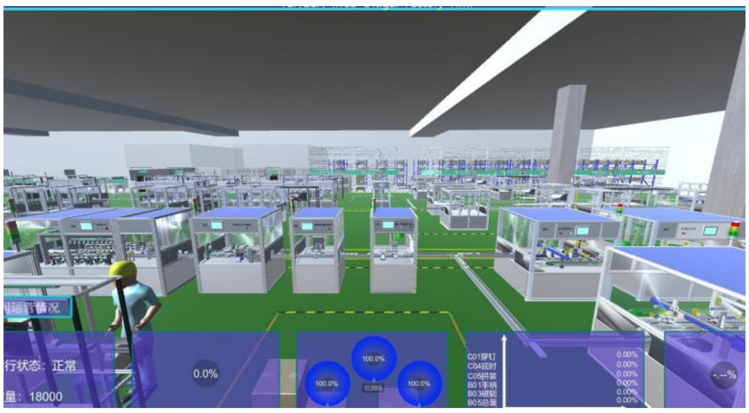
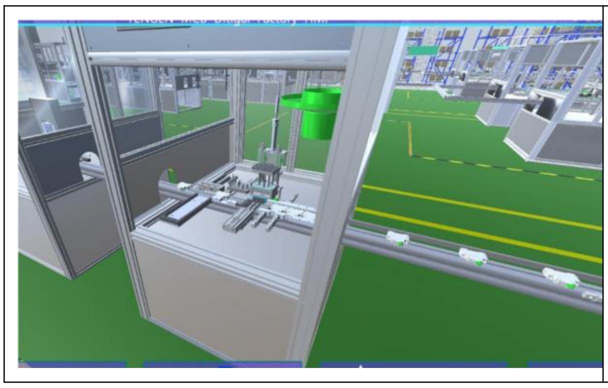
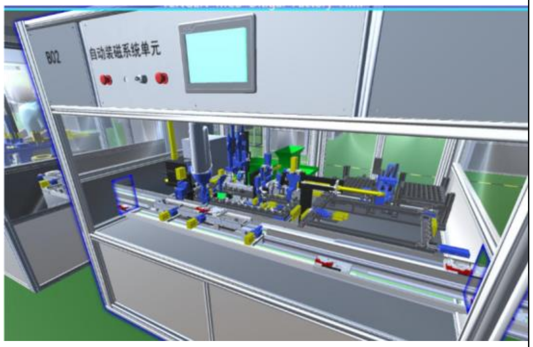
My boss asked me to render a virtual workshop for circuit breakers from one host to four hosts. The circuit breaker workshop is developed based on Unity. I will introduce in detail some specific situations of this project and the defects of the methods I have adopted?

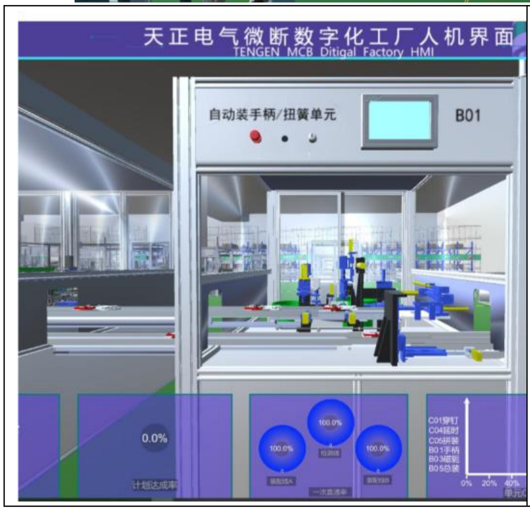
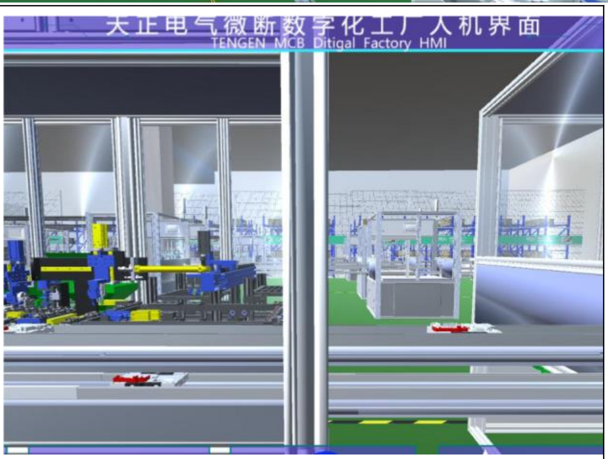






As you can see, there are many moving objects and stationary objects in this scene,The moving objects include cylinders and circuit breakers, and this circuit breaker is the cuboid piece by piece. The motion of the cylinder is animated in 3dmax,When the circuit breaker moves to a specific position, the trigger is placed in this position in advance, and the cylinder will call the animation made in advance to drive the breaker object to move, so as to highly restore the working situation of the real workshop. Generally speaking, this project is real-time rendering (my understanding is this, and I am not very sure),There are many moving objects, so the computer needs to constantly calculate the real-time position of the three-dimensional moving objects, and then render it on the screen.

