Using Avatars to create NPCs in High Fidelity

Overview

You can use High Fidelity's extensive avatar system to record yourself in different avatars to create non-player characters (NPCs) that you can then include in your High Fidelity domains. In this tutorial, we'll walk you through the process of:

- Changing your avatar to another model
- Using recorder.js to capture your actions in-world
- Writing a script to play back your avatar's actions on a separate model

Choosing a new Avatar

In High Fidelity, you have the option of choosing from pre-made avatars in the Marketplace, or creating your own avatars with tools like Blender, Maya, or Fuse. Avatar data is stored in a .fst format, and can be changed as often as you'd like. When recording NPCs for your domain, you may want to use different models to create different characters to record.

Choosing a new avatar from the Marketplace

Open up the Marketplace from the main HUD menu by selecting the 'Market' icon. In the Marketplace, find the avatar model you'd like to record with and choose 'Get' to download and equip the new skin.

Changing to a custom avatar

If you have custom avatars that you've created in other applications, you can generate a .fst file from the .fbx models directly in High Fidelity. To package a .fbx model to use in High Fidelity:

1. In Interface, make sure you've checked 'Advanced' in the 'Settings' menu
2. Go to Edit > Package Model
3. Click 'Browse' for the Model file and select your .fbx
4. If your textures aren't embedded into your .fbx model, select the Texture Directory on the next menu and click 'OK'
5. Save the resulting .fst file someplace where it is easily accessible

Once your custom avatar has an .fst file:

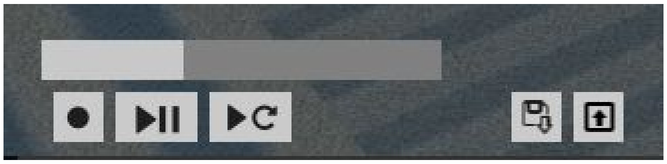
1. Upload your .fst file to a publicly accessible file server, such as a public Dropbox folder or AWS storage instance.
2. Copy the link to your .fst file
3. In Interface, go to Settings > Avatar and paste your .fst URL into the Appearance text box, then save changes

Recording Actions with Recorder.js

The Interface client provides a built-in mechanism for recording avatar action with Recorder.js, a script that can be found in your High Fidelity scripts folder under /developer/utility/record/. To launch the recorder:

1. Go to Edit > Open and Run Script File
2. Expand developer > utility > record and choose recorder.js
3. Hit ‘Load’ to run the recorder script in your Interface client

When you run the recorder script, a small UI will appear on screen with several buttons. This is your control panel for your recorded actions, and will let you create new clips by recording your avatar.



From left to right: start/stop recording, play/pause (playback), restart playback, save recording, load recording. The top bar is a progress bar that shows the recorded animation.

To begin recording an interaction with your avatar, click the recording button. It will turn red when the avatar motions are being recorded.Click the recording button a second time to end the recording. If you plan on looping your NPC behavior, take care to try and get your ending frame as close as possible to your starting frame position.

Playing back a recording on your avatar may result in some unexpected behaviors, since both you and the recording will be controlling your avatar at the same time. If you’re using hand controllers to record gestures, you may not see them when they’re being played back from recorder.js. You can also record audio, as long as your microphone is not muted.

When you have a recording that you like, save it to your local machine using the save recording button. The format for our output recording is a .hfr extension. We’ll want to access this from a script that will run on an *assignment client* on our domain server, so we’ll also want to add our script to our asset server.

Note: If you plan on using your recording in multiple domains, you will need to host your .hfr file on a file hosting service or upload it to each domain’s file server separately. Asset server entities are only available in the specific domain they’re hosted on.

1. Open up the Asset Browser with Edit > Asset Browser, or opening the Edit menu and choosing “Assets”.
2. Upload your .hfr file and save it into the Asset Server

3. Right click on the file in the asset browser and get the link to the file, e.g. atp:/filename.hfr

Once we have our recording saved, we're going to be setting up a script that runs continuously on our domain to play it back as a new NPC.

Understanding Assignment Client Scripts for Building NPCs
Assignment client scripts are specific types of scripts that run as part of your domain server. These scripts can take on a variety of different tasks in a domain, including playing the part of an avatar in the world. We'll be writing a specific assignment client agent that will load in our recorded animation, assign it to a location and avatar, set characteristics of playing back the animation, and then tell our agent to act like an avatar.

To start with a basic animation playback that loops, we'll load up the animation and keep the other information about the recording in-tact.

