

HVWC Multiplayer Platform

1.0

Generated by Doxygen 1.8.6

Thu Jan 30 2014 15:17:39

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	AnimationController Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Function Documentation	5
3.1.2.1	ActivateAnimation	5
3.1.2.2	ActivateAnimation	6
3.1.3	Property Documentation	6
3.1.3.1	Animations	6
3.1.3.2	CurrentAnimation	6
3.2	MainMenu.Avatar Class Reference	6
3.2.1	Detailed Description	6
3.2.2	Member Data Documentation	6
3.2.2.1	icon	6
3.2.2.2	name	7
3.2.2.3	prefab	7
3.3	C3VoiceChat Class Reference	7
3.3.1	Detailed Description	7
3.3.2	Member Function Documentation	7
3.3.2.1	JoinVoice	7
3.4	Chat Class Reference	7
3.4.1	Detailed Description	8
3.4.2	Member Function Documentation	8
3.4.2.1	SendChat	8
3.4.2.2	SendChat	8
3.4.3	Member Data Documentation	8
3.4.3.1	Messages	8

3.5	ClickTeleport Class Reference	9
3.5.1	Detailed Description	9
3.5.2	Member Data Documentation	9
3.5.2.1	radius	9
3.6	MainMenu Class Reference	9
3.6.1	Detailed Description	10
3.6.2	Member Data Documentation	10
3.6.2.1	avatars	10
3.6.2.2	background	10
3.6.2.3	CustomSkin	10
3.6.2.4	selectedAvatar	10
3.6.2.5	spawnRadius	10
3.7	NameTag Class Reference	10
3.7.1	Detailed Description	11
3.7.2	Member Data Documentation	11
3.7.2.1	fadeDistance	11
3.7.2.2	nameTagColor	11
3.8	NetworkManager Class Reference	11
3.8.1	Detailed Description	12
3.8.2	Member Function Documentation	12
3.8.2.1	Connect	12
3.8.2.2	CreateRoom	12
3.8.2.3	Instantiate	13
3.8.2.4	JoinRandomRoom	13
3.8.2.5	JoinRoom	13
3.8.2.6	LeaveRoom	13
3.8.2.7	LoadLevel	13
3.8.2.8	LoadLevel	13
3.8.3	Member Data Documentation	13
3.8.3.1	version	13
3.8.4	Property Documentation	14
3.8.4.1	Connected	14
3.8.4.2	Instance	14
3.8.4.3	IsMessageQueueRunning	14
3.8.4.4	OfflineMode	14
3.8.4.5	PlayerName	14
3.8.4.6	Room	14
3.8.4.7	RoomList	14
3.9	NetworkPlayer Class Reference	14
3.9.1	Detailed Description	15

3.10 PlayerController Class Reference	15
3.10.1 Detailed Description	16
3.10.2 Member Enumeration Documentation	16
3.10.2.1 CharacterState	16
3.10.3 Member Function Documentation	16
3.10.3.1 Fly	16
3.10.4 Member Data Documentation	16
3.10.4.1 animationController	16
3.10.4.2 flying	16
3.10.4.3 flySpeed	17
3.10.4.4 gravity	17
3.10.4.5 grounded	17
3.10.4.6 groundingDistance	17
3.10.4.7 groundLayers	17
3.10.4.8 heightOffset	17
3.10.4.9 runSpeed	17
3.10.4.10 walkSpeed	17
3.10.5 Property Documentation	17
3.10.5.1 CharState	17
3.11 PlayerHUD Class Reference	17
3.11.1 Detailed Description	18
3.11.2 Member Data Documentation	18
3.11.2.1 C3_Tex	18
3.11.2.2 CustomSkin	18
3.11.2.3 ExitTex	18
3.11.2.4 HelpTex	18
3.11.2.5 loadingScreen	18
3.11.2.6 MapTex	19
3.11.2.7 VolumeTex	19
3.12 PlayerList Class Reference	19
3.12.1 Detailed Description	19
3.12.2 Property Documentation	19
3.12.2.1 LocalPlayer	19
3.12.2.2 Players	19
3.12.2.3 RemotePlayers	20
3.13 ProximityTrigger Class Reference	20
3.13.1 Detailed Description	20
3.13.2 Member Data Documentation	20
3.13.2.1 url	20
3.14 QuickPlayBypass Class Reference	20

