Requirements / Packages

- Text Mesh Pro from Unity Package Manager
- NetCodeForGameObjects (Only If you want to check out the 3D demo scene)

Included Plugins

Zenject - dependency Injection framework
VivoxAccessToken (VAT) - creates VAT's for secure server requirements

New Features

Added Dynamic events to make subscribing to Vivox events from any class easier
Added Dynamic Async Events with support for normal Dynamic Synchronous Events
Added support for multiple logins and support to use it with existing Login() Method instead of creating separate methods.

Added support for VivoxConfig so users can pass in their own Vivox configuration
Added Zenject as my dependency injection framework

Added support for Cross Muting Users. In the Demo scene only 1 player can be cross muted at one time, but EasyCode API supports Vivox's Cross muting functionality for multiple players at a time (Located in EasyManager.cs or EasyMute.cs)

Added support for setting Auto/Custom Voice Activity Detection (VAD)
Added support for Updating Login Properties after user has logged into Vivox
Added Online and Offline documentation

Improvements

I have added Assembly definitions for faster compile time when I make changes and to keep EasyCodeForVivox separate from the main default Assembly CSharp

Added support for more runtime/dynamic events with exception handling and logging

Updated Dynamic Event cache to store dynamic event attributes methods at startup instead of checking at Runtime for faster Dynamic events

Added parameter checking for Dynamic Events to reduce errors and increase performance

Updated enum AttributeOptions for faster Dynamic Event checking based on Synchronous or Async methods

Added optional check for Dynamic events so if users opt out it doesn't slow down their game/app by checking for Dynamic events at runtime / startup

- Added Refresh Audio Devices to EasyManager
- Added Display name optional parameter for logging in

Decided against using Parallel.ForEach with async/await for concurrency when executing event methods dynamically (see why on this post by Stephen Cleary) and am using a List<Task> in combination with Task.Run() so I can use Task.WhenAll to still achieve concurrency and prevent thread blocking. Read here for more info

Demo scenes now using TextMeshProUGUI in and its own Content element in the Viewport, with a Content Size Fitter component for a dynamically updated Text UI instead of old way using Image and super expanded Text element

Separated Chat Scene UI into panels and added some simple explanations for each functionality

Added 3D demo scene to test 3D Positional Audio

Added Helper methods for Android / IOS permissions

Added Helper methods for getting /setting Transmitting channels

Added EasySettings ScriptableObject for Logging and different settings for EasyCode

Added colored Debug.Logs

Added more code examples on how to use EasyCode features

Added queue check for TTSMessages (if over 10 they may be lost have yet to handle overflow of TTS messages)

You can now choose Vivox Voice Gender for TTS in EasySettings instead of it being hardcoded

Added Participant Update Frequency options to EasySettings and added some tooltips.

Added Region/Shard support for channels

Added [RequireComponent] for Scene Context on EasyManager so users don't have to worry about setting up Zenject

Added more helpful VivoxExtensions

Added LoginAdded, LoginRemoved, LoginValuesUpdated events for handling multiple LoginSessions

Added Cross Mute overridable events to EasyManager

Bug Fixes

Fixed a bug when trying to join more than 1 3d positional channel. Added additional Channel3dProperties parameter for when joining channel

Fixed Dynamic Events not working if Key wasn't found in dictionary, but the event existed and was found on method

Fixed - originally was unsubscribing from the event reference that was passed in when invoking event handlers but realized after messing with the UI that it was not properly unsubscribing because I would have duplicate values when re-logging in. Have realized that you need to Unsubscribe using the original object (ex. loginSession, channelSession) and it is best to do so when Loggin out or leaving a channel for example and not from the invoked method called by event delegates

Fixed bug when matching parameters for Dynamic Events
Fixed TTS bug where Remote Transmission would play with local playback

Notes

Tried removing dependencies from EasyManager.cs but ended up still injecting everything. It is basically a super class. I plan on making examples on how to break apart easy manager for more modular design

Breaking Changes

Changed

Updated namespaces for EasyCodeForVivox classes

Removed EasyManager base class and went back to inheriting from MonoBehaviour

Rearranged classes for better organization

Changed Subscribe and Unsubscribe methods from public to private in VivoxBackend classes

Changed methods/etc. that contained "Voice" to "Audio" to keep consistent naming

Added Mirror 3D Positional Voice example as its own class

Added additional Channel3dProperties parameter for when joining channel

Changed some naming for Properties in EasySession.cs

Decided to keep EasyManager and refactored some methods into other classes for better SOLID design. Added interfaces for potential replacements classes in the future

Added namespace for events

Using dependency injection without forcing users of EasyCode to use it for the most part. Refactored code in EasyManager to the individual classes for more SOLIDness

Changed AudioSettings class name to Audio

Changed TTS extension method names.

Removed public static bool IsClientInitialized = false; from EasySession and checking exposed Initialized variable of the VivoxClient instead

Changed methods names SetTextActiveInChannel and SetVoiceActiveInChannel to ToggleTextInChannel and ToggleAudioInChannel

Changed applicationStanzaNamespace and applicationStanzaBody to Web API lingo and am using header and body as variable names

Added the rest of the overrideable callback methods for Vivox events in Easy3DExample and EasyChatExample

MuteAllUsers in the EasyMute class is now called LocalMuteAllUsers and same thing for Unmute methods

Changed public void SetAudioDeviceInput(IAudioDevice device, VivoxUnity.Client client) to public void SetAudioDeviceInput(string deviceName, VivoxUnity.Client client) so it easier to search for Audio Input / Output Devices

- TTSChooseVoice has been renamed to ChooseVoiceGender
- TTS events handler methods are now private in EasyTextToSpeech
- Changed InjectAudio to StartAudioInjection in EasyManager
- Changed SpeakTTS to PlayTTSMessage
- Changed TTSMsgLocalReplaceCurrentPlaying to TTSMsgLocalReplaceCurrentMessagePlaying
- Changed RefreshAudioDevices to have an Input and Output device

Removed

Removed static events from each Vivox functionality class and put into its own class of static events. I did this to avoid confusing using the asset, to subscribe to custom events and having to know which class which event is in, even though it breaks SRP and modularity a bit. I may switch it back in the future and provide both options since they can work separately/independently of each other

Removed Presence and Subscriptions methods and events because Unity is coming out with a separate service to manage Friends. Refer to this forum. Since they are not supporting the Presence feature I have decided not to either

Removed old Image Container and Text that I used for Chat in Demo Scene.

Removed bool parameter from IsLocalMuted event

Removed EasyExtensions and OldEasyExtensions

Removed public static bool IsClientInitialized = false; from EasySession

removed public void LocalToggleMuteRemoteUser(string userName, IChannelSession channelSession) from EasyMute in favor of LocalMuteRemoteUser(string userName, IChannelSession channelSession, bool mute) to give users the option to mute or unmute. Also refactored EasyManager wrapper methods to reflect these changes

Deprecated

EasyExtensions and OldEasyExtensions