

# Legend of the Great Unwashed

v0.1

Generated by Doxygen 1.8.9.1

Mon Nov 23 2015 13:07:47



# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Namespace Index</b>                        | <b>1</b>  |
| 1.1      | Namespace List . . . . .                      | 1         |
| <b>2</b> | <b>Hierarchical Index</b>                     | <b>3</b>  |
| 2.1      | Class Hierarchy . . . . .                     | 3         |
| <b>3</b> | <b>Class Index</b>                            | <b>5</b>  |
| 3.1      | Class List . . . . .                          | 5         |
| <b>4</b> | <b>File Index</b>                             | <b>7</b>  |
| 4.1      | File List . . . . .                           | 7         |
| <b>5</b> | <b>Namespace Documentation</b>                | <b>9</b>  |
| 5.1      | MainNS Namespace Reference . . . . .          | 9         |
| 5.1.1    | Function Documentation . . . . .              | 9         |
| 5.1.1.1  | logError . . . . .                            | 9         |
| 5.2      | mediawrap Namespace Reference . . . . .       | 9         |
| 5.2.1    | Detailed Description . . . . .                | 9         |
| 5.3      | teamusa Namespace Reference . . . . .         | 10        |
| 5.3.1    | Detailed Description . . . . .                | 11        |
| 5.3.2    | Typedef Documentation . . . . .               | 11        |
| 5.3.2.1  | ActorList . . . . .                           | 11        |
| 5.3.2.2  | AudioID . . . . .                             | 11        |
| 5.3.2.3  | BaseActorPtr . . . . .                        | 11        |
| 5.3.2.4  | Region . . . . .                              | 11        |
| 5.3.2.5  | TextureID . . . . .                           | 11        |
| 5.3.3    | Enumeration Type Documentation . . . . .      | 11        |
| 5.3.3.1  | ActorEventType . . . . .                      | 11        |
| 5.3.3.2  | CursorStyle . . . . .                         | 12        |
| <b>6</b> | <b>Class Documentation</b>                    | <b>13</b> |
| 6.1      | teamusa::ActorEvent Class Reference . . . . . | 13        |
| 6.1.1    | Detailed Description . . . . .                | 13        |

|         |  |    |
|---------|--|----|
| 6.1.2   | Constructor & Destructor Documentation . . . . . | 13 |
| 6.1.2.1 | ActorEvent . . . . .                             | 13 |
| 6.1.3   | Member Data Documentation . . . . .              | 13 |
| 6.1.3.1 | type . . . . .                                   | 13 |
| 6.1.3.2 | value . . . . .                                  | 13 |
| 6.2     | teamusa::ActorVideo Struct Reference . . . . .   | 13 |
| 6.2.1   | Constructor & Destructor Documentation . . . . . | 14 |
| 6.2.1.1 | ActorVideo . . . . .                             | 14 |
| 6.2.2   | Member Data Documentation . . . . .              | 14 |
| 6.2.2.1 | layer . . . . .                                  | 14 |
| 6.2.2.2 | textureID . . . . .                              | 14 |
| 6.3     | teamusa::AudioEngine Class Reference . . . . .   | 14 |
| 6.3.1   | Detailed Description . . . . .                   | 15 |
| 6.3.2   | Member Function Documentation . . . . .          | 15 |
| 6.3.2.1 | deleteSound . . . . .                            | 15 |
| 6.3.2.2 | deleteSoundGroup . . . . .                       | 15 |
| 6.3.2.3 | loadSound . . . . .                              | 15 |
| 6.3.2.4 | playSound . . . . .                              | 15 |
| 6.3.2.5 | playStream . . . . .                             | 15 |
| 6.3.3   | Member Data Documentation . . . . .              | 15 |
| 6.3.3.1 | audioPlayer . . . . .                            | 15 |
| 6.3.3.2 | coreResources . . . . .                          | 15 |
| 6.3.3.3 | levelResources . . . . .                         | 15 |
| 6.3.3.4 | MAX_RESERVED_ID . . . . .                        | 16 |
| 6.4     | mediawrap::AudioPlayer Class Reference . . . . . | 16 |
| 6.4.1   | Detailed Description . . . . .                   | 17 |
| 6.4.2   | Member Typedef Documentation . . . . .           | 17 |
| 6.4.2.1 | AudioID . . . . .                                | 17 |
| 6.4.3   | Constructor & Destructor Documentation . . . . . | 17 |
| 6.4.3.1 | AudioPlayer . . . . .                            | 17 |
| 6.4.3.2 | ~AudioPlayer . . . . .                           | 17 |
| 6.4.4   | Member Function Documentation . . . . .          | 17 |
| 6.4.4.1 | clear_samples . . . . .                          | 17 |
| 6.4.4.2 | delete_sample . . . . .                          | 17 |
| 6.4.4.3 | load_sample . . . . .                            | 17 |
| 6.4.4.4 | load_stream . . . . .                            | 17 |
| 6.4.4.5 | play_sample . . . . .                            | 18 |
| 6.4.4.6 | stream_audio . . . . .                           | 18 |
| 6.4.5   | Member Data Documentation . . . . .              | 18 |
| 6.4.5.1 | audio_buffer . . . . .                           | 18 |

|         |  |    |
|---------|--|----|
| 6.4.5.2 | <a href="#">audio_channels</a>                             | 18 |
| 6.4.5.3 | <a href="#">audio_format</a>                               | 18 |
| 6.4.5.4 | <a href="#">audio_rate</a>                                 | 18 |
| 6.4.5.5 | <a href="#">audio_samples</a>                              | 18 |
| 6.4.5.6 | <a href="#">audio_stream</a>                               | 18 |
| 6.5     | <a href="#">teamusa::AudioStreamActor Class Reference</a>  | 18 |
| 6.5.1   | <a href="#">Detailed Description</a>                       | 19 |
| 6.5.2   | <a href="#">Constructor &amp; Destructor Documentation</a> | 19 |
| 6.5.2.1 | <a href="#">AudioStreamActor</a>                           | 19 |
| 6.5.2.2 | <a href="#">~AudioStreamActor</a>                          | 19 |
| 6.5.3   | <a href="#">Member Function Documentation</a>              | 19 |
| 6.5.3.1 | <a href="#">getPath</a>                                    | 19 |
| 6.5.3.2 | <a href="#">step</a>                                       | 19 |
| 6.5.4   | <a href="#">Member Data Documentation</a>                  | 20 |
| 6.5.4.1 | <a href="#">activated</a>                                  | 20 |
| 6.5.4.2 | <a href="#">path</a>                                       | 20 |
| 6.6     | <a href="#">teamusa::BaseActor Class Reference</a>         | 20 |
| 6.6.1   | <a href="#">Detailed Description</a>                       | 22 |
| 6.6.2   | <a href="#">Constructor &amp; Destructor Documentation</a> | 22 |
| 6.6.2.1 | <a href="#">BaseActor</a>                                  | 22 |
| 6.6.2.2 | <a href="#">~BaseActor</a>                                 | 22 |
| 6.6.3   | <a href="#">Member Function Documentation</a>              | 22 |
| 6.6.3.1 | <a href="#">getLayer</a>                                   | 22 |
| 6.6.3.2 | <a href="#">getRegion</a>                                  | 22 |
| 6.6.3.3 | <a href="#">getTextureID</a>                               | 22 |
| 6.6.3.4 | <a href="#">hasVideo</a>                                   | 22 |
| 6.6.3.5 | <a href="#">isInBounds</a>                                 | 22 |
| 6.6.3.6 | <a href="#">onClick</a>                                    | 23 |
| 6.6.3.7 | <a href="#">onHover</a>                                    | 23 |
| 6.6.3.8 | <a href="#">setRegion</a>                                  | 23 |
| 6.6.3.9 | <a href="#">step</a>                                       | 23 |
| 6.6.4   | <a href="#">Member Data Documentation</a>                  | 24 |
| 6.6.4.1 | <a href="#">mAudioID</a>                                   | 24 |
| 6.6.4.2 | <a href="#">mRegion</a>                                    | 24 |
| 6.6.4.3 | <a href="#">mVideo</a>                                     | 24 |
| 6.7     | <a href="#">teamusa::DelayedAudioActor Class Reference</a> | 24 |
| 6.7.1   | <a href="#">Detailed Description</a>                       | 25 |
| 6.7.2   | <a href="#">Constructor &amp; Destructor Documentation</a> | 25 |
| 6.7.2.1 | <a href="#">DelayedAudioActor</a>                          | 25 |
| 6.7.2.2 | <a href="#">~DelayedAudioActor</a>                         | 25 |

|          |  |    |
|----------|--|----|
| 6.7.3    | Member Function Documentation              | 25 |
| 6.7.3.1  | step                                       | 25 |
| 6.7.4    | Member Data Documentation                  | 25 |
| 6.7.4.1  | audiold                                    | 25 |
| 6.7.4.2  | currentStep                                | 25 |
| 6.7.4.3  | delaySteps                                 | 25 |
| 6.8      | teamusa::DelayedVideoActor Class Reference | 25 |
| 6.8.1    | Detailed Description                       | 26 |
| 6.8.2    | Constructor & Destructor Documentation     | 26 |
| 6.8.2.1  | DelayedVideoActor                          | 26 |
| 6.8.2.2  | ~DelayedVideoActor                         | 26 |
| 6.8.3    | Member Function Documentation              | 26 |
| 6.8.3.1  | step                                       | 26 |
| 6.8.4    | Member Data Documentation                  | 27 |
| 6.8.4.1  | currentStep                                | 27 |
| 6.8.4.2  | delaySteps                                 | 27 |
| 6.8.4.3  | disappear                                  | 27 |
| 6.8.4.4  | textureId                                  | 27 |
| 6.9      | teamusa::Engine Class Reference            | 27 |
| 6.9.1    | Detailed Description                       | 28 |
| 6.9.2    | Member Typedef Documentation               | 28 |
| 6.9.2.1  | ActorEventHandler                          | 28 |
| 6.9.3    | Constructor & Destructor Documentation     | 28 |
| 6.9.3.1  | Engine                                     | 28 |
| 6.9.3.2  | ~Engine                                    | 28 |
| 6.9.4    | Member Function Documentation              | 28 |
| 6.9.4.1  | freeAndLoadLevel                           | 28 |
| 6.9.4.2  | getMouseClickState                         | 28 |
| 6.9.4.3  | getMouseCoordinates                        | 29 |
| 6.9.4.4  | handleEvent                                | 29 |
| 6.9.4.5  | onChangeScene                              | 29 |
| 6.9.4.6  | onDisplayText                              | 29 |
| 6.9.4.7  | onExitGame                                 | 29 |
| 6.9.4.8  | onLoadGame                                 | 29 |
| 6.9.4.9  | onLoadLevel                                | 29 |
| 6.9.4.10 | onNewGame                                  | 30 |
| 6.9.4.11 | onPlayAudio                                | 30 |
| 6.9.4.12 | onStreamAudio                              | 30 |
| 6.9.4.13 | render                                     | 30 |
| 6.9.4.14 | run  | 30 |

|          |   |    |
|----------|---|----|
| 6.9.5    | Member Data Documentation . . . . .                   | 30 |
| 6.9.5.1  | mActorEventHandlers . . . . .                         | 30 |
| 6.9.5.2  | mAudioEngine . . . . .                                | 30 |
| 6.9.5.3  | mCurrentLevelID . . . . .                             | 30 |
| 6.9.5.4  | mIsRunning . . . . .                                  | 30 |
| 6.9.5.5  | mLevel . . . . .                                      | 30 |
| 6.9.5.6  | mMainMenu . . . . .                                   | 31 |
| 6.9.5.7  | mPlayer . . . . .                                     | 31 |
| 6.9.5.8  | mSerializer . . . . .                                 | 31 |
| 6.9.5.9  | mVideoEngine . . . . .                                | 31 |
| 6.10     | teamusa::GameSaveSerializer Class Reference . . . . . | 31 |
| 6.10.1   | Detailed Description . . . . .                        | 31 |
| 6.10.2   | Constructor & Destructor Documentation . . . . .      | 31 |
| 6.10.2.1 | GameSaveSerializer . . . . .                          | 31 |
| 6.10.2.2 | ~GameSaveSerializer . . . . .                         | 31 |
| 6.10.3   | Member Function Documentation . . . . .               | 31 |
| 6.10.3.1 | load . . . . .  | 31 |
| 6.10.3.2 | save . . . . .  | 32 |
| 6.10.3.3 | saveInThread . . . . .                                | 32 |
| 6.10.3.4 | setSlot . . . . .                                     | 32 |
| 6.10.4   | Member Data Documentation . . . . .                   | 32 |
| 6.10.4.1 | fileLock . . . . .                                    | 32 |
| 6.10.4.2 | slot . . . . .  | 32 |
| 6.11     | teamusa::InventoryItemActor Class Reference . . . . . | 32 |
| 6.11.1   | Detailed Description . . . . .                        | 33 |
| 6.11.2   | Constructor & Destructor Documentation . . . . .      | 33 |
| 6.11.2.1 | InventoryItemActor . . . . .                          | 33 |
| 6.11.2.2 | ~InventoryItemActor . . . . .                         | 33 |
| 6.11.3   | Member Function Documentation . . . . .               | 33 |
| 6.11.3.1 | onClick . . . . .                                     | 33 |
| 6.11.3.2 | onHover . . . . .                                     | 33 |
| 6.11.3.3 | step . . . . .  | 33 |
| 6.11.4   | Member Data Documentation . . . . .                   | 34 |
| 6.11.4.1 | itemID . . . . .                                      | 34 |
| 6.11.4.2 | pickedUp . . . . .                                    | 34 |
| 6.12     | teamusa::Level Class Reference . . . . .              | 34 |
| 6.12.1   | Detailed Description . . . . .                        | 35 |
| 6.12.2   | Constructor & Destructor Documentation . . . . .      | 35 |
| 6.12.2.1 | Level . . . . .                                       | 35 |
| 6.12.2.2 | Level . . . . .                                       | 35 |

|           |  |    |
|-----------|--|----|
| 6.12.3    | Member Function Documentation          | 35 |
| 6.12.3.1  | changeScene                            | 35 |
| 6.12.3.2  | clearAll                               | 35 |
| 6.12.3.3  | getActors                              | 35 |
| 6.12.3.4  | getBGImageID                           | 36 |
| 6.12.3.5  | getScene                               | 36 |
| 6.12.3.6  | loadLevel                              | 36 |
| 6.12.3.7  | parseAudioStreamActor                  | 36 |
| 6.12.3.8  | parseDelayedAudioActor                 | 36 |
| 6.12.3.9  | parseDelayedVideoActor                 | 36 |
| 6.12.3.10 | parseInventoryItemActor                | 36 |
| 6.12.3.11 | parseLevelLink                         | 36 |
| 6.12.3.12 | parseMovingActor                       | 36 |
| 6.12.3.13 | parseResponsiveAudioActor              | 36 |
| 6.12.3.14 | parseResponsiveVideoActor              | 36 |
| 6.12.3.15 | parseSceneLink                         | 36 |
| 6.12.3.16 | parseTextboxSpawnActor                 | 36 |
| 6.12.3.17 | parseVideoActor                        | 36 |
| 6.12.3.18 | parseVideoEventActor                   | 36 |
| 6.12.4    | Member Data Documentation              | 36 |
| 6.12.4.1  | activeScene                            | 36 |
| 6.12.4.2  | scenes                                 | 36 |
| 6.12.4.3  | startScene                             | 36 |
| 6.13      | teamusa::LevelLink Class Reference     | 37 |
| 6.13.1    | Constructor & Destructor Documentation | 37 |
| 6.13.1.1  | LevelLink                              | 37 |
| 6.13.1.2  | ~LevelLink                             | 37 |
| 6.13.2    | Member Function Documentation          | 37 |
| 6.13.2.1  | getSceneID                             | 37 |
| 6.13.2.2  | getText                                | 38 |
| 6.13.2.3  | onClick                                | 38 |
| 6.13.2.4  | onHover                                | 38 |
| 6.13.2.5  | step                                   | 38 |
| 6.13.3    | Member Data Documentation              | 38 |
| 6.13.3.1  | itemRequiredText                       | 38 |
| 6.13.3.2  | levelID                                | 38 |
| 6.13.3.3  | requiredItemID                         | 38 |
| 6.13.3.4  | sceneID                                | 38 |
| 6.14      | teamusa::MovingActor Class Reference   | 39 |
| 6.14.1    | Detailed Description                   | 39 |



