WANG, XiangZhi (Eric)
E-mail: mdwxz@icloud.com ｜ **Tel:** +86 15507482495/ +852 65289897

EDUCATION BACKGROUND

The Hong Kong Polytechnic University (Yau Tsim Mong, HK)   09/2019 – 06/2023

Bachelor's Degree of Engineering, Bachelor of Science (HONS) in Computer Science

Core courses: Introduction to Computer Systems, Introduction to Data Analysis, Computer Networking, Operating Systems, Discrete Mathematics, Database System, Data Structure, Object-Oriented Programming, Software Engineering, System Programming, Human-Computer Interaction, Design and Analysis of Algorithms, Computer System Security, Machine Learning, Computer Vision, Legal Aspects and Ethics of Computing.

Field Orientation: Software Engineering, System/Architecture/Flowchart Design, VR Development, System/Web Development

GPA: 3.37

PROJECT EXPERIENCE (IN CHRONOLOGICAL ORDER)

Automated Examination System – Course Individual Project (`GitHub Link`)   24/12/2020 – 08/01/21
- Program Language: Java, MySQL, Python3
- Designed the entity-relationship between three entities of teachers, classes, and subjects
- Built a database by using MySQL to represent the relationship between teachers, classes, and subjects
- Inserted randomly generated data in the database with Python
- Designed UI for the automated examination system for better user interaction

Information Management System – Course Group Project (`GitHub Link`)   23/06/2020 – 05/07/20
- Role: Leader, System Designer, Coder
- Program Language: Python3, SQLite
- Designed a system for managing staff information, which included staff ID, name, department ID etc.
- Inserted randomly generated data in the database with Python
- Designed GUI pages in Python, which mainly contain welcome interface, help interface, partial information interface, employee information interface, department information editing interface, employee information editing interface
- Applied sqLite3 to store staff information data
- Uploaded images were stored in BLOB format

A CAVE-VR System for Tertiary Architecture, Engineering, and Construction (AEC) Education – Research Group Project (Has not been published yet)

10/09/2021 – 15/03/22
- Role: System Designer, Coder
- Program Language: C#
- Designed a CAVE-VR based prototype education system for undergraduate students who study AEC used in class.
- The system is developed via Microsoft Visual Studio, Unity 3D, and a CAVE System SDK, being tested, and evaluated on a CAVE system
- Two main scenes in the system: The Exploration Scene - Users follow the guide as they explore a restored interior scene of a real building; the Inspection Scene - Users observe from multiple angles, disassemble the building components, and learn the name and function of the parts of the building component. The player first appears on the Exploration Scene and is asked (tasks assignment) to find the specified building component (doors, stairs, and windows in this prototype). When the building component is found, the player is transported to the Inspection Scene for detailed observation and study of the component. After this, the player will be transported back to the Exploration Scene for further exploration. Use ends when all building components have been found and learned

- 3D models are built and used extensively in the system, along with 3D conversation boxes and status bars
- State Transition Network helps develop the system by indicating the system workflow
- A timer is added to limit the time. Each group of students is allowed to use the system for 13 minutes at most. Each Task has a time limit, and if the time limit is exceeded, the user is transported to the next task. The time limit can still be adjusted to ensure that all students experience the system before the end of the course
- The player moves with Teleporter to reduce 3D vertigo
- The program has been proven to improve the quality of teaching of construction-related subjects in controlled experiments

A Command-Line (CMD) Based Multiplayer Monopoly Game – Course Group Project (GitHub Link**)**

15/10/2021 – 18/11/21

- Role: Leader, System Designer, Coder
- Program Language: Java
- Designed a Multiplayer Monopoly game running at OS X or Linux Terminal (Windows CMD is not compatible with some features)
- The system is developed under Modal-View-Controller (MVC) Architecture
- A quasi-GUI framework is developed to support the User Interface of the Terminal. Window, Frame, Button, Label, and Input Box are all recreated
- Refreshed interface: Although the command line interface is used as the display, the whole interface is refreshed in real-time every 0.1s. ANSI escapes are used to move the Terminal cursor so that the terminal window position remains fixed when the interface is refreshed. That is why the game does not work on Windows
- Key listen without Echo: In Terminal of OS of the Unix family, keyboard input usually results in an echo i.e. The key input would be displayed directly in Terminal, but the echo would mess with the UI and the game did not need this feature, so Echo is blocked
- Animations: Since the rules and mechanics of the game are relatively simple, the game runs as fast as rock-paper-scissors. Animations are added to slow down the pace of the game and make it more playable.
- Multithread: The refreshed interface, listening for input, and game flow control are run by three separate threads
- Robot player: Robot players can automatically judge their situation and make choices. The addition of robot players allows a single player to play the game. The robots' play levels are chosen by the user

A Digital Artwork Sharing and Selling Website – Course Group Project (GitHub Link**)**

20/03/2022 – 18/04/21

- Role: Leader, System Designer, Coder
- Program Language: Python3, SQLite, JavaScript, HTML
- Designed a website for safe digital artwork sharing and selling
- The backend is built with Django Python Server and SQLite, the frontend is built with Ajax jQuery for sending requests and Viewer.js for viewing the preview of artworks
- A user may register an account, sign in with the account, see other's artwork collections on the marketplace and other's personal pages, recharge balance, buy other's artworks, upload his/her own artworks (and set prices), download his/her artwork collections, accept/reject other's buying requests
- All the collections in the database are encrypted with an AES cipher. The AES key is stored in the user's validation file, and the validation file is encrypted with an RSA cipher. The RSA private key is held by the user as his/her sign in password (unchangeable in this prototype)
- The communication between clients/browsers and sever is protected by SSL/TLS1.3
- After signing in, the session token will be returned and stored as cookies. In fact, the real session token (containing the user's private key and expiration time) is encrypted by an AES cipher and stored in variables on the server, and the temporary AES key of the session token is returned to the user. The expiration time is

updated with each request. If a user does not send a request to the server more than 20 minutes after signing in, the server automatically logs out the user account.

✧ During an artwork transaction, once the owner accepts other's purchasing request, this requested artwork will be decrypted by the owner's AES key and be encrypted by the buyer's RSA public key. At the next signing in of the buyer, the artwork will be decrypted by the buyer's private key (i.e., user password in this prototype) automatically. At this point, the transaction is truly completed

TECHNICAL SKILLS

Software: GitHub**,** Conda, Qt Designer, Office365, Adobe Photoshop, Apple Keynote, Apple iMovie
Programming Skills: Python (Advanced), Java (Advanced), C/C++ (Advanced), PHP+HTML+JavaScript (Advanced), C# (Advanced) and Unity
IDE: Microsoft Visual Studio, JetBrains Family, Sublime Text3, Microsoft VS Code
Specialties: Coding, UI Design, Network Communication Building, Database Building, Website Building, Algorithm Design, Graphic Designing, Photography, Video Editing