3.14.1 Detailed Description	21
3.15 TeleportObject Class Reference	21
3.15.1 Detailed Description	21
3.16 TeleportToObject Class Reference	21
3.16.1 Detailed Description	21
3.17 WebLink Class Reference	21
3.17.1 Detailed Description	22
3.17.2 Member Data Documentation	22
3.17.2.1 url	22
Index	23

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

MainMenu.Avatar	6
MonoBehaviour	
AnimationController	5
C3VoiceChat	7
Chat	7
ClickTeleport	9
MainMenu	9
NameTag	10
NetworkManager	11
NetworkPlayer	14
PlayerController	15
PlayerHUD	17
PlayerList	19
ProximityTrigger	20
QuickPlayBypass	20
TeleportObject	21
TeleportToObject	21
WebLink	21

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AnimationController	This class handles the activation and deactivation of animations on the player.	5
MainMenu.Avatar	This class constructs an Avatar object.	6
C3VoiceChat	This class allows for the creation or joining of rooms in the Vivox C3 voice chat application. . .	7
Chat	This class handles public and private chat messaging between players.	7
ClickTeleport	This class allows for teleportation by double-clicking on a collider.	9
MainMenu	This class displays the main menu used in the lobby.	9
NameTag	This class handles displaying the player's name tag.	10
NetworkManager	This singleton class manages the network and wraps useful network methods.	11
NetworkPlayer	This class handles the networking of important player properties.	14
PlayerController	This class handles the movement and animations of the local player.	15
PlayerHUD	This class displays the local player's HUD.	17
PlayerList	This class wraps useful network player methods.	19
ProximityTrigger	This class allows a web page to be opened when the player enters a trigger.	20
QuickPlayBypass	This class allows for the enabling of quickplay in the web player.	20
TeleportObject	This class allows for teleportation by clicking on a map icon.	21
TeleportToObject	This class allows for teleportation by clicking on a collider.	21
WebLink	This class allows for hyperlinking clicks on colliders.	21

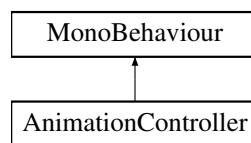
Chapter 3

Class Documentation

3.1 AnimationController Class Reference

This class handles the activation and deactivation of animations on the player.

Inheritance diagram for AnimationController:



Public Member Functions

- void [ActivateAnimation](#) (string animationToActivate)
A method to set activate an animation on our player.
- void [ActivateAnimation](#) (int index)
A method to set activate an animation on our player.

Properties

- string[] [Animations](#) [get, set]
A property to get/set the animations we have on our player.
- string [CurrentAnimation](#) [get, set]
A property to get/set the currently playing animation.

3.1.1 Detailed Description

This class handles the activation and deactivation of animations on the player.

3.1.2 Member Function Documentation

3.1.2.1 void AnimationController.ActivateAnimation (string animationToActivate)

A method to set activate an animation on our player.

Parameters

<i>animationTo-Activate</i>	The name of the animation to activate.
-----------------------------	--

3.1.2.2 void AnimationController.ActivateAnimation (int *index*)

A method to set activate an animation on our player.

Parameters

<i>index</i>	The index of the animation to activate.
--------------	---

3.1.3 Property Documentation

3.1.3.1 string [] AnimationController.Animations [get], [set]

A property to get/set the animations we have on our player.

3.1.3.2 string AnimationController.CurrentAnimation [get], [set]

A property to get/set the currently playing animation.

The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Player/AnimationController.cs

3.2 MainMenu.Avatar Class Reference

This class constructs an [Avatar](#) object.

Public Attributes

- string [name](#)
The name of the avatar prefab.
- GameObject [prefab](#)
The prefab for this avatar.
- Texture [icon](#)
The icon for this avatar.

