photon

Search

This document is about: **QUANTUM 2**

SWITCH TO

Input & Connection Flags

Introduction

The `DeterministicInputFlags` are used by Quantum to:

- detect whether a player is ***present***, i.e. connected, to the simulation;
- decide how to ***predict*** the next tick's input for a given player; and,
- know whether the input on a verified frame was provided by a client or was ***replaced*** by the server.

It is possible to automate the checks by implementing `PlayerConnectedSystem`, for more information please refer to its entry on the Player page.

Types

C#

```
public enum DeterministicInputFlags : byte {
  Repeatable = 1 << 0,
  PlayerNotPresent = 1 << 1,
  ReplacedByServer = 1 << 2
}
```

- `PlayerNotPresent` = means there is no client connected for this player index.
- `ReplacedByServer` = means the player index is controlled by a client, but the client did not send the input in time which resulted in the server repeating or replacing/zeroing out the input.

This can be set by the developer from Unity when injecting player input and should be used on direct-control-like input such as movement; it is not meant for command-like input (e.g. buy item).

Implementation example

IMPORTANT: `DeterministicInputFlags` can only be trusted on *verified* frames.

The code snippet below is an extra from the LittleGuys sample found on the BotSDK page.

C#

```csharp
private void UpdateIsBot(Frame f, EntityRef littleGuyEntity)
{
  // Return if players shouldn't be replaced by bots
  if (!f.RuntimeConfig.ReplaceOnDisconnect)
    return;

  // Only update this information if this frame is Verified.
  if (!f.IsVerified) return;

  var littleGuyComponent = f.Unsafe.GetPointer<LittleGuyComponent

  // Get the input flags for that player
  var inputFlags = f.GetPlayerInputFlags(littleGuyComponent->Play

  // Bitwise operations to see if the PlayerNotPresent flag is ac
  var playerDisconnected = (inputFlags & DeterministicInputFlags.

  // Store it in the IsBot field so this can be evaluated in othe
  littleGuyComponent->IsBot = playerDisconnected;

  // Only initialize the entity as a bot if it doesn't have the H
  if (playerDisconnected && f.TryGet<HFSMAgent>(littleGuyEntity,
  {
    // We're replacing players only by the HFSM, but this could e

    HFSMHelper.SetupHFSM(f, littleGuyEntity, f.RuntimeConfig.Repl
```

Back to top

We Make Multiplayer Simple

Products

Fusion
Quantum
Realtime
Chat
Voice
PUN

Memberships

Gaming Circle
Industries Circle

Documentation

Fusion
Quantum
Realtime
Chat
Voice
PUN
Bolt
Server
VR | AR | MR

Support

Gaming Circle
Industries Circle
Circle Discord
Circle Stack Overflow

Resources

Dashboard
Samples
SDK Downloads
Cloud Status

Connect

Public Discord
YouTube
Facebook
Twitter

Languages

English
日本語
한국어
简体中文