DPhysics Server Documentation

John Pan
Version 1.0
5/5/2015

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Package DPhysics.Lockstep

Classes

- class **Core** The core of the Lockstep Plugins that manages rooms and matchmaking.
- class **IDGen** *Used for generating room IDs, and potentially IDs for other objects*.
- class **Player** *Represents a networked player.*
- class **Room** *Represents a room where every player receives the same input. In other words, every player in a room plays the same game.*

Class Documentation

DPhysics.Lockstep.Core Class Reference

Member Data Documentation

Dictionary<ConnectionService, Player> DPhysics.Lockstep.Core.Connections;

Used for accessing a ConnectionService's Player instance.

Timer DPhysics.Lockstep.Core.GlobalTimer `[static]`

This Timer is responsible for advancing the server's simulation.

List<Player> DPhysics.Lockstep.Core.Players = new List<Player>()

All Players connected to the server.

const double DPhysics.Lockstep.Core.SimulationRate = 100

The rate at which the server runs its logic, in milliseconds. The server's rate should be synced with clients' simulation rate.

DPhysics.Lockstep.Player Class Reference

Detailed Description

Represents a networked player.

DPhysics.Lockstep.Room Class Reference

Public Member Functions

- **Room** (ushort ID)
  *Constructs a **Room** and sets its identification number as ID.*
- void **Simulate** ()
  *Simulates the room, distributing information to players as necessary and performing any **Room** logic.*
- void **ProcessInformation** (byte[] data)
  A player's game data gets routed to his room here.
- void **DistributeFrame** ()
  *Distributes a frame to every player in the **Room**.*
- bool **AddPlayer** (**Player** player)
  Returns true if the room is full and can start, false if it still needs more players.
- bool **RemovePlayer** (**Player** player)
  Returns true if room has no players and can end, false if it still has players

Public Attributes

- List< **Player** > **MyPlayers** = new List<**Player**>()
  The players in this room.
- List< byte > **CurrentFrame** = new List<byte>(20)
  The package to send the next frame.
- bool **Started** = false
  *Describes whether or not this **Room** has started yet.*
- int **JoinedCount** = 0
  Describes the amount of joined players.
- ushort **RoomID**
  *This **Room**'s unique ID. Used for accessing the **Room** in Core.AllRooms.*

Static Public Attributes

- static int **MaxPlayers** = 1
  Defines the maximum amount of players that can be in a room. Once this amount is met, the room starts and no more players can join.

Detailed Description

Represents a room where every player receives the same input. In other words, every player in a room plays the same game.

Constructor & Destructor Documentation

DPhysics.Lockstep.Room.Room (ushort *ID*)

Constructs a **Room** and sets its identification number as ID.

Parameters:

ID	The unique ID of the room, generated by GenID

Member Function Documentation

bool DPhysics.Lockstep.Room.AddPlayer (Player *player*)

Returns true if the room is full and can start, false if it still needs more players.

Parameters:

con	The ConnectionService of the Player.

Returns:

Whether or not the Room is full and can start.

void DPhysics.Lockstep.Room.DistributeFrame ()

Distributes a frame to every player in the **Room**.

void DPhysics.Lockstep.Room.ProcessInformation (byte[] *data*)

A player's game data gets routed to his room here.

Parameters:

data	The data received from the Player.

bool DPhysics.Lockstep.Room.RemovePlayer (Player *player*)

Remove a player from the Room and returns true if room has no players and can end, false if it still has players

Returns:

Whether or not the Room can shut down.

void DPhysics.Lockstep.Room.Simulate ()

Simulates the room, distributing information to players as necessary and performing any **Room** logic.

Member Data Documentation

List<byte> DPhysics.Lockstep.Room.CurrentFrame = new List<byte>(20)

The package to send the next frame.

int DPhysics.Lockstep.Room.JoinedCount = 0

Describes the amount of joined players.

int DPhysics.Lockstep.Room.MaxPlayers = 1 `[static]`

Defines the maximum amount of players that can be in a room. Once this amount is met, the room starts and no more players can join.

List<Player> DPhysics.Lockstep.Room.MyPlayers = new List<Player>()

The players in this room.

ushort DPhysics.Lockstep.Room.RoomID

This **Room**'s unique ID. Used for accessing the **Room** in Core.AllRooms.

bool DPhysics.Lockstep.Room.Started = false

Describes whether or not this **Room** has started yet.