

Design Specification for Connectome

Version 2.0

Prepared by:

Jon Schuck

Kevin Garrone

Khaled Alhendi

Sheik Hassan

Nate Bender

Kennan Meyer

Bairavi Venkatesh

Advisor:

Jeff Salvage

Stakeholder: Als Hope Foundation

Table of Contents

Contents

Revision History	3
1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Design Goals	4
1.4 Definitions	4
1.5 References	5
2. System Overview	6
2.1 Technologies Used	6
2.1.1 Hardware Interfaces	
The application will need to interface with the Emotiv hardware through the use of the developer SDK provided by the company. The Emotiv will require a connection to the user's computer through the use of bluetooth. The user will be required to train their mental commands in order to establish the ability to navigate the application.	6
2.1.2 Software Interfaces	6
C# and .NET	6
Unity	6
2.2 Context Diagram	7
2.3 Deployment Diagram	8
2.3 Architecture Diagram	9
2.4 Sequence Diagram	10
3. Detailed System Architecture	12
3.1 Primitive Types	12
3.2 Connectome Classes	14
4. Detailed Class Description	14
4.1 DisplayObjects	15
4.2 Device Reader	42
4.3 Selection Highlighter	53
4.4 Training	63
4.5 MenuBuilder	86

4.6 UITest	101
4.7 Calculator	109
4.8 DataSampler	113
4.9 TestRunner	118
4.10 Twitter	126
5. Traceability Matrix	165

Revision History

Version	Date	Reason For Changes
Version 1	10/21/16	Original Copy
Version 2	4/19/17	Update for SRS changes

1. Introduction

1.1 Purpose

This document specifies the software architecture and design specifications for connectome. The design decisions outlined in this document were made in compliance with the design constraints and functionality requirements outlined in the Connectome Software Requirements Specification Documents.

1.2 Scope

The document describes the software architecture for the initial release of Connectome. The implementation of Connectome adheres to the models and design decisions outlined in this report. The audience for this report is the developers and stakeholders of Connectome.

1.3 Design Goals

- **Modular** - The architecture of this application is broken down into separate software components with the goal of low coupling and high cohesion. This makes the application easier to both maintain as well as grow as new features are implemented later in the application's life cycle.
- **Extensibility** - The application is designed in a way that the software should be easily extensible. New features should be easily added to the existing system with minimum need for refactoring or changes to the existing code.
- **Reusability** - Every component implemented in the application is designed with code reuse in mind. The code is designed so that each component made can be used in multiple times throughout the life of the application.

1.4 Definitions

Brain Computer Interface (BCI) - a non-invasive technique wherein neural signals are used to communicate or control a desired output

Amyotrophic Lateral Sclerosis (ALS) - A nervous system disease that weakens muscles and impacts physical function.

Active Command - A distinct, user-trained neural signal that triggers a specific action (click, keyboard pop-up etc) in the Connectome interface.

Passive Command - A resting state, baseline neural signal that does not elicit a specific action in the Connectome interface.

Electroencephalogram (EEG) - the measurement of electrical activity in different parts of the brain

Emotiv - The name of the company that creates BCI devices used to pick up brain activity through EEG.

SSVEP – In neurology and neuroscience research, steady state visually evoked potentials (SSVEP) are signals that are natural responses to visual stimulation at specific frequencies

Timeline – A feed of social media catered to the users wishes

Profile Picture – A picture used on social media to represent the user to others on the site.

Twitter Handle – A unique name for the user that others can use to contact them.

Real Twitter Name – Nonunique name that represents the user

1.5 References

[Emotiv Hardware Manual](#)

For more information about the Emotiv hardware, please refer to the manual referenced above.

2. System Overview

2.1 Technologies Used

2.1.1 Hardware Interfaces

The application will need to interface with the Emotiv hardware through the use of the developer SDK provided by the company. The Emotiv will require a connection to the user's computer through the use of bluetooth. The user will be required to train their mental commands in order to establish the ability to navigate the application.

2.1.2 Software Interfaces

C# and Dot Net

The application will run on version 6.0 of the C# language. This is the most recent version of the C# programming language. The C# programming language will be used for all of the backend programming for the application. Visual Studio 2015 will be used to compile all of the code that is developed in C#. Version 3.5 of the .NET framework will be used for development in C# because of limitations imposed by Unity game engine.

Unity

The application will use the Unity game engine for the user interface of the Connectome application. While the application will not contain any game components, the framework makes the interface with the emotiv easy. A 32-bit personal version of the unity game engine will be used for the project. None of the components of the professional version should be needed for the application. Unity version 5.5 will be used for the development of the project.

2.2 Context Diagram

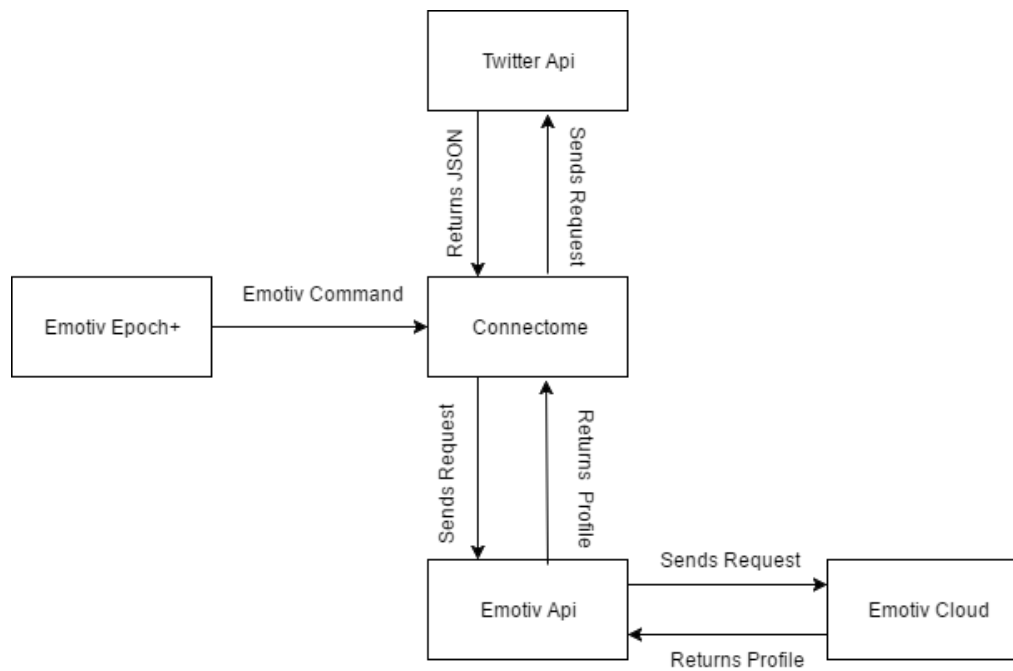


Figure 2.2.A : Connectome Context Diagram

Figure 2.2.A shows the main components of Connectome. The user interacts with the Connectome application using the Emotiv Epoch+ headset. When the user gives a mental command, it is sent to the connectome application. The application processes the command and performs the appropriate action. When the user logs into Connectome, the application will send a request to Emotiv's Cloud through its sdk to load the user's profile. Which will be returned and the application will start. If the user is trying to interact with twitter, the application will use the Twitter API to send or request information. The Twitter API will respond by sending JSON back to Connectome.

2.3 Deployment Diagram

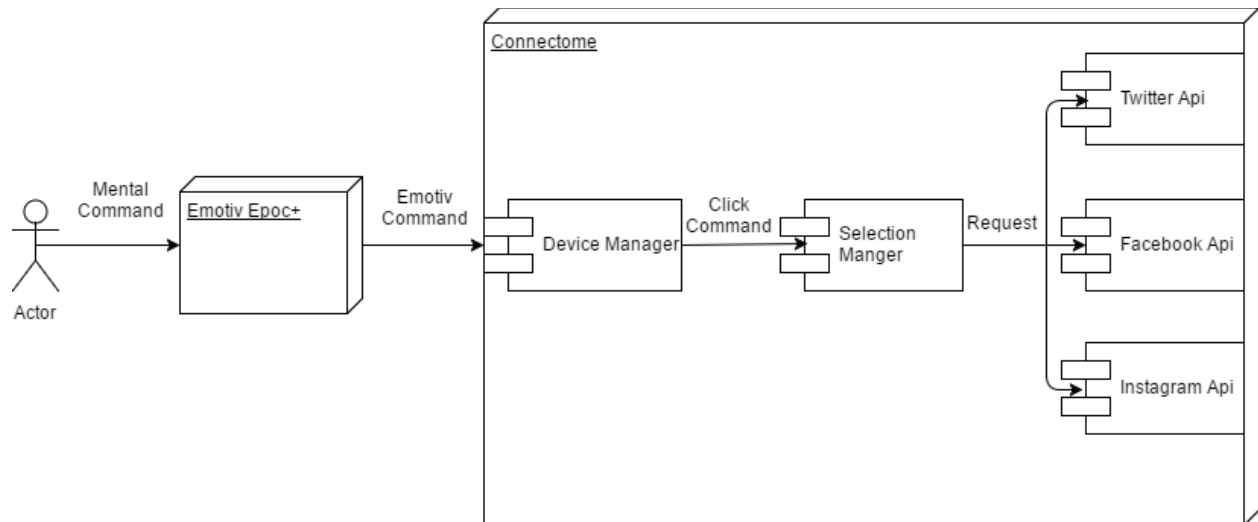


Figure 2.3.A Connectome Deployment Diagram

Figure 2.3.A shows a high level overview of the deployment of the Connectome Application. There are two components that are required. The Emotiv Epoch+ and the connectome application. The user wears the Emotiv and use it to pick up their mental commands. The Emotiv will then translate the mental commands into Emotiv commands that are sent to the Connectome Application. Connectome will pick up the Emotiv commands in the Device Manager module. Which will use an interpreter to decide if the user is trying to click or not. If the interpreter determines that the user is trying to click, the click command will be sent to the selection manager which will click on the component that is currently selected. If the component is linked to a Social Media Api, a requested will be sent.

2.3 Architecture Diagram

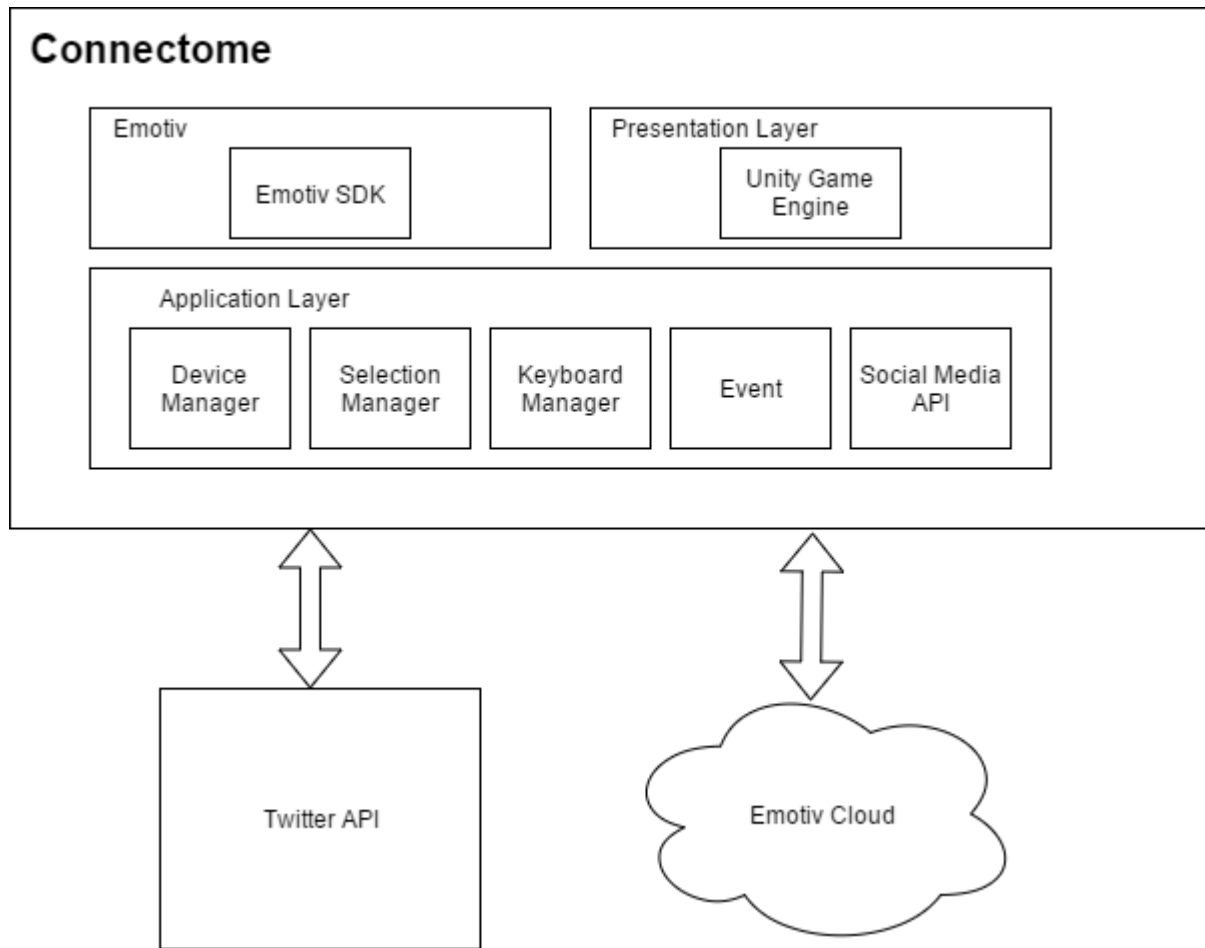


Figure 2.3.A Connectome Architecture Diagram

Figure 3 shows the general architecture overview of the Connectome application. There are two main layers within the Connectome application. The presentation layer, which was created in unity, and the application layer, which is written in C#. There are 5 main components within the application layer. The Device Manager which reads the emotiv command, the Selection Manager which rotates through the different elements on the screen for the user to interact with, the Keyboard Manager which handles interacting with the different keyboards the users types with, and the different social media Apis. The application also interacts with Twitter's Api to to read and post the user's information. As well as the

Emotiv Cloud to load the user's different profiles.

2.4 Sequence Diagram

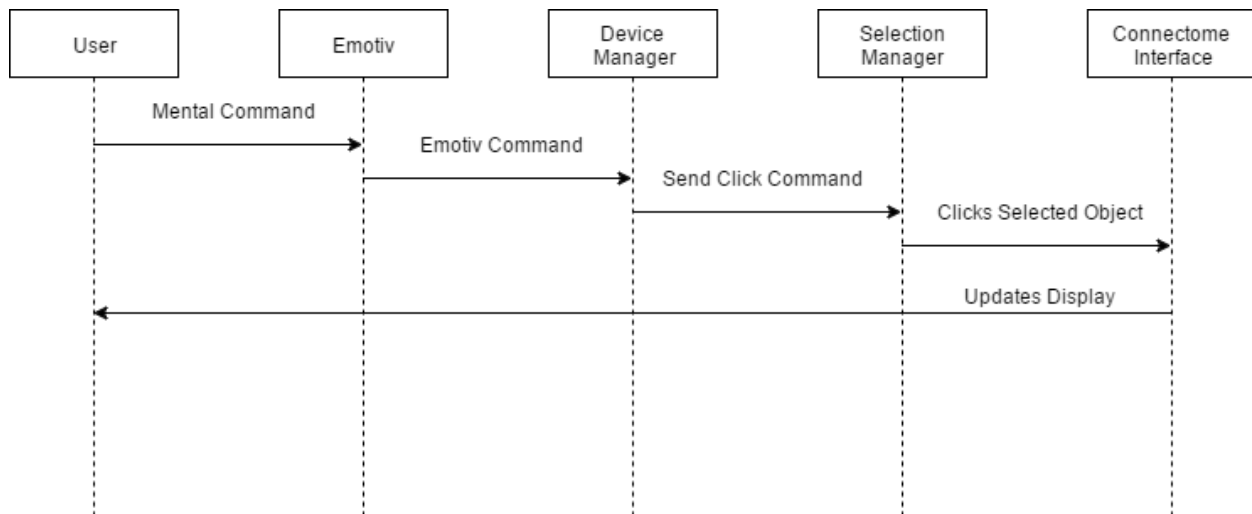


Figure 2.4.A: Sequence Diagram for pushing a button

Figure 2.4.A shows the process of pressing a button using mental commands. The user gives a mental command when the button the user wants to select is highlighted by the selection manager. Which rotates through every button on screen in a timed sequence. The mental command is picked up by the Emotiv Epoch+ and is converted into an Emotiv command the is sent to the application. The device reader picks up the commands and interprets what to do with them. When a click is interpreted a click command is sent to the selection manager. Which clicks the object that is currently highlighted, The connectome interface is then updated with the action of the button that is clicked.