3.2.1 Detailed Description

This class constructs an [Avatar](#) object.

3.2.2 Member Data Documentation

3.2.2.1 Texture MainMenu.Avatar.icon

The icon for this avatar.

3.2.2.2 string MainMenu.Avatar.name

The name of the avatar prefab.

3.2.2.3 GameObject MainMenu.Avatar.prefab

The prefab for this avatar.

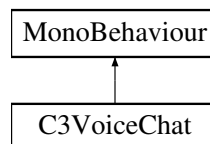
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/GUI/MainMenu.cs

3.3 C3VoiceChat Class Reference

This class allows for the creation or joining of rooms in the Vivox C3 voice chat application.

Inheritance diagram for C3VoiceChat:



Static Public Member Functions

- static void [JoinVoice](#) (string roomName, string userName)
Joins or creates a C3 voice channel with the player's username.

3.3.1 Detailed Description

This class allows for the creation or joining of rooms in the Vivox C3 voice chat application.

3.3.2 Member Function Documentation

3.3.2.1 static void C3VoiceChat.JoinVoice (string roomName, string userName) [static]

Joins or creates a C3 voice channel with the player's username.

Parameters

<i>roomName</i>	The room name.
<i>userName</i>	The player's username

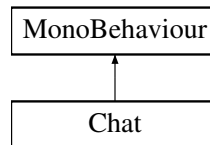
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/GUI/C3VoiceChat.cs

3.4 Chat Class Reference

This class handles public and private chat messaging between players.

Inheritance diagram for Chat:



Public Member Functions

- void [SendChat](#) (PhotonTargets targets, string message)
A method to send a public chat message.
- void [SendChat](#) (PhotonPlayer target, string message)
A method to send a private chat message.

Public Attributes

- List< string > [Messages](#) = new List<string>()
A list of the messages that have been sent.

3.4.1 Detailed Description

This class handles public and private chat messaging between players.

3.4.2 Member Function Documentation

3.4.2.1 void Chat.SendChat (PhotonTargets targets, string message)

A method to send a public chat message.

Parameters

<i>targets</i>	The players to whom this message should be sent.
<i>message</i>	The message to send.

3.4.2.2 void Chat.SendChat (PhotonPlayer target, string message)

A method to send a private chat message.

Parameters

<i>target</i>	The player to whom this message should be sent.
<i>message</i>	The message to send.

3.4.3 Member Data Documentation

3.4.3.1 List<string> Chat.Messages = new List<string>()

A list of the messages that have been sent.

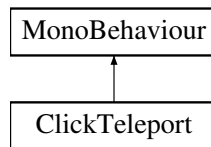
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Player/Chat.cs

3.5 ClickTeleport Class Reference

This class allows for teleportation by double-clicking on a collider.

Inheritance diagram for ClickTeleport:



Public Attributes

- float [radius](#) = 100f
The radius of the teleportation range.

3.5.1 Detailed Description

This class allows for teleportation by double-clicking on a collider.

3.5.2 Member Data Documentation

3.5.2.1 float ClickTeleport.radius = 100f

The radius of the teleportation range.

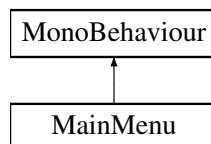
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Functionality/ClickTeleport.cs

3.6 MainMenu Class Reference

This class displays the main menu used in the lobby.

Inheritance diagram for MainMenu:



Classes

- class [Avatar](#)
This class constructs an [Avatar](#) object.

Public Attributes

- [Avatar](#)[] [avatars](#)

- An array of [Avatar](#).
- string `selectedAvatar` = ""
The selected avatar.
- float `spawnRadius` = 3f
The spawn radius.
- GUISkin [CustomSkin](#)
The custom skin.
- Texture [background](#)
The background for the main menu.

3.6.1 Detailed Description

This class displays the main menu used in the lobby.

3.6.2 Member Data Documentation

3.6.2.1 Avatar [] MainMenu.avatars

An array of [Avatar](#).

3.6.2.2 Texture MainMenu.background

The background for the main menu.

