photon

Search

This document is about: **QUANTUM 2**

SWITCH TO

Scene Loading

Introduction

Loading a scene in the context of Quantum involves two parts:

1. the scene view (Unity); and,
2. the scene data for the simulation (Quantum).

This document will present 3 valid approaches on how to implement scene loading. In all of them, you will want to load the Unity Scene and MapData yourself (via the `OnMapChanged` callback for the later).

N.B.: The `Simulation Config` asset offers an *Auto Load Scene from Map* option. This is fine for *prototyping*, for *production* we strongly advise you to write your own game specific loader.

Load while Offline

The Unity Scene can be loaded offline/locally before a player enters matchmaking or starts a game. This is a good option for casual games with direct matchmaking.

Sequence:

1. Load the gameplay scene in the background even while offline (Unity)
2. Enter matchmaking and find a room (Photon Realtime)
3. Start the simulation (Quantum) with the gameplay scene already pre-loaded

photon

Load While in Photon Room

In case you need a lobby to inform players about pre-match information, you can coordinate the loading via the Photon Realtime APIs and launch the Quantum simulation as soon as everybody has loaded the gameplay scene.

Sequence:

1. Player joins a Photon ROOM
2. Coordinate the scene loading for all clients who joined the room via Realtime:
   1. Signal the expected load time with (custom Photon message or room property)
   2. Wait for all clients to confirm they have loaded the scene (player property), **OR** the master client signals a time out
   3. Signal the Quantum simulation start (room property)

When game starts, the simulation and gameplay kick-off immediately without any additional loading being required; this makes it fair for all players involved.

Load After Quantum Started

This options is similar to the previous one with one crucial difference: the lobby rules are controlled and enforced by the Quantum simulation. Going this route allows you to benefit from the determinism right away, this is particularly useful when pre-match rules need to be enforced - for instance character selection rules in MOBAs.

Sequence:

1. Join a Room (Realtime)
2. Start simulation (Quantum)
3. Load gameplay scene (Unity)
4. The gameplay scene has a lobby controlled by the simulation logic
5. Players can use `SendPlayerData` with their selections
6. A time out is enforced by the lobby system (Quantum) thus maintaining determinism in case a client fails
7. Disable the lobby systems in Quantum and enable the gameplay related ones

Back to top

photon

photon

We Make Multiplayer Simple

Products

Fusion
Quantum
Realtime
Chat
Voice
PUN

Memberships

Gaming Circle
Industries Circle

Support

Gaming Circle
Industries Circle
Circle Discord
Circle Stack Overflow

Connect

Public Discord
YouTube
Facebook
Twitter
Blog
Contact Us

Documentation

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

Resources

Dashboard
Samples
SDK Downloads
Cloud Status

Languages

English
日本語
한국어
简体中文
繁体中文

photon