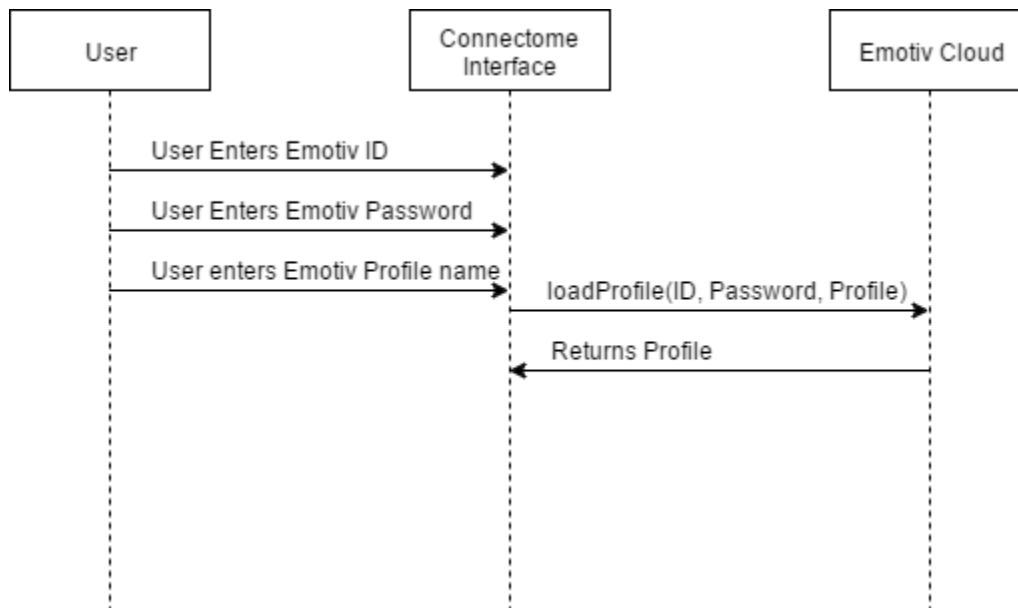


Figure 2.4.B: Sequence Diagram loading Emotiv profile

Figure 2.4.B shows the process of loading an emotiv profile into connectome. In the login screen the user will be prompted for the Emotiv Id, Emotiv Password, and the name of the Profile. When the user hits enter, they will be passed to a method in the Emotiv SDK that will make a call to get the profile. The Emotiv Cloud will then verify the information and return the profile.

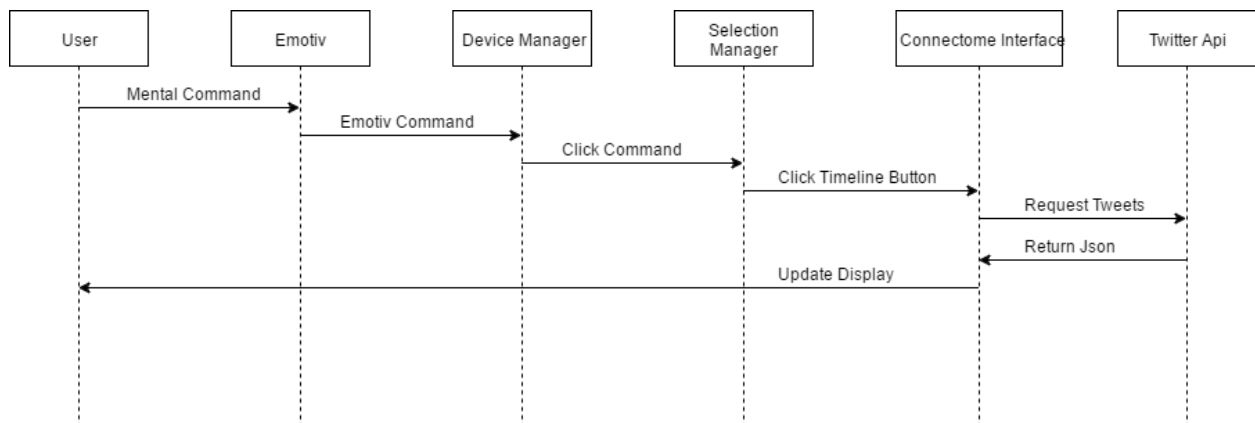


Figure 2.4.C: Sequence Diagram for accessing the user's timeline

Figure 2.4.C shows the process of loading the Twitter timeline. The user gives a mental command that is picked up by the Emotiv device. The command is then translated into an Emotiv command. Which is sent

to the Device Manager. The device manager interprets if there was a click, and if so, sends a click command to the Selection Manager. Which in this instance is highlighting the Timeline Button that is click. The Connectome Application then calls the Twitter Api to load the User's timeline. The api returns the tweets in the form of Json that is used to update the display for the user.

3. Detailed System Architecture

3.1 Primitive Types

The following Data Types are built in and do not require implementation

Short Name	.NET Class	Type	Width	Range (bits)
byte	Byte	Unsigned integer	8	0-255
sbyte	SByte	Signed integer	8	-128-127
int	Int32	Signed integer	32	-2,147,483,648 to 2,147,483,647
uint	UInt32	Signed integer	32	0 to 4294967295
short	Int16	Signed integer	16	-32,768 to 32,767
ushort	UInt16	Unsigned integer	16	0 to 65535
long	Int64	Signed integer	64	-9223372036854775808 to 9223372036854775807
ulong	UInt64	Unsigned integer	64	0 to 18446744073709551615
float	Single	Single-precision floating point type	32	-3.402823e38 to 3.402823e38
double	Double	Double-precision floating point	64	-1.79769313486232e308

		type		to 1.79769313486232e308
char	Char	A single Unicode character	16	Unicode symbols used in text
bool	Boolean	Logical Boolean type	8	True or false
object	Object	Base type of all other types		
string	String	A sequence of characters		
decimal	Decimal	Precise fractional or integral type	128	$\pm 1.0 \times 10e-28$ to $\pm 7.9 \times 10e28$

3.2 Connectome Classes

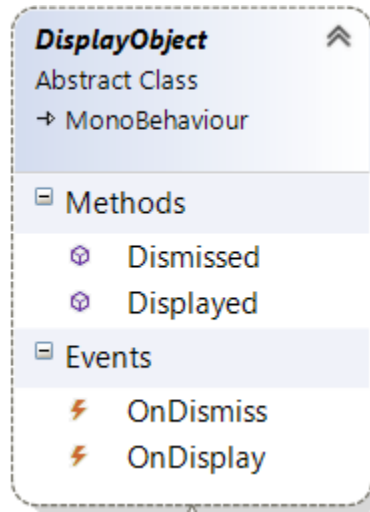


4. Detailed Class Description

4.1 DisplayObjects



4.1.1. Display Object



DisplayObject is an abstract class that is used for the creation of Menus and Panels.

4.1.1.1. Attributes

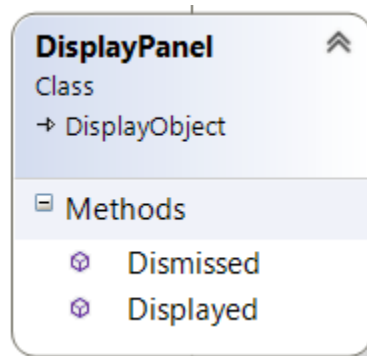
Name	Access	Type	Description
None	None	None	None

4.1.1.2. Methods

Name:	Dismissed
Input:	None
Output:	None
Description:	Gets called after object is dismissed. Calls OnDismiss Event

Name:	Displayed
Input:	None
Output:	None
Description:	Gets called after object is displayed. Calls OnDisplay Event

4.1.2. Display Panel



Display Panel is a high level class used for creating Panels.

4.1.2.1. Attributes

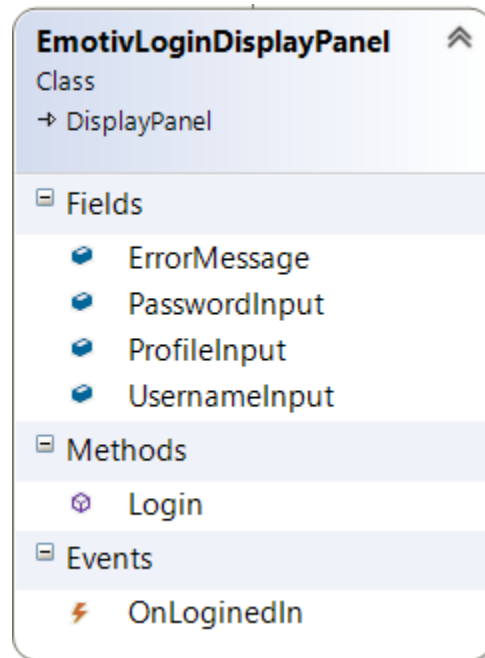
Name	Access	Type	Description
None	None	None	None

4.1.2.2. Methods

Name:	Dismissed
Input:	None
Output:	None
Description:	Gets called after object is dismissed. Calls OnDismiss Event

Name:	Displayed
Input:	None
Output:	None
Description:	Gets called after object is displayed. Calls OnDisplay Event

4.1.3. EmotivLoginDisplayPanel



EmotivLoginDisplayPanel is the class used in the login screen to login the user into the application.

4.1.3.1. Attribute

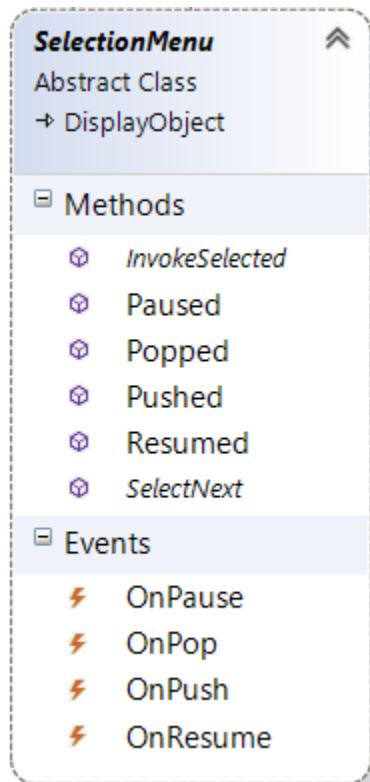
Name	Access	Type	Description
ErrorMessage	Public	Text	Used to display error messages to the user
PasswordInput	Public	InputField	Used to collect the user's password
ProfileInput	Public	InputField	Used to collect the user's profile name

UsernameInput	Public	InputField	Used to collect the user's username
---------------	--------	------------	-------------------------------------

4.1.3.2. Methods

Name:	Login
Input:	None
Output:	None
Description:	Attempt to create a device using given credentials

4.1.4. Selection Menu



SelectionMenu is an abstract class used to create Selectable Menus

4.1.4.1. Attribute

Name	Access	Type	Description
None	None	None	None

4.1.4.2. Methods

Name:	InvokeSelected
Input:	None
Output:	ISelectionMenu
Description:	Invokes currentpointed selection

Name:	Popped
Input:	None
Output:	None
Description:	Called after menu is popped. Calls popped event.

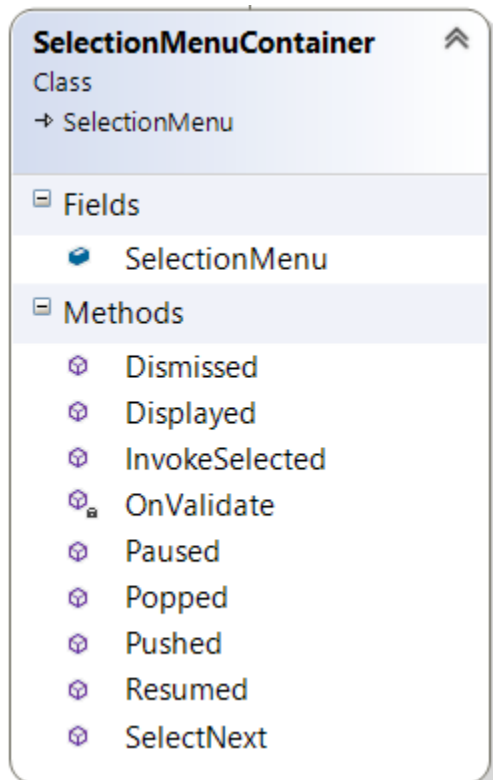
Name:	Paused
Input:	None
Output:	None
Description:	Invoked when menu is paused. Calls OnPause event.

Name:	Paused
Input:	None
Output:	None
Description:	Invoked when menu is paused. Calls OnPause event.

Name:	Resumed
Input:	None
Output:	None
Description:	Invoked when menu is paused. Calls OnResume event.

Name:	SelectNext
Input:	ISelectionHighlighter
Output:	None
Description:	Moves pointer to next selection

4.1.5. SelectionMenuContainer



4.1.5.1. Attributes

Name	Access	Type	Description
SelectionMenu	Public	SelectionMenu	A selectionmenu described in 4.2.3

4.1.5.2. Methods

Name:	Dismissed
Input:	None
Output:	None
Description:	Invokes SelectionMenu's Dismissed Method.

Name:	InvokeSelection
Input:	None
Output:	ISelectionMenu
Description:	Invokes SelectionMenu's InvokeSelected Method.

Name:	Popped
Input:	None
Output:	None
Description:	Invokes SelectionMenu's Popped Method.

Name:	Pushed
Input:	None
Output:	None
Description:	Invokes SelectionMenu's Pushed Method.

Name:	Paused
Input:	None
Output:	None
Description:	Invokes SelectionMenu's Paused Method.

Name:	Resumed
Input:	None
Output:	None
Description:	Invokes SelectionMenu's Resumed Method.

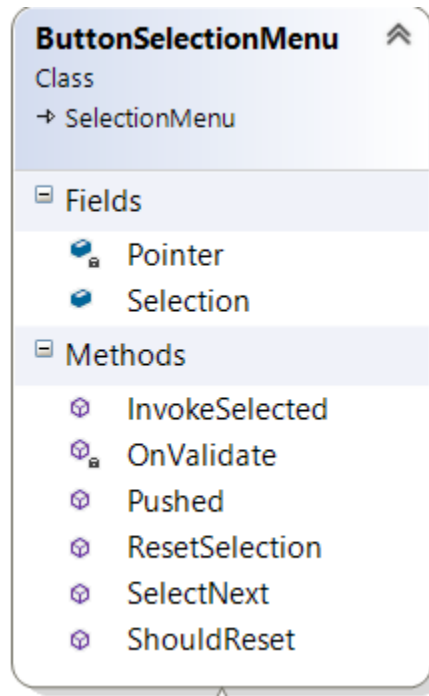
Name:	Displayed
--------------	-----------

Input:	None
Output:	None
Description:	Invokes SelectionMenu's Displayed Method.

Name:	SelectNext
Input:	ISelectionHighlighter
Output:	None
Description:	Invokes SelectionMenu's SelectNext Method.

Name:	OnValidate
Input:	None
Output:	None
Description:	Checks if class contains a SelectionMenu or itself.

4.1.6. ButtonSelectionMode



4.1.6.1. Attributes

Name	Access	Type	Description
Pointer	Public	int	Points to the current button in the array
Selection	Public	Button[]	Array of buttons on the current menu

4.1.6.2. Methods

Name:	OnValidate
Input:	None
Output:	None
Description:	Checks if there are buttons to select

Name:	InvokeSelected
Input:	None
Output:	ISelectionMenu
Description:	Clicks currently selected button

Name:	SelectNext
Input:	ISelectionHighlighter
Output:	None
Description:	Moves pointer to next button

Name:	ResetSelection
Input:	None

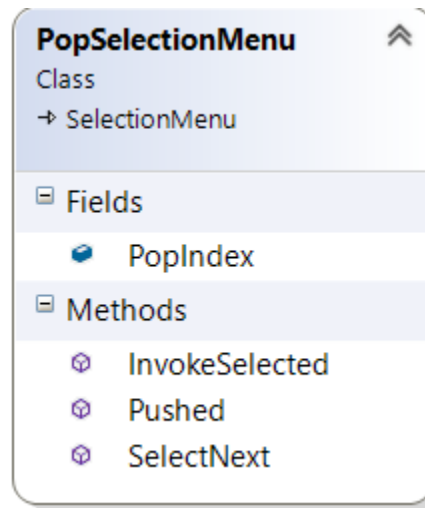
Output:	None
Description:	Sets the pointer to the beginning

Name:	Pushed
Input:	None
Output:	None
Description:	Adds next menu to the stack

Name:	Resumed
Input:	None
Output:	None
Description:	Resumes menu

Name:	ShouldReset
Input:	None
Output:	None
Description:	Checks if pointer has reached the end of the list.

4.1.7. PopSelectionMenu



4.1.7.1. Attributes

Name	Access	Type	Description
PopIndex	Public	int	The number of times the selection manager should pop

4.1.7.2. Methods

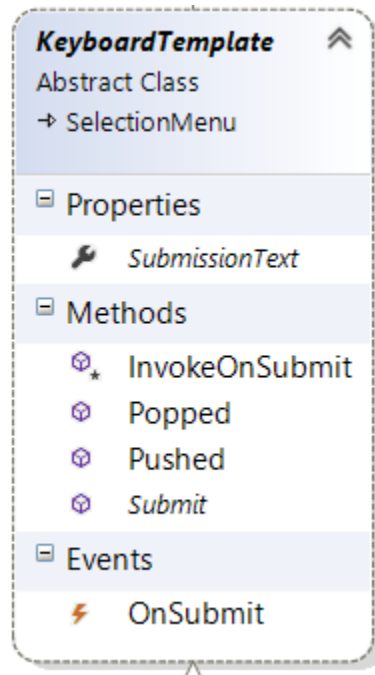
Name:	Pushed
Input:	None

Output:	None
Description:	Pops itself and then the number of times desired.

