Howe Yang

Unity developer focused on providing unparalleled experiences
Graduate of Computer Science at Waterloo

howeyang.github.io
647 - 863 - 5685
howechyang@gmail.com

Work Experience

Senior Integration and Support at Upsight

Sept. 2016- Oct. 2018

- **Served as the technical point of contact and escalation lead for AAA clients** such as Niantic, Ubisoft, Activision, EA, Nickelodeon, Scientific Games, Twitch and more; successfully resolved over 100 high priority incidents
- **Helped clients optimize the use of Mobile SDKs** which ranged from writing code, debugging issues and investigating problems
- **Provided expertise in Big Data analytics** by providing best practices and writing advanced analytical queries to improve performance
- **Collaborated with Data Engineering, QA, Mobile Ad, and Customer Success teams** to investigate, debug and monitor back-end software infrastructure
- **Assisted in maintaining Hadoop and SQL databases** which lead to shorter disruptions and quicker resolutions

Tools :
Unity
Android Studio
Xcode
SQL
Apache Hive
Airflow

Key Contributions

- **Employee of the month** within the first year of employment
- **Led to contract renewal of key client** due to my contributions in issue resolution
- **Documented Big Data ingestion pipeline** from end-to-end, setting the foundation for tracking and improving overall workflow and as entry information for new staff

Developer and Designer at Hcyang Studios

Nov. 2016- present

- **Created a unit defense game** inspired by Auto Chess using Unity, state machines and path finding
- **Prototyped with shaders, particle effects and game concepts** that were inspired from contemporary games like Breath of the Wild and Path of Exile
- **Utilized Unity's UI Canvas to replicate unique UI designs** such as the PS4 background theme

Tools :
Unity
C#
JS
CSS

Education

Bachelor of Computer Science (Honours) - University of Waterloo

Sept. 2012- Apr. 2016

- **Researched and implemented graphical techniques to build games** by writing techniques involving ray tracing, vertex and fragment shaders in Open Gl
- **Built a Python Bot for a Prisoner's Dilemma Competition**, which performed above the ideal algorithm benchmark by trialing different behaviours
- **Studied modern user interfaces** and how to create user-friendly and intuitive user interfaces by using Gesalt's design principles

Tools:
C
C++
Java
Open Gl
Lua