|          |  |    |
|----------|--|----|
| 6.14.2   | Constructor & Destructor Documentation | 40 |
| 6.14.2.1 | MovingActor                            | 40 |
| 6.14.2.2 | ~MovingActor                           | 40 |
| 6.14.3   | Member Function Documentation          | 40 |
| 6.14.3.1 | onClick                                | 40 |
| 6.14.3.2 | onHover                                | 40 |
| 6.14.3.3 | step                                   | 40 |
| 6.14.4   | Member Data Documentation              | 40 |
| 6.14.4.1 | currentStep                            | 40 |
| 6.14.4.2 | endRegion                              | 40 |
| 6.14.4.3 | hGrowth                                | 41 |
| 6.14.4.4 | isActive                               | 41 |
| 6.14.4.5 | transitionSteps                        | 41 |
| 6.14.4.6 | wGrowth                                | 41 |
| 6.14.4.7 | xSpeed                                 | 41 |
| 6.14.4.8 | ySpeed                                 | 41 |
| 6.15     | teamusa::Player Class Reference        | 41 |
| 6.15.1   | Detailed Description                   | 42 |
| 6.15.2   | Member Typedef Documentation           | 42 |
| 6.15.2.1 | Inventory                              | 42 |
| 6.15.3   | Constructor & Destructor Documentation | 42 |
| 6.15.3.1 | Player                                 | 42 |
| 6.15.3.2 | ~Player                                | 42 |
| 6.15.4   | Member Function Documentation          | 42 |
| 6.15.4.1 | addItem                                | 42 |
| 6.15.4.2 | getCursorTextureID                     | 42 |
| 6.15.4.3 | getInventory                           | 42 |
| 6.15.4.4 | getPosition                            | 43 |
| 6.15.4.5 | hasItem                                | 43 |
| 6.15.4.6 | setCursor                              | 43 |
| 6.15.4.7 | setInventory                           | 43 |
| 6.15.4.8 | setPosition                            | 43 |
| 6.15.4.9 | setPosition                            | 43 |
| 6.15.5   | Member Data Documentation              | 44 |
| 6.15.5.1 | CURSOR_DEFAULT_ID                      | 44 |
| 6.15.5.2 | CURSOR_DOWN_ID                         | 44 |
| 6.15.5.3 | CURSOR_LEFT_ID                         | 44 |
| 6.15.5.4 | CURSOR_RIGHT_ID                        | 44 |
| 6.15.5.5 | CURSOR_SELECT_ID                       | 44 |
| 6.15.5.6 | CURSOR_UP_ID                           | 44 |

|           |   |    |
|-----------|---|----|
| 6.15.5.7  | FLASHLIGHT_ID                                 | 44 |
| 6.15.5.8  | mCursorStyle                                  | 44 |
| 6.15.5.9  | mInventory                                    | 44 |
| 6.15.5.10 | mLayer  | 44 |
| 6.15.5.11 | MOUSE_CLICK_ID                                | 44 |
| 6.15.5.12 | mPosition                                     | 44 |
| 6.15.5.13 | mRegion                                       | 44 |
| 6.15.5.14 | mTextureID                                    | 44 |
| 6.16      | teamusa::Point Class Reference                | 44 |
| 6.16.1    | Detailed Description                          | 45 |
| 6.16.2    | Constructor & Destructor Documentation        | 45 |
| 6.16.2.1  | Point   | 45 |
| 6.16.2.2  | Point   | 45 |
| 6.16.3    | Member Data Documentation                     | 45 |
| 6.16.3.1  | x   | 45 |
| 6.16.3.2  | y   | 45 |
| 6.17      | teamusa::ResponsiveAudioActor Class Reference | 45 |
| 6.17.1    | Detailed Description                          | 46 |
| 6.17.2    | Constructor & Destructor Documentation        | 46 |
| 6.17.2.1  | ResponsiveAudioActor                          | 46 |
| 6.17.2.2  | ~ResponsiveAudioActor                         | 46 |
| 6.17.3    | Member Function Documentation                 | 46 |
| 6.17.3.1  | onClick                                       | 46 |
| 6.17.3.2  | onHover                                       | 46 |
| 6.17.3.3  | step  | 46 |
| 6.17.4    | Member Data Documentation                     | 47 |
| 6.17.4.1  | clickAudioId                                  | 47 |
| 6.17.4.2  | hoverAudioId                                  | 47 |
| 6.18      | teamusa::ResponsiveVideoActor Class Reference | 47 |
| 6.18.1    | Constructor & Destructor Documentation        | 47 |
| 6.18.1.1  | ResponsiveVideoActor                          | 47 |
| 6.18.1.2  | ~ResponsiveVideoActor                         | 47 |
| 6.18.2    | Member Function Documentation                 | 48 |
| 6.18.2.1  | onClick                                       | 48 |
| 6.18.2.2  | onHover                                       | 49 |
| 6.18.2.3  | setTextureId                                  | 49 |
| 6.18.2.4  | step  | 49 |
| 6.18.3    | Member Data Documentation                     | 49 |
| 6.18.3.1  | clickTexture                                  | 49 |
| 6.18.3.2  | defaultTextureId                              | 49 |

|          |  |    |
|----------|--|----|
| 6.18.3.3 | hoverTexture                               | 49 |
| 6.19     | teamusa::Level::Scene Class Reference      | 50 |
| 6.19.1   | Detailed Description                       | 50 |
| 6.19.2   | Member Data Documentation                  | 50 |
| 6.19.2.1 | actors                                     | 50 |
| 6.19.2.2 | bglImageID                                 | 50 |
| 6.20     | teamusa::SceneLink Class Reference         | 50 |
| 6.20.1   | Detailed Description                       | 51 |
| 6.20.2   | Constructor & Destructor Documentation     | 51 |
| 6.20.2.1 | SceneLink                                  | 51 |
| 6.20.2.2 | ~SceneLink                                 | 51 |
| 6.20.3   | Member Function Documentation              | 51 |
| 6.20.3.1 | getText                                    | 51 |
| 6.20.3.2 | onClick                                    | 51 |
| 6.20.3.3 | onHover                                    | 51 |
| 6.20.3.4 | step                                       | 52 |
| 6.20.4   | Member Data Documentation                  | 53 |
| 6.20.4.1 | cursorStyle                                | 53 |
| 6.20.4.2 | itemRequiredText                           | 53 |
| 6.20.4.3 | requiredItemID                             | 53 |
| 6.20.4.4 | sceneID                                    | 53 |
| 6.21     | teamusa::TextboxSpawnActor Class Reference | 53 |
| 6.21.1   | Detailed Description                       | 54 |
| 6.21.2   | Constructor & Destructor Documentation     | 54 |
| 6.21.2.1 | TextboxSpawnActor                          | 54 |
| 6.21.2.2 | ~TextboxSpawnActor                         | 54 |
| 6.21.3   | Member Function Documentation              | 54 |
| 6.21.3.1 | getText                                    | 54 |
| 6.21.3.2 | onClick                                    | 54 |
| 6.21.3.3 | step                                       | 54 |
| 6.21.4   | Member Data Documentation                  | 54 |
| 6.21.4.1 | activated                                  | 54 |
| 6.21.4.2 | text                                       | 54 |
| 6.22     | teamusa::Timer Class Reference             | 55 |
| 6.22.1   | Detailed Description                       | 55 |
| 6.22.2   | Constructor & Destructor Documentation     | 55 |
| 6.22.2.1 | Timer                                      | 55 |
| 6.22.2.2 | ~Timer                                     | 55 |
| 6.22.3   | Member Function Documentation              | 55 |
| 6.22.3.1 | getTicks                                   | 55 |

|           |   |    |
|-----------|---|----|
| 6.22.3.2  | pause                                   | 56 |
| 6.22.3.3  | start                                   | 56 |
| 6.22.3.4  | stop                                    | 56 |
| 6.22.3.5  | unpause                                 | 56 |
| 6.22.4    | Member Data Documentation               | 56 |
| 6.22.4.1  | mPaused                                 | 56 |
| 6.22.4.2  | mPauseTicks                             | 56 |
| 6.22.4.3  | mStarted                                | 56 |
| 6.22.4.4  | mStartTicks                             | 56 |
| 6.23      | teamusa::VideoActor Class Reference     | 56 |
| 6.23.1    | Constructor & Destructor Documentation  | 57 |
| 6.23.1.1  | VideoActor                              | 57 |
| 6.23.1.2  | ~VideoActor                             | 57 |
| 6.23.2    | Member Function Documentation           | 57 |
| 6.23.2.1  | step                                    | 57 |
| 6.24      | mediawrap::VideoContext Class Reference | 57 |
| 6.24.1    | Detailed Description                    | 58 |
| 6.24.2    | Member Typedef Documentation            | 59 |
| 6.24.2.1  | Region                                  | 59 |
| 6.24.2.2  | texture_iter                            | 59 |
| 6.24.2.3  | TextureID                               | 59 |
| 6.24.3    | Member Enumeration Documentation        | 59 |
| 6.24.3.1  | BlendMode                               | 59 |
| 6.24.3.2  | DebugColor                              | 59 |
| 6.24.3.3  | Flip                                    | 59 |
| 6.24.4    | Constructor & Destructor Documentation  | 60 |
| 6.24.4.1  | VideoContext                            | 60 |
| 6.24.4.2  | ~VideoContext                           | 61 |
| 6.24.5    | Member Function Documentation           | 61 |
| 6.24.5.1  | create_texture                          | 61 |
| 6.24.5.2  | delete_texture                          | 61 |
| 6.24.5.3  | display                                 | 61 |
| 6.24.5.4  | fill_texture                            | 61 |
| 6.24.5.5  | load_font                               | 62 |
| 6.24.5.6  | load_texture                            | 62 |
| 6.24.5.7  | render                                  | 62 |
| 6.24.5.8  | render_clear                            | 62 |
| 6.24.5.9  | render_clear                            | 62 |
| 6.24.5.10 | render_onto                             | 63 |
| 6.24.5.11 | render_rotate                           | 63 |

|           |   |    |
|-----------|---|----|
| 6.24.5.12 | render_text                             | 63 |
| 6.24.5.13 | renderDebugBox                          | 63 |
| 6.24.5.14 | swapFullscreen                          | 63 |
| 6.24.6    | Member Data Documentation               | 63 |
| 6.24.6.1  | font                                    | 64 |
| 6.24.6.2  | renderer                                | 64 |
| 6.24.6.3  | textures                                | 64 |
| 6.24.6.4  | video_display                           | 64 |
| 6.25      | mediawrap::VideoDisplay Class Reference | 64 |
| 6.25.1    | Detailed Description                    | 64 |
| 6.25.2    | Constructor & Destructor Documentation  | 64 |
| 6.25.2.1  | VideoDisplay                            | 64 |
| 6.25.2.2  | ~VideoDisplay                           | 65 |
| 6.25.3    | Member Function Documentation           | 65 |
| 6.25.3.1  | get_renderer                            | 65 |
| 6.25.3.2  | swapFullscreen                          | 65 |
| 6.25.4    | Member Data Documentation               | 65 |
| 6.25.4.1  | window                                  | 65 |
| 6.26      | teamusa::VideoEngine Class Reference    | 65 |
| 6.26.1    | Detailed Description                    | 66 |
| 6.26.2    | Constructor & Destructor Documentation  | 66 |
| 6.26.2.1  | VideoEngine                             | 66 |
| 6.26.2.2  | ~VideoEngine                            | 67 |
| 6.26.3    | Member Function Documentation           | 67 |
| 6.26.3.1  | clearLayers                             | 67 |
| 6.26.3.2  | deleteResourceGroup                     | 67 |
| 6.26.3.3  | deleteTexture                           | 67 |
| 6.26.3.4  | display                                 | 67 |
| 6.26.3.5  | hideTextbox                             | 67 |
| 6.26.3.6  | isShowingTextbox                        | 67 |
| 6.26.3.7  | loadTexture                             | 68 |
| 6.26.3.8  | render                                  | 69 |
| 6.26.3.9  | renderDebugBox                          | 69 |
| 6.26.3.10 | renderRotate                            | 69 |
| 6.26.3.11 | showTextbox                             | 69 |
| 6.26.3.12 | swapFullscreen                          | 69 |
| 6.26.4    | Member Data Documentation               | 69 |
| 6.26.4.1  | coreResources                           | 69 |
| 6.26.4.2  | layers                                  | 69 |
| 6.26.4.3  | levelResources                          | 69 |

|           |  |           |
|-----------|--|-----------|
| 6.26.4.4  | MAX_RESERVED_ID                          | 69        |
| 6.26.4.5  | NUM_LAYERS                               | 69        |
| 6.26.4.6  | SHADOW_LAYER                             | 70        |
| 6.26.4.7  | TEXT_LAYER                               | 70        |
| 6.26.4.8  | textboxActive                            | 70        |
| 6.26.4.9  | textboxPadding                           | 70        |
| 6.26.4.10 | textboxRegion                            | 70        |
| 6.26.4.11 | videoContext                             | 70        |
| 6.27      | teamusa::VideoEventActor Class Reference | 70        |
| 6.27.1    | Detailed Description                     | 70        |
| 6.27.2    | Constructor & Destructor Documentation   | 71        |
| 6.27.2.1  | VideoEventActor                          | 71        |
| 6.27.2.2  | ~VideoEventActor                         | 71        |
| 6.27.3    | Member Function Documentation            | 71        |
| 6.27.3.1  | onClick                                  | 71        |
| 6.27.3.2  | onHover                                  | 71        |
| 6.27.3.3  | step                                     | 71        |
| 6.27.4    | Member Data Documentation                | 71        |
| 6.27.4.1  | actorEvent                               | 71        |
| <b>7</b>  | <b>File Documentation</b>                | <b>73</b> |
| 7.1       | ActorEvent.h File Reference              | 73        |
| 7.1.1     | Detailed Description                     | 73        |
| 7.2       | Assert.h File Reference                  | 73        |
| 7.2.1     | Detailed Description                     | 74        |
| 7.2.2     | Macro Definition Documentation           | 74        |
| 7.2.2.1   | Assert                                   | 74        |
| 7.3       | AudioEngine.cpp File Reference           | 74        |
| 7.4       | AudioEngine.hpp File Reference           | 74        |
| 7.5       | AudioPlayer.cpp File Reference           | 74        |
| 7.6       | AudioPlayer.hpp File Reference           | 75        |
| 7.7       | AudioStreamActor.cpp File Reference      | 75        |
| 7.7.1     | Detailed Description                     | 75        |
| 7.8       | AudioStreamActor.h File Reference        | 75        |
| 7.8.1     | Detailed Description                     | 76        |
| 7.9       | BaseActor.cpp File Reference             | 76        |
| 7.10      | BaseActor.h File Reference               | 76        |
| 7.10.1    | Detailed Description                     | 76        |
| 7.11      | CursorStyle.h File Reference             | 76        |
| 7.11.1    | Detailed Description                     | 77        |

|  |    |
|--|----|
| 7.12 DelayedAudioActor.cpp File Reference . . . . .  | 77 |
| 7.12.1 Detailed Description . . . . .                | 77 |
| 7.13 DelayedAudioActor.h File Reference . . . . .    | 77 |
| 7.13.1 Detailed Description . . . . .                | 77 |
| 7.14 DelayedVideoActor.cpp File Reference . . . . .  | 77 |
| 7.14.1 Detailed Description . . . . .                | 78 |
| 7.15 DelayedVideoActor.h File Reference . . . . .    | 78 |
| 7.15.1 Detailed Description . . . . .                | 78 |
| 7.16 Engine.cpp File Reference . . . . .             | 78 |
| 7.16.1 Detailed Description . . . . .                | 79 |
| 7.16.2 Macro Definition Documentation . . . . .      | 79 |
| 7.16.2.1 BIND . . . . .                              | 79 |
| 7.16.3 Variable Documentation . . . . .              | 79 |
| 7.16.3.1 FRAME_TIME . . . . .                        | 79 |
| 7.17 Engine.h File Reference . . . . .               | 79 |
| 7.17.1 Detailed Description . . . . .                | 79 |
| 7.18 GameSaveSerializer.cpp File Reference . . . . . | 79 |
| 7.19 GameSaveSerializer.h File Reference . . . . .   | 80 |
| 7.19.1 Detailed Description . . . . .                | 80 |
| 7.20 Headers.h File Reference . . . . .              | 80 |
| 7.20.1 Detailed Description . . . . .                | 80 |
| 7.21 InventoryItemActor.cpp File Reference . . . . . | 80 |
| 7.21.1 Detailed Description . . . . .                | 81 |
| 7.22 InventoryItemActor.h File Reference . . . . .   | 81 |
| 7.22.1 Detailed Description . . . . .                | 81 |
| 7.23 Level.cpp File Reference . . . . .              | 81 |
| 7.23.1 Detailed Description . . . . .                | 82 |
| 7.23.2 Function Documentation . . . . .              | 82 |
| 7.23.2.1 loadError . . . . .                         | 82 |
| 7.23.2.2 operator>> . . . . .                        | 82 |
| 7.23.2.3 operator>> . . . . .                        | 82 |
| 7.24 Level.h File Reference . . . . .                | 82 |
| 7.24.1 Detailed Description . . . . .                | 82 |
| 7.25 LevelLink.cpp File Reference . . . . .          | 82 |
| 7.25.1 Detailed Description . . . . .                | 83 |
| 7.26 LevelLink.h File Reference . . . . .            | 83 |
| 7.26.1 Detailed Description . . . . .                | 83 |
| 7.27 main.cpp File Reference . . . . .               | 83 |
| 7.27.1 Detailed Description . . . . .                | 83 |
| 7.27.2 Function Documentation . . . . .              | 84 |

|          |   |    |
|----------|---|----|
| 7.27.2.1 | main                                    | 84 |
| 7.28     | MovingActor.cpp File Reference          | 84 |
| 7.29     | MovingActor.h File Reference            | 84 |
| 7.29.1   | Detailed Description                    | 84 |
| 7.30     | Player.cpp File Reference               | 84 |
| 7.30.1   | Detailed Description                    | 84 |
| 7.31     | Player.h File Reference                 | 84 |
| 7.31.1   | Detailed Description                    | 85 |
| 7.32     | Point.h File Reference                  | 85 |
| 7.32.1   | Detailed Description                    | 85 |
| 7.33     | ResourceGroup.hpp File Reference        | 85 |
| 7.33.1   | Enumeration Type Documentation          | 86 |
| 7.33.1.1 | ResourceGroup                           | 86 |
| 7.34     | ResponsiveAudioActor.cpp File Reference | 86 |
| 7.34.1   | Detailed Description                    | 86 |
| 7.35     | ResponsiveAudioActor.h File Reference   | 86 |
| 7.35.1   | Detailed Description                    | 86 |
| 7.36     | ResponsiveVideoActor.cpp File Reference | 86 |
| 7.36.1   | Detailed Description                    | 86 |
| 7.37     | ResponsiveVideoActor.h File Reference   | 87 |
| 7.37.1   | Detailed Description                    | 87 |
| 7.38     | SceneLink.cpp File Reference            | 87 |
| 7.38.1   | Detailed Description                    | 87 |
| 7.39     | SceneLink.h File Reference              | 87 |
| 7.39.1   | Detailed Description                    | 88 |
| 7.40     | TextboxSpawnActor.cpp File Reference    | 88 |
| 7.40.1   | Detailed Description                    | 88 |
| 7.41     | TextboxSpawnActor.h File Reference      | 88 |
| 7.41.1   | Detailed Description                    | 88 |
| 7.42     | Timer.cpp File Reference                | 88 |
| 7.42.1   | Detailed Description                    | 88 |
| 7.43     | Timer.h File Reference                  | 89 |
| 7.43.1   | Detailed Description                    | 89 |
| 7.44     | VideoActor.cpp File Reference           | 89 |
| 7.44.1   | Detailed Description                    | 89 |
| 7.45     | VideoActor.h File Reference             | 89 |
| 7.45.1   | Detailed Description                    | 90 |
| 7.46     | VideoContext.cpp File Reference         | 90 |
| 7.47     | VideoContext.hpp File Reference         | 90 |
| 7.48     | VideoDisplay.cpp File Reference         | 90 |