1. Create a new script on your computer called ACPlaybackScript.js.
2. Open up ACPlaybackScript.js and paste in the following code, replacing the RECORDINGNAME with your own recording file:

```
var clip_url = "atp:/RECORDINGNAME.hfr";
```

We'll use the next few lines of code to set up our playback channel and a few settings for our recording.

```
var PLAYBACK_CHANNEL = "playbackChannel";
var playFromCurrentLocation = false;
var useDisplayName = true;
var useAttachments = true;
var useAvatarModel = true;
var loop = true;
```

Variable	Description
playFromCurrentLocation	Whether the avatar should play from the original recorded location or a current position that is set elsewhere
useDisplayName	If the avatar should have the name set from the original recording
useAttachments	If the recording should include any attachments that

	were on the original avatar
`useAvatarModel`	Whether or not the original avatar skin should be used in the playback
`loop`	If the recording should repeat or only play once

Next, we'll set up our recording according to the parameters that we set above. For the time being, we're keeping things simple, but we'll dive into changing up the avatar, adding triggers for different behaviors, and changing some other settings with our NPCs later on.

```
Recording.loadRecording(clip_url);
Recording.setPlayFromCurrentLocation(playFromCurrentLocation);
Recording.setPlayerUseDisplayName(useDisplayName);
Recording.setPlayerUseAttachments(useAttachments);
Recording.setPlayerUseHeadModel(false);
Recording.setPlayerLoop(loop);
Recording.setPlayerUseSkeletonModel(useAvatarModel);
```

The next thing that we'll do is set up our agent to act as an avatar - by default, agents don't really take on a personified part of our world, but using the `.isAvatar` function tells the assignment client to treat our recording as an NPC. We also kick off our recording loop, and specify that it should run with an update loop as soon as it connects to the domain.

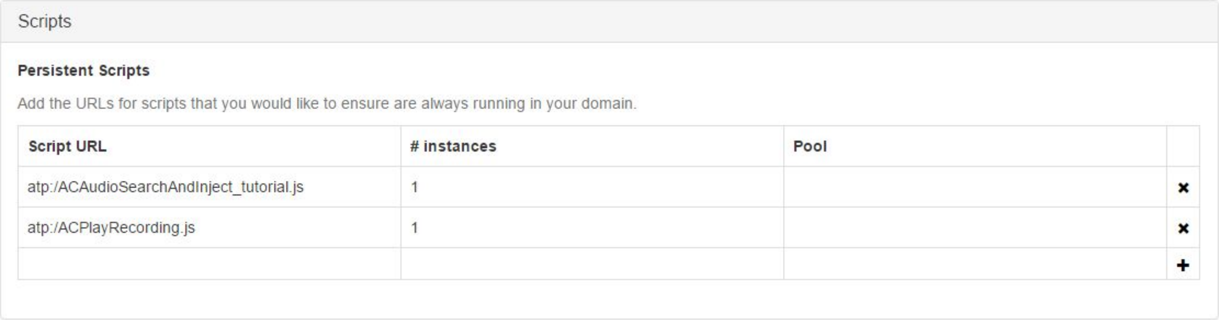
```
Agent.isAvatar = true;

if (!Recording.isPlaying()) {
  Recording.setPlayerTime(0.0);
  Recording.startPlaying();
}

Script.update.connect(update);
```

Save your script, and following the instructions above for the recording, add your script to your asset server. Copy the url (atp:/ACPlaybackScript.js) - we'll need to tell our domain to make sure it's always running when our domain is up.

Running your Assignment Client Script in a Domain

Our NPC script is a specific type of agent that will run on our server on an assignment client, meaning that it will run consistently in our domain whenever the domain itself is running. Running an assignment client script is different than an entity script on an item - we'll need to hop into our server settings to set our client script to run.

1. With Sandbox running, enter http://localhost:40100 in your desktop browser to connect to your server
2. Select ‘Settings’ from the top of the server page and scroll down / select ‘Scripts’ from the sidebar to see running assignment scripts
3. Add your script url to the table of persistent scripts by selecting the + icon on the last table row



Scripts

Persistent Scripts

Add the URLs for scripts that you would like to ensure are always running in your domain.

Script URL	# instances	Pool	
atp:/ACAudioSearchAndInject_tutorial.js	1		✖
atp:/ACPlayRecording.js	1		✖
			+

4. Save your changes and reset the server by clicking the green ‘Save and Restart’ button.

If you don’t see your agent appear in your domain, double-check your script for any errors and try reloading the content in your Sandbox by going to Edit > Reload Content (Clears all Caches).

Next Steps

In this tutorial, we walked through the process of changing avatars, recording different behaviors, and playing them back on an assignment client script from our domain. Play around with the different agent settings, and experiment how you can use different triggers to create more interactive NPCs using your own recordings!

Additional Resources

Download Fuse for Mixamo Avatars
Avatar Standards Documentation
Create Avatars
Blender Workflow for Avatars
Script Type Overview