**RecordManager:**

1. all the functions related to the recording
2. all the functions related to uploading the audio data to the server

Init(): init the recording timer text to “00:00:00”

PrepareRecord(): when the start record button in the home page is clicked, check whether the user joined at least one event to start recording

RequestRecord(): request microphone permission

CheckPermission(): check whether the user allow recording

TryGainPermission(): try to gain the permission of recording

StartRecord(): start to record, maximum record length is one hour

StopRecord(): stop to record and update the length of the audioclip

PlayAudio(): play the audio

ClipAudio(): capture the current clip data

UpdateTime(): update the recording timer text, change the raw time into ”XX:XX:XX” format

OnUploadBtnClick(): upload audio data with information to the server

UploadAudioToServer(): upload the audio to the server with category/soundlabel/eventid

OnCustomizeLabel(): turn the customize label checkbox on

ResetValuesInSoundLabelling(): when home or save is clicked

OnSoundLabelOptionChange(): change the sound label option

GamepieceUpdate(): update the game piece

UploadGamePieceInServer(): upload the game piece in the server

**CrossAudioList:**

RequestSoundList(): request the sound list from the server for cross-validation

GetSoundList(): analyze the data from the server to get the sound list

DownLoadClipRoutine(): download the 10 sounds once a time to store in the cache

DownloadAudioFromSoundList(): download the audio from the sound list

**TagManager:**

loadTagList(): load 3 options for cross-validation, one contains the original label

randomMnumber(): random algorithm

CompareTag(): compare whether the user’s choice match to the original label

ReportQuestion(): report problematic sounds

PlayAudio(): play the audio sound

PauseAudio(): Pause the audio sound

UpdateTime(): change the raw time into ”XX:XX:XX” format

SkipAudio(): Stop the audio and skip the current validation

**JasonClassN:**

parse the downloaded sound data for cross-validation

**EventDetails:**

StartGettingDetail(): event details including event title/streamer name/event date

GetEventDetails(): start coroutine GetEventsDetailsFromServer()

GetEventsDetailsFromServer(): get the event details from the server

**EventID:**

SetID()

GetID()

OnClickEventCard()

**CheckEvents:**

NoEvents(): no event exist

EventsExist(): events exist

**CheckValidLogin:**

CheckToken(): check the user’s token

sendRequest(): request authorization from the server

GoToConsentPage(): after first login, show the consent page

GoToWebView(): go to the sample web view page

GoToNextPage(): go to the home page

**ClickedSoundLabel:**

OnClickLabelRecord(): when the user click the microphone button on the eventdetailspage, go to the recording page

**EventJoin:**

OnClickJoinEventButton(): trigger the function JoinEvent()

JoinEvent(): send the user’s data to the server to join the event

**GetEvents:**

StartGettingEvents(): Start status, trigger the function GetJoinedEventsInfoFromServer() & GetEventsInfoFromServer()

UpdateJoinedEvents(): trigger the function GetJoinedEventsInfoFromServer()

UpdateEvents(): trigger the function GetEventsInfoFromServer()

GetEventsInfoFromServer(): get current valid events detailed information from the server

GetJoinedEventsInfoFromServer(): get the joined event detailed information from the server

**GetUserName:**

Getusername(): get the user’s display name

**JasonClass:** parse the event data for event details

**OnClickEventCard:**

OnClickEventCardButton(): when the user click the event card, show the detail page of this event

**OnClickSoundLabelRecordButton:**

OnClick(): trigger the function ClickedSoundLabel-OnClickLabelRecord()

**ClickedSoundLabel:**

OnClickLabelRecord(): when the user click the microphone button on the eventdetailspage, go to the recording page

**RotateGamePiece:**

OnClickLeftButton(): go to the previous game piece

OnClickRightButton(): go to the next game piece

OnConfirmButtonClick(): when the user click the confirm button

OnClickGamePiece(): rotate the game piece if on touch

**SoundGalleryBackButton:**

SetPreviousPage(): set the previous page, trigger the function BackButtonSet()

GoBackToPreviousPage(): set the previous page active

BackButtonSet(): go to the home page or go to the tasklist

**SoundItem:**

DeleteFile(): remove sound items

PlayAudioFile: trigger the function AddFileToClips()

AddFileToClips(): trigger the function ReadLocalAudioClip()/ReadServerlAudioClip():

ReadLocalAudioClip(): read local audio clips

ReadServerAudioClip(): read server audio clips

**SoundLabellingDetails:**

SoundObject(): get sound path/displayName/labelName/id

**SaveAudio:** not using

**ServerSaveAudio:** not using

**CrossValiManager:** not using

**WebvViewObject:** not using