---

|              |  |           |
|--------------|--|-----------|
| 7.49         | VideoDisplay.hpp File Reference . . . . .    | 90        |
| 7.50         | VideoEngine.cpp File Reference . . . . .     | 91        |
| 7.51         | VideoEngine.hpp File Reference . . . . .     | 91        |
| 7.52         | VideoEventActor.cpp File Reference . . . . . | 91        |
| 7.52.1       | Detailed Description . . . . .               | 91        |
| 7.53         | VideoEventActor.h File Reference . . . . .   | 92        |
| 7.53.1       | Detailed Description . . . . .               | 92        |
| <b>Index</b> |  | <b>93</b> |



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

|                           |  |    |
|---------------------------|--|----|
| <a href="#">MainNS</a>    | .....  | 9  |
| <a href="#">mediawrap</a> |  |    |
|                           | Provides basic audio playing capabilities with WAV files                       | 9  |
| <a href="#">teamusa</a>   |  |    |
|                           | Provides project-specific audio functionality for Legend of the Great Unwashed | 10 |



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

|   |    |
|---|----|
| teamusa::ActorEvent . . . . .           | 13 |
| teamusa::ActorVideo . . . . .           | 13 |
| teamusa::AudioEngine . . . . .          | 14 |
| mediawrap::AudioPlayer . . . . .        | 16 |
| teamusa::BaseActor . . . . .            | 20 |
| teamusa::AudioStreamActor . . . . .     | 18 |
| teamusa::DelayedAudioActor . . . . .    | 24 |
| teamusa::DelayedVideoActor . . . . .    | 25 |
| teamusa::InventoryItemActor . . . . .   | 32 |
| teamusa::LevelLink . . . . .            | 37 |
| teamusa::MovingActor . . . . .          | 39 |
| teamusa::ResponsiveAudioActor . . . . . | 45 |
| teamusa::ResponsiveVideoActor . . . . . | 47 |
| teamusa::SceneLink . . . . .            | 50 |
| teamusa::TextboxSpawnActor . . . . .    | 53 |
| teamusa::VideoActor . . . . .           | 56 |
| teamusa::VideoEventActor . . . . .      | 70 |
| teamusa::Engine . . . . .               | 27 |
| teamusa::GameSaveSerializer . . . . .   | 31 |
| teamusa::Level . . . . .                | 34 |
| teamusa::Player . . . . .               | 41 |
| teamusa::Point . . . . .                | 44 |
| teamusa::Level::Scene . . . . .         | 50 |
| teamusa::Timer . . . . .                | 55 |
| mediawrap::VideoContext . . . . .       | 57 |
| mediawrap::VideoDisplay . . . . .       | 64 |
| teamusa::VideoEngine . . . . .          | 65 |



## Chapter 3

# Class Index

### 3.1 Class List

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

|   |   |    |
|---|---|----|
| <a href="#">teamusa::ActorEvent</a>           | Event data generated by Actors, handled by <a href="#">Engine</a> . . . . .   | 13 |
| <a href="#">teamusa::ActorVideo</a>           | . . . . .   | 13 |
| <a href="#">teamusa::AudioEngine</a>          | Provides project-specific audio functionality for Legend of the Great Unwashed . . . . .  | 14 |
| <a href="#">mediawrap::AudioPlayer</a>        | Provides basic audio playing capabilities with WAV files . . . . .  | 16 |
| <a href="#">teamusa::AudioStreamActor</a>     | If this actor is not activated, it will emit a StreamAudio event and set its status to activated when the step method is called . . . . . | 18 |
| <a href="#">teamusa::BaseActor</a>            | Abstract class which all actors must derive from . . . . .  | 20 |
| <a href="#">teamusa::DelayedAudioActor</a>    | Will increment a counter every time the step method is called . . . . .   | 24 |
| <a href="#">teamusa::DelayedVideoActor</a>    | Will increment a counter every time the step method is called . . . . .   | 25 |
| <a href="#">teamusa::Engine</a>               | Processes all components of the game each frame . . . . .   | 27 |
| <a href="#">teamusa::GameSaveSerializer</a>   | Provides multithreaded save, single-thread load of save files . . . . .   | 31 |
| <a href="#">teamusa::InventoryItemActor</a>   | InventoryItemActor creates a collectible item in the game environment . . . . .   | 32 |
| <a href="#">teamusa::Level</a>                | A <a href="#">Level</a> is a container of Scenes and Actors corresponding to those scenes . . . . .                                       | 34 |
| <a href="#">teamusa::LevelLink</a>            | . . . . .   | 37 |
| <a href="#">teamusa::MovingActor</a>          | Will transition from one region to the next by calculating the distance to move each frame for a set number of frames . . . . .           | 39 |
| <a href="#">teamusa::Player</a>               | Handles all data relevant to the player engaging the game . . . . .   | 41 |
| <a href="#">teamusa::Point</a>                | An (x,y) coordinate within the rendering window . . . . .   | 44 |
| <a href="#">teamusa::ResponsiveAudioActor</a> | \ Brief: Will increment the value of stepCount until it is equal to durationSteps for each call to the step method . . . . .              | 45 |
| <a href="#">teamusa::ResponsiveVideoActor</a> | . . . . .   | 47 |
| <a href="#">teamusa::Level::Scene</a>         | A scene is a collection of images (Actors) that is displayed on the screen . . . . .  | 50 |

|   |    |
|---|----|
| <a href="#">teamusa::SceneLink</a>  |    |
| Allows the player to transition between scenes . . . . .                                | 50 |
| <a href="#">teamusa::TextboxSpawnActor</a>  |    |
| Spawns a textbox that gives the player written information . . . . .                    | 53 |
| <a href="#">teamusa::Timer</a>  |    |
| A timer that counts up from zero in milliseconds . . . . .                              | 55 |
| <a href="#">teamusa::VideoActor</a> . . . . .   | 56 |
| <a href="#">mediawrap::VideoContext</a>   |    |
| Provides basic 2D rendering capabilities . . . . .                                      | 57 |
| <a href="#">mediawrap::VideoDisplay</a>   |    |
| Creates a window and initializes SDL2 and SDL2_IMG . . . . .                            | 64 |
| <a href="#">teamusa::VideoEngine</a>  |    |
| Provides video capabilities that are specific to Legend of the Great Unwashed . . . . . | 65 |
| <a href="#">teamusa::VideoEventActor</a>  |    |
| Will display a texture and perform no action until clicked . . . . .                    | 70 |



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

|  |  |    |
|--|--|----|
| <a href="#">ActorEvent.h</a>           | Declares ActorEvent struct . . . . .             | 73 |
| <a href="#">Assert.h</a>               | Declares custom Assert macro . . . . .           | 73 |
| <a href="#">AudioEngine.cpp</a>        | . . . . .  | 74 |
| <a href="#">AudioEngine.hpp</a>        | . . . . .  | 74 |
| <a href="#">AudioPlayer.cpp</a>        | . . . . .  | 74 |
| <a href="#">AudioPlayer.hpp</a>        | . . . . .  | 75 |
| <a href="#">AudioStreamActor.cpp</a>   | Implements AudioStreamActor class . . . . .      | 75 |
| <a href="#">AudioStreamActor.h</a>     | Declares AudioStreamActor class . . . . .        | 75 |
| <a href="#">BaseActor.cpp</a>          | . . . . .  | 76 |
| <a href="#">BaseActor.h</a>            | Implements BaseActor class . . . . .             | 76 |
| <a href="#">CursorStyle.h</a>          | Declares CursorStyle enumerations . . . . .      | 76 |
| <a href="#">DelayedAudioActor.cpp</a>  | Implements DelayedAudioActor class . . . . .     | 77 |
| <a href="#">DelayedAudioActor.h</a>    | Declares DelayedAudioActor class . . . . .       | 77 |
| <a href="#">DelayedVideoActor.cpp</a>  | Declares DelayedVideoActor class . . . . .       | 77 |
| <a href="#">DelayedVideoActor.h</a>    | Declares DelayedVideoActor class . . . . .       | 78 |
| <a href="#">Engine.cpp</a>             | Implements Engine class . . . . .                | 78 |
| <a href="#">Engine.h</a>               | Declares Engine class . . . . .                  | 79 |
| <a href="#">GameSaveSerializer.cpp</a> | . . . . .  | 79 |
| <a href="#">GameSaveSerializer.h</a>   | Declares save file serializer class . . . . .    | 80 |
| <a href="#">Headers.h</a>              | Easy way to include all headers needed . . . . . | 80 |
| <a href="#">InventoryItemActor.cpp</a> | Implements InventoryItemActor class . . . . .    | 80 |
| <a href="#">InventoryItemActor.h</a>   | Declares InventoryItemActor class . . . . .      | 81 |

|  |  |    |
|--|--|----|
| <a href="#">Level.cpp</a>                | Implements Level class . . . . .   | 81 |
| <a href="#">Level.h</a>                  | Declares Level class . . . . .   | 82 |
| <a href="#">LevelLink.cpp</a>            | Implements LevelLink class . . . . .   | 82 |
| <a href="#">LevelLink.h</a>              | Declares LevelLink class . . . . .   | 83 |
| <a href="#">main.cpp</a>                 | Entry point of program . . . . .   | 83 |
| <a href="#">MovingActor.cpp</a>          | . . . . .  | 84 |
| <a href="#">MovingActor.h</a>            | Declares movingActor class . . . . .   | 84 |
| <a href="#">Player.cpp</a>               | Implements Player class . . . . .  | 84 |
| <a href="#">Player.h</a>                 | Declares Player class . . . . .  | 84 |
| <a href="#">Point.h</a>                  | Declares Point struct . . . . .  | 85 |
| <a href="#">ResourceGroup.hpp</a>        | . . . . .  | 85 |
| <a href="#">ResponsiveAudioActor.cpp</a> | Implements ResponsiveAudioActor class . . . . .  | 86 |
| <a href="#">ResponsiveAudioActor.h</a>   | Declares ResponsiveAudioActor class . . . . .  | 86 |
| <a href="#">ResponsiveVideoActor.cpp</a> | Will display the default TextureID . . . . .   | 86 |
| <a href="#">ResponsiveVideoActor.h</a>   | Declares ResponsivevideoActor class . . . . .  | 87 |
| <a href="#">SceneLink.cpp</a>            | Implements SceneLink class . . . . .   | 87 |
| <a href="#">SceneLink.h</a>              | Declares SceneLink class . . . . .   | 87 |
| <a href="#">TextboxSpawnActor.cpp</a>    | Declares TextboxSpawnActor class . . . . .   | 88 |
| <a href="#">TextboxSpawnActor.h</a>      | Declares TextboxSpawnActor class . . . . .   | 88 |
| <a href="#">Timer.cpp</a>                | Implements Timer class . . . . .   | 88 |
| <a href="#">Timer.h</a>                  | Declares Timer class . . . . .   | 89 |
| <a href="#">VideoActor.cpp</a>           | Implements VideoActor class . . . . .  | 89 |
| <a href="#">VideoActor.h</a>             | This module makes sure An actor that will only display a texture at a given region . . . . . | 89 |
| <a href="#">VideoContext.cpp</a>         | . . . . .  | 90 |
| <a href="#">VideoContext.hpp</a>         | . . . . .  | 90 |
| <a href="#">VideoDisplay.cpp</a>         | . . . . .  | 90 |
| <a href="#">VideoDisplay.hpp</a>         | . . . . .  | 90 |
| <a href="#">VideoEngine.cpp</a>          | . . . . .  | 91 |
| <a href="#">VideoEngine.hpp</a>          | . . . . .  | 91 |
| <a href="#">VideoEventActor.cpp</a>      | Declares VideoEventActor class . . . . .   | 91 |
| <a href="#">VideoEventActor.h</a>        | Declares VideoEventActor class . . . . .   | 92 |

## Chapter 5

# Namespace Documentation

### 5.1 MainNS Namespace Reference

#### Functions

- static void [logError](#) (const std::string &desc)  
*Writes an error message to the log file.*

#### 5.1.1 Function Documentation

5.1.1.1 static void MainNS::logError ( const std::string & desc ) [static]

Writes an error message to the log file.

##### Parameters

|             |  |
|-------------|--|
| <i>desc</i> | The string containing the error message. |
|-------------|--|

### 5.2 mediawrap Namespace Reference

Provides basic audio playing capabilities with WAV files.

#### Classes

- class [AudioPlayer](#)  
*Provides basic audio playing capabilities with WAV files.*
- class [VideoContext](#)  
*Provides basic 2D rendering capabilities.*
- class [VideoDisplay](#)  
*Creates a window and initializes SDL2 and SDL2\_IMG.*

#### 5.2.1 Detailed Description

Provides basic audio playing capabilities with WAV files.

Creates a window and initializes SDL2 and SDL2\_IMG.

Provides basic 2D rendering capabilities.

## 5.3 teamusa Namespace Reference

Provides project-specific audio functionality for Legend of the Great Unwashed.

### Classes

- class [ActorEvent](#)  
*Event data generated by Actors, handled by [Engine](#).*
- struct [ActorVideo](#)
- class [AudioEngine](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*
- class [AudioStreamActor](#)  
*If this actor is not activated, it will emit a [StreamAudio](#) event and set its status to activated when the step method is called.*
- class [BaseActor](#)  
*Abstract class which all actors must derive from.*
- class [DelayedAudioActor](#)  
*Will increment a counter every time the step method is called.*
- class [DelayedVideoActor](#)  
*Will increment a counter every time the step method is called.*
- class [Engine](#)  
*Processes all components of the game each frame.*
- class [GameSaveSerializer](#)  
*Provides multithreaded save, single-thread load of save files.*
- class [InventoryItemActor](#)  
*InventoryItemActor creates a collectible item in the game environment.*
- class [Level](#)  
*A [Level](#) is a container of Scenes and Actors corresponding to those scenes.*
- class [LevelLink](#)
- class [MovingActor](#)  
*Will transition from one region to the next by calculating the distance to move each frame for a set number of frames.*
- class [Player](#)  
*Handles all data relevant to the player engaging the game.*
- class [Point](#)  
*An (x,y) coordinate within the rendering window.*
- class [ResponsiveAudioActor](#)  
*\ Brief: Will increment the value of stepCount until it is equal to durationSteps for each call to the step method.*
- class [ResponsiveVideoActor](#)
- class [SceneLink](#)  
*Allows the player to transition between scenes.*
- class [TextboxSpawnActor](#)  
*Spawns a textbox that gives the player written information.*
- class [Timer](#)  
*A timer that counts up from zero in milliseconds.*
- class [VideoActor](#)
- class [VideoEngine](#)  
*Provides video capabilities that are specific to Legend of the Great Unwashed.*
- class [VideoEventActor](#)  
*Will display a texture and perform no action until clicked.*

## Typedefs

- typedef [mediawrap::AudioPlayer::AudioID](#) AudioID
- typedef std::shared\_ptr< [BaseActor](#) > [BaseActorPtr](#)
- typedef std::vector< [BaseActorPtr](#) > [ActorList](#)
- typedef [mediawrap::VideoContext::TextureID](#) TextureID
- typedef [mediawrap::VideoContext::Region](#) Region

## Enumerations

- enum [ActorEventType](#) {  
[Nil](#) = -1, [ChangeScene](#), [LoadLevel](#), [PlayAudio](#),  
[NewGame](#), [LoadGame](#), [DisplayText](#), [ExitGame](#),  
[StreamAudio](#) }  
*Events that actors can trigger.*
- enum [CursorStyle](#) {  
[CursorStyle::CURSOR\\_DEFAULT](#), [CursorStyle::CURSOR\\_SELECT](#), [CursorStyle::CURSOR\\_LEFT](#),  
[CursorStyle::CURSOR\\_RIGHT](#),  
[CursorStyle::CURSOR\\_UP](#), [CursorStyle::CURSOR\\_DOWN](#) }  
*The possible styles for the mouse cursor.*

### 5.3.1 Detailed Description

Provides project-specific audio functionality for Legend of the Great Unwashed.

