

Network for racing game

1. Introduction

In order to make the racing game support for the online play and larger number of players (16 players), some network techniques like ghost object system, data compression, prediction and interpolation algorithms are implemented in the network system used by the game. In this report, the ghost object technique will be discussed in detail, and prediction and interpolation algorithm will also be mentioned.

2. Ghost object system

The structure of the ghost object system is like following figure. Each player has their own object (master object) and the ghost object of the other players, where the ghost object has the same properties as the master object. With this, when the master object is created, updated or deleted, the corresponding ghost object will also be created, updated, or deleted.

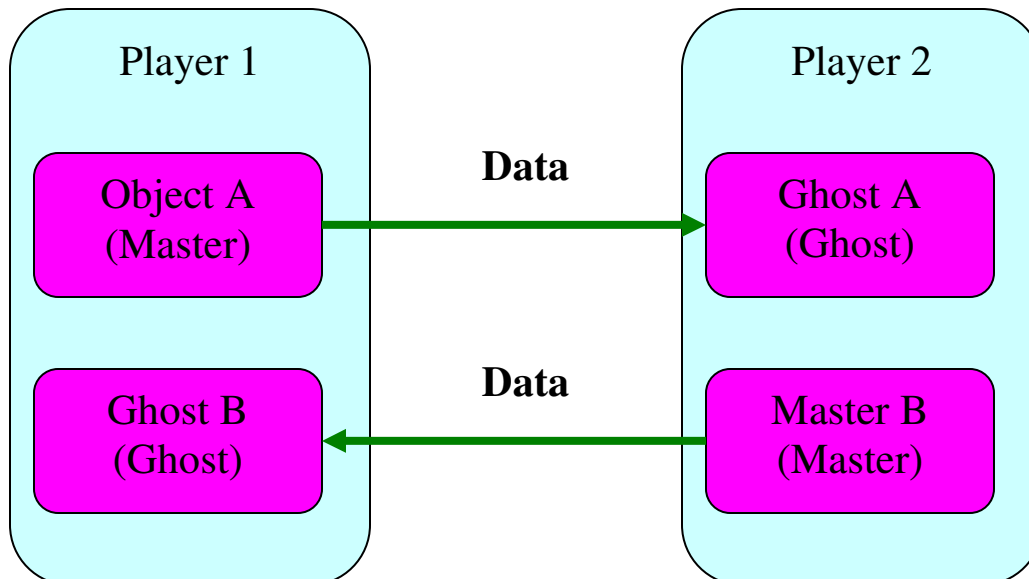


Figure 1: The structure of the ghost object system. When player 1's object (Master A) is updated, the corresponding ghost object A in Player 2 will be updated.

2.1 Network object creating part

When a player's network object needs to be created, a master object is created on that player's side, and the corresponding ghost object will be created on all other players. For example, in figure 1, when network object A needs to be created on player 1, the player 1's object A is created as a master object in its own side, and corresponding ghost object A is created on player 2's side respectively.

2.2 Network object's data transmit part

The ghost object system supports for transmitting data easily. When one player's network object updates its properties, the object's ghost on all other players will be also updated.

For instance, if master object A in player 1 updates its properties, the ghost object A will also updates its properties correspondingly.

2.3 Network object deleting part

If one player's network object needs to be deleted, the master object on that player will be deleted first, and then the corresponding ghost object will also be deleted respectively. For example, player1's object A is deleted, the master object A will first be deleted, and the player 2's ghost object A will be deleted after while.

2.4 Techniques used to save bandwidth for the ghost object system

Network object's properties can be divided into parts, for example, vehicle's properties can be divided into position, velocity, vehicle id, vehicle texture, etc. In the system, all the properties do not need to be sent every time, only the useful information is sent to save bandwidth.

The system implemented three ways to send data, which are guaranteed ordered, guaranteed unordered, and unguaranteed. The first two require the data to be sent without loss. With these three ways, the important data can be sent through the first two ways, like the vehicle id. The often sending data or unimportant data can be sent through the unguaranteed way, such as vehicle position, velocity, etc.

The data transmitted are compressed in bit. And some of the data are sent by using 8-bit or 16-bit floating-point instead of 32-bit. The position and orientation data are sent less frequently instead of sending every frame as the client side can use prediction or interpolation algorithm to estimate the position of the vehicle by using the history data gotten from the network.

2.5 Remote Procedural Call (RPC) framework

To simplify the network implementation process, remote procedural call method is used, the functions in the server side and client side have the same interface, when one side call the function, the function will be run in the other side. For example, when Player 1 calls RPC function to update Object A's properties, that function will be run on Player 2, and the ghost object A's properties will be updated.

3. Client Side Interpolation and prediction

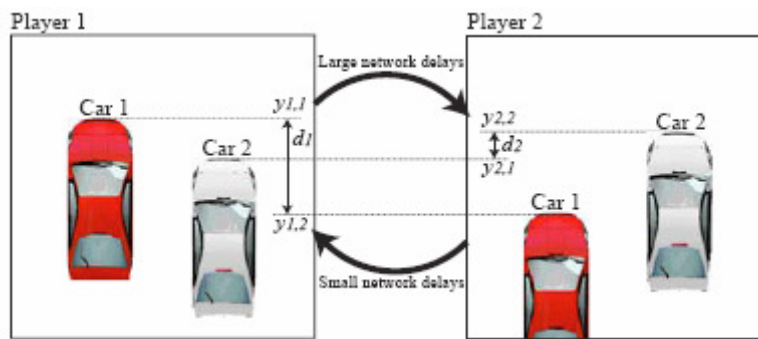


Figure 2: The synchronization problem caused by network delay

The client side interpolation and prediction is used to estimate the ghost object's movement. It uses the history data (position and velocity) to predict the current position of the ghost object in order to solve the synchronization problem caused by the network delay, which is shown in the figure 2.

3.1 Client side interpolation

Interpolation is used to smooth the movement of the ghost object and synchronize the ghost object with the master object. It gets the difference between the ghost object's position and the transmitted master object's position, and use the difference to update each history frame's position to let the ghost object moves much nearer to the actual position frame by frame.

3.2 Client side prediction

In this algorithm, the position and time value transmitted from the master object will be stored as history data. Then the history data can be used to calculate the current position by using the Newton's law. By using the current position, the ghost object's future position can also be predicted. Then the difference between the ghost object's position and the transmitted master object's position can be gotten. The ghost object can be moved to the master object's position smoothly by using some corrections algorithms.

4. Network systems

The client-server architecture and peer-to-peer architecture are researched and implemented in the game.

4.1 Client-server architecture

There is a centralized server to relay the updates of clients to all other clients, and the server is responsible to create the session for player to play and manage the joining and quitting of players.

4.2 Peer-to-peer architecture

In peer-to-peer mode, the server is one of the players, we call it as master player; and whose responsibility is to create game to let other players to join. Each player sends update messages to all other players. The master player is responsible for joining and quitting of the players.

5. Conclusion

With the ghost object system, the design and implementation of the networked game is much simpler, and the system also saves network bandwidth by using techniques like sending only useful information, compressing data, etc. The prediction and interpolation algorithms are also used to smooth the ghost object's movement and synchronize the ghost with the master more closely.