A METHOD FOR IMPLEMENTING MULTI-PLAYER GAMES

FIELD OF THE INVENTION

[0001] The invention pertains to multi-player games, and in particular, first person shooter (FPS) online multiplayer games.

BACKGROUND

[0002] When playing an online multi-player first person shooter (FPS) game, all actions (moving, activating weapon, driving a vehicle) has a consequence (e.g. avatar is moved to position X, hit or missed target T, vehicle is driven to position X).

[0003] The traditional way of implementing online multi-player FPS games is that all action-consequence pairs need to be verified by a server, before the consequence of the action can be rendered in the game. This is to prevent cheating or to mitigate any dishonest behavior. For example, if a player provides the command to perform the action “move to the left”, (1) a data package will have to be sent to the server. (2) From the content of the data package, the server then determines that the consequence of the action is “avatar is moved to position X”. (3) The server then sends the consequence of the action to the client device. (4) Upon receipt, the client device then renders the consequence in the game.

[0004] In other words, in the traditional method, the server has to determine the consequence of the action, and provide the consequence to the client device, before the consequence can be rendered on the client device. This scheme suffers from certain disadvantages. Firstly, the time taken to complete steps (1) to (4) can result in high latency in the game. This is because even though the action has already been performed by the player, the result or consequence of said action can only be rendered in the game on the client device after steps (1) to (4) have been completed. Therefore, steps (1) to (4) and any delay of said steps can result in intermittent pauses and jerky gameplay, which can have a negative effect on user experience.

[0005] Furthermore, in the traditional method, in order to compute the consequence of every action, a vast number of data packages will have to be sent from the client device to the server, constantly, and at very high frequencies. This leads to extremely high data transfers which will consume a lot of data and bandwidth. The problem is compounded especially in mobile

CLAIMS

1. A method for implementing a multi-player game comprising the steps of:

launching by a first client application on a first client device, a first instance of the multi-player game;

receiving at the first client application, a command to perform an action;

calculating at the first client application, a consequence of the action;

rendering by the first client application, the consequence in the first instance of the multi-player game;

determining by the first client application, that the consequence has a game-altering impact;

sending a data package from the first client device to a server, the data package comprising the action, the consequence, and the game-altering impact;

receiving by the server, the data package;

determining by the server, that the consequence is valid;

determining by the server, that the game-altering impact is valid;

disseminating by the server, the data package to other client devices, so that client applications in the other client devices can render the consequence and the game-altering impact in other instances of the multi-player game;

sending by the server, a valid notification to the first client device;

receiving by the first client device, the valid notification; and

rendering by the first client application, the game-altering impact in the first instance of the multi-player game;

wherein there is a time lag between the rendering of the consequence and the rendering of the game-altering impact by the first client application in the first instance of the multi-player game, and wherein the effect of the time lag is reflected in the first instance of the multi-player game.

2. The method of claim 1 further comprising the steps of:

determining by the server, that the consequence is invalid;

sending by the server, a reinstate notification to the first client device;

receiving by the first client device, the reinstate notification; and

reinstating by the first client application, a state of the first instance of the multi-player game to one in which the consequence had never been rendered.

3. The method of claim 1 further comprising the steps of:

determining by the server, that the game-altering impact is invalid;

sending by the server, an invalid notification to the first client device; and

receiving by the first client device, the invalid notification, which instructs the first client application not to render the game-altering impact in the first instance of the multi-player game.

4. The method of claim 1 wherein the step of determining by the server, that the consequence is valid comprises the step of:

determining by the server, that the consequence satisfies a set of predefined conditions.

5. The method of claim 2 further comprising the steps of:

calculating by the server, a correct consequence;

sending by the server, the correct consequence to the first client device, so that the first client application can render the correct consequence in the first instance of the multi-player game; and

disseminating by the server, the correct consequence to the other client devices, so that the client applications in the other client devices can render the correct consequence in the other instances of the multi-player game.

6. The method of claim 1 wherein the step of determining by the server, that the game-altering impact is valid, comprises the steps of:-

calculating by the server, a correct game-altering impact; and

determining by the server, that a variance between the game-altering impact and the correct game-altering impact falls within an acceptable threshold.

7. The method of claim 1 wherein the step of determining by the server, that the game-altering impact is valid, comprises the step of:-

determining by the server, that the game-altering impact satisfies another set of predefined conditions.

8. The method of claim 3 further comprising the steps of:

calculating by the server, a correct game-altering impact;

sending by the server, the correct game-altering impact to the first client device, so that the first client application can render the correct game-altering impact in the first instance of the multi-player game; and

disseminating by the server, the correct game-altering impact to the other client devices, so that the client applications in the other client devices can render the correct game-altering impact in the other instances of the multi-player game.

9. The method of claim 1 further comprising the steps of:

receiving at the first client application, a command to perform a series of actions, the series of actions comprising a first action;

sending by the first client device, an initialization data package to the server, the initialization data package indicating a start of the series of actions;

calculating at the first client application, a first consequence of the first action;

rendering by the first client application, the first consequence in the first instance of the multi-player game;

determining by the first client application that the first consequence does not have a game-altering impact, and bypassing sending a first data package from the first client device to the server; and

sending by the first client device, a finalization data package to the server, the finalization data package indicating an end to the series of actions.

10. The method of claim 9 wherein the series of actions is a sustained activation of a weapon, and the first action is a first activation of the weapon, and the first consequence is the shot missed a target, and the first data package contains data pertaining to the first activation of the weapon.

11. The method of claim 1 wherein if the action has an associated series of animation, the method comprises the steps of:

rendering by the first client application, the series of animation in the first instance of the multi-player game;

sending another data package from the first client device to the server, the another data package comprising the action;

receiving by the server, the another data package;

disseminating by the server, the another data package to the other client devices, so that the client applications in the other client devices can render the series of animation in the other instances of the multi-player game;

calculating by the server, a correct consequence of the action;

calculating by the server, a correct game-altering impact of the consequence;

sending by the server, the correct consequence and the correct game-altering impact to the first client device, so that the first client application can render the correct consequence and the correct game-altering impact in the first instance of the multi-player game; and

disseminating by the server, the correct consequence and the correct game-altering impact to the other client devices, so that the client applications in the other client devices can render the correct consequence and the correct game-altering impact in the other instances of the multi-player game.

12. The method of claim 11 wherein the series of animation comprises an animation for a grenade being thrown, an animation for the grenade landing on a location, and an animation for the grenade exploding to cover a blast radius.

13. The method of any one of the preceding claims wherein the multi-player game is an online first person shooter (FPS) game.

14. The method of any one of the preceding claims wherein the first client device and the other client devices are mobile devices.