Provides video capabilities that are specific to Legend of the Great Unwashed.

### 5.3.2 Typedef Documentation

5.3.2.1 typedef std::vector<[BaseActorPtr](#)> [teamusa::ActorList](#)

5.3.2.2 typedef [mediawrap::AudioPlayer::AudioID](#) [teamusa::AudioID](#)

5.3.2.3 typedef std::shared\_ptr<[BaseActor](#)> [teamusa::BaseActorPtr](#)

5.3.2.4 typedef [mediawrap::VideoContext::Region](#) [teamusa::Region](#)

5.3.2.5 typedef [mediawrap::VideoContext::TextureID](#) [teamusa::TextureID](#)

### 5.3.3 Enumeration Type Documentation

5.3.3.1 enum [teamusa::ActorEventType](#)

Events that actors can trigger.

#### Enumerator

***Nil***

***ChangeScene***

***LoadLevel***

***PlayAudio***

***NewGame***

***LoadGame***

***DisplayText***

***ExitGame***

***StreamAudio***

#### 5.3.3.2 enum teamusa::CursorStyle [strong]

The possible styles for the mouse cursor.

Enumerator

***CURSOR\_DEFAULT*** Default cursor.

***CURSOR\_SELECT*** Offers the ability to select an object.

***CURSOR\_LEFT*** Points left.

***CURSOR\_RIGHT*** Points right.

***CURSOR\_UP*** Points up.

***CURSOR\_DOWN*** Points down.

## Chapter 6

# Class Documentation

### 6.1 teamusa::ActorEvent Class Reference

Event data generated by Actors, handled by [Engine](#).

```
#include <ActorEvent.h>
```

#### Public Member Functions

- [ActorEvent](#) (void)

#### Public Attributes

- [int32\\_t value](#)
- [ActorEventType type](#)

#### 6.1.1 Detailed Description

Event data generated by Actors, handled by [Engine](#).

#### 6.1.2 Constructor & Destructor Documentation

6.1.2.1 [teamusa::ActorEvent::ActorEvent \( void \)](#) `[inline]`

#### 6.1.3 Member Data Documentation

6.1.3.1 [ActorEventType](#) [teamusa::ActorEvent::type](#)

6.1.3.2 [int32\\_t](#) [teamusa::ActorEvent::value](#)

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

- [ActorEvent.h](#)

### 6.2 teamusa::ActorVideo Struct Reference

```
#include <BaseActor.h>
```

## Public Member Functions

- [ActorVideo](#) (void)

## Public Attributes

- int32\_t [layer](#)
- int32\_t [textureID](#)

### 6.2.1 Constructor & Destructor Documentation

6.2.1.1 `teamusa::ActorVideo::ActorVideo ( void ) [inline]`

### 6.2.2 Member Data Documentation

6.2.2.1 `int32_t teamusa::ActorVideo::layer`

6.2.2.2 `int32_t teamusa::ActorVideo::textureID`

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

- [BaseActor.h](#)

## 6.3 teamusa::AudioEngine Class Reference

Provides project-specific audio functionality for Legend of the Great Unwashed.

```
#include <AudioEngine.hpp>
```

## Public Member Functions

- void [loadSound](#) (const std::string &path, [AudioID](#) id, [ResourceGroup](#) group)  
*Loads the given sound file and associates it with the given id.*
- void [playSound](#) ([AudioID](#) id)  
*Plays the sound associated with the given id.*
- void [playStream](#) (const std::string &path)  
*Plays the given stream in a loop continuously.*
- void [deleteSound](#) ([AudioID](#) id)  
*Deletes the given sound from memory.*
- void [deleteSoundGroup](#) ([ResourceGroup](#) resourceGroup)  
*Deletes the entire group of sounds.*

## Private Attributes

- std::vector< [AudioID](#) > [coreResources](#)
- std::vector< [AudioID](#) > [levelResources](#)
- [AudioPlayer](#) [audioPlayer](#)

## Static Private Attributes

- static const [AudioID](#) [MAX\\_RESERVED\\_ID](#) = 1000



### 6.3.1 Detailed Description

Provides project-specific audio functionality for Legend of the Great Unwashed.

### 6.3.2 Member Function Documentation

#### 6.3.2.1 void teamusa::AudioEngine::deleteSound ( AudioID *id* )

Deletes the given sound from memory.

Parameters

|           |                                |
|-----------|--------------------------------|
| <i>id</i> | The id of the audio to delete. |
|-----------|--------------------------------|

#### 6.3.2.2 void teamusa::AudioEngine::deleteSoundGroup ( ResourceGroup *resourceGroup* )

Deletes the entire group of sounds.

Parameters

|  |  |
|--|--|
|  |  |
|--|--|

#### 6.3.2.3 void teamusa::AudioEngine::loadSound ( const std::string & *path*, AudioID *id*, ResourceGroup *group* )

Loads the given sound file and associates it with the given id.

Parameters

|             |  |
|-------------|--|
| <i>path</i> | The relative path of the sound file to load.   |
| <i>id</i>   | The id to associate with the given sound file. |

#### 6.3.2.4 void teamusa::AudioEngine::playSound ( AudioID *id* )

Plays the sound associated with the given id.

Parameters

|           |                              |
|-----------|------------------------------|
| <i>id</i> | The id of the sound to play. |
|-----------|------------------------------|

#### 6.3.2.5 void teamusa::AudioEngine::playStream ( const std::string & *path* )

Plays the given stream in a loop continuously.

Parameters

|             |                                  |
|-------------|----------------------------------|
| <i>path</i> | The path of the audio to stream. |
|-------------|----------------------------------|

### 6.3.3 Member Data Documentation

#### 6.3.3.1 AudioPlayer teamusa::AudioEngine::audioPlayer [private]

#### 6.3.3.2 std::vector<AudioID> teamusa::AudioEngine::coreResources [private]

#### 6.3.3.3 std::vector<AudioID> teamusa::AudioEngine::levelResources [private]

6.3.3.4 `const AudioID teamusa::AudioEngine::MAX_RESERVED_ID = 1000` `[static], [private]`

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

- [AudioEngine.hpp](#)
- [AudioEngine.cpp](#)

## 6.4 mediawrap::AudioPlayer Class Reference

Provides basic audio playing capabilities with WAV files.

```
#include <AudioPlayer.hpp>
```

### Public Types

- typedef unsigned int [AudioID](#)  
*Used to uniquely identify each audio sample.*

### Public Member Functions

- [AudioPlayer](#) ()  
*Constructs a new audio player.*
- [~AudioPlayer](#) ()  
*Deletes the audio player and all of its samples and streams.*
- void [load\\_stream](#) (const std::string &file\_path)  
*Loads the given audio file and prepares it for streaming.*
- void [stream\\_audio](#) (int loops=-1)  
*Plays the loaded audio stream loop+1 times.*
- void [load\\_sample](#) ([AudioID](#) id, const std::string &file\_path)  
*Loads the given audio sample into memory.*
- void [play\\_sample](#) ([AudioID](#) id)  
*Plays the given audio sample in the first available channel.*
- void [delete\\_sample](#) ([AudioID](#) id)  
*Deletes the sample created by a call to [load\\_sample\(\)](#).*
- void [clear\\_samples](#) ()  
*Deletes all samples created by a call to [load\\_sample\(\)](#).*

### Private Attributes

- std::unordered\_map< [AudioID](#), Mix\_Chunk \* > \* [audio\\_samples](#)
- Mix\_Music \* [audio\\_stream](#)

### Static Private Attributes

- static const int [audio\\_rate](#) = 44100
- static const int [audio\\_channels](#) = 1
- static const int [audio\\_buffer](#) = 4096
- static const Uint16 [audio\\_format](#) = AUDIO\_S16

### 6.4.1 Detailed Description

Provides basic audio playing capabilities with WAV files.

Acts as an abstraction layer for SDL2.

### 6.4.2 Member Typedef Documentation

#### 6.4.2.1 typedef unsigned int mediawrap::AudioPlayer::AudioID

Used to uniquely identify each audio sample.

### 6.4.3 Constructor & Destructor Documentation

#### 6.4.3.1 mediawrap::AudioPlayer::AudioPlayer ( )

Constructs a new audio player.

Enables SDL audio functionality.

#### 6.4.3.2 mediawrap::AudioPlayer::~~AudioPlayer ( )

Deletes the audio player and all of its samples and streams.

Disables SDL audio functionality.

### 6.4.4 Member Function Documentation

#### 6.4.4.1 void mediawrap::AudioPlayer::clear\_samples ( )

Deletes all samples created by a call to [load\\_sample\(\)](#).

#### 6.4.4.2 void mediawrap::AudioPlayer::delete\_sample ( AudioID id )

Deletes the sample created by a call to [load\\_sample\(\)](#).

Parameters

|           |                                 |
|-----------|---------------------------------|
| <i>id</i> | The id of the sample to delete. |
|-----------|---------------------------------|

#### 6.4.4.3 void mediawrap::AudioPlayer::load\_sample ( AudioID id, const std::string & file\_path )

Loads the given audio sample into memory.

Loading a sample into an existing id will delete the sample associated with it before the new sample is loaded.

Parameters

|                  |   |
|------------------|---|
| <i>id</i>        | The unique id to store the sample under.        |
| <i>file_path</i> | The path of the audio file to load into memory. |

#### 6.4.4.4 void mediawrap::AudioPlayer::load\_stream ( const std::string & file\_path )

Loads the given audio file and prepares it for streaming.

Only one audio stream can be loaded at a time. The previously loaded stream will be deleted if this method is called multiple times.

#### Parameters

|                  |                               |
|------------------|-------------------------------|
| <i>file_path</i> | The path of the file to load. |
|------------------|-------------------------------|

#### 6.4.4.5 void mediawrap::AudioPlayer::play\_sample ( AudioID id )

Plays the given audio sample in the first available channel.

#### Parameters

|           |                                     |
|-----------|-------------------------------------|
| <i>id</i> | The id of the audio sample to play. |
|-----------|-------------------------------------|

#### 6.4.4.6 void mediawrap::AudioPlayer::stream\_audio ( int loops = -1 )

Plays the loaded audio stream loop+1 times.

If set to -1, the audio will loop indefinitely. Only one audio stream can be played at a time.

#### Parameters

|              |   |
|--------------|---|
| <i>loops</i> | The number of times to play the audio. A value of -1 is infinite. Defaults to looping infinitely. |
|--------------|---|

### 6.4.5 Member Data Documentation

6.4.5.1 `const int mediawrap::AudioPlayer::audio_buffer = 4096` [static], [private]

6.4.5.2 `const int mediawrap::AudioPlayer::audio_channels = 1` [static], [private]

6.4.5.3 `const Uint16 mediawrap::AudioPlayer::audio_format = AUDIO_S16` [static], [private]

6.4.5.4 `const int mediawrap::AudioPlayer::audio_rate = 44100` [static], [private]

6.4.5.5 `std::unordered_map<AudioID, Mix_Chunk*>* mediawrap::AudioPlayer::audio_samples` [private]

6.4.5.6 `Mix_Music* mediawrap::AudioPlayer::audio_stream` [private]

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

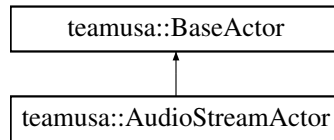
- [AudioPlayer.hpp](#)
- [AudioPlayer.cpp](#)

## 6.5 teamusa::AudioStreamActor Class Reference

If this actor is not activated, it will emit a StreamAudio event and set its status to activated when the step method is called.

```
#include <AudioStreamActor.h>
```

Inheritance diagram for teamusa::AudioStreamActor:



## Public Member Functions

- [AudioStreamActor](#) (std::string [path](#))
- virtual [~AudioStreamActor](#) (void) override
- virtual const [ActorEvent](#) [step](#) ([Player](#) &player) override  
*This method updates the player on every frame.*
- std::string [getPath](#) ()  
*This method gets the path to the requested audio file.*

## Private Attributes

- std::string [path](#)
- bool [activated](#)

## Additional Inherited Members

### 6.5.1 Detailed Description

If this actor is not activated, it will emit a StreamAudio event and set its status to activated when the step method is called.

The engine can then retrieve the path to the audio file by a call to this actor's getPath method.

### 6.5.2 Constructor & Destructor Documentation

6.5.2.1 [AudioStreamActor::AudioStreamActor](#) ( std::string *path* ) [explicit]

6.5.2.2 [AudioStreamActor::~~AudioStreamActor](#) ( void ) [override],[virtual]

### 6.5.3 Member Function Documentation

6.5.3.1 std::string [AudioStreamActor::getPath](#) ( )

This method gets the path to the requested audio file.

#### Returns

Returns a file path formatted as a string.

6.5.3.2 const [ActorEvent](#) [AudioStreamActor::step](#) ( [Player](#) & *player* ) [override],[virtual]

This method updates the player on every frame.

## Parameters

|                        |            |
|------------------------|------------|
| <a href="#">Player</a> | The player |
|------------------------|------------|

## Returns

Returns an [ActorEvent](#) that triggers an action from one or more actors.

Implements [teamusa::BaseActor](#).

## 6.5.4 Member Data Documentation

6.5.4.1 `bool teamusa::AudioStreamActor::activated` `[private]`

6.5.4.2 `std::string teamusa::AudioStreamActor::path` `[private]`

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

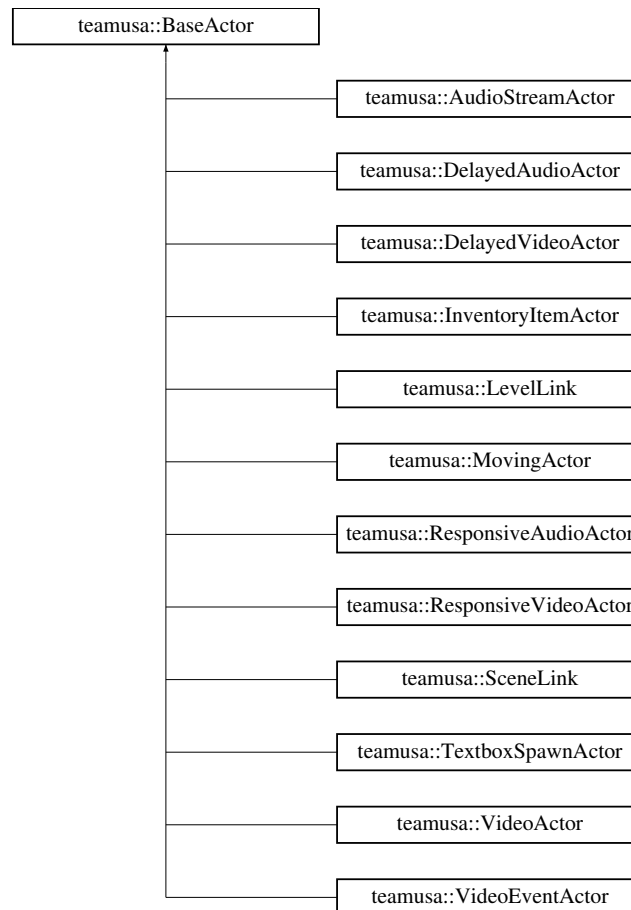
- [AudioStreamActor.h](#)
- [AudioStreamActor.cpp](#)

## 6.6 teamusa::BaseActor Class Reference

Abstract class which all actors must derive from.

```
#include <BaseActor.h>
```

Inheritance diagram for teamusa::BaseActor:



## Public Member Functions

- **BaseActor** (const **Region** &region=**Region**())
- virtual **~BaseActor** (void)=0
- virtual const **ActorEvent** **onClick** (**Player** &player)  
*Called when the actor is clicked on.*
- virtual const **ActorEvent** **onHover** (**Player** &player)  
*Called when the actor is hovered over with the mouse.*
- virtual const **ActorEvent** **step** (**Player** &player)=0  
*Called each frame, each derived actor should handle this.*
- virtual const bool **isInBounds** (const **Point** &point)  
*Calculates if point is in bounds of actor's region.*
- virtual void **setRegion** (const **Region** &region)  
*Sets the actor's region (can be used by **Level** when loading).*
- virtual const **Region** **getRegion** (void) const  
*Gets the actor's Region.*
- virtual const int32\_t **getLayer** (void) const  
*Gets the layer the actor should be rendered on.*
- virtual const int32\_t **getTextureID** (void) const  
*Gets the texture ID of the actor.*
- const bool **hasVideo** (void) const  
*Returns true if the actor has a video component.*

## Protected Attributes

- [Region](#) mRegion
- [AudioID](#) mAudioID
- [ActorVideo](#) \* mVideo

### 6.6.1 Detailed Description

Abstract class which all actors must derive from.