Name:	InvokeSelected
Input:	None
Output:	ISelectionMenu
Description:	Throws NotImplementedException

Name:	SelectNext
Input:	None
Output:	None
Description:	Throws NotImplementedException

4.1.8. KeyboardTemplate



4.1.8.1. Attributes

Name	Access	Type	Description
SubmissionText	Public	String	The users current input into the keyboard

4.1.8.2. Methods

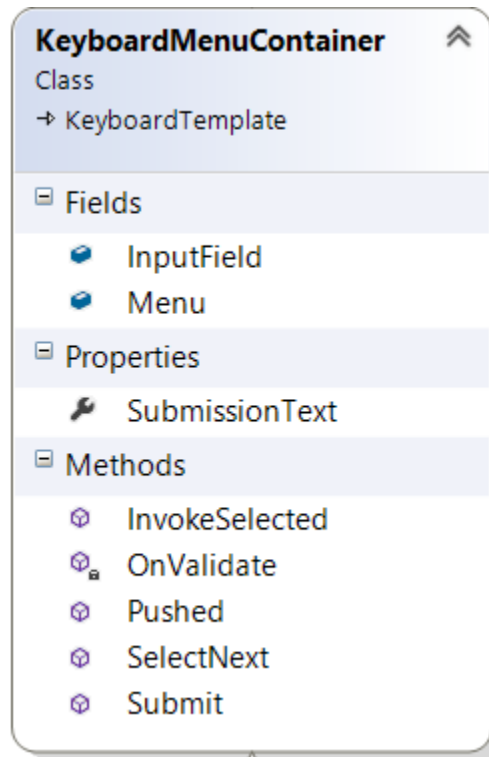
Name:	InvokeOnSubmit
Input:	None
Output:	None

Description:	Submits SubmissionText
---------------------	------------------------

Name:	Popped
Input:	None
Output:	None
Description:	Pops menu on the stack

Name:	Pushed
Input:	None
Output:	None
Description:	reset Text field and enable gameobject when pushed by SelectionManager

4.1.9. KeyboardMenuContainer



4.1.10. Attribute

Name	Access	Type	Description
InputField	Public	InputField	Shows text inputted into keyboard
Menu	Public	SelectionMenu	Menu used to select buttons/keys

4.1.11. Methods

Name:	Submit
Input:	None
Output:	None
Description:	Sends the SubmissionText

Name:	InvokeSelected
Input:	None
Output:	ISelectionMenu
Description:	Returns current selectedobject

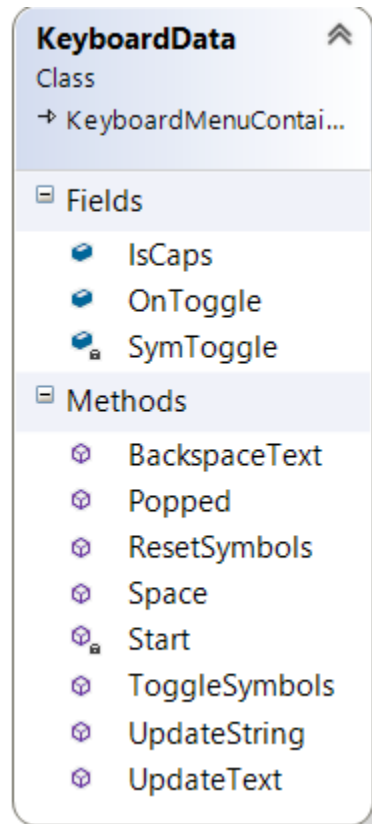
Name:	InvokeSelected
Input:	None
Output:	ISelectionMenu
Description:	Returns current selectedobject

Name:	SelectNext
Input:	ISelectionMenu

Output:	None
Description:	Selects the next object in line

Name:	Pushed
Input:	None
Output:	None
Description:	Pushes next menu on the stack

4.1.12. KeyboardData



4.1.12.1. Attributes

Name	Access	Type	Description
------	--------	------	-------------

IsCaps	Public	bool	Sets uppercase letter
SymToggle	Public	bool	Sets symbols
OnToggle	public	Action<string>	String of toggle actions

4.1.12.2. Methods

Name:	UpdateString
Input:	String
Output:	None
Description:	Adds selected text to current input

Name:	Space
Input:	None
Output:	None
Description:	Inserts space into string

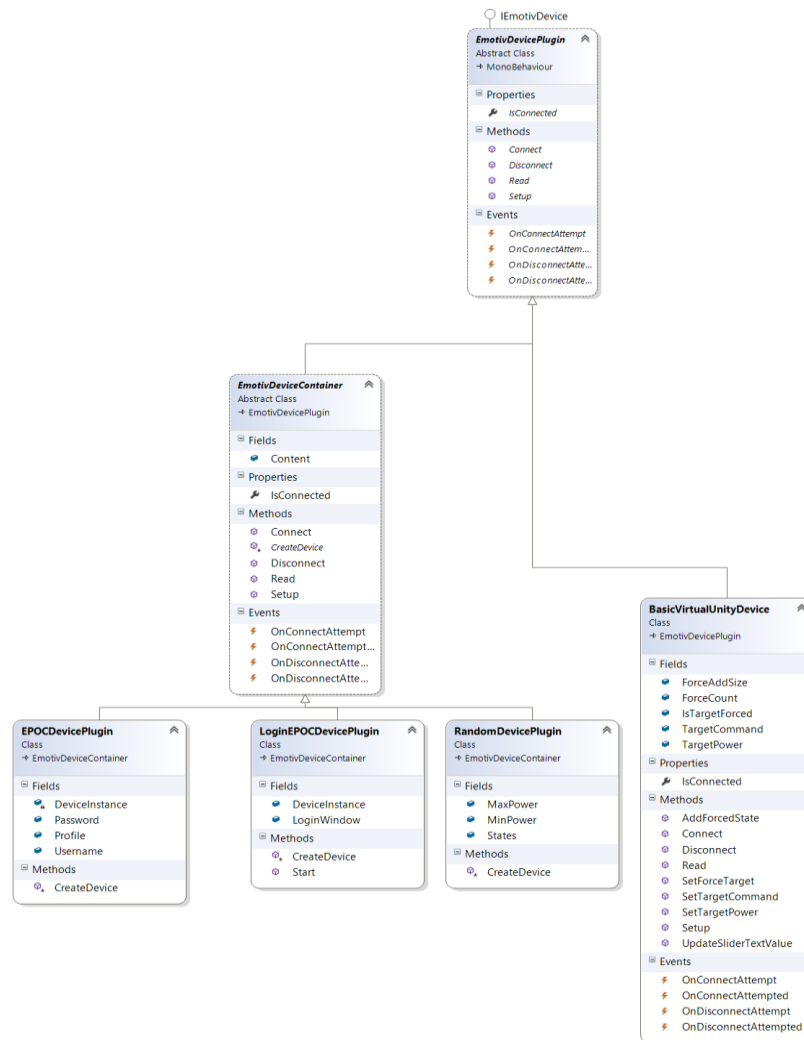
Name:	BackspaceText
Input:	None
Output:	None
Description:	Backspace last letter

Name:	Popped
Input:	None
Output:	None
Description:	Removes the last menu

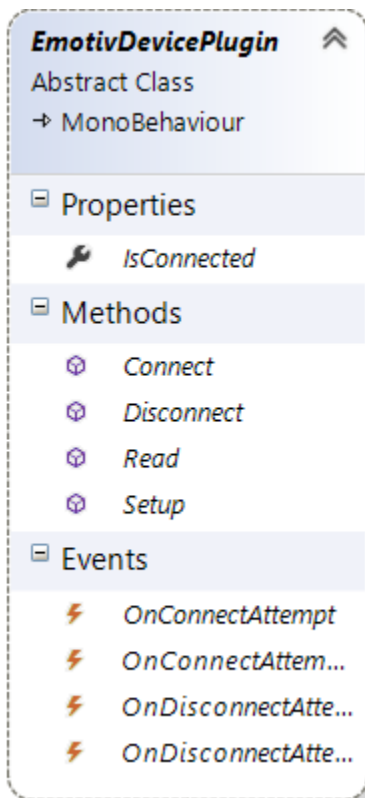
Name:	ResetSymbols
Input:	None
Output:	None
Description:	Reset back to default if the keyboard is exited while Symbols are up

Name:	ToggleSymbols
Input:	None
Output:	None
Description:	Brings up symbols

4.2 Device Reader



4.2. EmotivDevicePlugin



4.2.1.1. Attributes

Name	Access	Type	Description
IsConnected	Public	bool	Shows if device is connected

4.2.1.2. Methods

Name:	Connect
Input:	String
Output:	Bool
Description:	Connects to the device and returns if it was successful

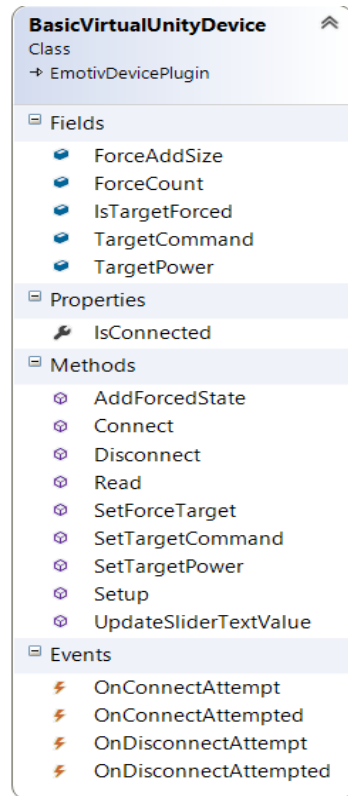
Name:	Disconnect
Input:	String
Output:	Bool
Description:	Disconnects the device and returns if it was successful

Name:	Read
Input:	Long
Output:	IEmotivState
Description:	Returns the most current state from the emotiv

Name:	Setup
Input:	None

Output:	None
Description:	Initializes the device

4.2.2. BasicVirtualUnityDevice



4.2.2.1. Attributes

Name	Access	Type	Description
------	--------	------	-------------

TargetCommandtype	Public	EmotivCommandType	Command the device will output
TargetPower	Public	Float	Power the device outputs
IsTargetForced	Public	bool	Forces device to output target command.
ForceAddSize	public	int	Number of commands given per interval
ForceCount	public	int	Number of forced commands

4.2.2.2. Methods

Name:	Connect
Input:	String
Output:	Bool
Description:	Connects to the device and returns if it was successful

Name:	Disconnect
Input:	String
Output:	Bool
Description:	Disconnects the device and returns if it was successful

Name:	Read
Input:	Long
Output:	IEmotivState
Description:	Returns the most current state from the emotiv

Name:	Setup
Input:	None
Output:	None
Description:	Initializes the device

Name:	SetTargetPower
Input:	Slider
Output:	None
Description:	Sets TagetPower through slider object

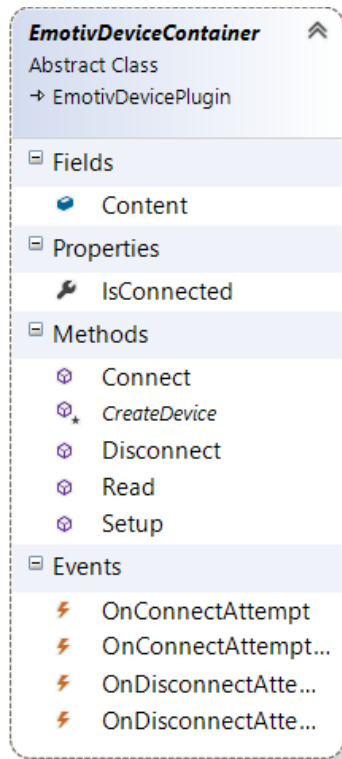
Name:	UpdateSliderTextValue
Input:	Text
Output:	None
Description:	Updates slider text value

Name:	SetTargetCommand
Input:	Dropdown
Output:	None
Description:	Updates target command

Name:	SetForceTarget
Input:	Toggle
Output:	None
Description:	Turns on or off forced states

Name:	AddForcedState
Input:	None
Output:	None
Description:	Updates forced count with new states

4.2.3. EmotivDeviceContainer



4.2.3.1. Attributes

Name	Access	Type	Description
Content	Public	IEmotivDevice	Emotiv Device

4.2.3.2. Methods

Name:	Connect
Input:	String
Output:	bool
Description:	Connects to the device and returns if it was successful

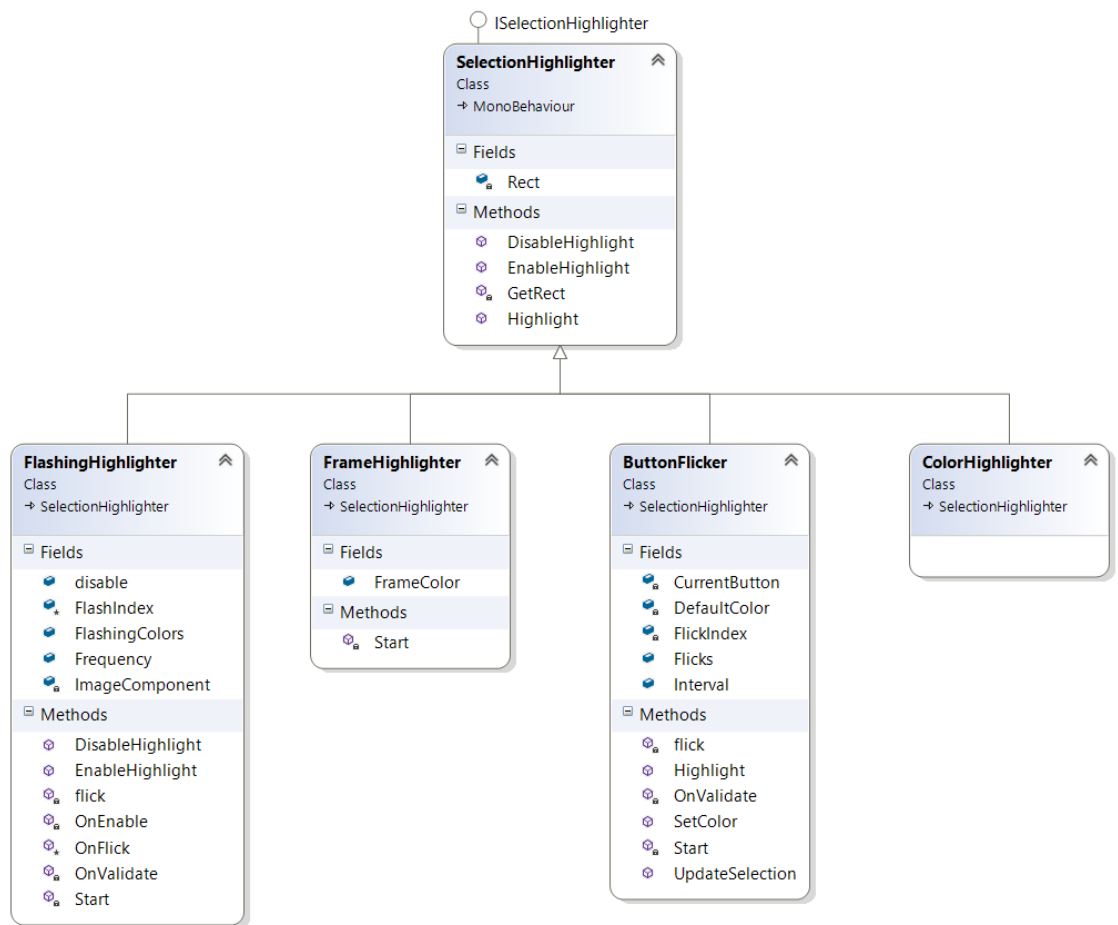
Name:	Disconnect
Input:	String
Output:	Bool
Description:	Disconnects the device and returns if it was successful

Name:	Read
Input:	Long
Output:	IEmotivState
Description:	Returns the most current state from the emotiv

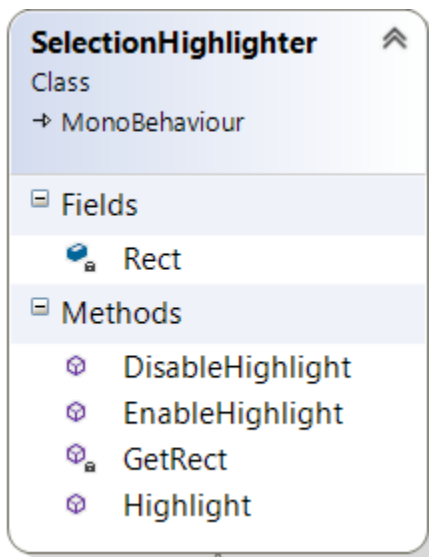
Name:	Setup
Input:	None

Output:	None
Description:	Initializes the device

4.3 Selection Highlighter



4.3. SelectionHighlighter



4.3.1.1. Attributes

Name	Access	Type	Description
Rect	Public	Rectangle	Rectangle that covers selected button