3.6.2.3 GUISkin MainMenu.CustomSkin

The custom skin.

3.6.2.4 string MainMenu.selectedAvatar = ""

The selected avatar.

3.6.2.5 float MainMenu.spawnRadius = 3f

The spawn radius.

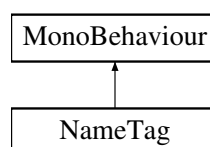
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/GUI/MainMenu.cs

3.7 NameTag Class Reference

This class handles displaying the player's name tag.

Inheritance diagram for NameTag:



Public Attributes

- Color [nameTagColor](#)
The color the text of name tag should be.
- float [fadeDistance](#)
The distance at which the name tag should start to fade.

3.7.1 Detailed Description

This class handles displaying the player's name tag.

3.7.2 Member Data Documentation

3.7.2.1 float NameTag.fadeDistance

The distance at which the name tag should start to fade.

3.7.2.2 Color NameTag.nameTagColor

The color the text of name tag should be.

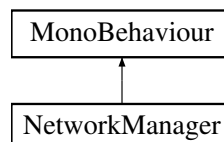
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Player/NameTag.cs

3.8 NetworkManager Class Reference

This singleton class manages the network and wraps useful network methods.

Inheritance diagram for NetworkManager:



Public Member Functions

- void [Connect](#) (string [version](#))
A method to connect to the Photon server. Uses the settings you set in the PUN Setup Wizard.
- void [CreateRoom](#) (string roomName, bool isVisible, bool isOpen, int maxPlayers)
A method to create a new room on the Photon server.
- void [JoinRoom](#) (string roomName)
A method to join an existing room on the Photon server.
- void [JoinRandomRoom](#) ()
A method to join a random room on the Photon server.
- void [LeaveRoom](#) ()
A method to leave the current room.
- GameObject [Instantiate](#) (string prefabName, Vector3 position, Quaternion rotation, int group)
A method to instantiate a prefab on the network.

- void [LoadLevel](#) (int levelID)
A method to load a level.
- void [LoadLevel](#) (string levelName)
A method to load a level.

Public Attributes

- string [version](#)
The version of this build. Photon separates versions on the network.

Properties

- static [NetworkManager Instance](#) [get, set]
A static instance of this script to make this gameobject a singleton.
- string [PlayerName](#) [get, set]
Gets or sets the name of the local player.
- Room [Room](#) [get]
Gets the current room.
- bool [Connected](#) [get]
Gets a value indicating whether PhotonNetwork is connected.
- RoomInfo[] [RoomList](#) [get]
Gets the room list.
- bool [OfflineMode](#) [get, set]
Gets or sets a value indicating whether PhotonNetwork is in offline mode.
- bool [IsMessageQueueRunning](#) [get, set]
Gets or sets a value indicating whether PhotonNetwork is or should be running the message queue.

3.8.1 Detailed Description

This singleton class manages the network and wraps useful network methods.

3.8.2 Member Function Documentation

3.8.2.1 void NetworkManager.Connect (string *version*)

A method to connect to the Photon server. Uses the settings you set in the PUN Setup Wizard.

Parameters

<i>version</i>	The version of this build. Photon uses this to separate different versions on the server.
----------------	---

3.8.2.2 void NetworkManager.CreateRoom (string *roomName*, bool *isVisible*, bool *isOpen*, int *maxPlayers*)

A method to create a new room on the Photon server.

Parameters

<i>roomName</i>	The name of the room we're creating.
-----------------	--------------------------------------

<i>isVisible</i>	Should this room be visible to others?
<i>isOpen</i>	Should this room be open to others?
<i>maxPlayers</i>	The maximum number of players that can be in this room at one time.

3.8.2.3 GameObject NetworkManager.Instantiate (string *prefabName*, Vector3 *position*, Quaternion *rotation*, int *group*)

A method to instantiate a prefab on the network.

Parameters

<i>prefabName</i>	The name of the prefab we're instantiating. NOTE: The prefab must be located in the Resources folder.
<i>position</i>	The position where the prefab should be instantiated.
<i>rotation</i>	How the prefab should be orientated.
<i>group</i>	The group the prefab. Can be used for team-based play.