### 6.6.2 Constructor & Destructor Documentation

6.6.2.1 `BaseActor::BaseActor ( const Region & region = Region() )` `[explicit]`

6.6.2.2 `BaseActor::~~BaseActor ( void )` `[pure virtual]`

### 6.6.3 Member Function Documentation

6.6.3.1 `const int32_t BaseActor::getLayer ( void ) const` `[virtual]`

Gets the layer the actor should be rendered on.

#### Returns

An integer containing the layer.

6.6.3.2 `const Region BaseActor::getRegion ( void ) const` `[virtual]`

Gets the actor's Region.

#### Returns

The actor's Region struct.

6.6.3.3 `const int32_t BaseActor::getTextureID ( void ) const` `[virtual]`

Gets the texture ID of the actor.

#### Returns

The integer containing the texture ID.

6.6.3.4 `const bool BaseActor::hasVideo ( void ) const`

Returns true if the actor has a video component.

6.6.3.5 `const bool BaseActor::isInBounds ( const Point & point )` `[virtual]`

Calculates if point is in bounds of actor's region.



## Parameters

|              |                    |
|--------------|--------------------|
| <i>point</i> | The point to test. |
|--------------|--------------------|

## Returns

True if point is within actor's region.

#### 6.6.3.6 const ActorEvent BaseActor::onClick ( Player & *player* ) [virtual]

Called when the actor is clicked on.

## Parameters

|               |                          |
|---------------|--------------------------|
| <i>player</i> | The player in the scene. |
|---------------|--------------------------|

## Returns

The [ActorEvent](#) to be handled by [Engine](#) when clicked on.

Reimplemented in [teamusa::MovingActor](#), [teamusa::InventoryItemActor](#), [teamusa::TextboxSpawnActor](#), [teamusa::ResponsiveAudioActor](#), [teamusa::VideoEventActor](#), [teamusa::SceneLink](#), [teamusa::LevelLink](#), and [teamusa::ResponsiveVideoActor](#).

#### 6.6.3.7 const ActorEvent BaseActor::onHover ( Player & *player* ) [virtual]

Called when the actor is hovered over with the mouse.

## Parameters

|               |                          |
|---------------|--------------------------|
| <i>player</i> | The player in the scene. |
|---------------|--------------------------|

## Returns

The [ActorEvent](#) to be handled by [Engine](#) when hovered over.

Reimplemented in [teamusa::MovingActor](#), [teamusa::ResponsiveAudioActor](#), [teamusa::VideoEventActor](#), [teamusa::SceneLink](#), [teamusa::LevelLink](#), [teamusa::ResponsiveVideoActor](#), and [teamusa::InventoryItemActor](#).

#### 6.6.3.8 void BaseActor::setRegion ( const Region & *region* ) [virtual]

Sets the actor's region (can be used by [Level](#) when loading).

## Parameters

|               |                    |
|---------------|--------------------|
| <i>region</i> | The Region to set. |
|---------------|--------------------|

#### 6.6.3.9 virtual const ActorEvent teamusa::BaseActor::step ( Player & *player* ) [pure virtual]

Called each frame, each derived actor should handle this.

## Parameters

|               |                          |
|---------------|--------------------------|
| <i>player</i> | The player in the scene. |
|---------------|--------------------------|

#### Returns

Any [ActorEvent](#) that should be handled immediately by [Engine](#).

Implemented in [teamusa::MovingActor](#), [teamusa::ResponsiveAudioActor](#), [teamusa::VideoEventActor](#), [teamusa::SceneLink](#), [teamusa::InventoryItemActor](#), [teamusa::LevelLink](#), [teamusa::TextboxSpawnActor](#), [teamusa::DelayedVideoActor](#), [teamusa::ResponsiveVideoActor](#), [teamusa::DelayedAudioActor](#), [teamusa::AudioStreamActor](#), and [teamusa::VideoActor](#).

### 6.6.4 Member Data Documentation

6.6.4.1 **AudioID** [teamusa::BaseActor::mAudioID](#) [protected]

6.6.4.2 **Region** [teamusa::BaseActor::mRegion](#) [protected]

6.6.4.3 **ActorVideo\*** [teamusa::BaseActor::mVideo](#) [protected]

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

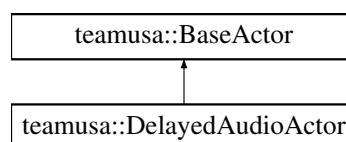
- [BaseActor.h](#)
- [BaseActor.cpp](#)

## 6.7 teamusa::DelayedAudioActor Class Reference

Will increment a counter every time the step method is called.

```
#include <DelayedAudioActor.h>
```

Inheritance diagram for [teamusa::DelayedAudioActor](#):



#### Public Member Functions

- [DelayedAudioActor](#) (int audioID, int [delaySteps](#))
- virtual [~DelayedAudioActor](#) (void) override
- virtual const [ActorEvent](#) [step](#) ([Player](#) &player) override

*Advances the actor one frame.*

#### Private Attributes

- int [audioid](#)
- int [delaySteps](#)
- int [currentStep](#)

## Additional Inherited Members

### 6.7.1 Detailed Description

Will increment a counter every time the step method is called.

After a specified number of steps have occurred, this actor will change its TextureID to a valid value and will be displayed. When the number of steps is equal to the disappearing step, the TextureID will be set to an ignored value, causing the actor to disappear.

### 6.7.2 Constructor & Destructor Documentation

6.7.2.1 `DelayedAudioActor::DelayedAudioActor ( int audioID, int delaySteps = 0 )` `[explicit]`

6.7.2.2 `DelayedAudioActor::~~DelayedAudioActor ( void )` `[override]`, `[virtual]`

### 6.7.3 Member Function Documentation

6.7.3.1 `const ActorEvent DelayedAudioActor::step ( Player & player )` `[override]`, `[virtual]`

Advances the actor one frame.

Parameters

|                               |                                     |
|-------------------------------|-------------------------------------|
| <a href="#"><i>Player</i></a> | The <a href="#"><i>Player</i></a> . |
|-------------------------------|-------------------------------------|

Returns

Returns an [ActorEvent](#) that triggers one or more actors to perform an action

Implements [teamusa::BaseActor](#).

### 6.7.4 Member Data Documentation

6.7.4.1 `int teamusa::DelayedAudioActor::audiold` `[private]`

6.7.4.2 `int teamusa::DelayedAudioActor::currentStep` `[private]`

6.7.4.3 `int teamusa::DelayedAudioActor::delaySteps` `[private]`

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

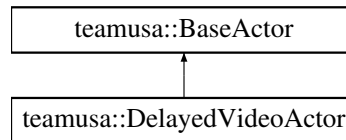
- [DelayedAudioActor.h](#)
- [DelayedAudioActor.cpp](#)

## 6.8 teamusa::DelayedVideoActor Class Reference

Will increment a counter every time the step method is called.

```
#include <DelayedVideoActor.h>
```

Inheritance diagram for teamusa::DelayedVideoActor:



## Public Member Functions

- [DelayedVideoActor](#) ([Region](#) region, int textureID, int delaysteps, int disappearStep, int layer)
- virtual [~DelayedVideoActor](#) (void) override
- virtual const [ActorEvent](#) step ([Player](#) &player) override

*Advances the actor one frame.*

## Private Attributes

- int [textureId](#)
- int [delaySteps](#)
- int [currentStep](#)
- int [disappear](#)

## Additional Inherited Members

### 6.8.1 Detailed Description

Will increment a counter every time the step method is called.

After a specified number of steps have occurred, this actor will change its TextureID to a valid value and will be displayed. When the number of steps is equal to the disappearing step, the TextureID will be set to an ignored value, causing the actor to disappear

### 6.8.2 Constructor & Destructor Documentation

**6.8.2.1** `DelayedVideoActor::DelayedVideoActor ( Region region, int textureID, int delaysteps, int disappearStep, int layer )` `[explicit]`

**6.8.2.2** `DelayedVideoActor::~~DelayedVideoActor ( void )` `[override],[virtual]`

### 6.8.3 Member Function Documentation

**6.8.3.1** `const ActorEvent DelayedVideoActor::step ( Player & player )` `[override],[virtual]`

Advances the actor one frame.

Parameters

|                        |                              |
|------------------------|------------------------------|
| <a href="#">Player</a> | The <a href="#">Player</a> . |
|------------------------|------------------------------|

Returns

Returns an [ActorEvent](#) that triggers one or more actors to perform an action.

Implements [teamusa::BaseActor](#).

## 6.8.4 Member Data Documentation

6.8.4.1 int teamusa::DelayedVideoActor::currentStep [private]

6.8.4.2 int teamusa::DelayedVideoActor::delaySteps [private]

6.8.4.3 int teamusa::DelayedVideoActor::disappear [private]

6.8.4.4 int teamusa::DelayedVideoActor::textureId [private]

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

- [DelayedVideoActor.h](#)
- [DelayedVideoActor.cpp](#)

## 6.9 teamusa::Engine Class Reference

Processes all components of the game each frame.

```
#include <Engine.h>
```

### Public Member Functions

- [Engine](#) (void)
- [~Engine](#) (void)
- void [run](#) (void)  
*Starts the game, runs until the player quits or there is an exception.*

### Private Types

- typedef std::function< void([BaseActorPtr](#) actor, const int32\_t value) > [ActorEventHandler](#)

### Private Member Functions

- const [Point](#) [getMouseCoordinates](#) (void) const  
*Retrieves the window mouse coordinates.*
- const int32\_t [getMouseClickedState](#) (void) const  
*Retrives the current mouse button state.*
- void [handleEvent](#) ([BaseActorPtr](#) actor, const [ActorEvent](#) &e)  
*Handles actor event on actor who triggered it.*
- void [render](#) (const [ActorList](#) &actors)  
*Renders all actors in the scene.*
- void [onChangeScene](#) ([BaseActorPtr](#) actor, const int32\_t value)  
*Handles scene change events triggered by [SceneLink](#) actors.*
- void [onLoadLevel](#) ([BaseActorPtr](#) actor, const int32\_t value)  
*Handles level change events triggered by [Levellink](#) actors.*
- void [onPlayAudio](#) ([BaseActorPtr](#) actor, const int32\_t value)  
*Handles audio events triggered by actors.*
- void [onNewGame](#) ([BaseActorPtr](#) actor, const int32\_t value)  
*Handles new game events triggered by main menu actors.*
- void [onLoadGame](#) ([BaseActorPtr](#) actor, const int32\_t value)

*Handles load game events triggered by main menu actors.*

- void [onDisplayText](#) ([BaseActorPtr](#) actor, const int32\_t value)

*Handles text display events triggered by actors.*

- void [onExitGame](#) ([BaseActorPtr](#) actor, const int32\_t value)

*Handles exit game events triggered by quit game button at main menu.*

- void [onStreamAudio](#) ([BaseActorPtr](#) actor, const int32\_t value)

*Handles stream audio events triggered by actors, calls into [AudioEngine](#).*

- void [freeAndLoadLevel](#) (const int32\_t id)

*Clears resource data for current level and loads the specified level.*

## Private Attributes

- std::shared\_ptr< [AudioEngine](#) > [mAudioEngine](#)
- std::shared\_ptr< [VideoEngine](#) > [mVideoEngine](#)
- [Level](#) [mLevel](#)
- int32\_t [mCurrentLevelID](#)
- [Player](#) [mPlayer](#)
- bool [mIsRunning](#)
- bool [mMainMenu](#)
- [GameSaveSerializer](#) [mSerializer](#)
- std::vector< [ActorEventHandler](#) > [mActorEventHandlers](#)

## 6.9.1 Detailed Description

Processes all components of the game each frame.

## 6.9.2 Member Typedef Documentation

- 6.9.2.1 typedef std::function< void( [BaseActorPtr](#) actor, const int32\_t value ) > [teamusa::Engine::ActorEventHandler](#) [private]

## 6.9.3 Constructor & Destructor Documentation

- 6.9.3.1 [Engine::Engine](#) ( void ) [explicit]

- 6.9.3.2 [Engine::~~Engine](#) ( void )

## 6.9.4 Member Function Documentation

- 6.9.4.1 void [Engine::freeAndLoadLevel](#) ( const int32\_t id ) [private]

Clears resource data for current level and loads the specified level.

- 6.9.4.2 const int32\_t [Engine::getMouseClickedState](#) ( void ) const [private]

Retrives the current mouse button state.

### Returns

Integer describing mouse state.

**6.9.4.3** `const Point Engine::getMouseCoordinates ( void ) const` `[private]`

Retrieves the window mouse coordinates.

**Returns**

A [Point](#) struct containing the x and y values of the mouse.

**6.9.4.4** `void Engine::handleEvent ( BaseActorPtr actor, const ActorEvent & e )` `[private]`

Handles actor event on actor who triggered it.

Looks up function pointer in table, calls the corresponding function.

**6.9.4.5** `void Engine::onChangeScene ( BaseActorPtr actor, const int32_t value )` `[private]`

Handles scene change events triggered by [SceneLink](#) actors.

**Parameters**

|              |  |
|--------------|--|
| <i>actor</i> | The actor who triggered the event.             |
| <i>value</i> | A value corresponding to the event, if needed. |

**6.9.4.6** `void Engine::onDisplayText ( BaseActorPtr actor, const int32_t value )` `[private]`

Handles text display events triggered by actors.

**Parameters**

|              |  |
|--------------|--|
| <i>actor</i> | The actor who triggered the event.             |
| <i>value</i> | A value corresponding to the event, if needed. |

**6.9.4.7** `void Engine::onExitGame ( BaseActorPtr actor, const int32_t value )` `[private]`

Handles exit game events triggered by quit game button at main menu.

**Parameters**

|              |  |
|--------------|--|
| <i>actor</i> | The actor who triggered the event.             |
| <i>value</i> | A value corresponding to the event, if needed. |

**6.9.4.8** `void Engine::onLoadGame ( BaseActorPtr actor, const int32_t value )` `[private]`

Handles load game events triggered by main menu actors.

**Parameters**

|              |  |
|--------------|--|
| <i>actor</i> | The actor who triggered the event.             |
| <i>value</i> | A value corresponding to the event, if needed. |

**6.9.4.9** `void Engine::onLoadLevel ( BaseActorPtr actor, const int32_t value )` `[private]`

Handles level change events triggered by [LevelLink](#) actors.

## Parameters

|              |  |
|--------------|--|
| <i>actor</i> | The actor who triggered the event.             |
| <i>value</i> | A value corresponding to the event, if needed. |

**6.9.4.10** `void Engine::onNewGame ( BaseActorPtr actor, const int32_t value )` [private]

Handles new game events triggered by main menu actors.

## Parameters

|              |  |
|--------------|--|
| <i>actor</i> | The actor who triggered the event.             |
| <i>value</i> | A value corresponding to the event, if needed. |

**6.9.4.11** `void Engine::onPlayAudio ( BaseActorPtr actor, const int32_t value )` [private]

Handles audio events triggered by actors.

Calls into the [AudioEngine](#).

## Parameters

|              |  |
|--------------|--|
| <i>actor</i> | The actor who triggered the event.             |
| <i>value</i> | A value corresponding to the event, if needed. |

**6.9.4.12** `void Engine::onStreamAudio ( BaseActorPtr actor, const int32_t value )` [private]

Handles stream audio events triggered by actors, calls into [AudioEngine](#).

## Parameters

|              |  |
|--------------|--|
| <i>actor</i> | The actor who triggered the event.             |
| <i>value</i> | A value corresponding to the event, if needed. |

**6.9.4.13** `void Engine::render ( const ActorList & actors )` [private]

Renders all actors in the scene.

**6.9.4.14** `void Engine::run ( void )`

Starts the game, runs until the player quits or there is an exception.