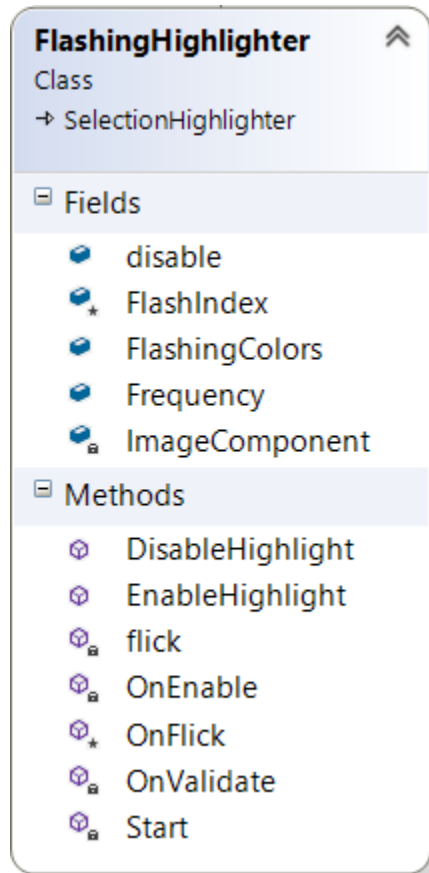
4.3.1.2. Methods

Name:	DisableHighlighter
Input:	None
Output:	None
Description:	Turns off highlighter

Name:	Highlight
Input:	GameObject
Output:	None
Description:	Highlight passed game object

Name:	EnableHighlighter
Input:	None
Output:	None
Description:	Turns on highlighter

4.3.2. FlashingHighlighter



4.3.2.1. Attributes

Name	Access	Type	Description
disable	Public	bool	True if highlighter is disabled
FlashIndex	Protected	int	Index of current gameobject

FlashColors	Public	Color[]	List of highlighter Colors
Frequency	Public	int	Hertz of the flashing highlighter
ImageComponent	Private	int	Highlighter Image

4.3.2.2. Methods

Name:	DisableHighlighter
Input:	None
Output:	None
Description:	Turns off highlighter

Name:	Start
Input:	None
Output:	None
Description:	Starts Flashing Highlighter

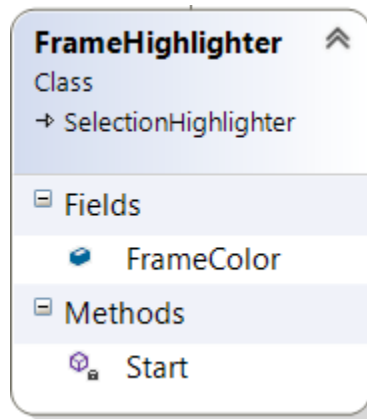
Name:	EnableHighlighter
Input:	None
Output:	None
Description:	Turns on highlighter

Name:	OnEnable
Input:	None
Output:	None
Description:	Starts highlight in coroutine

Name:	OnValidate
Input:	None
Output:	None
Description:	Checks that Highlighter is configured correctly

Name:	OnFlick
Input:	None
Output:	None
Description:	Flashes highlighter for a single iteration

4.3.3. FrameHighlighter



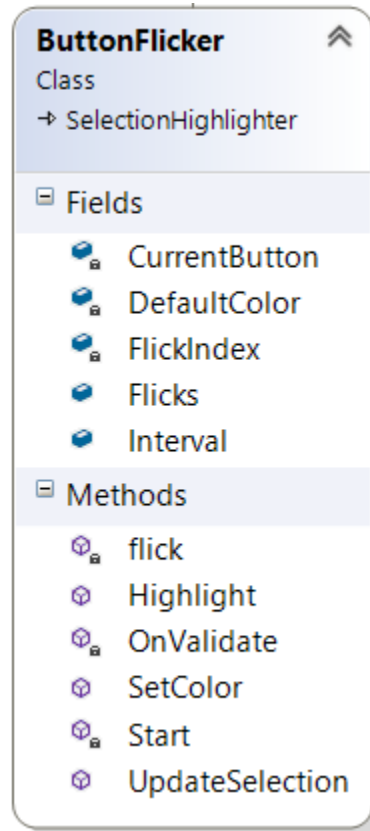
4.3.3.1. Attribute

Name	Access	Type	Description
FrameColor	Public	bool	True if highlighter is disabled

4.3.3.2. Methods

Name:	Start
Input:	None
Output:	None
Description:	Starts Flashing Highlighter

4.3.4. ButtonFlicker



4.3.4.1. Attributes

Name	Access	Type	Description
CurrentButton	Private	Button	Current Button Selected
DefaultColor	private	Color	Default Flicker Color

FlickIndex	private	int	Button Index
Flicks	public	Color[]	List of button colors
Intervals	public	int	Time between button rotations

4.3.4.2. Methods

Name:	Start
Input:	None
Output:	None
Description:	Starts Flashing Highlighter

Name:	Flick
Input:	None
Output:	IEnumerator
Description:	Flickers the button

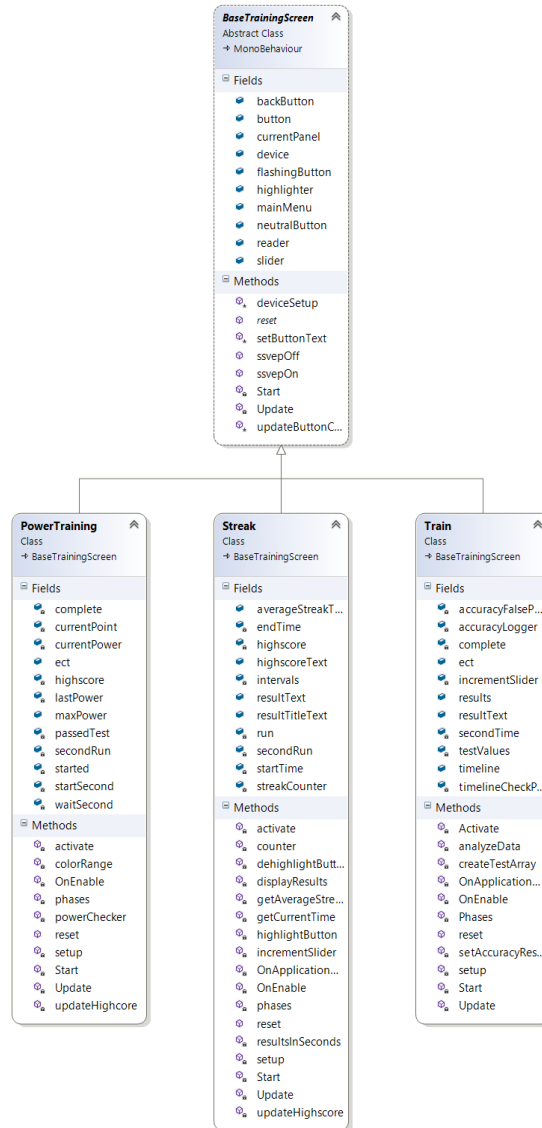
Name:	Highlight
Input:	GameObject
Output:	None
Description:	Highlights the passed GameObject

Name:	SetColor
Input:	Color
Output:	None
Description:	Sets the color of the button

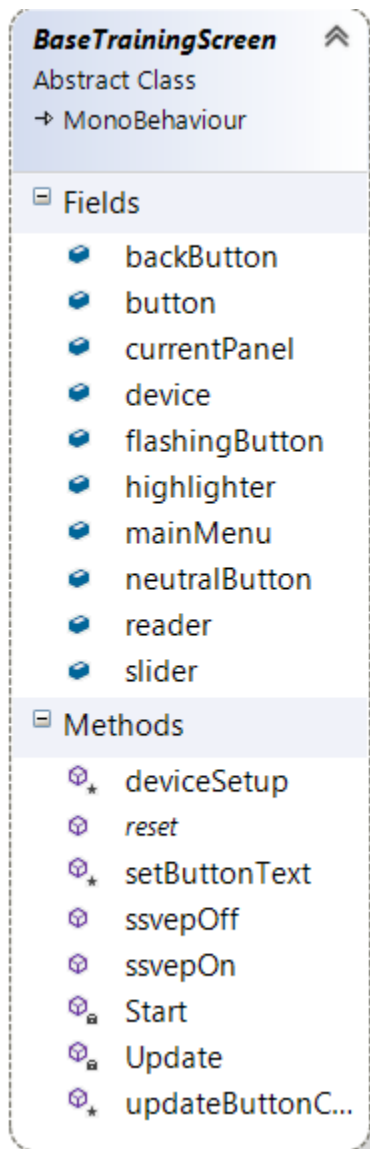
Name:	OnValidate
Input:	None
Output:	None
Description:	Checks that Highlighter is configured correctly

Name:	UpdateSelection
Input:	None
Output:	None
Description:	Updates current selection

4.4 Training



4.4. BaseTraining



4.4.1.1. Attributes

Name	Access	Type	Description
------	--------	------	-------------

device	Public	IEmotivDevice	Emotiv device object
reader	public	IEmotivReader	Reads from emotiv device
slider	public	Slider	Slider to track time
button	public	Button	Generic Button for training
backButton	public	Button	Button used to go back to previous screen.
highlighter	public	SelectionHighlighter	Highlights passed buttons
flashingButton	public	Button	Flashes when SSVEP is selected
neutralButton	public	Button	Default button for neutral
Mainmenu	public	GameObject	Holds Previous Screen
currentPanel	public	GameObject	Holds Current Screen

4.4.1.2. Methods

Name:	Start
Input:	None
Output:	None
Description:	Use this for initialization

Name:	Update
Input:	None
Output:	None

Description:	Update is called once per frame
---------------------	---------------------------------

Name:	deviceSetup
Input:	int
Output:	None
Description:	Initializes the device

Name:	updateButtonColor
Input:	Color
Output:	None
Description:	Updates the color of the main button

Name:	setButtonText
Input:	String
Output:	None
Description:	Updates the text of the main button

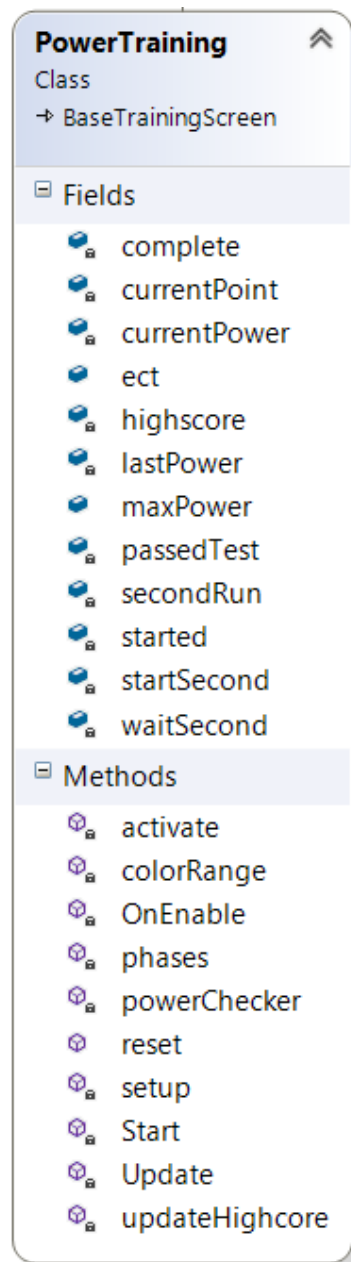
Name:	ssvepOff
Input:	None

Output:	None
Description:	Disables SSVEP and Neutral Buttons

Name:	ssvepOn
Input:	None
Output:	None
Description:	Enables SSVEP and Neutral Buttons

Name:	reset
Input:	None
Output:	None
Description:	Resets all components of the test to run again

4.4.2. PowerTraining



4.4.2.1. Attributes

Name	Access	Type	Description
ect	Public	float[]	Holds emotiv states
currentPoint	public	int	Current place int ect array
currentPower	public	float	Power of the last emotiv state
highscore	public	float	Highest power acheived
startSecond	public	float	Start time of current streak
passedTest	public	boolean	If last power goal was reached
started	public	boolean	If test has offically started
secondRun	public	boolean	If test has been run before
waitSecond	public	boolean	Must have consecutive pushes
maxPower	public	text	Display highest power to the user

4.4.2.2. Methods

Name:	Start
Input:	None
Output:	None
Description:	Start the power training

Name:	OnEnable
Input:	None
Output:	None
Description:	Start the power training after reset

Name:	Update
Input:	None
Output:	None
Description:	Updates display every frame

Name:	updateHighscore
Input:	None
Output:	None
Description:	Updates the highscore

Name:	activate
Input:	None
Output:	IEnumerator
Description:	Runs the training is a coroutine

Name:	phases
Input:	None
Output:	None
Description:	Initializes timing slider and reader

Name:	powerChecker
Input:	IEmotivState
Output:	None
Description:	Takes the current emotiv state and checks current score is surpassed for one second.

Name:	colorRange
Input:	float
Output:	None
Description:	Changes button color based off of current power

Name:	reset
Input:	None
Output:	None

Description:	Resets all variables for next test
---------------------	------------------------------------

Name:	setup
Input:	None
Output:	None
Description:	Initializes all variables at the start of a training

4.4.3. Streak

Streak	
Class	
→ BaseTrainingScreen	
Fields	
averageStreakT...	
endTime	
highscore	
highscoreText	
intervals	
resultText	
resultTitleText	
run	
secondRun	
startTime	
streakCounter	
Methods	
activate	
counter	
dehighlightButt...	
displayResults	
getAverageStre...	
getCurrentTime	
highlightButton	
incrementSlider	
OnApplication...	
OnEnable	
phases	
reset	
resultsInSeconds	
setup	
Start	
Update	
updateHighscore	

4.4.3.1. Attributes

Name	Access	Type	Description
resultText	Public	test	Test for results
highscoreText	public	text	Displays the highscore
resultTitleText	public	text	Displays the results title
averageStreakText	public	text	Displays the average streak length
streakCounter	private	int	Length of current streak
highscore	private	long	Longest streak recorded
startTime	private	long	Start time of streak
endTime	private	long	End time of streak
run	private	boolean	Used to delay start of training
secondRun	private	boolean	Used on second run of test
intervals	private	arraylist	Stores intervals of streaks

4.4.3.2. Methods

Name:	start
Input:	None
Output:	None
Description:	Starts training

Name:	onEnable
Input:	None
Output:	None
Description:	Starts training on second run

Name:	OnApplicationQuit
Input:	None
Output:	None
Description:	Stops reader at end of training

Name:	Update
Input:	None
Output:	None
Description:	Updates training every frame

Name:	activate
Input:	None
Output:	None
Description:	Runs training in coroutine

Name:	phases
Input:	None
Output:	IEnumerator
Description:	Stops reader at end of training

Name:	counter
Input:	IEmotivState
Output:	None
Description:	Counts consecutive push states

Name:	resultsInSeconds
Input:	long
Output:	String
Description:	Converts long to time in string format

Name:	incrementSlider
Input:	None
Output:	None
Description:	Increments sliders by the time that passes

Name:	updateHighscore
Input:	None
Output:	None
Description:	Updates highscore with new highscore

Name:	getCurrentTime
Input:	None
Output:	long
Description:	Gets current streak time

Name:	displayResults
Input:	None
Output:	None
Description:	Displays the results to the user

Name:	getAverageStreak
Input:	None
Output:	long
Description:	Returns average streak length

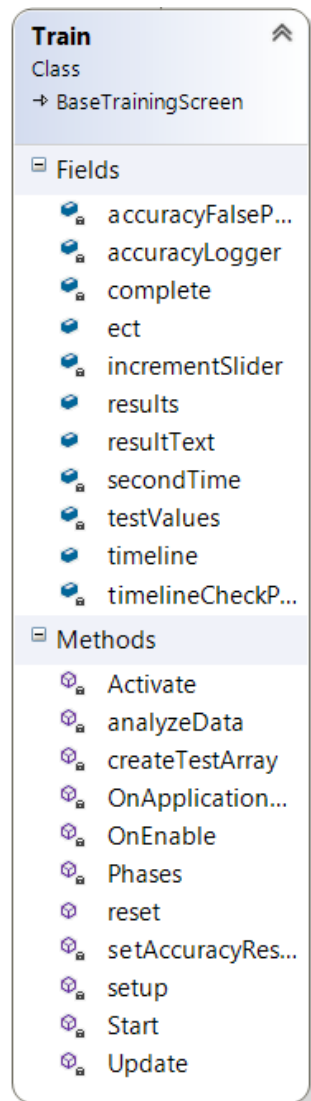
Name:	highlightButton
Input:	None
Output:	None
Description:	Highlights the button

Name:	dehighlightButton
Input:	None
Output:	None
Description:	Dehighlights the button