3.8.2.4 void NetworkManager.JoinRandomRoom ()

A method to join a random room on the Photon server.

3.8.2.5 void NetworkManager.JoinRoom (string *roomName*)

A method to join an existing room on the Photon server.

Parameters

<i>roomName</i>	The name of the room we're trying to join.
-----------------	--

3.8.2.6 void NetworkManager.LeaveRoom ()

A method to leave the current room.

3.8.2.7 void NetworkManager.LoadLevel (int *levelID*)

A method to load a level.

Parameters

<i>levelID</i>	The integer id of the level to load.
----------------	--------------------------------------

3.8.2.8 void NetworkManager.LoadLevel (string *levelName*)

A method to load a level.

Parameters

<i>levelName</i>	The name of the level to load.
------------------	--------------------------------

3.8.3 Member Data Documentation

3.8.3.1 string NetworkManager.version

The version of this build. Photon separates versions on the network.

3.8.4 Property Documentation

3.8.4.1 `bool NetworkManager.Connected` [get]

Gets a value indicating whether PhotonNetwork is connected.

true if connected; otherwise, false.

3.8.4.2 `NetworkManager NetworkManager.Instance` [static], [get], [set]

A static instance of this script to make this gameobject a singleton.

3.8.4.3 `bool NetworkManager.IsMessageQueueRunning` [get], [set]

Gets or sets a value indicating whether PhotonNetwork is or should be running the message queue.

true if this instance is message queue running; otherwise, false.

3.8.4.4 `bool NetworkManager.OfflineMode` [get], [set]

Gets or sets a value indicating whether PhotonNetwork is in offline mode.

true if in offline mode; otherwise, false.

3.8.4.5 `string NetworkManager.PlayerName` [get], [set]

Gets or sets the name of the local player.

The name of the local player.

3.8.4.6 `Room NetworkManager.Room` [get]

Gets the current room.

The current room.

3.8.4.7 `RoomInfo [] NetworkManager.RoomList` [get]

Gets the room list.

The room list.

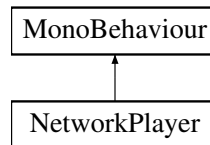
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/NetworkManager.cs

3.9 NetworkPlayer Class Reference

This class handles the networking of important player properties.

Inheritance diagram for NetworkPlayer:



3.9.1 Detailed Description

This class handles the networking of important player properties.

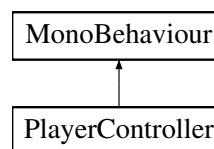
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Player/NetworkPlayer.cs

3.10 PlayerController Class Reference

This class handles the movement and animations of the local player.

Inheritance diagram for PlayerController:



Public Types

- enum [CharacterState](#) {
[CharacterState.IDLE](#) =0, [CharacterState.WALK](#) =1, [CharacterState.RUN](#) =2, [CharacterState.JUMP](#) =3,
[CharacterState.SIT](#) =4, [CharacterState.FLY](#) =5 }

This enumeration lists the different character states.

Public Member Functions

- void [Fly](#) (bool val)
A method to enable flying and set properties to allow flight.

Public Attributes

- [AnimationController](#) [animationController](#)
An instance of the Rigidbody component on this player.
- float [walkSpeed](#) =1.0f
The walk speed.
- float [runSpeed](#) =1.5f
The run speed.
- float [flySpeed](#) =2.0f
The fly speed.
- float [heightOffset](#) = 0.0f
The height offset for where the grounded raycast should originate.
- float [groundingDistance](#) = 0.1f

- The grounding distance.*

 - LayerMask [groundLayers](#)

The ground layers.
- float [gravity](#) = 50.0f

The artificial gravity.
- bool [grounded](#)

A boolean to check if the player is grounded.
- bool [flying](#)

A boolean to check if the player is flying.

Properties

- [CharacterState](#) [CharState](#) [get, set]
A property to get/set and trigger the character state of the player.

3.10.1 Detailed Description

This class handles the movement and animations of the local player.