## 6.9.5 Member Data Documentation

**6.9.5.1** `std::vector<ActorEventHandler> teamusa::Engine::mActorEventHandlers` [private]

**6.9.5.2** `std::shared_ptr<AudioEngine> teamusa::Engine::mAudioEngine` [private]

**6.9.5.3** `int32_t teamusa::Engine::mCurrentLevelID` [private]

**6.9.5.4** `bool teamusa::Engine::mIsRunning` [private]

**6.9.5.5** `Level teamusa::Engine::mLevel` [private]



6.9.5.6 `bool teamusa::Engine::mMainMenu` `[private]`

6.9.5.7 `Player teamusa::Engine::mPlayer` `[private]`

6.9.5.8 `GameSaveSerializer teamusa::Engine::mSerializer` `[private]`

6.9.5.9 `std::shared_ptr<VideoEngine> teamusa::Engine::mVideoEngine` `[private]`

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

- [Engine.h](#)
- [Engine.cpp](#)

## 6.10 teamusa::GameSaveSerializer Class Reference

Provides multithreaded save, single-thread load of save files.

```
#include <GameSaveSerializer.h>
```

### Public Member Functions

- [GameSaveSerializer](#) (void)
- [~GameSaveSerializer](#) (void)
- void [setSlot](#) (const int32\_t slot)  
*Sets the slot number to save/load in.*
- bool [load](#) (int &levelID, int &sceneID, [Player::Inventory](#) &inventory)  
*Loads a save file.*
- void [save](#) (const int &levelID, const int &sceneID, const [Player::Inventory](#) &inventory)  
*Saves a file.*
- void [saveInThread](#) (const int levelID, const int sceneID, const [Player::Inventory](#) inventory)  
*Saves a file in a separate thread.*

### Private Attributes

- std::mutex [fileLock](#)
- int32\_t [slot](#)

#### 6.10.1 Detailed Description

Provides multithreaded save, single-thread load of save files.

#### 6.10.2 Constructor & Destructor Documentation

6.10.2.1 `teamusa::GameSaveSerializer::GameSaveSerializer ( void )`

6.10.2.2 `teamusa::GameSaveSerializer::~~GameSaveSerializer ( void )`

#### 6.10.3 Member Function Documentation

6.10.3.1 `bool teamusa::GameSaveSerializer::load ( int & levelID, int & sceneID, Player::Inventory & inventory )`

Loads a save file.

**Returns**

True if save file was loaded successfully, false if it doesn't exist.

6.10.3.2 `void teamusa::GameSaveSerializer::save ( const int & levelID, const int & sceneID, const Player::Inventory & inventory )`

Saves a file.

6.10.3.3 `void teamusa::GameSaveSerializer::saveInThread ( const int levelID, const int sceneID, const Player::Inventory & inventory )`

Saves a file in a separate thread.

6.10.3.4 `void teamusa::GameSaveSerializer::setSlot ( const int32_t slot )`

Sets the slot number to save/load in.

**6.10.4 Member Data Documentation**

6.10.4.1 `std::mutex teamusa::GameSaveSerializer::fileLock` [private]

6.10.4.2 `int32_t teamusa::GameSaveSerializer::slot` [private]

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

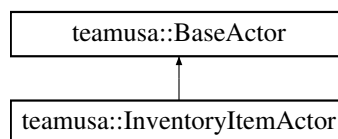
- [GameSaveSerializer.h](#)
- [GameSaveSerializer.cpp](#)

**6.11 teamusa::InventoryItemActor Class Reference**

InventoryItemActor creates a collectible item in the game environment.

```
#include <InventoryItemActor.h>
```

Inheritance diagram for teamusa::InventoryItemActor:

**Public Member Functions**

- [InventoryItemActor](#) ([Region](#) region, const int [itemID](#)==1, const int textureID==1, const int layer==1)
- virtual `~InventoryItemActor` (void) override
- virtual const [ActorEvent](#) [onHover](#) ([Player](#) &player) override  
*Generates an [ActorEvent](#) if the player hovers over the actors' region.*
- virtual const [ActorEvent](#) [onClick](#) ([Player](#) &player) override  
*Generates an [ActorEvent](#) if the player clicks in the actor's region.*
- virtual const [ActorEvent](#) [step](#) ([Player](#) &player) override  
*Advances the actor one frame and sends the appropriate [ActorEvent](#).*

## Private Attributes

- int [itemID](#)
- bool [pickedUp](#) = false

## Additional Inherited Members

### 6.11.1 Detailed Description

InventoryItemActor creates a collectible item in the game environment.

### 6.11.2 Constructor & Destructor Documentation

6.11.2.1 `InventoryItemActor::InventoryItemActor ( Region region, const int itemID = -1, const int textureID = -1, const int layer = -1 )` `[explicit]`

6.11.2.2 `InventoryItemActor::~~InventoryItemActor ( void )` `[override]`, `[virtual]`

### 6.11.3 Member Function Documentation

6.11.3.1 `const ActorEvent InventoryItemActor::onClick ( Player & player )` `[override]`, `[virtual]`

Generates an [ActorEvent](#) if the player clicks in the actor's region.

#### Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

#### Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.11.3.2 `const ActorEvent InventoryItemActor::onHover ( Player & player )` `[override]`, `[virtual]`

Generates an [ActorEvent](#) if the player hovers over the actors' region.

#### Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

#### Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.11.3.3 `const ActorEvent InventoryItemActor::step ( Player & player )` `[override]`, `[virtual]`

Advances the actor one frame and sends the appropriate [ActorEvent](#).

## Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

## Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Implements [teamusa::BaseActor](#).

## 6.11.4 Member Data Documentation

6.11.4.1 `int teamusa::InventoryItemActor::itemID` [private]

6.11.4.2 `bool teamusa::InventoryItemActor::pickedUp = false` [private]

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

- [InventoryItemActor.h](#)
- [InventoryItemActor.cpp](#)

## 6.12 teamusa::Level Class Reference

A [Level](#) is a container of Scenes and Actors corresponding to those scenes.

```
#include <Level.h>
```

### Classes

- class [Scene](#)

*A scene is a collection of images (Actors) that is displayed on the screen.*

### Public Member Functions

- [Level](#) (void)
- [Level](#) (int levelID, [AudioEngine](#) &audioEngine, [VideoEngine](#) &videoEngine)
- const [ActorList](#) & [getActors](#) (void) const  
*Returns the list of actors in the current scene.*
- const int [getBGImageID](#) (void) const  
*Returns the textureID of the background image in the current scene.*
- const int [loadLevel](#) (const std::string &path, [AudioEngine](#) &audioEngine, [VideoEngine](#) &videoEngine)  
*Parses the specified level file, loads textures, audio samples, and stores the actors in a hash table.*
- void [changeScene](#) (const int sceneID)  
*Changes the currently active scene.*
- const int [getScene](#) ()  
*Returns the index of the currently active scene.*
- void [clearAll](#) (void)  
*Removes all loaded scenes and actors from memory.*

## Private Member Functions

- [BaseActorPtr parseAudioStreamActor](#) (std::fstream &fs)
- [BaseActorPtr parseDelayedAudioActor](#) (std::fstream &fs)
- [BaseActorPtr parseDelayedVideoActor](#) (std::fstream &fs)
- [BaseActorPtr parseInventoryItemActor](#) (std::fstream &fs)
- [BaseActorPtr parseLevelLink](#) (std::fstream &fs)
- [BaseActorPtr parseMovingActor](#) (std::fstream &fs)
- [BaseActorPtr parseResponsiveAudioActor](#) (std::fstream &fs)
- [BaseActorPtr parseResponsiveVideoActor](#) (std::fstream &fs)
- [BaseActorPtr parseSceneLink](#) (std::fstream &fs)
- [BaseActorPtr parseTextboxSpawnActor](#) (std::fstream &fs)
- [BaseActorPtr parseVideoActor](#) (std::fstream &fs)
- [BaseActorPtr parseVideoEventActor](#) (std::fstream &fs)

## Private Attributes

- std::unordered\_map< int, [Scene](#) > [scenes](#)
- int [startScene](#)
- int [activeScene](#)

### 6.12.1 Detailed Description

A [Level](#) is a container of Scenes and Actors corresponding to those scenes.

### 6.12.2 Constructor & Destructor Documentation

6.12.2.1 `Level::Level ( void )`

6.12.2.2 `Level::Level ( int levelID, AudioEngine & audioEngine, VideoEngine & videoEngine )`

### 6.12.3 Member Function Documentation

6.12.3.1 `void Level::changeScene ( const int sceneID )`

Changes the currently active scene.

Subsequent calls to [getActors\(\)](#) will return the actors in that scene.

Parameters

|                |                          |
|----------------|--------------------------|
| <i>sceneID</i> | The ID of the new scene. |
|----------------|--------------------------|

6.12.3.2 `void Level::clearAll ( void )`

Removes all loaded scenes and actors from memory.

6.12.3.3 `const ActorList & Level::getActors ( void ) const`

Returns the list of actors in the current scene.

#### 6.12.3.4 `const int Level::getBGImageID ( void ) const`

Returns the textureID of the background image in the current scene.

#### 6.12.3.5 `const int Level::getScene ( )`

Returns the index of the currently active scene.

#### 6.12.3.6 `const int Level::loadLevel ( const std::string & path, AudioEngine & audioEngine, VideoEngine & videoEngine )`

Parses the specified level file, loads textures, audio samples, and stores the actors in a hash table.

Parameters

|                    |   |
|--------------------|---|
| <i>path</i>        | The file path to the .lvl file.             |
| <i>audioEngine</i> | A reference to the audio engine being used. |
| <i>videoEngine</i> | A reference to the video engine being used. |

#### 6.12.3.7 `BaseActorPtr Level::parseAudioStreamActor ( std::fstream & fs ) [private]`

#### 6.12.3.8 `BaseActorPtr Level::parseDelayedAudioActor ( std::fstream & fs ) [private]`

#### 6.12.3.9 `BaseActorPtr Level::parseDelayedVideoActor ( std::fstream & fs ) [private]`

#### 6.12.3.10 `BaseActorPtr Level::parseInventoryItemActor ( std::fstream & fs ) [private]`

#### 6.12.3.11 `BaseActorPtr Level::parseLevelLink ( std::fstream & fs ) [private]`

#### 6.12.3.12 `BaseActorPtr Level::parseMovingActor ( std::fstream & fs ) [private]`

#### 6.12.3.13 `BaseActorPtr Level::parseResponsiveAudioActor ( std::fstream & fs ) [private]`

#### 6.12.3.14 `BaseActorPtr Level::parseResponsiveVideoActor ( std::fstream & fs ) [private]`

#### 6.12.3.15 `BaseActorPtr Level::parseSceneLink ( std::fstream & fs ) [private]`

#### 6.12.3.16 `BaseActorPtr Level::parseTextboxSpawnActor ( std::fstream & fs ) [private]`

#### 6.12.3.17 `BaseActorPtr Level::parseVideoActor ( std::fstream & fs ) [private]`

#### 6.12.3.18 `BaseActorPtr Level::parseVideoEventActor ( std::fstream & fs ) [private]`

### 6.12.4 Member Data Documentation

#### 6.12.4.1 `int teamusa::Level::activeScene [private]`

#### 6.12.4.2 `std::unordered_map<int, Scene> teamusa::Level::scenes [private]`

#### 6.12.4.3 `int teamusa::Level::startScene [private]`

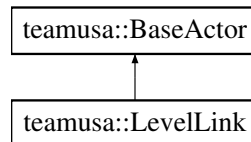
The documentation for this class was generated from the following files:

- [Level.h](#)
- [Level.cpp](#)

## 6.13 teamusa::LevelLink Class Reference

```
#include <LevelLink.h>
```

Inheritance diagram for teamusa::LevelLink:



### Public Member Functions

- [LevelLink](#) ([Region](#) region, const int Level\_ID, const int [sceneID](#), const std::string itemRequired\_Text, const int item\_ID=-1)
- virtual [~LevelLink](#) (void) override
- virtual const [ActorEvent](#) [onClick](#) ([Player](#) &player) override  
*Returns an actor event when the actor's region is clicked on.*
- virtual const [ActorEvent](#) [onHover](#) ([Player](#) &player) override  
*Returns an actor event when the actor's region is hovered over.*
- virtual const [ActorEvent](#) [step](#) ([Player](#) &player) override  
*Advances the actor one frame.*
- const int [getSceneID](#) (void) const  
*Gets the appropriate SceneID.*
- virtual const std::string [getText](#) ()  
*Generates text when the player attempts to traverse a scene without a required item.*

### Private Attributes

- int [sceneID](#)
- int [levelID](#)
- std::string [itemRequiredText](#)
- int [requiredItemID](#) = -1

### Additional Inherited Members

#### 6.13.1 Constructor & Destructor Documentation

6.13.1.1 `LevelLink::LevelLink ( Region region, const int Level_ID, const int sceneID, const std::string itemRequired_Text, const int item_ID = -1 ) [explicit]`

6.13.1.2 `LevelLink::~~LevelLink ( void ) [override],[virtual]`

#### 6.13.2 Member Function Documentation

6.13.2.1 `const int LevelLink::getSceneID ( void ) const`

Gets the appropriate SceneID.

##### Returns

Returns an integer representing the scene ID.

6.13.2.2 `const std::string LevelLink::getText ( )` [virtual]

Generates text when the player attempts to traverse a scene without a required item.

6.13.2.3 `const ActorEvent LevelLink::onClick ( Player & player )` [override],[virtual]

Returns an actor event when the actor's region is clicked on.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.13.2.4 `const ActorEvent LevelLink::onHover ( Player & player )` [override],[virtual]

Returns an actor event when the actor's region is hovered over.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.13.2.5 `const ActorEvent LevelLink::step ( Player & player )` [override],[virtual]

Advances the actor one frame.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Implements [teamusa::BaseActor](#).

### 6.13.3 Member Data Documentation

6.13.3.1 `std::string teamusa::LevelLink::itemRequiredText` [private]

6.13.3.2 `int teamusa::LevelLink::levelID` [private]

6.13.3.3 `int teamusa::LevelLink::requiredItemID = -1` [private]

6.13.3.4 `int teamusa::LevelLink::sceneID` [private]

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



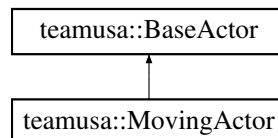
- [LevelLink.h](#)
- [LevelLink.cpp](#)

## 6.14 teamusa::MovingActor Class Reference

Will transition from one region to the next by calculating the distance to move each frame for a set number of frames.

```
#include <MovingActor.h>
```

Inheritance diagram for teamusa::MovingActor:



### Public Member Functions

- [MovingActor](#) ([Region](#) startRegion, [Region](#) endregion, int textureId, int layer, int transitionsteps, bool move↔ OnSpawn)
- virtual [~MovingActor](#) (void) override
- virtual const [ActorEvent](#) [onClick](#) ([Player](#) &player) override  
*Generates an [ActorEvent](#) when the actor's region is clicked.*
- virtual const [ActorEvent](#) [onHover](#) ([Player](#) &player) override  
*Generates an [ActorEvent](#) when the actor's region is hovered over.*
- virtual const [ActorEvent](#) [step](#) ([Player](#) &player)  
*Advances the actor one frame.*

### Private Attributes

- [Region](#) endRegion
- int [transitionSteps](#) = 1
- int [currentStep](#) = 0
- int [xSpeed](#) = 0
- int [ySpeed](#) = 0
- int [hGrowth](#) = 0
- int [wGrowth](#) = 0
- bool [isActive](#) = false

### Additional Inherited Members

#### 6.14.1 Detailed Description

Will transition from one region to the next by calculating the distance to move each frame for a set number of frames.

This allows for movement across the X and Y axis as well as scaling of the size of textures.

## 6.14.2 Constructor & Destructor Documentation

6.14.2.1 `MovingActor::MovingActor ( Region startRegion, Region endregion, int textureId, int layer, int transitionsteps, bool moveOnSpawn )` `[explicit]`

6.14.2.2 `MovingActor::~MovingActor ( void )` `[override],[virtual]`

## 6.14.3 Member Function Documentation

6.14.3.1 `const ActorEvent MovingActor::onClick ( Player & player )` `[override],[virtual]`

Generates an [ActorEvent](#) when the actor's region is clicked.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.14.3.2 `const ActorEvent MovingActor::onHover ( Player & player )` `[override],[virtual]`

Generates an [ActorEvent](#) when the actor's region is hovered over.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.14.3.3 `const ActorEvent MovingActor::step ( Player & player )` `[virtual]`

Advances the actor one frame.

Parameters

|                        |            |
|------------------------|------------|
| <a href="#">Player</a> | The player |
|------------------------|------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Implements [teamusa::BaseActor](#).