Name:	reset
Input:	None
Output:	None
Description:	Resets the variables at the end of the training

Name:	setup
Input:	None
Output:	None
Description:	Initializes the training

4.4.4. Train



4.4.4.1. Attributes

Name	Access	Type	Description
resultText	Public	test	Test for results
highscoreText	public	text	Displays the highscore
resultTitleText	public	text	Displays the results title
averageStreakText	public	text	Displays the average streak length
streakCounter	private	int	Length of current streak
highscore	private	long	Longest streak recorded
startTime	private	long	Start time of streak
endTime	private	long	End time of streak
run	private	boolean	Used to delay start of training
secondRun	private	boolean	Used on second run of test

4.4.4.2. Methods

Name:	Start
Input:	None
Output:	None
Description:	Starts the Training

Name:	OnEnable
Input:	None

Output:	None
Description:	Starts the training after the first run

Name:	createTestArray
Input:	None
Output:	None
Description:	Creates array of commands in random order

Name:	Update
Input:	None
Output:	None
Description:	Updates training every frame

Name:	OnApplicationQuit
Input:	None
Output:	None
Description:	Turns off reader on applications end

Name:	Activate
Input:	None

Output:	None
Description:	Runs training in coroutine

Name:	Phases
Input:	None
Output:	None
Description:	Core training code

Name:	setAccuracyResultText
Input:	Int[], int[]
Output:	None
Description:	Displays results from training

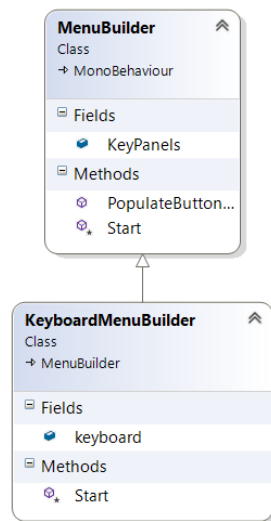
Name:	analyzeData
Input:	None
Output:	int[]
Description:	Processes the emotiv states collected from the training

Name:	reset
--------------	-------

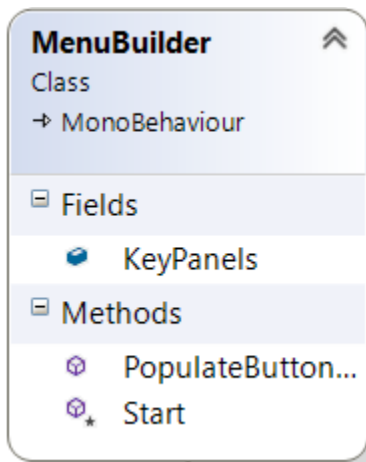
Input:	None
Output:	None
Description:	Resets the variables at the end of the training

Name:	Setup
Input:	None
Output:	None
Description:	Initializes variables at the beginning of the training

4.5 MenuBuilder



4.5. MenuBuilder



4.5.1.1. Attributes

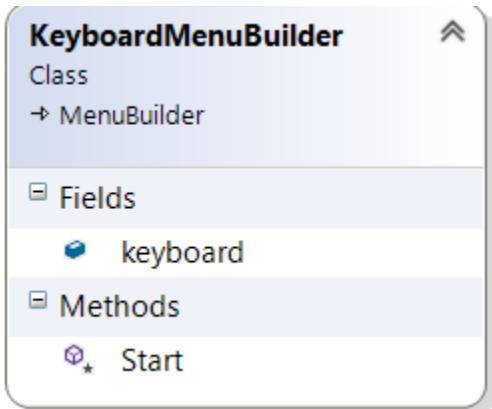
Name	Access	Type	Description
KeyPanels	Public	List<GameObject>	Holds objects on Menu

4.5.1.2. Methods

Name:	PopulateButtonsFromFile
Input:	String
Output:	None
Description:	Reads in a JSON file and begin populating the menu

Name:	PopulateButton
Input:	Button[], JSONArray, int
Output:	int
Description:	The index of the last child button that was set

4.5.2. KeyboardMenuBuilder



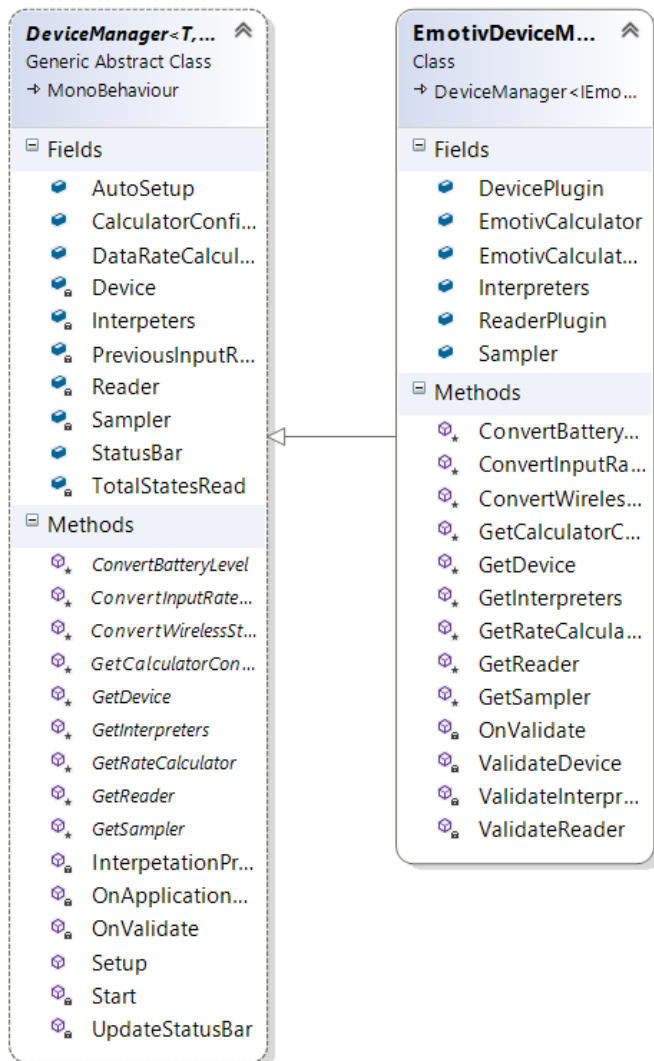
4.5.2.1. Attributes

Name	Access	Type	Description
keyboard	Public	List<GameObject>	Holds objects on Menu

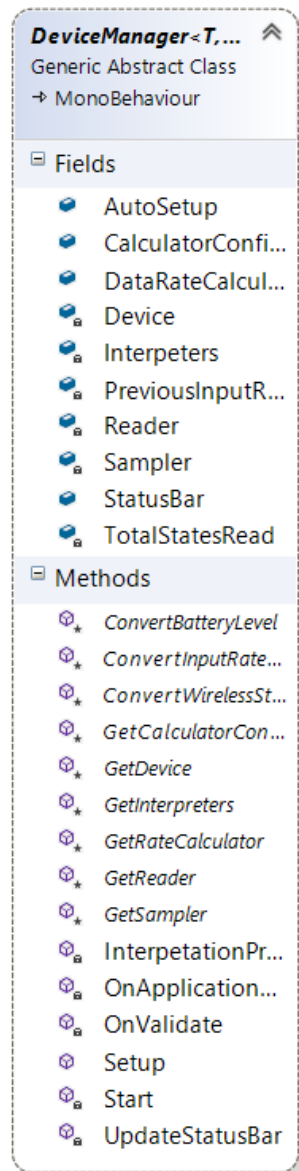
4.5.2.2.

Name:	Start
Input:	None
Output:	None
Description:	Starts and initializes keyboard

Device Manager



4.5.3. DeviceManager



4.5.3.1. Attributes

Name	Access	Type	Description
AutoSetup	public	Boolean	Use Default Config
StatusBar	public	DeviceStatusBar	Holds StatudBar which displays InputRate, BatteryLevel, and WirelessSignal.
DataRateCalculator	public	DataRateCalculator<T, C>	Holds Rate calculator which algorithm for rating sampled data
CalculatorConfiguratio	public	Generic	Holds calculator configuration
Device	private	ICnectomeDevice<T>	Holds device
Reader	private	ICnectomeReader<T>	Holds reader
Sampler	private	DataSampler<T>	Holds read data and samples it for interpreters
Interpreters	private	DataInterpreter<T>[]	Holds interpreters to be invoked
TotalStatesRead	private	int	Counts total states read
PreviousInputRate	private	int	Holds previously Recognised total states. Used to calculate input rate.

4.5.3.2. Methods

Name:	Setup
Input:	None
Output:	None
Description:	Sets up Device, Reader, and Sampler then start interpreters coroutine

Name:	Start
Input:	None
Output:	None
Description:	Auto runs Manager if AutoSetup is checked

Name:	OnApplicationQuit
Input:	None
Output:	None
Description:	Insures reader is disabled

Name:	InterpetationProcess
Input:	None
Output:	IEnumerator
Description:	Coroutine that Interpret every interpreter with an allocated sample.

Name:	UpdateStatusBar
Input:	None
Output:	IEnumerator

Description:	Updates DeviceStatusBar at a lower rate than Interpreters
---------------------	---

Name:	GetDevice
Input:	None
Output:	ICollectomeDevice<T>
Description:	Returns an initialized Device

Name:	GetReader
Input:	None
Output:	ICollectomeReader<T>
Description:	Gets an initialized Reader

Name:	GetInterpreters
Input:	None
Output:	DataInterpreter<T>[]
Description:	Gets an initialized Interpreters

Name:	GetRateCalculator
Input:	None

Output:	DataInterpreter<T>[]
Description:	Gets an initialized RateCalculator

Name:	GetCalculatorConfiguration
Input:	None
Output:	RateCalculatorConfiguration
Description:	Gets an initialized CalculatorConfiguration

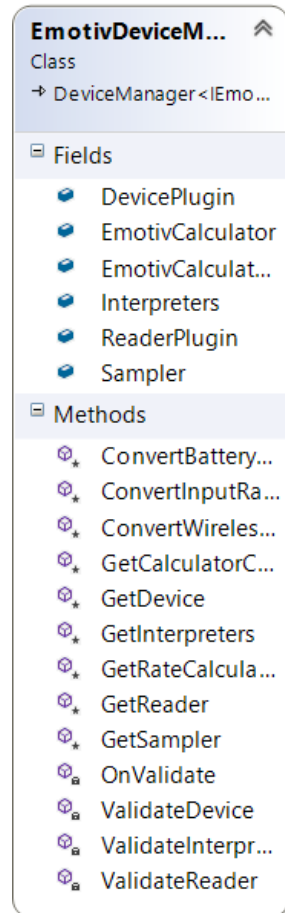
Name:	ConvertWirelessStrength
Input:	int
Output:	WirelessSignalStrengthLevel
Description:	Defines how to convert Device's wireless signal

Name:	ConvertBatteryLevel
Input:	int
Output:	BatteryLevel
Description:	Defines how to convert Device's battery level

Name:	ConvertInputRateLevel
--------------	-----------------------

Input:	int
Output:	InputRateLevel
Description:	Defines how to convert Device's input rate level

4.5.4. EmotivDeviceManager



4.5.4.1. Attributes

Name	Access	Type	Description
DevicePlugin	public	EmotivDevicePlugin	Hold Device
ReaderPlugin	public	EmotivReaderPlugin	Holds Redaer

Sampler	public	EmotivSampler	Holds Samplers with creates samples to interpret
EmotivCalculator	public	EmotivRateCalculator	Holds data calculator
EmotivCalculatorConfiguration	public	EmotivCalculatorConfiguration	Holds calculator configuration
Interpreters	public	EmotivInterpreter[]	Holds data sample interpreters

4.5.4.2. Methods

Name:	GetDevice
Input:	None
Output:	ICnectomeDevice<T>
Description:	Returns an initialized Device

Name:	GetReader
Input:	None
Output:	ICnectomeReader<T>
Description:	Gets an initialized Reader

Name:	GetInterpreters
Input:	None

Output:	DataInterpreter<T>[]
Description:	Gets an initialized Interpreters

Name:	GetRateCalculator
Input:	None
Output:	DataInterpreter<T>[]
Description:	Gets an initialized RateCalculator

Name:	GetCalculatorConfiguration
Input:	None
Output:	RateCalculatorConfiguration
Description:	Gets an initialized CalculatorConfiguration

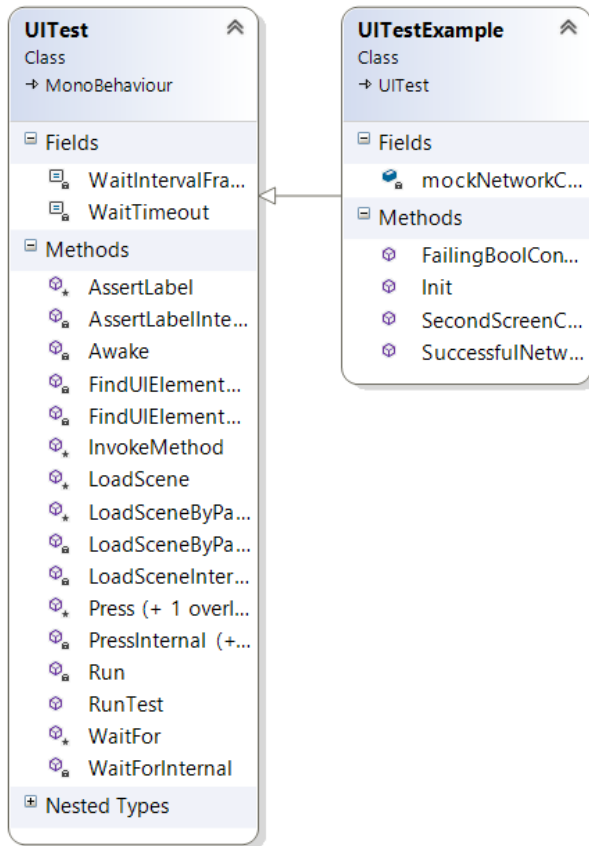
Name:	ConvertWirelessStrength
Input:	int
Output:	WirelessSignalStrengthLevel
Description:	Defines how to convert Device's wireless signal

Name:	ConvertBatteryLevel
--------------	---------------------

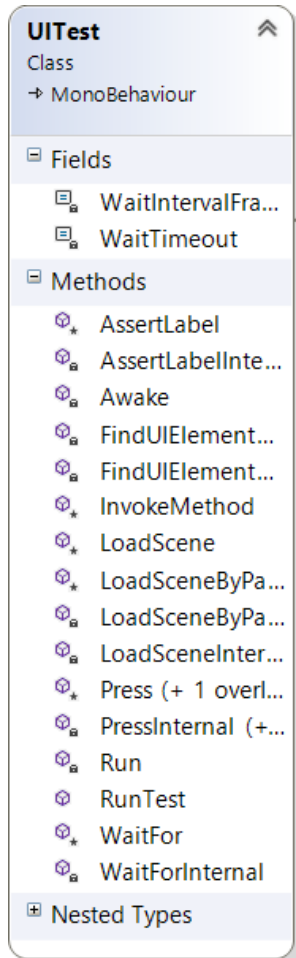
Input:	int
Output:	BatteryLevel
Description:	Defines how to convert Device's battery level

Name:	ConvertInputRateLevel
Input:	int
Output:	InputRateLevel
Description:	Defines how to convert Device's input rate level

4.6 UITest



4.6. UITest



4.6.1.1. Attrubutes

Name	Access	Type	Description
WaitTimeout	private	float	Time before failure
WaitIntervalFrames	private	float	Amount of frames to wait

4.6.1.2. Methods

Name:	Awake
Input:	None
Output:	None
Description:	Stops cleanup of class

Name:	RunTest
Input:	MethodInfo
Output:	IEnumerator
Description:	Starts test's coroutines

Name:	InvokeMethod
Input:	MethodInfo
Output:	IEnumerator
Description:	Passes class to passed method

Name:	Run
Input:	Type

Output:	IEnumerator
Description:	Runs methods in passed type

Name:	WaitFor
Input:	Condition
Output:	Coroutine
Description:	Pauses coroutine till condition is met

Name:	LoadScene
Input:	string
Output:	Coroutine
Description:	Loads scene with string name

Name:	LoadSceneByPath
Input:	string
Output:	Coroutine
Description:	Loads scene with string path

Name:	AssertLabel
--------------	-------------

Input:	String, String
Output:	Coroutine
Description:	Sets internal labels

Name:	FindUIElement
Input:	string
Output:	generic
Description:	Finds element by name

Name:	FindUIElementOrNull
Input:	string
Output:	generic
Description:	Finds element by name or returns null

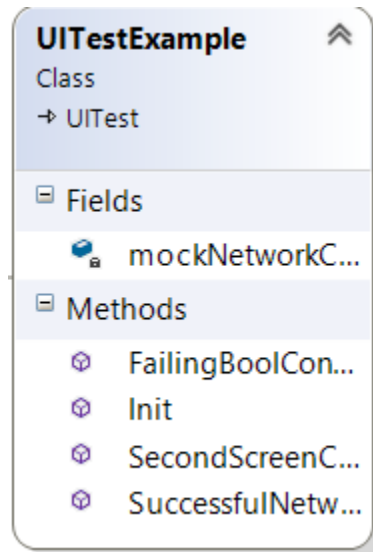
Name:	Press
Input:	string
Output:	Coroutine
Description:	Presses button with name

