1. Walk and talk application – chat bubbles above player’s heads

Folder Contents

MultiDemo -> Assets:	Contains game assets.
MultiDemo -> MainScene:	Game executable
MultiDemo -> Assets -> Images:	Contains textures used in the game.
MultiDemo -> Assets -> Materials:	Contains materials created using textures.
MultiDemo -> Assets -> Prefab:	Contains game prefabs for wooden crate and the FPS player.
MultiDemo -> Assets -> Scripts:	Contains unity game scripts.

Scripts

ChatMessage.cs:
This script contains server-client message handling functionalities and user-defined message which will be exchanged between players in a multiplayer mode.

ControlPlayer.cs:
This script contains functionalities to update the 3D text displayed above the player when typed using chat text field. The text is synchronized across the network by using command function.

FPSInputController.js:
This script is taken from the Unity Standard assets, which is available from the unity store and provides FPS character control motor functionalities.

MouseLook.cs:
This script is taken from the Unity Standard assets, which is available from the unity store and attached to the main camera and the FPS player to enable mouse look functionality.

PlayerMovie.cs and *UpdateUIText.cs*
These scripts are miscellaneous scripts to be used for simple character control and chat text updates. They are NOT used in this particular demo.

How to run the application

Running the application on the same PC as localhost

- Double-click open - MultiDemo -> MainScene.exe
- Choose a smaller resolution (can be 800x600) and preferably check the ‘windowed’ checkbox in the application pop-up window.
- Repeat the above two steps to open a player2 window.
- One game window can be used to demonstrate as a server(Player1) and the other window as a client(Player2).
- Click on ‘LAN Host’ in the server window. This will run the application as a server.
- Click on ‘LAN Client’ in the client window. This will run the application as client.
- Use the chat text field at left-bottom to give the server player a name or type in a chat text to send it across the network.
- Use step 7 above for client window.
- Click on ‘Send ’ button once you type in a message.
- Use arrow keys to move the players around the scene (Keys – W, A, S, D can also be used to move the player around).

- Use mouse to look around and walk during game play.

Running the application on different PCs

- Copy the game folder – MultiDemo to different PCs.
- Click-open the application MultiDemo -> MainScene.exe on both PCs.
- This time, the server can be run using 'LAN Host' on server PC.
- The client should be connected using the IP address of the server machine (In LAN).
- Type in the server IP in the text field next to 'LAN Client' instead of 'localhost'.
- Click on 'LAN Client' now in the client PC to connect with the server running on another PC.
- Use the chat text field at left-bottom to give the server player a name or type in a chat text to send it across the network.
- Use step 7 above for client window.
- Click on 'Send ' button once you type in a message.
- Use arrow keys to move the players around the scene (Keys – W, A, S, D can also be used to move the player around).
- Use mouse to look around and walk during game play.

To stop the applications, use the 'stop' button on the server window and then in the client window.
To re-run the applications close the game windows on both PCs or single PCs in case of localhost and click open the application windows again from the folder.

----

2. World-space (server) trigger. Change object color, move an object, destroy an object on all clients based on someone running into it server side.

The application is an updated version of the previously delivered demo.

Updates:

- Environment modifications – skybox, objects to demonstrate color change, object transform and object destruction on network based on player trigger
- Player model inclusion and network transform changes.

Scripts:

ColorCubeControl.cs

This particular script handles functionality related to the color change of the cube based on trigger entry by the player on the server side.

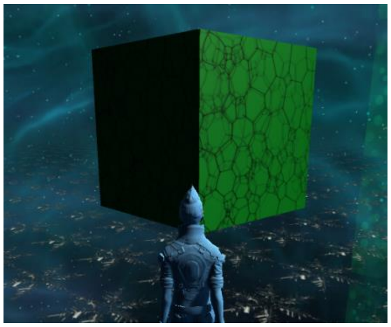
DestroyObject.cs

This script contains the implementation to destroy the cube when the player enters the cube trigger on the server side.

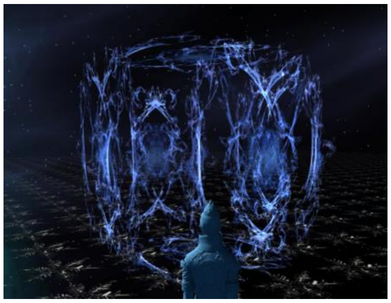
Interactions:

Player when run into the sphere object (marble textured object), the sphere rolls and the transform is updated on the network server and sync on all clients.

Player when run into the rotating cubes, the color of the cube changes randomly.



Player when run into the mystic textures cubes the cube vanishes on server and simultaneously on all clients in network.



3. Keys and doors. Client created objects. At least one object needs to be created when a player is standing in a volume and presses a button.

Scritps:
CreateObject.cs:
Contains functions to instantiate an object when the player is standing in the volume and presses a key.

TriggerObjCreate.cs:
Contains functions to check for enter and exit triggers of the player with respect to the volume.

Interactions:

Player when standing in the volume and presses the key – ‘K’, a sphere object will get instantiated right next to the volume.