Chengzhe Li

(413) 801-7111 chengzheli@umass.edu http://cunzhe.life

EDUCATION

UNIVERSITY OF MASSACHUSETTS AMHERST **Amherst, MA**

M.S. Computer Science Candidate (Expected graduation Dec 2021) *GPA:4.0*

- **Relevant Coursework:** Game Programming, Secure Distributed System, Distributed&Operating Systems, Thry & Practice/Software Engin, Machine Learning, Systems and ML, Computer Vision, Comp&Network Security

UNIVERSITY OF MASSACHUSETTS AMHERST **Amherst, MA**

B.S. Computer Science (Sep 2016 - Dec 2019) *GPA:3.6*

- **Relevant Coursework:** Artificial Intelligence, Pract & Appl of Data Management, Introduction to Algorithms, Operating System, Computer Networks, Software Engineering, Introduction to Modern Computer Graphics
- **Awards & Activity:** Bay State Master, Course contributor of Computer Networks, ESL Activity Assistant

SKILLS

- Proficient in C/C++, C# and Python, familiar about Java, and always willing to learn new languages
- Developing experience in Unity and UE, knowledge about game programming and underneath.
- Good Understanding of Data Structure, Operating System, Database Management, Networks, and Algorithms.
- Familiar with Agile developing and Software Engineering design and procedure.
- Knowledge and experience of implementing and using AI algorithms and ML models and concepts.
- Understanding of distributed systems, complicated operating systems, and secure decentralized systems.
- Programming experience with RESTful, RPC, multihreading distributed system, familiar with docker.
- A Good knowledge base of Computer Graphics and Linear Algebra.
- Passion on Games, Music, Technology ,and new areas. I'm always hungry to learn, and good at it!

PROFESSIONAL EXPERIENCE

TENCENT **SHENZHEN, CHINA**

Software Engineer Intern *May 2020 - July 2020*

- Worked as a part of IEG(Interactive Entertaining Group)-CTG(Core Technology Group), our project is a distributed framework for improving the efficiency of Unreal Engine. The framework is developed in .NET and can be used in UBT(Unreal Build Tools), Lightmass, and ShaderCompiling.
- Got my self familiar with the existing framework and code fastly, learned to write code in strict standards.
- My Job: Testing, Debugging, and fixing known problems. Researched, designed, and implemented new distributing methods to improve efficiency problems. Wrote test framework to evaluate the improvement of new methods.
- Worked with CI to make the project pipelined and efficient. Using data virtualization tools to analyze logs
- Learned: How to get into a large project, work with other brilliant people to deliver an amazing product, finding and fixing problems, develop and test with pipeline tools, adding new functions to existing framework, etc..

PROJECTS

CookingPapa

- Game developed using Unity. The idea is the player needs to fight with the monsterized animal or vegetable to get the food material. In this stage, some integrations will be acquired after defeating the boss, some others will be hidden in the scenes. Other materials like salt may also need to be acquired in specific ways. Then, the player needs to use the same way as cooking in real life to cook the material. They will finally win the game and learn about cooking.
- My Job: Choosing the assets we used, deign the game logic, and implementing the gameplay. Setting up the scenes, UI, etc.. Adding videos, BGMs, and sound effects to make the game more enjoyable.
- Added clear instructions, particle systems, and physical systems to make the game harder but understandable.
- The playable game and demo video are available on my Github.

Pygmy

- Pygmy is is simulated online bookstore micro servers. We built frontend server, order server, and catalog server as three separate RESTful micro servers by Flask, and a simple command-line user interface to interact with it. Worked with Sharuya.
- We used Sqlite3 as the lightweight database and storing information on catalog server.
- The frontend server support three operations, search(category), lookup(item), and buy(item). The first two will query the catalog server, the third one goed to order server, then the order server will update the catalog server and return.
- Besides the basic functionalities, we implemented support for replication and synchronization on backend servers, cache on frontend servers, load balancing, and crash fault tolerance and recovery. We also used docker to containerize our microservers so it's easier to deploy.

Bazaar

- P2P networks built in Python, used pyro4 as RPC library, nodes were able to rise "lookup" call to buy items from other nodes on the network. The message would be passed by RPC function. The structure of the network was configurable. The network can handle multiple transactions been processed at the same time. Worked with Sharuya.