Name:	Press
Input:	GameObject
Output:	Coroutine
Description:	Presses passed GameObject

Name:	WaitForInternal
Input:	Condition, string
Output:	IEnumerator
Description:	Pause process for waitinterval

Name:	WaitForInternal
Input:	Condition, string
Output:	IEnumerator
Description:	Pause process for waitinterval

4.6.2. UITestExample



4.6.2.1. Attributes

Name	Access	Type	Description
mockNetworkClient	private	MockNetworkClient	Fake network client

4.6.2.2. Methods

Name:	Init
Input:	None
Output:	IEnumerator

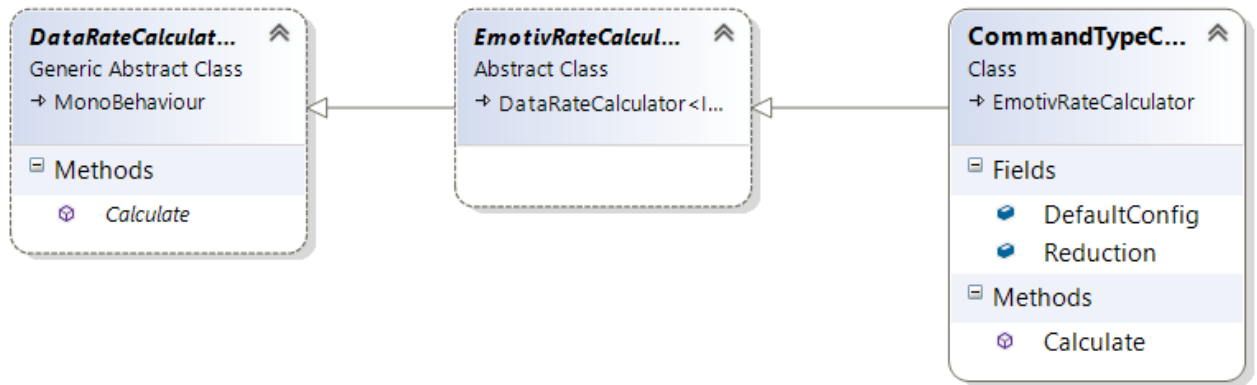
Description:	Initializes Mock Client
---------------------	-------------------------

Name:	SecondScreenCanBeOpenedFromTheFirstOne
Input:	None
Output:	IEnumerator
Description:	Opens new scene and posts a comment

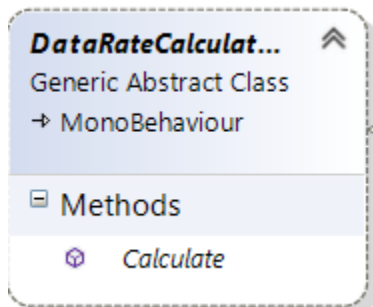
Name:	SuccessfulNetworkResponseIsDisplayedOnTheFirstScreen
Input:	None
Output:	IEnumerator
Description:	Displays network response on first scene

Name:	FailingBoolCondition
Input:	None
Output:	IEnumerator
Description:	Checks if condition fails

4.7 Calculator



4.7. DataRateCalculator



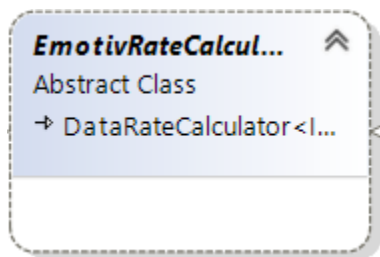
4.7.1.1. Attributes

Name	Access	Type	Description
None	None	None	None

4.7.1.2. Methods

Name:	Calculate
Input:	IEnumerable<T>, Generic
Output:	float
Description:	Calculates sampled data

4.7.2. EmotivRateCalculator



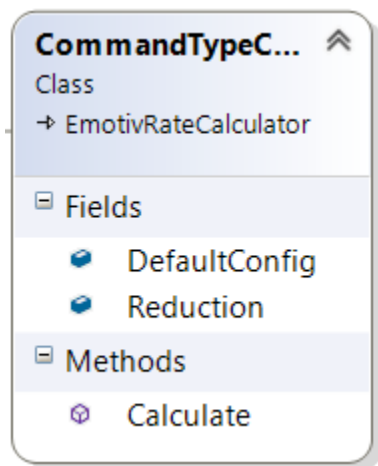
4.7.2.1. Attribute

Name	Access	Type	Description
None	None	None	None

4.7.2.2. Methods

Name:	None
Input:	None
Output:	None
Description:	None

4.7.3. CommandTypeCalculator



4.7.3.1. Attribute

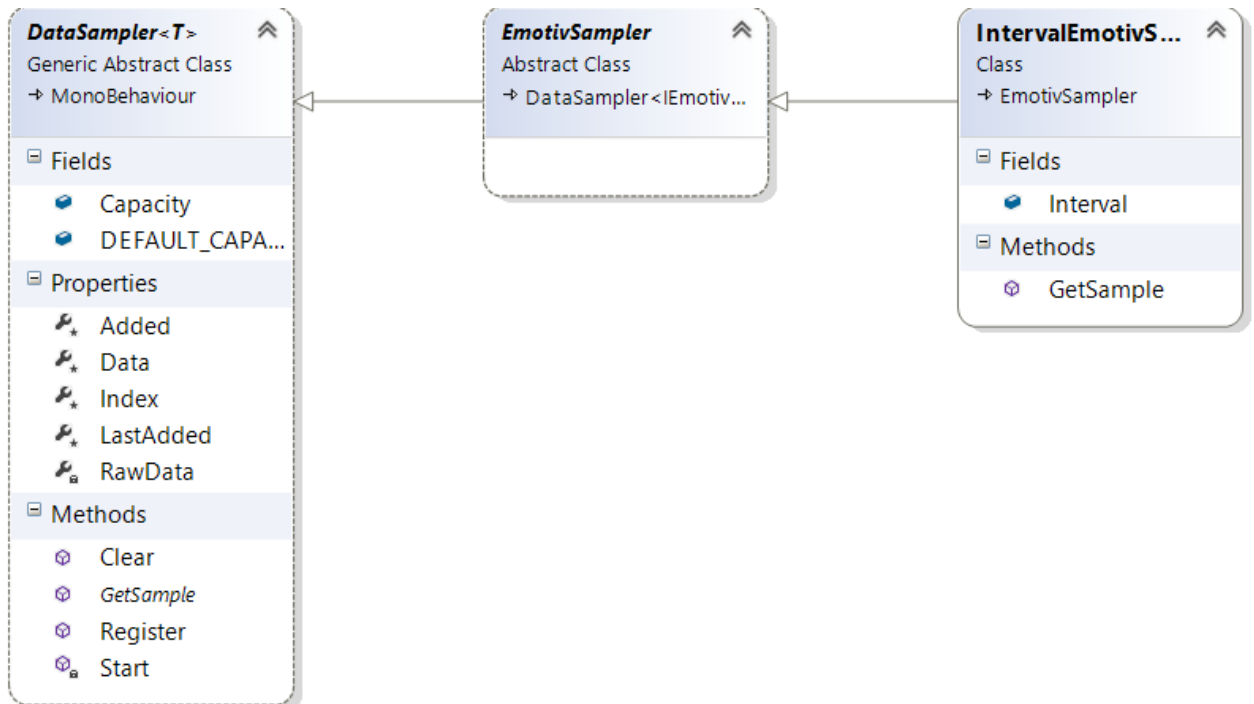
Name	Access	Type	Description
DefaultConfig	public	EmotivCalculatorConfigur	Default config

		ation	
Reduction	public	boolean	Enables and disabled rate reduction for samples below minimum size.

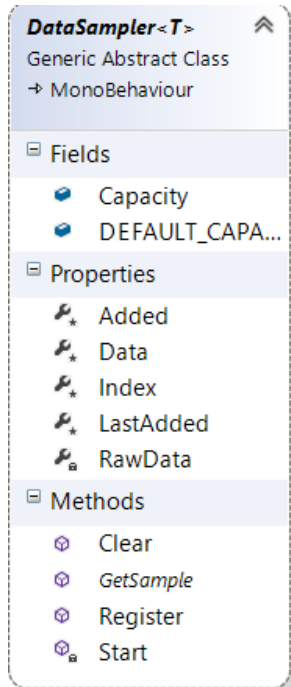
4.7.3.2. **Methods**

Name:	Calculate
Input:	IEnumerable<IEmotivState>, EmotivCalculatorConfiguration
Output:	float
Description:	Returns click percentage from data passed

4.8 DataSampler



4.8. DataSampler



4.8.1.1. Attributes

Name	Access	Type	Description
DEFAULT_CAPACITY	public	int	Default capacity
Capacity	public	int	Maximum amount of data held

4.8.1.2. Methods

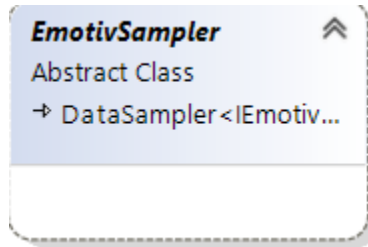
Name:	Start
-------	-------

Input:	None
Output:	None
Description:	Prepares Data entry

Name:	Register
Input:	Generic
Output:	None
Description:	Add data to list

Name:	Clear
Input:	None
Output:	None
Description:	Clear recorded data

4.8.2. EmotivSampler



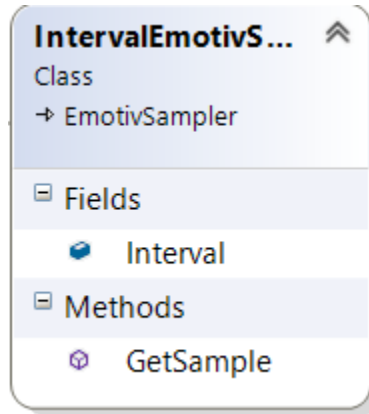
4.8.2.1. Attributes

Name	Access	Type	Description
None	None	None	None

4.8.2.2. Methods

Name:	None
Input:	None
Output:	None
Description:	None

4.8.3. IntervalEmotivSampler



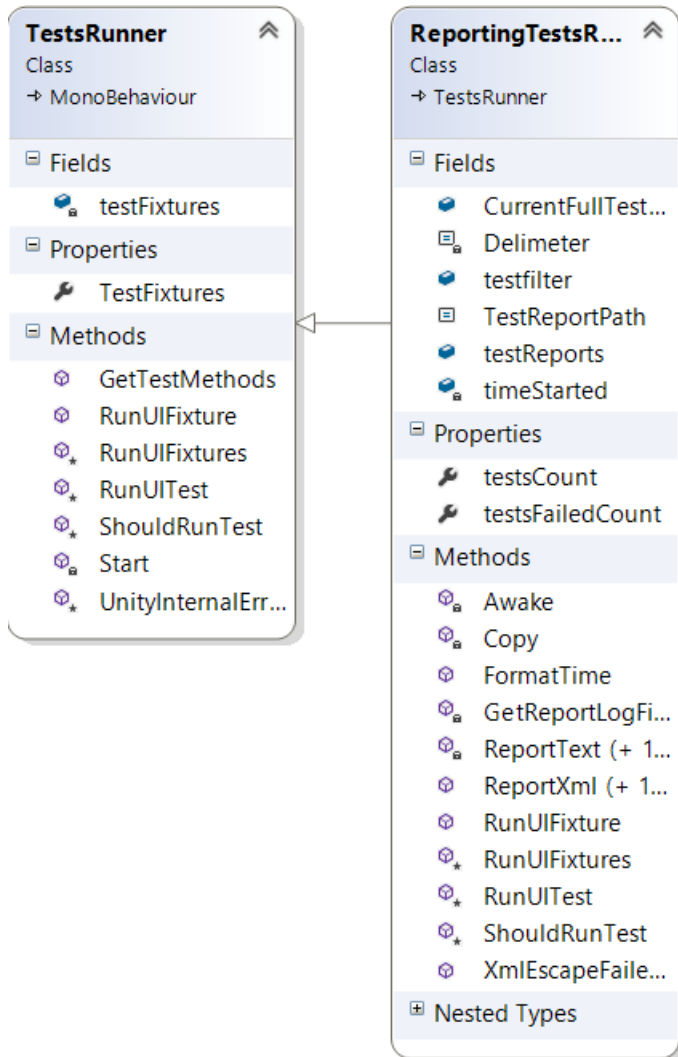
4.8.3.1. Attributes

Name	Access	Type	Description
Interval	public	int	Sampling Interval

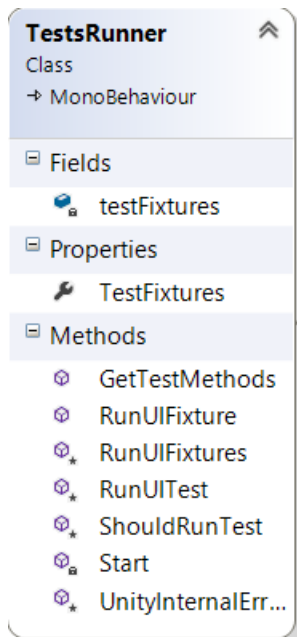
4.8.3.2. Methods

Name:	GetSample
Input:	None
Output:	IEnumerable<IEmotivState>
Description:	Samples latest interval.

4.9 TestRunner



4.9. TestRunner



4.9.1.1. Attributes

Name	Access	Type	Description
testFixtures	public	List<Type>	Holds objects to test

4.9.1.2. Methods

Name:	Start
Input:	None
Output:	IEnumerable<IEmotivState>
Description:	Starts tests

Name:	RunUIFixtures
Input:	None
Output:	None
Description:	Runs fixtures for each object

Name:	RunUIFixture
Input:	Type, MethodInfo[]
Output:	IEnumerable<IEmotivState>
Description:	Starts tests

Name:	GetTestMethods
Input:	FixtureType
Output:	MethodInfo[]

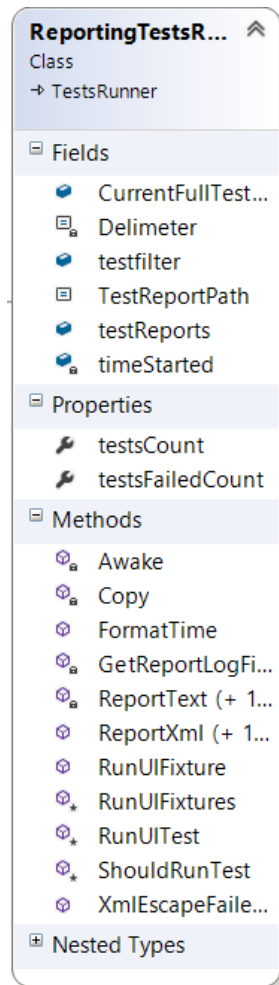
Description:	Gets Methods to test for Fixtures
---------------------	-----------------------------------

Name:	ShouldRunTest
Input:	String
Output:	IEnumerator
Description:	Returns a boolean

Name:	RunUITest
Input:	Type, MethodInfo
Output:	IEnumerator
Description:	Test each fixture passed

Name:	UnityInternalError
Input:	String
Output:	bool
Description:	Returns a Boolean if condition is thrown

4.9.2. ReportingTestsRunner



4.9.2.1. Attributes

Name	Access	Type	Description
------	--------	------	-------------

TestReportPath	public	String	Holds or
Delimiter	private	String	Const Delimiter
testfilter	public	String	Test filter string
testReports	public	Dictionary<Type, List<TestReport>>	Holds test reports
testsFailedCount	public	int	Number of tests failed
testsCount	public	int	Number of test
CurrentFullTestName	public	String	Full name of the test
timeStarted	public	DateTime	Hold time when report is started

4.9.2.2. Methods

Name:	Awoke
Input:	None
Output:	None
Description:	Reinitializes test when activated

Name:	RunUIFixtures
Input:	None
Output:	IEnumerator
Description:	Runs test for all Fixtures

Name:	RunUIFixtures
Input:	Type, MethodInfo[]
Output:	IEnumerator
Description:	Runs test for passed fixtures

Name:	ReportXML
Input:	Type, DateTime, DateTime
Output:	None
Description:	Generates XML Report

Name:	ReportXML
Input:	Type, TestReport, StreamWriter
Output:	None
Description:	Generates XML Report

Name:	Copy
Input:	StreamReader, StreamWriter
Output:	long
Description:	Readers target file and writes it to a new file