## 6.14.4 Member Data Documentation

6.14.4.1 `int teamusa::MovingActor::currentStep = 0` `[private]`

6.14.4.2 `Region teamusa::MovingActor::endRegion` `[private]`

- 6.14.4.3 `int teamusa::MovingActor::hGrowth = 0` `[private]`
- 6.14.4.4 `bool teamusa::MovingActor::isActive = false` `[private]`
- 6.14.4.5 `int teamusa::MovingActor::transitionSteps = 1` `[private]`
- 6.14.4.6 `int teamusa::MovingActor::wGrowth = 0` `[private]`
- 6.14.4.7 `int teamusa::MovingActor::xSpeed = 0` `[private]`
- 6.14.4.8 `int teamusa::MovingActor::ySpeed = 0` `[private]`

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

- [MovingActor.h](#)
- [MovingActor.cpp](#)

## 6.15 teamusa::Player Class Reference

Handles all data relevant to the player engaging the game.

```
#include <Player.h>
```

### Public Types

- `typedef std::vector< int32_t >` [Inventory](#)  
*[Player](#) inventory - an array of integer IDs.*

### Public Member Functions

- [Player](#) (void)
- [~Player](#) (void)
- `const bool` [hasItem](#) (const `int32_t` itemType) `const`  
*Tests if the player has an item in their inventory.*
- `void` [addItem](#) (const `int32_t` itemType)  
*Inserts an item into the player's inventory.*
- `void` [setCursor](#) (const [CursorStyle](#) style)  
*Sets the visual style of the player's mouse cursor.*
- `const int` [getCursorTextureID](#) (void) `const`  
*Returns the current cursor texture ID associated with the cursor style.*
- `void` [setPosition](#) (const `int32_t` x, const `int32_t` y)  
*Sets the position of the player's cursor.*
- `void` [setPosition](#) (const [Point](#) &position)  
*Sets the position of the player's cursor.*
- `const Point` [getPosition](#) (void) `const`  
*Gets the player's cursor position.*
- `const Inventory &` [getInventory](#) () `const`  
*Returns the player's inventory.*
- `void` [setInventory](#) (const [Inventory](#) &inventory)  
*Clears the player's current inventory and assigns the new one.*

## Static Public Attributes

- static const int [FLASHLIGHT\\_ID](#) = 1666
- static const int [CURSOR\\_DEFAULT\\_ID](#) = 1667
- static const int [CURSOR\\_SELECT\\_ID](#) = 1668
- static const int [CURSOR\\_UP\\_ID](#) = 1669
- static const int [CURSOR\\_DOWN\\_ID](#) = 1670
- static const int [CURSOR\\_LEFT\\_ID](#) = 1671
- static const int [CURSOR\\_RIGHT\\_ID](#) = 1672
- static const int [MOUSE\\_CLICK\\_ID](#) = 1700

## Private Attributes

- [Region](#) mRegion
- int32\_t mLayer
- int32\_t mTextureID
- [Point](#) mPosition
- [Inventory](#) mInventory
- [CursorStyle](#) mCursorStyle

## 6.15.1 Detailed Description

Handles all data relevant to the player engaging the game.

## 6.15.2 Member Typedef Documentation

### 6.15.2.1 typedef std::vector<int32\_t> teamusa::Player::Inventory

[Player](#) inventory - an array of integer IDs.

## 6.15.3 Constructor & Destructor Documentation

### 6.15.3.1 Player::Player ( void ) [explicit]

### 6.15.3.2 Player::~~Player ( void )

## 6.15.4 Member Function Documentation

### 6.15.4.1 void Player::addItem ( const int32\_t itemType )

Inserts an item into the player's inventory.

Parameters

|                 |                                |
|-----------------|--------------------------------|
| <i>itemType</i> | The item identifier to insert. |
|-----------------|--------------------------------|

### 6.15.4.2 const int Player::getCursorTextureID ( void ) const

Returns the current cursor texture ID associated with the cursor style.

### 6.15.4.3 const Player::Inventory & Player::getInventory ( ) const

Returns the player's inventory.

#### 6.15.4.4 const Point Player::getPosition ( void ) const

Gets the player's cursor position.

##### Returns

A [Point](#) struct containing the cursor position.

#### 6.15.4.5 const bool Player::hasItem ( const int32\_t itemType ) const

Tests if the player has an item in their inventory.

##### Parameters

|                 |                           |
|-----------------|---------------------------|
| <i>itemType</i> | The item type identifier. |
|-----------------|---------------------------|

##### Returns

True if the player has the item.

#### 6.15.4.6 void Player::setCursor ( const CursorStyle style )

Sets the visual style of the player's mouse cursor.

##### Parameters

|              |                                |
|--------------|--------------------------------|
| <i>style</i> | The style type for the cursor. |
|--------------|--------------------------------|

#### 6.15.4.7 void Player::setInventory ( const Inventory & inventory )

Clears the player's current inventory and assigns the new one.

##### Parameters

|                  |  |
|------------------|--|
| <i>inventory</i> | The inventory to assign to the player. |
|------------------|--|

#### 6.15.4.8 void Player::setPosition ( const int32\_t x, const int32\_t y )

Sets the position of the player's cursor.

##### Parameters

|          |                                 |
|----------|---------------------------------|
| <i>x</i> | The x-coordinate of the cursor. |
| <i>y</i> | The y-coordinate of the cursor. |

#### 6.15.4.9 void Player::setPosition ( const Point & position )

Sets the position of the player's cursor.

##### Parameters

|                 |  |
|-----------------|--|
| <i>position</i> | A <a href="#">Point</a> struct containing the cursor position. |
|-----------------|--|

### 6.15.5 Member Data Documentation

6.15.5.1 `const int Player::CURSOR_DEFAULT_ID = 1667` [static]

6.15.5.2 `const int Player::CURSOR_DOWN_ID = 1670` [static]

6.15.5.3 `const int Player::CURSOR_LEFT_ID = 1671` [static]

6.15.5.4 `const int Player::CURSOR_RIGHT_ID = 1672` [static]

6.15.5.5 `const int Player::CURSOR_SELECT_ID = 1668` [static]

6.15.5.6 `const int Player::CURSOR_UP_ID = 1669` [static]

6.15.5.7 `const int Player::FLASHLIGHT_ID = 1666` [static]

6.15.5.8 `CursorStyle teamusa::Player::mCursorStyle` [private]

6.15.5.9 `Inventory teamusa::Player::mInventory` [private]

6.15.5.10 `int32_t teamusa::Player::mLayer` [private]

6.15.5.11 `const int Player::MOUSE_CLICK_ID = 1700` [static]

6.15.5.12 `Point teamusa::Player::mPosition` [private]

6.15.5.13 `Region teamusa::Player::mRegion` [private]

6.15.5.14 `int32_t teamusa::Player::mTextureID` [private]

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

- [Player.h](#)
- [Player.cpp](#)

## 6.16 teamusa::Point Class Reference

An (x,y) coordinate within the rendering window.

```
#include <Point.h>
```

### Public Member Functions

- [Point](#) (void)
- [Point](#) (int32\_t x, const int32\_t y)

### Public Attributes

- int32\_t x
- int32\_t y

### 6.16.1 Detailed Description

An (x,y) coordinate within the rendering window.

### 6.16.2 Constructor & Destructor Documentation

6.16.2.1 teamusa::Point::Point ( void ) `[inline]`

6.16.2.2 teamusa::Point::Point ( int32\_t x, const int32\_t y ) `[inline]`

### 6.16.3 Member Data Documentation

6.16.3.1 int32\_t teamusa::Point::x

6.16.3.2 int32\_t teamusa::Point::y

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

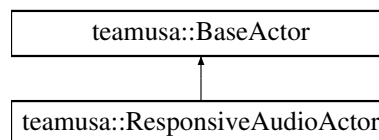
- [Point.h](#)

## 6.17 teamusa::ResponsiveAudioActor Class Reference

\ Brief: Will increment the value of stepCount until it is equal to durationSteps for each call to the step method.

```
#include <ResponsiveAudioActor.h>
```

Inheritance diagram for teamusa::ResponsiveAudioActor:



### Public Member Functions

- [ResponsiveAudioActor](#) ([Region](#) region, int [hoverAudioId](#), int [clickAudioId](#))
- virtual `~ResponsiveAudioActor` (void) override
- virtual const [ActorEvent](#) [onClick](#) ([Player](#) &player) override  
*Generates an [ActorEvent](#) when the actor's region is clicked.*
- virtual const [ActorEvent](#) [onHover](#) ([Player](#) &player) override  
*Generates an [ActorEvent](#) when the actor's region is hovered over.*
- virtual const [ActorEvent](#) [step](#) ([Player](#) &player) override  
*Advances the actor one frame.*

### Private Attributes

- int [hoverAudioId](#)
- int [clickAudioId](#)

## Additional Inherited Members

### 6.17.1 Detailed Description

\ Brief: Will increment the value of stepCount until it is equal to durationSteps for each call to the step method.

A call to onClick or onHover will set the value of stepCount to zero and emit an AudioID and value if stepCount is equal to durationSteps. The hoverAudioID or clickAudioID can be set to an invalid AudioID value to prevent sound from being played.

### 6.17.2 Constructor & Destructor Documentation

6.17.2.1 **ResponsiveAudioActor::ResponsiveAudioActor** ( **Region** *region*, **int** *hoverAudioId* = -1, **int** *clickAudioId* = -1 )  
[explicit]

6.17.2.2 **ResponsiveAudioActor::~~ResponsiveAudioActor** ( **void** ) [override],[virtual]

### 6.17.3 Member Function Documentation

6.17.3.1 **const ActorEvent** **ResponsiveAudioActor::onClick** ( **Player &** *player* ) [override],[virtual]

Generates an [ActorEvent](#) when the actor's region is clicked.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.17.3.2 **const ActorEvent** **ResponsiveAudioActor::onHover** ( **Player &** *player* ) [override],[virtual]

Generates an [ActorEvent](#) when the actor's region is hovered over.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.17.3.3 **const ActorEvent** **ResponsiveAudioActor::step** ( **Player &** *player* ) [override],[virtual]

Advances the actor one frame.

Parameters

|                        |            |
|------------------------|------------|
| <a href="#">Player</a> | The player |
|------------------------|------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Implements [teamusa::BaseActor](#).



### 6.17.4 Member Data Documentation

6.17.4.1 int teamusa::ResponsiveAudioActor::clickAudioId [private]

6.17.4.2 int teamusa::ResponsiveAudioActor::hoverAudioId [private]

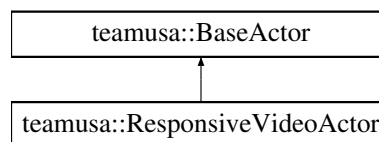
The documentation for this class was generated from the following files:

- [ResponsiveAudioActor.h](#)
- [ResponsiveAudioActor.cpp](#)

## 6.18 teamusa::ResponsiveVideoActor Class Reference

```
#include <ResponsiveVideoActor.h>
```

Inheritance diagram for teamusa::ResponsiveVideoActor:



### Public Member Functions

- [ResponsiveVideoActor](#) ([Region](#) region, int hoverTextureId, int clickTextureID, int defaultTextureID, int layer)
- virtual [~ResponsiveVideoActor](#) (void) override
- virtual const [ActorEvent](#) [onClick](#) ([Player](#) &player) override  
*Generates an [ActorEvent](#) when the actor's region is clicked.*
- virtual const [ActorEvent](#) [onHover](#) ([Player](#) &player) override  
*Generates an [ActorEvent](#) when the actor's region is hovered over.*
- virtual const [ActorEvent](#) [step](#) ([Player](#) &player) override  
*Advances the actor one frame.*
- void [setTextureId](#) (int TextureId)  
*Sets the requested texture ID.*

### Private Attributes

- int [hoverTexture](#)
- int [clickTexture](#)
- int [defaultTextureId](#)

### Additional Inherited Members

#### 6.18.1 Constructor & Destructor Documentation

6.18.1.1 [ResponsiveVideoActor::ResponsiveVideoActor](#) ( [Region](#) region, int hoverTextureId, int clickTextureID, int defaultTextureID, int layer ) [explicit]

6.18.1.2 [ResponsiveVideoActor::~~ResponsiveVideoActor](#) ( void ) [override],[virtual]

## 6.18.2 Member Function Documentation

6.18.2.1 `const ActorEvent ResponsiveVideoActor::onClick ( Player & player )` `[override],[virtual]`

Generates an [ActorEvent](#) when the actor's region is clicked.

## Parameters

|                               |             |
|-------------------------------|-------------|
| <a href="#"><i>Player</i></a> | The player. |
|-------------------------------|-------------|

## Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

#### 6.18.2.2 `const ActorEvent ResponsiveVideoActor::onHover ( Player & player ) [override],[virtual]`

Generates an [ActorEvent](#) when the actor's region is hovered over.

## Parameters

|                               |             |
|-------------------------------|-------------|
| <a href="#"><i>Player</i></a> | The player. |
|-------------------------------|-------------|

## Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action

Reimplemented from [teamusa::BaseActor](#).

#### 6.18.2.3 `void ResponsiveVideoActor::setTextureId ( int TextureId )`

Sets the requested texture ID.

## Parameters

|                  |  |
|------------------|--|
| <i>TextureID</i> | The integer ID of the requested texture. |
|------------------|--|

#### 6.18.2.4 `const ActorEvent ResponsiveVideoActor::step ( Player & player ) [override],[virtual]`

Advances the actor one frame.

## Parameters

|                               |            |
|-------------------------------|------------|
| <a href="#"><i>Player</i></a> | The player |
|-------------------------------|------------|

## Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Implements [teamusa::BaseActor](#).

### 6.18.3 Member Data Documentation

#### 6.18.3.1 `int teamusa::ResponsiveVideoActor::clickTexture [private]`

#### 6.18.3.2 `int teamusa::ResponsiveVideoActor::defaultTextureId [private]`

#### 6.18.3.3 `int teamusa::ResponsiveVideoActor::hoverTexture [private]`

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

- [ResponsiveVideoActor.h](#)
- [ResponsiveVideoActor.cpp](#)

## 6.19 teamusa::Level::Scene Class Reference

A scene is a collection of images (Actors) that is displayed on the screen.

### Public Attributes

- [ActorList](#) actors
- int [bgImageID](#)

### 6.19.1 Detailed Description

A scene is a collection of images (Actors) that is displayed on the screen.

### 6.19.2 Member Data Documentation

6.19.2.1 [ActorList](#) teamusa::Level::Scene::actors

6.19.2.2 int teamusa::Level::Scene::bgImageID

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

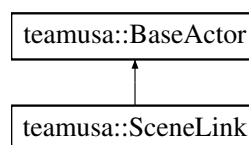
- [Level.h](#)

## 6.20 teamusa::SceneLink Class Reference

Allows the player to transition between scenes.

```
#include <SceneLink.h>
```

Inheritance diagram for teamusa::SceneLink:



### Public Member Functions

- [SceneLink](#) ([Region](#) region, const int scene\_ID, const std::string &itemRequired\_Text, const int item\_ID=-1)
- virtual [~SceneLink](#) (void) override
- virtual const [ActorEvent](#) [onClick](#) ([Player](#) &player) override  
*Generates an [ActorEvent](#) when the actor's region is clicked.*
- virtual const [ActorEvent](#) [onHover](#) ([Player](#) &player) override  
*Generates an [ActorEvent](#) when the actor's region is hovered over.*
- virtual const [ActorEvent](#) [step](#) ([Player](#) &player) override  
*Advances the actor one frame.*
- virtual const std::string [getText](#) ()  
*Displays the appropriate text when a player attempts to traverse a scene without the required item.*

## Private Attributes

- int [sceneID](#)
- std::string [itemRequiredText](#)
- int [requiredItemID](#)
- [CursorStyle](#) [cursorStyle](#)

## Additional Inherited Members

### 6.20.1 Detailed Description

Allows the player to transition between scenes.

### 6.20.2 Constructor & Destructor Documentation

6.20.2.1 `SceneLink::SceneLink ( Region region, const int scene_ID, const std::string & itemRequired_Text, const int item_ID = -1 )` `[explicit]`

6.20.2.2 `SceneLink::~SceneLink ( void )` `[override]`, `[virtual]`

### 6.20.3 Member Function Documentation

6.20.3.1 `const std::string SceneLink::getText ( )` `[virtual]`

Displays the appropriate text when a player attempts to traverse a scene without the required item.

6.20.3.2 `const ActorEvent SceneLink::onClick ( Player & player )` `[override]`, `[virtual]`

Generates an [ActorEvent](#) when the actor's region is clicked.

#### Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

#### Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.20.3.3 `const ActorEvent SceneLink::onHover ( Player & player )` `[override]`, `[virtual]`

Generates an [ActorEvent](#) when the actor's region is hovered over.

#### Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

#### Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.20.3.4 `const ActorEvent SceneLink::step ( Player & player )` `[override],[virtual]`

Advances the actor one frame.

## Parameters

|                        |            |
|------------------------|------------|
| <a href="#">Player</a> | The player |
|------------------------|------------|

## Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Implements [teamusa::BaseActor](#).

