# Contact

[atolbertsmith@outlook.com](mailto:atolbertsmith@outlook.com)

(301) 992-8865 (mobile)

# Website

[github.com/atolbert01](http://www.github.com/atolbert01)

# Languages

|  |  |
| --- | --- |
| C# | ● ● ● |
| C/C++ | ● ● ● |
| Java | ● ● ● |
| Python | ● ● ○ |
| Lua | ● ● ○ |
| JavaScript | ● ● ○ |
| HTML/CSS | ● ● ○ |

# Tools/Tech

|  |  |
| --- | --- |
| Git | ● ● ● |
| Windows | ● ● ● |
| Ubuntu | ● ● ○ |
| macOS | ● ○ ○ |
| OpenXR | ● ● ○ |
| Oculus | ● ● ○ |
| Android | ● ● ○ |
| Visual Studio | ● ● ● |
| Unity Engine | ● ● ● |
| Unreal Engine | ● ○ ○ |
| Autodesk Maya | ● ○ ○ |

Creative problem-solver, software engineer, and game developer. Using technology to improve and enhance the real lives of actual humans. Currently building skills in 3D graphics, VR/XR, and gameplay programming to advance the goal of creating immersive interactive experiences for training, education, and entertainment.

# Summary

* Software engineer with almost a decade of experience
* Team leadership experience
* Virtual Reality game development
* Networked multiplayer development
* Mobile game development
* Unity Engine, Unreal Engine, OpenGL competency

# Education

**M.S. Computer Science** – University of Maryland, Baltimore County, 2021

**B.S. Computer Science** – University of Maryland University College, 2016

**B.A. Audio Design and Production** – Columbia College Chicago, 2009

# Experience

**Lead VR Programmer**: Naval Nuclear Laboratory – February 2022 to Present

Lead a team of programmers in a fast-paced collaborative environment to deliver immersive training simulations to the US Navy using virtual reality technology and game development practices. Development leverages the Unity game engine and OpenXR standard targeting PCs and a variety of VR headsets including HTC VIVE and Varjo Aero and makes use of an array of VR peripherals such as VR treadmills (Virtualizer, Infinadeck), haptic gloves and vests, body trackers, and controllers. Launched a first-of-its-kind VR simulation trainer to nuclear shipyards featuring online networked multiplayer supporting up to 8 individuals.

# Experience (cont)

**Cryptologic Software Engineer:** U.S. Department of Defense – January 2019 to Present

Perform engineering functions critical to producing and delivering Information Assurance material including Java programming. Part of a high-performing, zero-defects team that develops, tests, generates, and manages cryptographic key material. Develop and test high-assurance software, and generate cryptographic keys/codes sent to customers in physical or electronic form.

**Application Developer:** PA Higher Education Assistance Agency – September 2016 to July 2018

Developed desktop and web-based productivity and automation applications in Microsoft .NET environment (C#, Visual Basic) as well as JavaScript and HTML. Backend via IBM DB2 SQL. Participated in agile cross-organization modernization effort to create web-based tools to improve call-center workflow. Individually developed applications that saved more than 100,000 dollars and more than 6000 call-center hours.

# Notable Projects

Ray Tracing Renderer (C++)

A ray tracing renderer that outputs images featuring accurate shadows and reflections. Supports ray-sphere as well as ray-triangle intersections.

Procedural Island Generator (C++/OpenGL)

An interactive 3D scene with procedurally generated mountainous terrain, GLSL shading, texture mapping, and player character controller with collision behavior for scaling sloped terrain.

The Sky Is Alive (C#/Unity Engine) - <https://youtu.be/p0AwMvCQr_U>

A 2D platformer game for Android devices. Players navigate obstacle courses by tapping to jump or swiping to dash across large expanses. In contrast to auto-runner style platformers, player movement is unrestricted allowing for free exploration without the use of onscreen joysticks or buttons. The device surface is divided into two regions – tapping the left or right side of the screen causes the player character to hop left or right respectively.

Battle Royal Rhythm (prototype) (C#/Unity) - <https://youtu.be/mgciPbGXg0E>

A prototype rhythm platformer game designed for Columbia College Alumni Game Jam 2021. Enemy movements are synced to the beat of the music and the player must time their movements for maximum effect. This game jam involved working together with a group of about 8 other Columbia Alumni to build a game in a short time period.

Galentha (C#/Unity Engine)

Implemented peer-to-peer multiplayer features for 3D open-world shooter game using Unity’s MLAPI multiplayer framework. Worked as part of a small team on an ongoing independent production.

Latch Key Quest (C#/Unity Engine)

A retro top-down adventure game for desktop PCs. Worked as director and programmer with a team of several computer science students. Implementation involved developing enemy AI behaviors, play mechanics, and pixel-perfect camera scripts.

Clove engine (C#/.NET/monogame)

A suite of game development tools including a 2D skeletal animation program as well as an in-game level editor.

Claudio QA Bot (Python/PyTorch)

A proof-of-concept implementation for a QA bot based on the BERT (Bidirectional Encoder Representations from Transformers) neural model, which reads text input non-sequentially (as opposed to directionally left-to-right or right-to-left) to learn the context of words within their total surroundings. Trained on Stanford’s SQuAD dataset, the QA bot is capable of answering questions on a variety of topics gleaned from a set of Wikipedia articles.

The Celebration of Nature: A Survey of Landscape Modeling Techniques (Paper)

Literature review examining the historical development of 3D landscape modeling techniques with an emphasis on the identification of future applications and areas of study beyond film and games to include scientific analysis of the geomorphological impacts of climate change. Topics covered include procedural modeling, simulations, example-based modeling, AI-assisted simulations, and full ecosystem modeling.