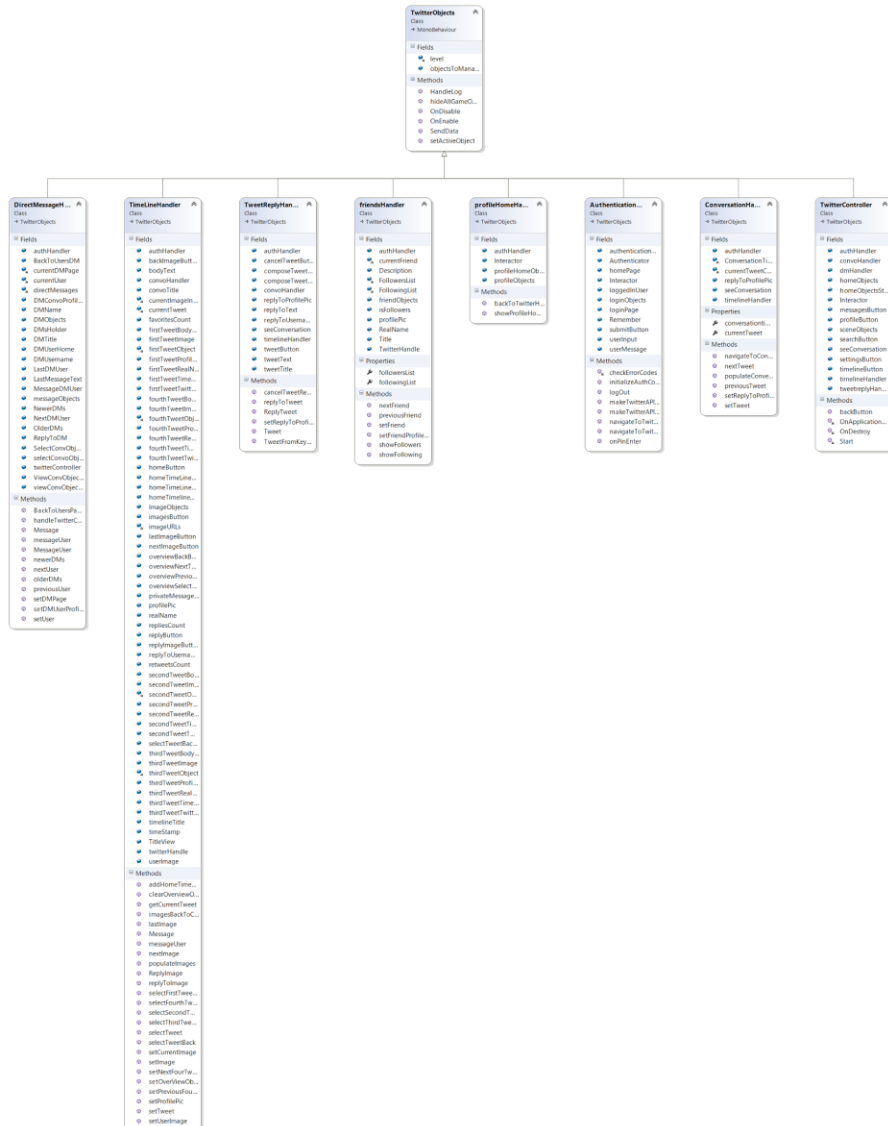
Name:	GetReportLogFilePath
Input:	String
Output:	String
Description:	Returns path to log file

Name:	ReportText
Input:	None
Output:	None
Description:	Prints out report text to debug stream

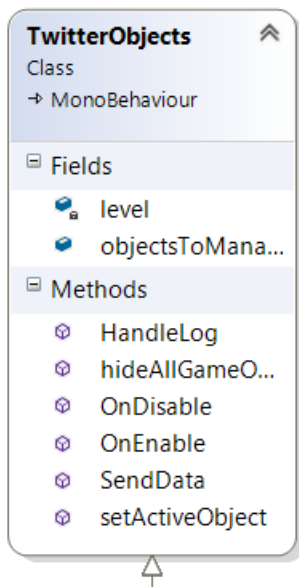
Name:	ReportText
Input:	Type, List<TestReport>, StringBuilder
Output:	None
Description:	Prints out report text to debug stream

Name:	ShouldRunTest
Input:	Type, List<TestReport>, StringBuilder
Output:	None
Description:	Prints out report text to debug stream

4.10 Twitter



4.10. TwitterObjects



4.10.1.1. Attributes

Name	Access	Type	Description
levels	public	String	Create a string to store log level in
objectsToManage	public	String[]	Contains names of object types

4.10.1.2. Methods

Name:	OnEnable
Input:	None
Output:	None
Description:	Register the HandleLog function on scene start to fire on debug.log events

Name:	OnDisable
Input:	None
Output:	None
Description:	Remove callback when object goes out of scope

Name:	HandleLog
Input:	String, String, LogType
Output:	None
Description:	Capture debug.log output, send logs to Loggly

Name:	SendData
Input:	String, String, String
Output:	IEnumerator

Description:	Uses Access token to connect to twitter
---------------------	---

Name:	setActiveObject
Input:	String
Output:	None
Description:	Changes object status to active

Name:	hideAllGameObjects
Input:	Nones
Output:	None
Description:	Hides all game objects

4.10.2. DirectMessageHandler

DirectMessageH...	
Class	
↳ TwitterObjects	
Fields	
authHandler	
BackToUsersDM	
currentDMPage	
currentUser	
directMessages	
DMConvoProfil...	
DMName	
DMSObjects	
DMSHolder	
DMTitle	
DMUserHome	
DMUsername	
LastDMUser	
LastMessageText	
MessageDMUser	
messageObjects	
NewerDMs	
NextDMUser	
OlderDMs	
ReplyToDM	
SelectConvObj...	
selectConvoObj...	
twitterController	
ViewConvObjec...	
viewConvObjec...	
Methods	
BackToUsersPa...	
handleTwitterC...	
Message	
messageUser	
MessageUser	
newerDMs	
nextUser	
olderDMs	
previousUser	
setDMPage	
setDMUserProfi...	
setUser	

4.10.2.1. Attributes

Name	Access	Type	Description
DMObjects	public	GameObject	Holds Custom DM Unity Object
SelectConvObjects	public	GameObject	Current Conversation Game Object
ViewConvObjects	public	GameObject	Current Convo Game Object shown
DMConvoProfilePic	public	Image	Profile Picture
DMName	public	Text	Other Person's name
DMUsername	public	Text	Other Person's Username
LastMessageText	public	Text	Last Message received
LastDMUser	public	Button	Button directing to previous user messaged
NextDMUser	public	Button	Button directing next person in list to DM
MessageDMUser	public	Button	Button directing to message current user
DMUserHome	public	Button	Returns to previous screen
DMsHolder	public	GameObject	Holds DM information
NewerDMs	public	Button	Loads newer DMs
OlderDMs	public	Button	Loads older DMs
ReplyToDM	public	Button	Loads Keyboard to reply
BackToUsersDM	public	Button	Returns to User DM Screen
DMTitle	public	Text	Title to DM page

messageObjects	public	List<GameObject>	List of Game Objects needed to message
directMessages	private	List<DirectMessage>	List of DMs in a chat
CurrentUser	private	currentUser	User who's messages are loaded
currentDMPage	private	int	Current page of DMs
authHandler	public	AuthenticationHandler	Handles twitter api authentication
twitterController	public	TwitterController	Handles possible twitter interactions
selectConvoObjectsString	public	String	Current ConvoObject selected name
viewConvObjectsString	public	String	Current ConvoObject view name

4.10.2.2. Methods

Name:	handleTwitterConversations
Input:	None
Output:	None
Description:	Make the menu displaying all conversations visible. This is a list of users and their most recent message

Name:	setUser
Input:	User
Output:	None
Description:	This function sets the current DM user

Name:	setDMUserProfilePic
Input:	String
Output:	IEnumerator
Description:	Loads User's profile picture

Name:	nextUser
Input:	None
Output:	None
Description:	This function populates the DM users page with the next oldest user.

Name:	previousUser
Input:	None
Output:	None
Description:	This function populates the DM users page with the next newest user.

Name:	messageUser
Input:	None
Output:	None
Description:	Populate the screen to view the entire conversation with a user.

Name:	setDMPage
Input:	None
Output:	None
Description:	Populate the screen with the current set of messages

Name:	BackToUsersPage
Input:	None
Output:	None
Description:	Go back from the conversation page to the dm users page

Name:	olderDMs
Input:	None
Output:	None
Description:	On click object to get the next page of older dms

Name:	newerDMs
Input:	None
Output:	None
Description:	On click object to get the next page of newer dms

Name:	Message
Input:	String
Output:	None
Description:	Function to contact the twitter api to message a user.

Name:	MessageUser
Input:	None
Output:	None
Description:	Pulls up the on screen keyboard to dm a user.

4.10.3. TimelineHandler

TimelineHandler	
Class	
↳ TwitterObjects	
Fields	
authHandler	
backImageButt...	
bodyText	
convoHandler	
convoTitle	
currentImageIn...	
currentTweet	
favoritesCount	
firstTweetBody...	
firstTweetImage	
firstTweetObject	
firstTweetProfil...	
firstTweetRealI...	
firstTweetTime...	
firstTweetTwitt...	
fourthTweetBo...	
fourthTweetIm...	
fourthTweetObj...	
fourthTweetPro...	
fourthTweetRe...	
fourthTweetTi...	
fourthTweetTwi...	
homeButton	
homeTimeLine...	
homeTimeLine...	
homeTimeLine...	
ImageObjects	
imagesButton	
imageURLs	
lastImageButton	
nextImageButton	
overviewBackB...	
overviewNextT...	
overviewPrevio...	
overviewSelect...	
privateMessage...	
profilePic	
realName	
repliesCount	
replyButton	
replyImageButt...	
replyToUserna...	
retweetsCount	
secondTweetBo...	
secondTweetIm...	
secondTweetO...	
secondTweetPr...	
secondTweetRe...	
secondTweetTi...	
secondTweetT...	
selectTweetBac...	
thirdTweetBody...	
thirdTweetImage	
thirdTweetObject	
thirdTweetProfi...	
thirdTweetReal...	
thirdTweetTime...	
thirdTweetTwitt...	
timelineTitle	
timeStamp	
TitleView	
twitterHandle	
userImage	
Methods	
addHomeTime...	
clearOverviewO...	
getCurrentTweet	
imagesBackToC...	
lastImage	
Message	
messageUser	
nextImage	
populateImages	
ReplyImage	
replyToImage	
selectFirstTwee...	
selectFourthTw...	
selectSecondT...	
selectThirdTwe...	
selectTweet	
selectTweetBack	
setCurrentImage	
setImage	
setNextFourTw...	
setOverViewOb...	
setPreviousFou...	
setProfilePic	
setTweet	
setUserImage	

4.10.3.1. Attributes

Name	Access	Type	Description
homeTimelineOverviewObjects	public	GameObject	Holds all Timeline Game Objects
firstTweetObject	public	Status	First Tweet Displayed
secondTweetObject	public	Status	Second Tweet Displayed
thirdTweetObject	public	Status	Third Tweet Displayed
fourthTweetObject	public	Status	Fourth Tweet Displayed
firstTweetProfilePic	public	Image	First Tweet Profile Pic
secondTweetProfilePic	public	Image	SecondTweet Profile Pic
thirdTweetProfilePic	public	Image	Third Tweet Profile Pic
fourthTweetProfilePic	public	Image	Fourth Tweet Profile Pic
firstTweetRealName	public	Text	First Tweet Real Name
secondTweetRealName	public	Text	Second Tweet Real Name
thirdTweetRealName	public	Text	ThirdTweet Real Name
fourthTweetRealName	public	Text	Fourth Tweet Real Name
firstTweetTwitterHandle	public	Text	First Tweet Twitter Handle
secondTweetTwitterHandle	public	Text	Second Tweet Twitter Handle
thirdTweetTwitterHandle	public	Text	Third Tweet Twitter Handle
fourthTweetTwitterHandle	public	Text	Fourth Tweet Twitter Handle
firstTweetBodyText	public	Text	First Tweet Message Body

secondTweetBodyText	public	Text	Second Tweet Message Body
thirdTweetBodyText	public	Text	Third Tweet Message Body
fourthTweetBodyText	public	Text	Fourth Tweet Message Body
firstTweetTimeStamp	public	Text	Timestamp for first tweet
secondTweetTimeStamp	public	Text	Timestamp for second tweet
thirdTweetTimeStamp	public	Text	Timestamp for third tweet
fourthTweetTimeStamp	public	Text	Timestamp for fourth tweet
firstTweetImage	public	Image	Image to first Tweet
secondTweetImage	public	Image	Image to second Tweet
thirdTweetImage	public	Image	Image to third Tweet
fourthTweetImage	public	Image	Image to fourthTweet
overviewNextTweetButton	public	Button	Button leading to next set of tweets
overviewPreviousTweetButton	public	Button	Button leading to previous set of Tweets
overviewSelectTweetButton	public	Button	Button to select a tweet
overviewBackButton	public	Button	Button to go back from overview
selectTweetBackButton	public	Button	Button to go back from tweet select

4.10.3.2. Methods

Name:	addHomeTimeLine
Input:	None

Output:	None
Description:	This function populates the user homepage

Name:	selectTweetBack
Input:	None
Output:	None
Description:	hide back button, show tweet navigation buttons

Name:	selectTweet
Input:	None
Output:	None
Description:	hide tweet navigation buttons, hide select tweet back button

Name:	setPreviousFourTweets
Input:	None
Output:	None
Description:	Loads tweet from last screen

Name:	setNextFourTweets
Input:	None
Output:	None

Description:	Loads four tweets for next screen
---------------------	-----------------------------------

Name:	setOverViewObjects
Input:	None
Output:	None
Description:	Loads Overview Game Objects

Name:	clearOverviewObjects
Input:	None
Output:	None
Description:	Clears Overview Game Objects

Name:	setImage
Input:	Image, String, int, int
Output:	IEnumerator
Description:	Loads and sets image

Name:	selectFirstTweetObject
Input:	None
Output:	None
Description:	Selects first tweet game object in list

Name:	selectSecondTweetObject
Input:	None
Output:	None
Description:	Selects second tweet game object in list

Name:	selectThirdTweetObject
Input:	None
Output:	None
Description:	Selects third tweet game object in list

Name:	selectFourthTweetObject
Input:	None
Output:	None
Description:	Selects fourth tweet game object in list

Name:	setTweet
Input:	Status
Output:	None
Description:	This function sets the current tweet for the user.

Name:	setProfilePic
Input:	String
Output:	IEnumerator
Description:	Loads and sets the user's profile pic

Name:	getCurrentTweet
Input:	Status
Output:	None
Description:	Loads current tweet

Name:	populateImages
Input:	None
Output:	None
Description:	Handle the twitter page to display the images for the current post

Name:	imagesBackToCurrentTweet
Input:	None
Output:	None
Description:	Go back from the images menu to the twitter timeline

Name:	setCurrentImage
Input:	None
Output:	None
Description:	Set the current image being displayed

Name:	nextImage
Input:	None
Output:	None
Description:	Go to the next image

Name:	lastImage
Input:	None
Output:	None
Description:	Go to the last image

Name:	setUserImage
Input:	String
Output:	IEnumerator
Description:	Loads and set's the User's Image

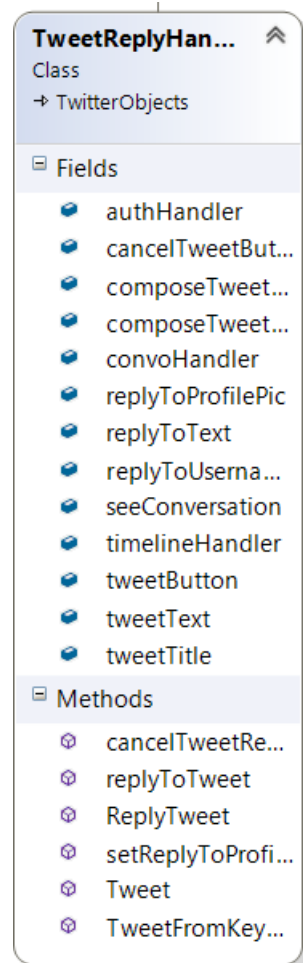
Name:	Message
Input:	String
Output:	None
Description:	Contact the twitter api to send a message to the user

Name:	messageUser
Input:	None
Output:	None
Description:	Pop up the on screen keyboard to message a user

Name:	ReplyImage
Input:	String
Output:	None
Description:	Contact the twitter api to send the tweet reply for an image

Name:	replyToImage
Input:	None
Output:	None
Description:	This function brings the user to a screen that allows them to reply to a tweet.

4.10.4. TweetReplyHandler



4.10.4.1. Attributes

Name	Access	Type	Description
authHandler	public	AuthenticationHandler	Handlers Authentication with twitter
timelineHandler	public	TimelineHandler	Handles interacting with the timeline
convoHandler	public	ConversationHandler	Handles conversations in twitter
composeTweetObjects	public	GameObject	GameObject that holds tweets written by user
tweetTitle	public	Text	Title of tweet
replyToProfilePic	public	Image	Profile picture of other user the current user is replying too
replyToUsername	public	Text	Username name of person user is replying too
replyToText	public	Text	Text of person the user is replying too
tweetText	public	Inputfield	Field where user inputs their reply
tweetButton	public	Button	Button to send tweet
cancelTweetButton	public	Button	Button to cancel tweet
seeConversation	public	Button	Button to show full conversation
composeTweetObjectsString	public	String	Name of Tweet Object

4.10.4.2. Methods

Name:	ReplyTweet
Input:	String
Output:	None

Description:	Function to contact the twitter api to reply to a tweet
---------------------	---

Name:	replyToTweet
Input:	None
Output:	None
Description:	This function brings the user to a screen that allows them to reply to a tweet.

