The first time I wanted to dig into the world of Computer Science was a year ago when I encountered my favourite anime, which showed me how splendid a game could be with advanced technologies: background stories are automatically generated by AI which has independent thinking abilities. Delving deeper into it, CS became more and more charming, particularly those spectacular AI, VR, etc., offering the possibility to break the limit of the material world and provide players with an immersive game environment.

Last year, I attended a summer school at the University of Oxford, which was my first time to attend a CS course. Surprisingly, programming attracted me. It was like communicating with a friend who would give me expected responses, which was even more comfortable than interacting with a human. We had several tasks, involving designing websites and implementing an encryption and decryption program. While the tutor only taught us fundamental programming skills like selection statements and loops and I had no basis of programming, I had to learn coding from some online materials the tutor recommended, from which I gradually developed my self-learning ability. To accomplish the program, I searched online for functions I might need and came up with several plans. At first, I was considering writing a dictionary. However, it was impossible to cover the characters of all existing languages. As I learned more, I then knew that all characters are uniquely encoded as numbers using Unicode. Hence, I decided to encode strings into numbers, which can be encrypted with mathematical functions. Meanwhile, I used various functions for characters of different positions in a string to increase the complexity of my encryption. Similarly, the decryption processes used corresponding inverse functions. This project expanded my knowledge of encryption, and I did realise that cryptography is ubiquitous and indispensable; it is widely used for encryption of websites, accounts, etc.

Later on, tutors introduced us about the history of cryptography like enigma in World War II by the final presentation, which aroused my curiosity to learn more effective encryption methods. After the summer school, I still immersed in the ocean of encryption and learned another two algorithms. One is the XOR encryption, a relatively simple symmetric encryption, and the other is a widely used algorithm: RSA. The theory of the XOR encryption was easy to understand: each binary digit of the numerical encoding of the plaintext is applied with the key using XOR operation. When the receiver decrypts the cipher, they use the same method with the same key to get the plaintext as the operation is reversible. This is convenient but also vulnerable, for example, if the plaintext has sequences of repeated characters. On the other hand, key distribution is crucial to all symmetric encryption methods, which also increases the vulnerability of the XOR encryption. The RSA algorithm, however, is more robust, because breaking its key requires large number prime factorisation. As this is an NP-hard problem, it is not solvable by any practical algorithm whose time complexity does not grow exponentially. More specifically, when the key is short, a simple algorithm is adequate to break it, whereas with a key of length more than 1024 bits, breaking the key cannot be done within a reasonably short time. Therefore, the RSA encryption’s security is guaranteed unless the P=NP question is solved or a usable quantum computer is invented. I believe this is one of the reasons why it is the most commonly used. Besides, the digital signature is also widely used using the RSA algorithm. Meanwhile, it is an asymmetric encryption, which means that it avoids danger in the process of key distribution as private key should be kept secret completely while only public key is published for encryption or signature verification. This experience confirmed and stimulated my ambition to learn more brilliant CS theories.

As I discovered I was short of knowledge about algorithms when learning encryption algorithms, I read a book named Aha Algorithm. After reading, I realised that Apart from the feasibility of algorithms, their time complexity is also a vital consideration not only for encryption but all kinds of algorithms. Algorithms originated from our life rather than profound concepts I used to consider. Basic events and details that we naturally neglect are arranged systematically and turned into well-regulated approaches to address problems efficiently, which fascinated me further.

In 2019, I participated in the Euclid Contest and STEP 1 examination (grade 1), harvesting quite good results. While I was preparing, I acquired extra mathematical knowledge beyond the A-level syllabus and I found some of them were related to CS such as number theory, which helped me gain better comprehension when learning encryption algorithms where it is widely used. I also attended the BPhO competition (silver award), in which I practiced logical thinking and problem-solving ability for science questions.

Gradually knowing more about complicated theories hidden behind seemingly simple computer operations，I want to expand my knowledge in university and reach my goal of being a game developer.