The first time I wanted to dig deep into the world of Computer Science was a year ago when I encountered my favorite animation, which showed me how splendid a game could be in the future with advanced computer technologies that the background stories are automatically generated by system and AIs have independent thinking abilities. Delving deeper into it, computer science became more and more charming, particularly those spectacular artificial intelligence and virtual reality, etc, which offer the possibility to break the limit of material world and allow players to enjoy immersive game environment. As I learnt more, CS became more charming to me and since then, I have got a strong desire for being one of game and AI industry staffs who establish and support surreal worlds in games.

Last year, I attended a summer school at University of Oxford and it was the first time I attended a CS course. Surprisingly, I was greatly interested in programming. It was like communicating with a friend and the computer could give me many expected responses, which was even more comfortable than interacting with a person. We had several tasks, involving designing websites and implementing a program of encryption and decryption. While the tutor only taught us fundamental programming skills like selection statements and loops and I had not learnt programming before, we had to learn coding from some online materials recommended by the tutor, from which I developed a strong self-learning ability from the course.

In order to accomplish the encryption program, I searched online for functions I probably needed and came up with several plans. At first, I was considering of writing a dictionary however it was impossible to cover all characters of all languages in the world. As I researched more and I finally knew that all characters are uniquely encoded as numbers using Unicode. Hence, I decided to transform strings into numbers, so that it can be encrypted with mathematical functions. Meanwhile, for the characters of different positions in a string, I used distinct functions to increase the complexity of my cipher. For example, the letters on positions whose indexes can be divided exactly would use y=x^2 while those have factor 7 would choose another. Similarly, the decryption processes used corresponding inverse functions. This project expanded my knowledge of encryption quite a few, and I did realize that cryptography is indispensable and ubiquitous, it is widely used for encryption of application or website accounts, etc; the data of a game also require protection to prevent leakage of information.

After our presentations, tutors introduced us about some encryption history like enigma in the World War II, which aroused my curiosity to search for more effective approaches to encrypt nowadays. After the summer school, I still immersed in the ocean of encryption and learned another two algorithms. One is the XOR encryption, a symmetric encryption algorithm, which is relatively simple to implement. The other is the currently safest encryption algorithm, the RSA encryption. Through my researches, I found that the theory of XOR encryption is easy to understand: first a certain key is set, and then each binary digit of the plaintext’s code, usually ASCII, is applied with the given key, using XOR operation. When decrypt messages, receivers who own the same key can apply XOR calculation to the cipher to gain original cipher as XOR operations are reversible. This is convenient but can also be vulnerable, for example, if the plaintext has sequences of repeated characters, rules of the cipher could be discovered by attackers. However, when the plaintext has the same length as its cipher it will be considerably secure. About RSA encryption I learnt that it is robust because the private key is a product of two concealed prime numbers, generally 1024 to 4096 bits long, which could only be cracked with brute force and hence requires lengthy time. More specifically, when the private key is small, a simple algorithm is adequate to break it, however, as the key becomes greater the time consumption increases exponentially because its time complexity is O(2^n) if the key has n bits, which is a NP-hard question and therefore guarantees RSA ciphers’ security unless the development of quantum computer has a breakthrough or P=NP question is solved.

I have read a book named Aha Algorithm, which introduces various algorithms based on readily comprehensible daily cases such as sorting algorithms. After finishing reading, I realised that the algorithms are in 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. For example, the algorithm of calculating the shortest path is fundamental to the navigation function of mapping applications. It encouraged me to explore more theoretical and fundamental aspects behind applications.

In June, I participated in STEP 1 and 2 examinations, whose questions were difficult and complicated. The sub-questions involved instructions which were helpful for dealing with the next. Following them, I was able to get basic ideas and solve most part of a problem, which enhanced my problem-solving and mathematical skills. During preparation, I acquired some extra mathematical knowledge beyond A-level textbooks and I found some of which are closely related to computer science because it is based on a wide range of principles of mathematics. I learnt the Fibonacci sequence and factorial which use the technique of recursion, which makes the structure clear and easier to understand and is frequently applied in many algorithms.

Also I took part in several competitions and harvested some awards, including Euclid Contest(top 25%), BPhO physics competition(silver award), from which I got practice of logic thinking ability because I needed to analyze conditions provided, then deduce the appropriate equations for ultimate solutions.