There are two cameras in the scene, one is panorama camera and the other is automatic roaming camera, as shown below,The upper one is panorama camera and the next two are roaming cameras. When there is no keyboard and mouse operation in panorama camera for a period of time, it will directly switch to the roaming camera, which will automatically roam by the fixed roaming line.


天正电气微断数字化工厂人机界面
TENGEN MCB Ditigal Factory HMI
自动装手柄/扭簧单元
B01
0.0%
100.0%
100.0%
100.0%
计划达成率

天正电气微断数字化工厂人机界面
TENGEN MCB Ditigal Factory HMI

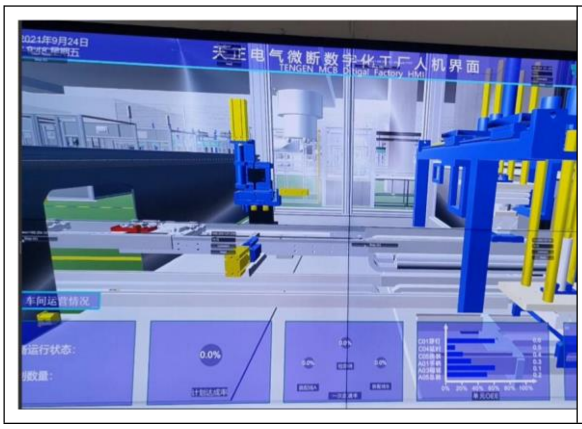
Now my boss wants me to calculate this project from one host computer to four hosts. The following is our hardware situation. Four high-performance image workstations and four projection screens.

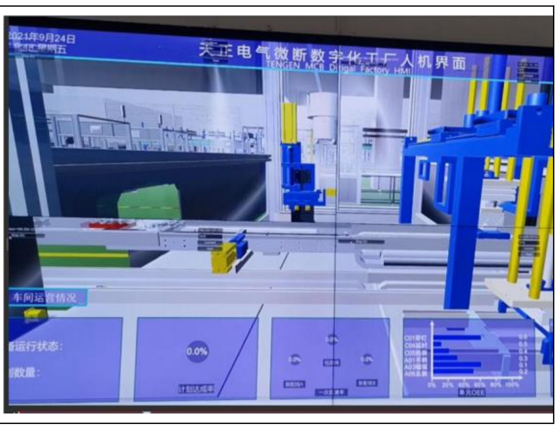
The following is the synchronous rendering of my current scheme,Four screens project four areas in a world. I used to import the same assets into four hosts, one of which is the master terminal and the other three are the slave terminals,The slave terminals cut off the motion power of the circuit breaker and the switching code of the camera, but kept the trigger and animation of the cylinder, and the master terminal was responsible for all the logical calculations. Through the Transform component of Unity, the master sends the position and rotation information of each moving circuit breaker to the client in real time, and then through rpc mechanism, the master tells the slave when to switch cameras. To put it simply, there are four identical three-dimensional worlds in the workshop in four workstations,The master terminal drives the circuit breaker to move and then sends the information such as the position of the circuit breaker to the other three slave terminals through the local area network,When the circuit breaker at the slave terminal reaches the position of the trigger, it triggers the movement of the cylinder, so as to achieve the goal of synchronizing the four worlds. And then rendered to four screens by projection segmentation of the camera.

Unfortunately, my mechanism has encountered several problems.

1. Using RPC to switch cameras from end to end has encountered considerable delay, which leads to the dislocation of four screens.
2. The jitter visible to the naked eye will occur at the slave end of the circuit breaker, which should be caused by the position signal transmitted by the network.

The following picture shows the image of wandering screen dislocation, which is due to the problem that RPC is not fast enough to transmit camera switching information.


2021年9月24日
天正电气微断数字化工厂人机界面
TENGEN MCB Digital Factory HMI
车间运营情况
运行状态:
数量:
0.0%


2021年9月24日
天正电气微断数字化工厂人机界面
TENGEN MCB Digital Factory HMI
车间运营情况
运行状态:
数量:
0.0%

My boss asked me to further optimize the above two problems, but I was pessimistic about the delay caused by this network, and thought it was an inherent defect of software and hardware. Recently, I read about rendering pipeline, rendering farm and so on, and considered whether I could solve this distributed problem for rendering pipeline.