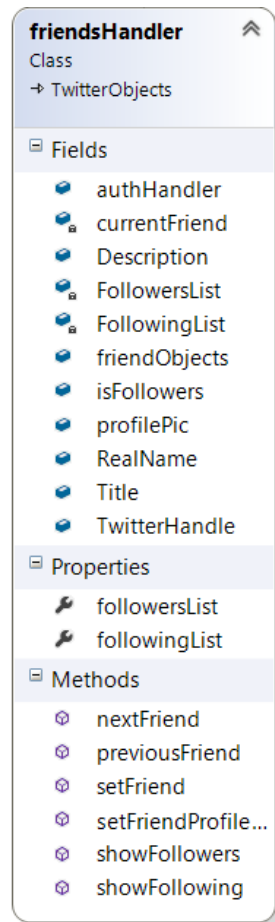
Name:	cancelTweetReplyButton
Input:	None
Output:	None
Description:	Hides tweet reply button

Name:	setReplyToProfilePic
Input:	String
Output:	IEnumerator
Description:	Loads profile picture to reply

Name:	Tweet
Input:	String
Output:	IEnumerator
Description:	Contacts the twitter api to tweet a message

Name:	TweetFromKeyboard
Input:	None
Output:	None
Description:	Pull up the on screen keyboard to tweet

4.10.5. FreindsHandler



4.10.5.1. Attributes

Name	Access	Type	Description
------	--------	------	-------------

friendObjects	public	GameObject	Holds friend's information
Description	public	Text	Holds description of friend
Title	public	Text	Holds title to friend's profile
profilePic	public	Image	Holds friend's profile picture
RealName	public	Text	Holds friend's real name
TwitterHandle	public	Text	Holds friend's twitter handle
isFollowers	public	bool	Check's if friend is a follower
FollowingList	private	List<User>	List of user's followers
followingList	public	List<User>	List of who the user is following
FollowersList	private	List<User>	List of user's followers
followersList	public	List<User>	List of user's followers
currentFriend	private	int	Number of User's Friends
authHandler	public	AuthenticationHandler	Handles twitter's authentication

4.10.5.2. Methods

Name:	showFollowers
Input:	None
Output:	None
Description:	Shows user's followers

Name:	showFollowing
--------------	---------------

Input:	None
Output:	None
Description:	Shows who is following the user

Name:	setFriend
Input:	User
Output:	None
Description:	This function sets the current tweet for the user.

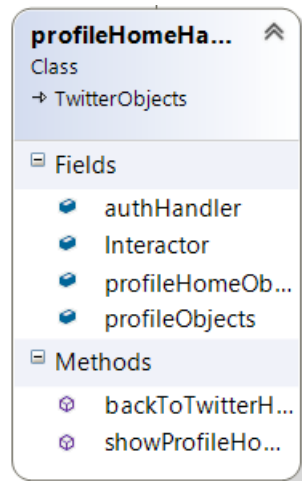
Name:	setFriendProfilePic
Input:	String
Output:	IEnumerator
Description:	Loads and Sets friend's profile picture

Name:	nextFriend
Input:	None
Output:	None
Description:	Loads Next Friend

Name:	previousFriend
Input:	None

Output:	None
Description:	Loads last friend viewed

4.10.6. ProfileHandHandler



4.10.6.1. Attributes

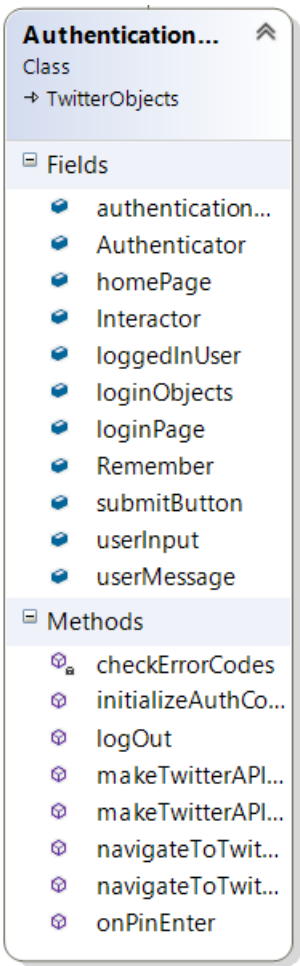
Name	Access	Type	Description
authHandler	public	AuthenticationHandler	Handle's Twitter Authentication
profileHomeObjects	public	GameObject	Profile Home Game Object
Interactor	public	TwitterInteractor	Interacts with Twitter
profileObjects	public	String	Name of Profile Object

4.10.6.2. Methods

Name:	showProfileHome
Input:	None
Output:	None
Description:	Makes profile home scene visible

Name:	backToTwitterHome
Input:	None
Output:	None
Description:	Disables current scene

4.10.7. AuthenticationHandler



4.10.7.1. Attributes

Name	Access	Type	Description
------	--------	------	-------------

Remember	public	bool	Hold current credentials
loginObjects	public	GameObject	Holds Login Game Object
authenticationURL	public	Text	Holds url to authenticate the account
userMessage	public	Text	Holds User's message
userInput	public	InputField	Textbox for user to input message
submitButton	public	Text	Button to send user's message
loginPage	public	String	Name of login page
homePage	public	String	Name of home page
loggedInUser	public	String	Name of current User
Authenticator	public	TwitterAuthenticator	Used to Authenticate twitter interactions
Interactor	public	TwitterInteractor	Interacts with Twitter

4.10.7.2. Methods

Name:	initializeAuthComponent
Input:	None
Output:	None
Description:	Intializes the AuthenticationHandler

Name:	onPinEnter
Input:	None
Output:	None

Description:	This is the on click event for when the user enters their pin code that they received from the twitter website
---------------------	--

Name:	navigateToTwitterHome
Input:	None
Output:	None
Description:	This function navigates the user to the twitter home page. From this page, the user can choose whether they wish to navigate to their timeline, their own profile

Name:	navigateToTwitterAuthPage
Input:	String
Output:	None
Description:	This function navigates the user back to the user authentication page. Here the user will need to open the link that is copied to their clipboard in a browser, and then enter the PIN code shown

Name:	makeTwitterAPICallNoReturnVal
Input:	Action
Output:	None
Description:	This function allows us to make a call to the API for functions that do not have a return value

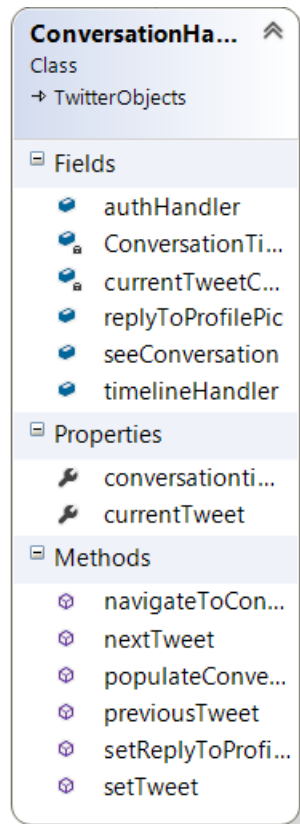
Name:	makeTwitterAPICall
Input:	Func<T>

Output:	T
Description:	This function allows us to make calls to a twitter api function that DOES have a return value.

Name:	checkErrorCodes
Input:	Exception
Output:	None
Description:	Checks error and responds with appropriate action

Name:	logOut
Input:	Exception
Output:	None
Description:	Logs the user out

4.10.8. ConversationHandler



4.10.8.1. Attributes

Name	Access	Type	Description
authHandler	public	AuthenticationHandler	Handles Twitter Authentication
timelineHandler	public	TimeLineHandler	Handles interacting with timeline
ConversationTimeline	private	List<Status>	List of conversations on timeline

conversationTimelineStatuses	public	List<Status>	List of statuses on timeline
currentTweetConvo	private	int	Index of current tweet
currentTweet	public	int	Index of current tweet
replyToProfilePic	public	Image	Profile picture of person the user is replying to
seeConversation	public	Button	Button used to display full conversation

4.10.8.2. Methods

Name:	populateConversation
Input:	None
Output:	None
Description:	Populate the unity conversation objects

Name:	setTweet
Input:	int
Output:	None
Description:	This function sets the current tweet for the user

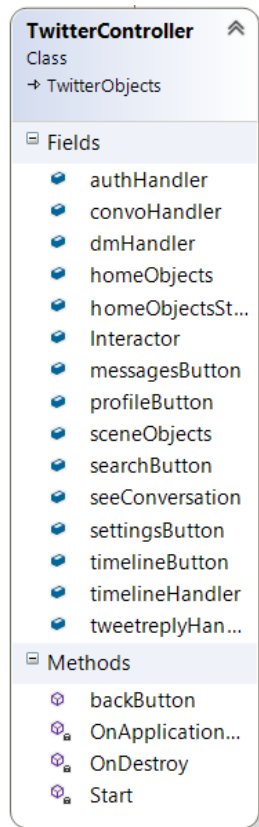
Name:	setReplyToProfilePic
Input:	String
Output:	IEnumerator

Description:	Loads and set's profile picture to for profile the user is replying to
---------------------	--

Name:	nextTweet
Input:	None
Output:	None
Description:	This function populates the timeline ui with the next tweet in the list

Name:	previousTweet
Input:	None
Output:	None
Description:	This function populates the timeline ui with the last tweet in the list

4.10.9. TwitterController



4.10.9.1. Attributes

Name	Access	Type	Description
sceneObjects	public	GameObject	Current Scene
homeObjectsString	public	String	Name of home object
homeObjects	public	GameObject	Home object for scene

timelineButton	public	Button	Button to show timeline
profileButton	public	Button	Button to show profile
messagesButton	public	Button	Button to show messages
settingsButton	public	Button	Button to show settings
searchButton	public	Button	Button to start search
seeConversation	public	Button	Button to show full conversation
authHandler	public	AuthenticationHandler	Handles Twitter Authentication
timelineHandler	public	TimeLineHandler	Handles interactions with the timeline
convoHandler	public	ConversationHandler	Handles conversations
dmHandler	public	DirectMessageHandler	Handles direct messages
tweetreplyHandler	public	TweetReplyHandler	Handles creating a tweet
Interactor	public	TwitterInteractor	Handles interacting with Twitter

4.10.9.2. Methods

Name:	Start
Input:	None
Output:	None
Description:	Loads and initializes twitter scene

Name:	OnApplicationQuit
Input:	None

Output:	None
Description:	Closes threads on application exit

Name:	OnDestroy
Input:	None
Output:	None
Description:	Closes threads on scene destruction

Name:	backButton
Input:	None
Output:	None
Description:	Disables current scene

5. Traceability Matrix

Description	Color
Section Head	
Requirement Head	
Handled by Unity	

Requirement	Description	Class Number	Class Name
R3.1	User Interface		
R3.1.1	Login Page		
R3.1.1.1	Username field		
R3.1.1.2	Password Field		
R3.1.1.3	Facebook Button		
R3.1.1.4	Twitter Button		
R3.1.1.5	Instagram Button		
R3.1.1.6	Login Button		
R3.1.1.7	Remember me checkbox		
R3.1.2	Logged in Menu		
R3.1.2.1	Button to go to Facebook Scene		
R3.1.2.2	Button to go to Twitter Scene		
R3.1.2.3	Button to go to Instagram Scene		
R3.1.3	Twitter Home Menu		
R3.1.3.1	Tweet Button	4.1.6	ButtonSelectionMenu
R3.1.3.2	Timeline Button	4.1.6	ButtonSelectionMenu
R3.1.3.3	Profile Button	4.1.6	ButtonSelectionMenu
R3.1.3.4	Messages Button	4.1.6	ButtonSelectionMenu
R3.1.3.5	Settings Button		

R3.1.3.6	Connection Status Orb	4.6.1	DeviceManager
R3.1.3.7	Battery Indicator	4.6.1	DeviceManager
R3.1.3.8	Signal Strength Indicator	4.6.1	DeviceManager
R3.1.3.9	Manual Entry Button		
R3.1.4	Timeline Page		
R3.1.4.1	Newer Tweets Button	4.1.6	ButtonSelectionMenu
R3.1.4.2	Older Tweets Button	4.1.6	ButtonSelectionMenu
R3.1.4.3	Four Tweet Elements	4.1.6	ButtonSelectionMenu
R3.1.4.4	Back Button	4.1.6	ButtonSelectionMenu
R3.1.4.5	Select Tweet	4.1.6	ButtonSelectionMenu
R3.1.5	Tweet Details Page		
R3.1.5.1	Profile Picture	4.11.3	TimelineHandler
R3.1.5.2	Twitter Name	4.11.3	TimelineHandler
R3.1.5.3	Twitter Handle	4.11.3	TimelineHandler
R3.1.5.4	Twitter Text	4.11.3	TimelineHandler
R3.1.5.5	Home Button	4.11.3	TimelineHandler
R3.1.5.6	Reply Button	4.11.3	TimelineHandler
R3.1.5.7	Message Button	4.11.3	TimelineHandler
R3.1.5.8	Images Button	4.11.3	TimelineHandler
R3.1.6	Image Page		
R3.1.6.1	Image View	4.11.3	TimelineHandler
R3.1.6.2	Last Image Button	4.11.3	TimelineHandler
R3.1.6.3	Back Button	4.11.3	TimelineHandler
R3.1.6.4	Reply Button	4.11.3	TimelineHandler
R3.1.6.5	Next Button	4.11.3	TimelineHandler
R3.1.7	Profile Home Menu		
R3.1.7.1	Tweets Button	4.1.6	ButtonSelectionMenu
R3.1.7.2	Following Button	4.1.6	ButtonSelectionMenu
R3.1.7.3	Followers Button	4.1.6	ButtonSelectionMenu
R3.1.7.4	Likes Button	4.1.6	ButtonSelectionMenu
R3.1.7.5	Back Button	4.1.6	ButtonSelectionMenu
R3.1.8	User's Tweet Timeline Page		
R3.1.8.1	Profile Picture	4.11.3	TimelineHandler
R3.1.8.2	Twitter Name	4.11.3	TimelineHandler
R3.1.8.3	Twitter Handle	4.11.3	TimelineHandler

R3.1.8.4	Twitter Text Body	4.11.3	TimelineHandler
R3.1.8.5	Previous Tweet Button	4.1.6	ButtonSelectionMenu
R3.1.8.6	Home Button	4.1.6	ButtonSelectionMenu
R3.1.8.7	Reply Button	4.1.6	ButtonSelectionMenu
R3.1.8.8	Retweet Button	4.1.6	ButtonSelectionMenu
R3.1.8.9	Image Button	4.1.6	ButtonSelectionMenu
R3.1.8.10	Next Tweet Button	4.1.6	ButtonSelectionMenu
R3.1.9	Browse Follower/Following Screen		
R3.1.9.1	Twitter Name	4.11.3	TimelineHandler
R3.1.9.2	Twitter Handle	4.11.3	TimelineHandler
R3.1.9.3	Previous Button	4.1.6	ButtonSelectionMenu
R3.1.9.4	Back Button	4.1.6	ButtonSelectionMenu
R3.1.9.5	Mute Button	4.1.6	ButtonSelectionMenu
R3.1.9.6	Block Button	4.1.6	ButtonSelectionMenu
R3.1.9.7	Unfollow Button	4.1.6	ButtonSelectionMenu
R3.1.9.8	Next Button	4.1.6	ButtonSelectionMenu
R3.1.10	Direct Message Home Page		
R3.1.10.1	Twitter Name	4.11.3	TimelineHandler
R3.1.10.2	Twitter Handle	4.11.3	TimelineHandler
R3.1.10.3	Text Body	4.11.3	TimelineHandler
R3.1.10.4	Newer Button	4.1.6	ButtonSelectionMenu
R3.1.10.5	Home Button	4.1.6	ButtonSelectionMenu
R3.1.10.6	Message Button	4.1.6	ButtonSelectionMenu
R3.1.10.7	Older Button	4.1.6	ButtonSelectionMenu
R3.1.11	Browse Messages Page		
R3.1.11.1	Blue Text Box		
R3.1.11.2	White Text Box		
R3.1.11.3	Back Button	4.1.6	ButtonSelectionMenu
R3.1.11.4	Older Button	4.1.6	ButtonSelectionMenu
R3.1.11.5	Newer Button	4.1.6	ButtonSelectionMenu
R3.1.11.6	Reply Button	4.1.6	ButtonSelectionMenu
R3.1.12	Popup Keyboard		
R3.1.12.1	Enter Text Field		
R3.1.12.2	Exit Button	4.1.6	ButtonSelectionMenu
R3.1.12.3	Submit Button	4.1.6	ButtonSelectionMenu

R3.1.12.4	Pharse/Letter Buttons	4.1.6	ButtonSelectionMenu
-----------	-----------------------	-------	---------------------

