game Jams

TECHNICAL PROFICENCIES

- C++
- C#
- Java
- Unity
- Unreal
- GitHub

These languages and tools are a short selection based on personal experience and preference.

PROFILE

Technology enthusiast recently graduated from Northumbria University, with First Class Honours in BSc (Hons) Computer Science with Games Development, with a final grade of 75%. I thrive on not only solving problems, but digging deep into all possible solutions, deciding which will be best for the given context, and why.

NOTABLE MODULES

The following modules have been selected for their relevance in providing a solid foundation for my existing knowledge and experience. Further examples can be found under the "Portfolio" section of my website.

Software Architecture for Games (82%)
Using C++ to extend a DirectX wrapper into a functional 2D game engine, applying an optional advanced element. My engine was built on the concept of generic, flexible components, similar to the Unity framework. Gameplay programmers can add their own game objects, attaching both core and custom components to alter the behavior of and breathe context into the game object. This module focused on the underpinning programming patterns of a game engine, and the strengths/weaknesses of different approaches, along with a supporting commentary on how this can greatly affect the working environment of a large-scale development team.

Software Engineering Practice (72%)
Working within a team to design, build and evaluate a game via the Unreal Engine. Tasks were split between members, with my "missions" including "Environment Creation" and "Game Design", along with the group mission "Interaction Programming". Our game was a top-down, offering the player the exclusive choice of either a gun or a melee weapon, before they must overcome puzzles and enemies before the final battle atop an unstable volcano.

Computer Networks and Control Systems (92%)
Using a combination of C++ and Java to control a sprite with a custom peripheral (MBED board) across a local network (via ethernet), the on-screen program displayed a game inspired by Atari's 1979 *Lunar Lander*. The player must manage limited fuel resources whilst finding a flat area to land safely, controlling the craft using the board's joystick & potentiometers. Flashing LEDs and sound effects gave feedback to the player, simulating the ship's on-board systems (speedometer, altimeter, fuel gage, etc.)