3.10.2 Member Enumeration Documentation

3.10.2.1 enum [PlayerController.CharacterState](#)

This enumeration lists the different character states.

Enumerator

- IDLE** The IDLE character state.
- WALK** The WALK character state.
- RUN** The RUN character state.
- JUMP** The JUMP character state.
- SIT** The SIT character state.
- FLY** The FLY character state.

3.10.3 Member Function Documentation

3.10.3.1 void [PlayerController.Fly](#) (bool *val*)

A method to enable flying and set properties to allow flight.

3.10.4 Member Data Documentation

3.10.4.1 [AnimationController](#) [PlayerController.animationController](#)

An instance of the Rigidbody component on this player.

An instance of the AnimationController component on this player.

3.10.4.2 bool [PlayerController.flying](#)

A boolean to check if the player is flying.

3.10.4.3 float PlayerController.flySpeed = 2.0f

The fly speed.

3.10.4.4 float PlayerController.gravity = 50.0f

The artificial gravity.

3.10.4.5 bool PlayerController.grounded

A boolean to check if the player is grounded.

3.10.4.6 float PlayerController.groundingDistance = 0.1f

The grounding distance.

3.10.4.7 LayerMask PlayerController.groundLayers

The ground layers.

3.10.4.8 float PlayerController.heightOffset = 0.0f

The height offset for where the grounded raycast should originate.

3.10.4.9 float PlayerController.runSpeed = 1.5f

The run speed.

3.10.4.10 float PlayerController.walkSpeed = 1.0f

The walk speed.

3.10.5 Property Documentation

3.10.5.1 CharacterState PlayerController.CharState [get], [set]

A property to get/set and trigger the character state of the player.

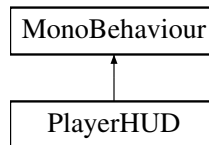
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Player/PlayerController.cs

3.11 PlayerHUD Class Reference

This class displays the local player's HUD.

Inheritance diagram for PlayerHUD:



Public Attributes

- `GUISkin` [CustomSkin](#)
The custom skin used to style the player HUD.
- `Texture2D` [C3_Tex](#)
The C3 button texture.
- `Texture2D` [MapTex](#)
The map button texture.
- `Texture2D` [VolumeTex](#)
The volume button texture.
- `Texture2D` [HelpTex](#)
The help button texture.
- `Texture2D` [ExitTex](#)
The exit button texture.
- `Texture2D` [loadingScreen](#)
The loading screen texture.

3.11.1 Detailed Description

This class displays the local player's HUD.

3.11.2 Member Data Documentation

3.11.2.1 `Texture2D PlayerHUD.C3_Tex`

The C3 button texture.

3.11.2.2 `GUISkin PlayerHUD.CustomSkin`

The custom skin used to style the player HUD.

3.11.2.3 `Texture2D PlayerHUD.ExitTex`

The exit button texture.

3.11.2.4 `Texture2D PlayerHUD.HelpTex`

The help button texture.

3.11.2.5 `Texture2D PlayerHUD.loadingScreen`

The loading screen texture.

3.11.2.6 Texture2D PlayerHUD.MapTex

The map button texture.

3.11.2.7 Texture2D PlayerHUD.VolumeTex

The volume button texture.

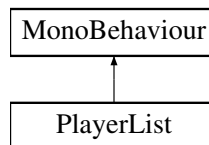
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Player/PlayerHUD.cs

3.12 PlayerList Class Reference

This class wraps useful network player methods.

Inheritance diagram for PlayerList:



Properties

- PhotonPlayer[] **Players** [get]
Gets all networked players.
- PhotonPlayer **LocalPlayer** [get]
Gets the local player.
- PhotonPlayer[] **RemotePlayers** [get]
Gets the remote players.

3.12.1 Detailed Description

This class wraps useful network player methods.

3.12.2 Property Documentation

3.12.2.1 PhotonPlayer PlayerList.LocalPlayer [get]

Gets the local player.

The local player.

3.12.2.2 PhotonPlayer [] PlayerList.Players [get]

Gets all networked players.

