

Station Setup

This section will go over the concepts of the technique and how it works, but not a specific example using it. See examples chapter for how to make something specific using the technique.

- [Default Chair](#)
- [Station Mobility](#)
- ["Desynced Station"](#)
- [Station Annoyances](#)

Default Chair

[VRCChair inspector](#)

In VRChat, there is a component called VRC Station. This is the basis for all chairs in worlds. The default setting allows a player to enter and sit in the station. At a technical level, when a player is in a station, the player's position is set to the station's position. This leads us to the first observation: Player positions are normally forced synced, but while in a station, the station does not need to be in the same location for everyone. The sitting player will still follow the station where you see it.

The most basic version of repositioning remote players is a default chair with some additional logic to move it. As an example, you can create a simple Udon program such that when a remote player is in the chair, the station is moved -1000 below the floor. This creates a situation where the player sitting in the station is still in the same location, but for everyone else, the player has vanished. Privacy concerns aside, the player sitting in the station is no longer at the same location for everyone.

[Desynced Chair](#)

UdonGraph

[Desynced Chair UdonGraph](#)

UdonSharp

[Desynced Chair UdonSharp](#)

CyanTrigger

[Desynced Chair CyanTrigger](#)

Station Mobility

The default settings for a station force the player into sitting animation and move that player's position to the station's location. There are additional settings to change this behavior.

[Default VRCStation Inspector](#)

1. Player Mobility

The first is the "Player Mobility" setting. Default value is immobile. There are three total settings:

1. Immobilize
2. Immobilize For Vehicle
3. Mobile

The first two options, Immobilize and Immobilize for Vehicle, are very similar (I don't actually know the differences). Both options will force the player's position to the station's position. Note that the station's position is actually the "Player Enter Location" transform. This can be set to the station, but it can also be any transform in the scene. This transform can be moved to any position and rotation and the player will be moved to it. Outside of stations, players cannot be rotated on the x or z axes, but only the y axis. While in an immobile station, players can be rotated along all axes.

The last option, Mobile, allows the player to "enter" the station, but the player's position is not forced to the station's. This option alone is not enough to allow the player to move around while in the station though. The "seated" setting must be disabled and the "Disable Station Exit" must be enabled.

2. Seated

The seated option determines if the player's animations should change while in the station. By default, Seated is enabled, which will force the player to perform seated animations. When seated is disabled, animations are not changed, which allow the player to continue standing.

3. Disable Station Exit

This setting determines if the player will automatically exit the station when moving around. It defaults off, meaning attempting to move will exit the station.

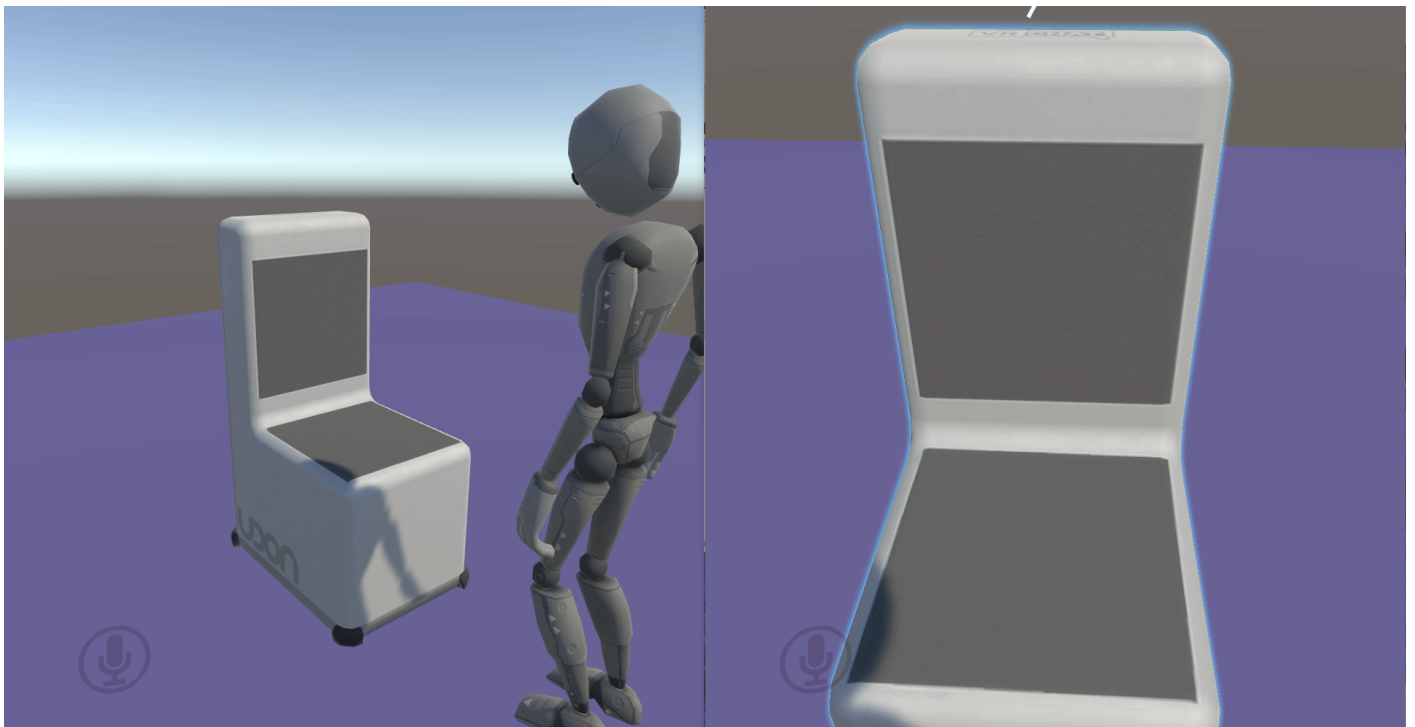
To have a "mobile" station, the station is required to have player mobility set to "Mobile", seated unchecked, and Disable Station Exit checked. This will allow the player to move while staying in the station.

[Mobile VRCStation Inspector](#)

"Desynced Station"

By themselves, mobile stations don't offer much. The power of a mobile station is when you "desync" the values, or have different settings for the local player compared to remote. As mentioned in part 1, you can move a station to different locations if you are the one sitting vs everyone else watching. With mobile stations, this won't do anything since the player isn't moving with the station. The biggest trick here is that you can set the station to mobile for the player sitting, and for all other players, set the station to immobile. In this case, the player in the station will be able to move around locally (allowing Network IK to show their animations to everyone else too), but all other players will see the one in the station stuck at the station's position.

From here, it is up to you and your specific world on how the player should be moved around.



[Desynced Mobile Station UdonGraph](#)
[Desynced Mobile Station UdonSharp](#)
[Desynced Mobile Station CyanTrigger](#)

Station Annoyances

In SDK3, teleporting the player will force exit stations. Depending on your setup and uses, you may want to have a check for if the player exits the station at an unexpected moment. If you initiate the teleport, then you can try to force the player back into the station. Make sure to do this in a delayed way since you cannot force a player back into a station in the same event as `OnStationExited`. Note that Respawn will also force exit the station, but it is not recommended to force the player back into the station as this is a good way to exit any unusual situations.

Force entering stations only work for humanoid avatars. If a player is in a generic avatar (or questionable client user), then you will need to check for this.

Stations only allow one user to be in them at a time. If you want to apply this technique to everyone in the instance, you will need to give every player their own unique station. This can be achieved using object pools. It is recommended to use a system like CyanLaser's

[CyanPlayerObjectPool](#).