Unity Multiplayer

Presented by Michael Gassman
Unity Multiplayer

This is a network that is made of:
• 1 server
• Multiple clients (local client, remote client)

The server can be:
• Host - This is a server and a client in a single process
• A dedicated server with no active player(s). (Soon)

http://docs.unity3d.com/Manual/UNetConcepts.html
Network Topologies

Dedicated Server

Peer to Peer (Mesh)

Hybrid – Host Based

Server

Client

Client

Client

Client

Client

Client

Client

Host Client

Client

Client

Client
Server, Client, and Host
# Host Migration

## Network Migration Manager (Script)

- **Use Host Migration**: Checked
- **Show GUI**: Checked
- **Offset X**: 10
- **Offset Y**: 300
- **Disconnected From Host**: True
- **Waiting to become New Host**: False
- **Waiting to Reconnect to New Host**: False
- **Your Connection ID**: 1
- **New Host Address**

### Peers

- **Peer**:
  - Peer Info: conn:0 addr:<host>:7777 host:True isYou:False
  - Peer Info: conn:1 addr:<host>:127.0.0.1:60250 host:False isYou:False

### Pending Players

- **Player netId:2**
  - Connection: 0
  - **Ship2D(Clone)**

- **Player netId:18**
  - Connection: 1
  - **Ship2D(Clone)**
Instantiate vs Spawn

Single-instance games:
• GameObjects are Instantiated to create an instance of them in the game.

Networked games:
• Objects must also be “spawned” on the network
Instantiate vs Spawn

Spawning:
• Makes GameObjects active on the network
• Can only be done on the server
• Causes the objects to be created on connected clients.

The Spawning System supports:
• Distributed object life-cycle management
• State synchronization.
Player Objects

- Player objects are special.
- There is a player object associated with each person playing the game.
- Commands are routed to that object. (client-to-server remote procedure calls)
- Each player object is associated with a connection.
- Switching Players for connection
Local Players

• A person cannot invoke a command on another person’s player object - only their own.
• This is the concept of “my” player object.
• This player object becomes a “local player” object on the client of that player.
Local Players

- Only the player object that is “yours” will have the `isLocalPlayer` flag set.
- `isLocalPlayer` that is set to true, and a callback `OnStartLocalPlayer()` that is invoked on the object on the client.
Authority

- GameObjects can have “local authority”.
- This means that the object on its owner’s client is responsible for the object.
- This is used most commonly for controlling movement.

Example:
- The NetworkTransform component will send movement from the client if it has authority.
Authority

- Authority can reside on the server if the GameObject is not spawned with client authority
Authority

Network Identity
How the code works

Game State Management
Spawning Management
Scene Management
Debugging Information
Matchmaking
Customization
How the code works

• High level APIs. Integrated into the Unity component based workflow; implement and iterate multiplayer features today.

• Low Level APIs. Full access to Multiplayer network code; completely optimize your game.
**How the code works**

- **Match Maker** lets players advertise their game for others to join.
- **Relay Server** ensures network traffic reaches all players in a game.
- **Host** is a client that is also the server for a particular game session (yes, we do host migration).
- **Infrastructure** is load balanced, globally balanced, for quality of service.
Multiplayer Demo
Multiplayer is for

1. **Real Time, Session Based Games** e.g. Action RPG, Racing.
2. **Rapid Prototyping.** Integrated in Unity so it’s fast and easy implement networked features.
3. **Global Audiences.** Infrastructure is distributed across continents for consistent quality.
4. **Flexibility.** Reutilize UMP subcomponents with ANY network solution.
Multiplayer is not (yet) for

1. **Asynchronous Games** that require a dedicated database, socket server.
2. **Platform-Specific Features** like Steamworks. No native support today.
3. **Cheat-Sensitive Games** that require a controlled dedicated server
4. **High Bandwidth Games** that require > 4500 bps e.g. multiplayer physics. (only through Relay Server)
Thank you!