All networked players.

3.12.2.3 PhotonPlayer [] PlayerList.RemotePlayers [get]

Gets the remote players.

The remote players.

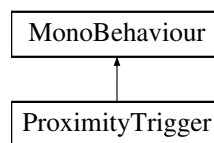
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Player/PlayerList.cs

3.13 ProximityTrigger Class Reference

This class allows a web page to be opened when the player enters a trigger.

Inheritance diagram for ProximityTrigger:



Public Attributes

- string [url](#)
The URL that should be opened.

3.13.1 Detailed Description

This class allows a web page to be opened when the player enters a trigger.

3.13.2 Member Data Documentation

3.13.2.1 string ProximityTrigger.url

The URL that should be opened.

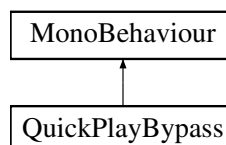
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Functionality/ProximityTrigger.cs

3.14 QuickPlayBypass Class Reference

This class allows for the enabling of quickplay in the web player.

Inheritance diagram for QuickPlayBypass:



3.14.1 Detailed Description

This class allows for the enabling of quickplay in the web player.

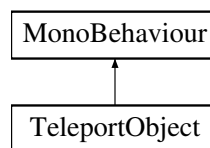
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/QuickPlayBypass.cs

3.15 TeleportObject Class Reference

This class allows for teleportation by clicking on a map icon.

Inheritance diagram for TeleportObject:



3.15.1 Detailed Description

This class allows for teleportation by clicking on a map icon.

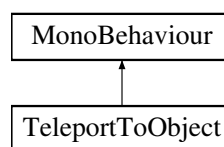
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Functionality/TeleportObject.cs

3.16 TeleportToObject Class Reference

This class allows for teleportation by clicking on a collider.

Inheritance diagram for TeleportToObject:



3.16.1 Detailed Description

This class allows for teleportation by clicking on a collider.

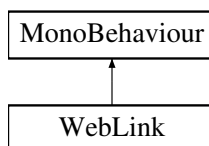
The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Functionality/TeleportToObject.cs

3.17 WebLink Class Reference

This class allows for hyperlinking clicks on colliders.

Inheritance diagram for WebLink:



Public Attributes

- string [url](#)
The URL.

3.17.1 Detailed Description

This class allows for hyperlinking clicks on colliders.

3.17.2 Member Data Documentation

3.17.2.1 string `WebLink.url`

The URL.

The documentation for this class was generated from the following file:

- C:/Users/VULCAN/Documents/GitHub/HVWC/Assets/Scripts/IDIA/Functionality/WebLink.cs

Index

- ActivateAnimation
 - AnimationController, [5](#), [6](#)
- AnimationController, [5](#)
 - ActivateAnimation, [5](#), [6](#)
 - Animations, [6](#)
 - CurrentAnimation, [6](#)
- animationController
 - PlayerController, [16](#)
- Animations
 - AnimationController, [6](#)
- avatars
 - MainMenu, [10](#)
- background
 - MainMenu, [10](#)
- C3_Tex
 - PlayerHUD, [18](#)
- C3VoiceChat, [7](#)
 - JoinVoice, [7](#)
- CharState
 - PlayerController, [17](#)
- CharacterState
 - PlayerController, [16](#)
- Chat, [7](#)
 - Messages, [8](#)
 - SendChat, [8](#)
- ClickTeleport, [9](#)
 - radius, [9](#)
- Connect
 - NetworkManager, [12](#)
- Connected
 - NetworkManager, [14](#)
- CreateRoom
 - NetworkManager, [12](#)
- CurrentAnimation
 - AnimationController, [6](#)
- CustomSkin
 - MainMenu, [10](#)
 - PlayerHUD, [18](#)
- ExitTex
 - PlayerHUD, [18](#)
- FLY
 - PlayerController, [16](#)
- fadeDistance
 - NameTag, [11](#)
- Fly
 - PlayerController, [16](#)
- flySpeed
 - PlayerController, [16](#)
- flying
 - PlayerController, [16](#)
- gravity
 - PlayerController, [17](#)
- groundLayers
 - PlayerController, [17](#)
- grounded
 - PlayerController, [17](#)
- groundingDistance
 - PlayerController, [17](#)
- heightOffset
 - PlayerController, [17](#)
- HelpTex
 - PlayerHUD, [18](#)
- IDLE
 - PlayerController, [16](#)
- icon
 - MainMenu::Avatar, [6](#)
- Instance
 - NetworkManager, [14](#)
- Instantiate
 - NetworkManager, [13](#)
- IsMessageQueueRunning
 - NetworkManager, [14](#)
- JUMP
 - PlayerController, [16](#)
- JoinRandomRoom
 - NetworkManager, [13](#)
- JoinRoom
 - NetworkManager, [13](#)
- JoinVoice
 - C3VoiceChat, [7](#)
- LeaveRoom
 - NetworkManager, [13](#)
- LoadLevel
 - NetworkManager, [13](#)
- loadingScreen
 - PlayerHUD, [18](#)
- LocalPlayer
 - PlayerList, [19](#)
- MainMenu, [9](#)
 - avatars, [10](#)
 - background, [10](#)

- CustomSkin, 10
- selectedAvatar, 10
- spawnRadius, 10
- MainMenu.Avatar, 6
- MainMenu::Avatar
 - icon, 6
 - name, 6
 - prefab, 7
- MapTex
 - PlayerHUD, 18
- Messages
 - Chat, 8
- name
 - MainMenu::Avatar, 6
- NameTag, 10
 - fadeDistance, 11
 - nameTagColor, 11
- nameTagColor
 - NameTag, 11
- NetworkManager, 11
 - Connect, 12
 - Connected, 14
 - CreateRoom, 12
 - Instance, 14
 - Instantiate, 13
 - IsMessageQueueRunning, 14
 - JoinRandomRoom, 13
 - JoinRoom, 13
 - LeaveRoom, 13
 - LoadLevel, 13
 - OfflineMode, 14
 - PlayerName, 14
 - Room, 14
 - RoomList, 14
 - version, 13
- NetworkPlayer, 14
- OfflineMode
 - NetworkManager, 14
- PlayerController
 - FLY, 16
 - IDLE, 16
 - JUMP, 16
 - RUN, 16
 - SIT, 16
 - WALK, 16
- PlayerController, 15
 - animationController, 16
 - CharState, 17
 - CharacterState, 16
 - Fly, 16
 - flySpeed, 16
 - flying, 16
 - gravity, 17
 - groundLayers, 17
 - grounded, 17
 - groundingDistance, 17
 - heightOffset, 17
 - runSpeed, 17
 - walkSpeed, 17
- PlayerHUD, 17
 - C3_Tex, 18
 - CustomSkin, 18
 - ExitTex, 18
 - HelpTex, 18
 - loadingScreen, 18
 - MapTex, 18
 - VolumeTex, 19
- PlayerList, 19
 - LocalPlayer, 19
 - Players, 19
 - RemotePlayers, 19
- PlayerName
 - NetworkManager, 14
- Players
 - PlayerList, 19
- prefab
 - MainMenu::Avatar, 7
- ProximityTrigger, 20
 - url, 20
- QuickPlayBypass, 20
- RUN
 - PlayerController, 16
- radius
 - ClickTeleport, 9
- RemotePlayers
 - PlayerList, 19
- Room
 - NetworkManager, 14
- RoomList
 - NetworkManager, 14
- runSpeed
 - PlayerController, 17
- SIT
 - PlayerController, 16
- selectedAvatar
 - MainMenu, 10
- SendChat
 - Chat, 8
- spawnRadius
 - MainMenu, 10
- TeleportObject, 21
- TeleportToObject, 21
- url
 - ProximityTrigger, 20
 - WebLink, 22
- version
 - NetworkManager, 13
- VolumeTex
 - PlayerHUD, 19

WALK

 PlayerController, [16](#)

walkSpeed

 PlayerController, [17](#)

WebLink, [21](#)

 url, [22](#)