## 6.20.4 Member Data Documentation

6.20.4.1 [CursorStyle](#) [teamusa::SceneLink::cursorStyle](#) [private]

6.20.4.2 [std::string](#) [teamusa::SceneLink::itemRequiredText](#) [private]

6.20.4.3 [int](#) [teamusa::SceneLink::requiredItemID](#) [private]

6.20.4.4 [int](#) [teamusa::SceneLink::sceneID](#) [private]

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

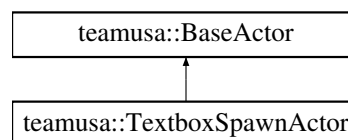
- [SceneLink.h](#)
- [SceneLink.cpp](#)

## 6.21 teamusa::TextboxSpawnActor Class Reference

Spawns a textbox that gives the player written information.

```
#include <TextboxSpawnActor.h>
```

Inheritance diagram for [teamusa::TextboxSpawnActor](#):



## Public Member Functions

- [TextboxSpawnActor](#) ([Region](#) region, [std::string](#) text)
- virtual [~TextboxSpawnActor](#) (void)
- virtual const [ActorEvent](#) [onClick](#) ([Player](#) &player)  
*Generates an [ActorEvent](#) when the actor's region is clicked.*
- virtual const [ActorEvent](#) [step](#) ([Player](#) &player)  
*Generates an [ActorEvent](#) when the actor's region is hovered over.*
- [std::string](#) [getText](#) (void)  
*Retrieves the text for the textbox from the level file.*

## Private Attributes

- [std::string](#) text
- bool activated

## Additional Inherited Members

### 6.21.1 Detailed Description

Spawns a textbox that gives the player written information.

Will emit a DisplayText event when the onClick method is called.

The actor can then have its text accessed by the engine for display through a call to the getText method.

### 6.21.2 Constructor & Destructor Documentation

6.21.2.1 `TextboxSpawnActor::TextboxSpawnActor ( Region region, std::string text )` `[explicit]`

6.21.2.2 `TextboxSpawnActor::~~TextboxSpawnActor ( void )` `[virtual]`

### 6.21.3 Member Function Documentation

6.21.3.1 `std::string TextboxSpawnActor::getText ( void )`

Retrieves the text for the textbox from the level file.

6.21.3.2 `const ActorEvent TextboxSpawnActor::onClick ( Player & player )` `[virtual]`

Generates an [ActorEvent](#) when the actor's region is clicked.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.21.3.3 `const ActorEvent TextboxSpawnActor::step ( Player & player )` `[virtual]`

Generates an [ActorEvent](#) when the actor's region is hovered over.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Implements [teamusa::BaseActor](#).

### 6.21.4 Member Data Documentation

6.21.4.1 `bool teamusa::TextboxSpawnActor::activated` `[private]`

6.21.4.2 `std::string teamusa::TextboxSpawnActor::text` `[private]`

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



- [TextboxSpawnActor.h](#)
- [TextboxSpawnActor.cpp](#)

## 6.22 teamusa::Timer Class Reference

A timer that counts up from zero in milliseconds.

```
#include <Timer.h>
```

### Public Member Functions

- [Timer](#) (void)
- [~Timer](#) (void)
- const uint32\_t [start](#) (void)  
*Starts the timer.*
- void [stop](#) (void)  
*Stops the timer.*
- void [pause](#) (void)  
*Pauses the timer.*
- void [unpause](#) (void)  
*Unpauses the timer.*
- const uint32\_t [getTicks](#) (void) const  
*Gets the time in milliseconds since the timer was started.*

### Private Attributes

- uint32\_t [mStartTicks](#)
- uint32\_t [mPauseTicks](#)
- bool [mPaused](#)
- bool [mStarted](#)

### 6.22.1 Detailed Description

A timer that counts up from zero in milliseconds.

### 6.22.2 Constructor & Destructor Documentation

6.22.2.1 [Timer::Timer \( void \)](#) `[explicit]`

6.22.2.2 [Timer::~~Timer \( void \)](#)

### 6.22.3 Member Function Documentation

6.22.3.1 [const uint32\\_t Timer::getTicks \( void \) const](#)

Gets the time in milliseconds since the timer was started.

#### Returns

The elapsed time.

#### 6.22.3.2 void Timer::pause ( void )

Pauses the timer.

#### 6.22.3.3 const uint32\_t Timer::start ( void )

Starts the timer.

#### 6.22.3.4 void Timer::stop ( void )

Stops the timer.

#### 6.22.3.5 void Timer::unpause ( void )

Unpauses the timer.

### 6.22.4 Member Data Documentation

#### 6.22.4.1 bool teamusa::Timer::mPaused [private]

#### 6.22.4.2 uint32\_t teamusa::Timer::mPauseTicks [private]

#### 6.22.4.3 bool teamusa::Timer::mStarted [private]

#### 6.22.4.4 uint32\_t teamusa::Timer::mStartTicks [private]

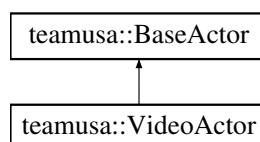
The documentation for this class was generated from the following files:

- [Timer.h](#)
- [Timer.cpp](#)

## 6.23 teamusa::VideoActor Class Reference

```
#include <VideoActor.h>
```

Inheritance diagram for teamusa::VideoActor:



### Public Member Functions

- [VideoActor](#) ([Region](#) region, int textureId, int layer)
- virtual [~VideoActor](#) (void) override
- virtual const [ActorEvent](#) step ([Player](#) &player) override

*Advances the actor one frame.*

## Additional Inherited Members

### 6.23.1 Constructor & Destructor Documentation

6.23.1.1 `VideoActor::VideoActor ( Region region, int textureId = -1, int layer = 1 )` `[explicit]`

6.23.1.2 `VideoActor::~~VideoActor ( void )` `[override]`, `[virtual]`

### 6.23.2 Member Function Documentation

6.23.2.1 `const ActorEvent VideoActor::step ( Player & player )` `[override]`, `[virtual]`

Advances the actor one frame.

Parameters

|                        |            |
|------------------------|------------|
| <a href="#">Player</a> | The player |
|------------------------|------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Implements [teamusa::BaseActor](#).

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

- [VideoActor.h](#)
- [VideoActor.cpp](#)

## 6.24 mediawrap::VideoContext Class Reference

Provides basic 2D rendering capabilities.

```
#include <VideoContext.hpp>
```

### Public Types

- enum [Flip](#) { [FLIP\\_NONE](#) = SDL\_FLIP\_NONE, [FLIP\\_HORIZONTAL](#) = SDL\_FLIP\_HORIZONTAL, [FLIP\\_VERTICAL](#) = SDL\_FLIP\_VERTICAL }
- Used to designate how an image should be flipped across an axis.*
- enum [BlendMode](#) { [BLENDMODE\\_NONE](#) = SDL\_BLENDMODE\_NONE, [BLENDMODE\\_BLEND](#) = SDL\_BLENDMODE\_BLEND, [BLENDMODE\\_ADD](#) = SDL\_BLENDMODE\_ADD, [BLENDMODE\\_MOD](#) = SDL\_BLENDMODE\_MOD }
- Used to specify how a texture should behave when objects are rendered onto it.*
- enum [DebugColor](#) { [RED](#) = 0, [GREEN](#), [BLUE](#) }
- typedef `SDL_Rect` [Region](#)
- Used to specify x, y, width, height of an texture source or destination region.*
- typedef `unsigned int` [TextureID](#)
- Used to identify each texture uniquely.*
- typedef `std::unordered_map< TextureID, SDL_Texture * >::iterator` [texture\\_iter](#)
- Used to access elements in the texture map.*

## Public Member Functions

- [VideoContext](#) (const std::string &title, unsigned int width, unsigned int height)  
*Constructs a new rendering context that includes a window and the renderer associated with it.*
- [~VideoContext](#) (void)  
*Deletes the renderer and window associated with this context.*
- void [display](#) (void)  
*Displays the rendered textures on screen.*
- [Region load\\_texture](#) (TextureID id, const std::string &image\_path, BlendMode blend=BLENDMODE\_BLEND)  
*Loads a texture from the filename into the specified texture id.*
- [Region create\\_texture](#) (TextureID id, int width, int height, BlendMode blend=BLENDMODE\_BLEND)  
*Creates a blank texture, which should be filled completely or cleared before rendering to prevent old fragments from appearing.*
- void [delete\\_texture](#) (TextureID id)  
*The deletes the given texture from this context.*
- void [render](#) (TextureID id, Region \*dest, Region \*src)  
*Draws the given texture onto the canvas.*
- void [renderDebugBox](#) (const Region &region, const DebugColor color, const TextureID layer)
- void [render\\_onto](#) (TextureID dest\_id, TextureID src\_id, const Region \*dest\_region, Region \*src\_region)  
*Draws the given source texture onto the destination texture.*
- void [render\\_rotate](#) (TextureID dest\_id, TextureID src\_id, Region \*dest\_region, Region \*src\_region, double angle=0.0, Flip flip=FLIP\_NONE)  
*Draws the given source texture onto the destination texture after applying a rotate and flip operation.*
- void [render\\_clear](#) ()  
*Clears the canvas with the default clear color.*
- void [render\\_clear](#) (TextureID id)  
*Clears the given texture with the default clear color.*
- void [fill\\_texture](#) (TextureID id, int r, int g, int b, int a)  
*Fills the given texture with the given rgba value.*
- void [load\\_font](#) (const std::string &font\_path, int font\_size)  
*Loads the given font from the path specified.*
- void [render\\_text](#) (TextureID dest\_id, Region \*dest\_region, const std::string &text, Uint8 r, Uint8 g, Uint8 b, Uint8 a)  
*Renders the given text onto the the destination texture.*
- void [swapFullscreen](#) (void)  
*Checks the state of the window and swaps to fullscreen or windowed mode.*

## Private Attributes

- std::unordered\_map< [TextureID](#), SDL\_Texture \* > \* [textures](#)
- [VideoDisplay](#) \* [video\\_display](#)
- SDL\_Renderer \* [renderer](#)
- TTF\_Font \* [font](#)

## 6.24.1 Detailed Description

Provides basic 2D rendering capabilities.

Acts as an abstraction layer to the SDL2 video library.

## 6.24.2 Member Typedef Documentation

### 6.24.2.1 typedef SDL\_Rect mediawrap::VideoContext::Region

Used to specify x, y, width, height of an texture source or destination region.

### 6.24.2.2 typedef std::unordered\_map<TextureID, SDL\_Texture\*>::iterator mediawrap::VideoContext::texture\_iter

Used to access elements in the texture map.

### 6.24.2.3 typedef unsigned int mediawrap::VideoContext::TextureID

Used to identify each texture uniquely.

Each texture loaded is to be assigned a key of this type.

## 6.24.3 Member Enumeration Documentation

### 6.24.3.1 enum mediawrap::VideoContext::BlendMode

Used to specify how a texture should behave when objects are rendered onto it.

Enumerator

***BLENDMODE\_NONE***

***BLENDMODE\_BLEND***

***BLENDMODE\_ADD***

***BLENDMODE\_MOD***

### 6.24.3.2 enum mediawrap::VideoContext::DebugColor

Enumerator

***RED***

***GREEN***

***BLUE***

### 6.24.3.3 enum mediawrap::VideoContext::Flip

Used to designate how an image should be flipped across an axis.

These two values can be ORed together to achive both effects.

Enumerator

***FLIP\_NONE***

***FLIP\_HORIZONTAL***

***FLIP\_VERTICAL***

## 6.24.4 Constructor & Destructor Documentation

### 6.24.4.1 VideoContext::VideoContext ( const std::string & *title*, unsigned int *width*, unsigned int *height* )

Constructs a new rendering context that includes a window and the renderer associated with it.

Provides utilities for loading textures and storing them in an internal mapping.

## Parameters

|               |  |
|---------------|--|
| <i>title</i>  | The title to display at the top of the window. |
| <i>width</i>  | The width of the window created.               |
| <i>height</i> | The height of the window created.              |

## 6.24.4.2 VideoContext::~~VideoContext ( void )

Deletes the renderer and window associated with this context.

Also deletes all textures currently loaded by this context.

## 6.24.5 Member Function Documentation

6.24.5.1 VideoContext::Region VideoContext::create\_texture ( TextureID *id*, int *width*, int *height*, BlendMode *blend* = BLENDMODE\_BLEND )

Creates a blank texture, which should be filled completely or cleared before rendering to prevent old fragments from appearing.

Must be deleted using delete\_texture.

## Parameters

|               |  |
|---------------|--|
| <i>id</i>     | The id to assign to this texture. If this id is already in use, it deletes the existing texture first before loading this new one. |
| <i>width</i>  | The width of the new texture   |
| <i>height</i> | The height of the next texture   |
| <i>blend</i>  | The blending mode which decides how to react with other textures. Defaults to BLENDMODE_BLEND.                                     |

## Returns

The source region of the new texture created.

6.24.5.2 void VideoContext::delete\_texture ( TextureID *id* )

The deletes the given texture from this context.

## Parameters

|           |                                  |
|-----------|----------------------------------|
| <i>id</i> | The id of the texture to delete. |
|-----------|----------------------------------|

## 6.24.5.3 void VideoContext::display ( void )

Displays the rendered textures on screen.

6.24.5.4 void VideoContext::fill\_texture ( TextureID *id*, int *r*, int *g*, int *b*, int *a* )

Fills the given texture with the given rgba value.

## Parameters

|           |   |
|-----------|---|
| <i>id</i> | The id of the texture to fill with the specified color. |
| <i>r</i>  | The red value 0-255                                     |
| <i>g</i>  | The green value 0-255                                   |
| <i>b</i>  | The blue value 0-255                                    |
| <i>a</i>  | The alpha value 0-255                                   |

## 6.24.5.5 void VideoContext::load\_font ( const std::string &amp; font\_path, int font\_size )

Loads the given font from the path specified.

Only one font may be loaded at any given time. Repeated calls to this function will delete the previous font before creating a new one.

## Parameters

|                  |   |
|------------------|---|
| <i>font_path</i> | The path to the ttf file to load as a font. |
| <i>font_size</i> | The size of the font to load.               |

## 6.24.5.6 VideoContext::Region VideoContext::load\_texture ( TextureID id, const std::string &amp; image\_path, BlendMode blend = BLENDMODE\_BLEND )

Loads a texture from the filename into the specified texture id.

Must be deleted using delete\_texture.

## Parameters

|                   |  |
|-------------------|--|
| <i>id</i>         | The id to assign to this texture. If this id is already in use, it deletes the existing texture first before loading this new one. |
| <i>image_path</i> | The path of the file to load as a texture.   |
| <i>blend</i>      | The blending mode which decides how to react with other textures. Defaults to BLENDMODE_BLEND.                                     |

## Returns

The auto-detected source rectangle for this image.

## 6.24.5.7 void VideoContext::render ( TextureID id, Region \* dest, Region \* src )

Draws the given texture onto the canvas.

## Parameters

|             |   |
|-------------|---|
| <i>id</i>   | The id of the texture to draw onto the canvas.  |
| <i>dest</i> | The destination region to draw onto the canvas. |
| <i>src</i>  | The source region to copy from when drawing.    |

## 6.24.5.8 void VideoContext::render\_clear ( )

Clears the canvas with the default clear color.

## 6.24.5.9 void VideoContext::render\_clear ( TextureID id )

Clears the given texture with the default clear color.



## Parameters

|           |                                 |
|-----------|---------------------------------|
| <i>id</i> | The id of the texture to clear. |
|-----------|---------------------------------|

6.24.5.10 `void VideoContext::render_onto ( TextureID dest_id, TextureID src_id, const Region * dest_region, Region * src_region )`

Draws the given source texture onto the destination texture.

## Parameters

|                    |  |
|--------------------|--|
| <i>dest_id</i>     | The id of the texture that will act as a canvas and be drawn on. |
| <i>src_id</i>      | The id of the texture to draw over the destination Texture.      |
| <i>dest_region</i> | The region to draw the source texture into.                      |
| <i>src_region</i>  | The region to copy the source texture from.                      |

6.24.5.11 `void VideoContext::render_rotate ( TextureID dest_id, TextureID src_id, Region * dest_region, Region * src_region, double angle = 0.0, Flip flip = FLIP_NONE )`

Draws the given source texture onto the destination texture after applying a rotate and flip operation.

## Parameters

|                    |  |
|--------------------|--|
| <i>dest_id</i>     | The id of the texture that will act as a canvas and be drawn on.   |
| <i>src_id</i>      | The id of the texture to draw over the destination Texture.        |
| <i>dest_region</i> | The region to draw the source texture into.                        |
| <i>src_region</i>  | The region to copy the source texture from.                        |
| <i>angle</i>       | The angle in degrees to rotate the source image. Defaults to zero. |
| <i>flip</i>        | The direction to flip the source texture in. Defaults to none.     |

6.24.5.12 `void VideoContext::render_text ( TextureID dest_id, Region * dest_region, const std::string & text, Uint8 r, Uint8 g, Uint8 b, Uint8 a )`

Renders the given text onto the the destination texture.

A successful call to `load_font` must be performed before this method should be called.

## Parameters

|                    |  |
|--------------------|--|
| <i>dest_id</i>     | The destination texture to render onto.                        |
| <i>dest_region</i> | The region on the destination texture to render the font into. |
| <i>text</i>        | The string to render.  |
| <i>r</i>           | The red value 0-255  |
| <i>g</i>           | The green value 0-255  |
| <i>b</i>           | The blue value 0-255   |
| <i>a</i>           | The alpha value 0-255  |

6.24.5.13 `void VideoContext::renderDebugBox ( const Region & region, const DebugColor color, const TextureID layer )`

6.24.5.14 `void VideoContext::swapFullscreen ( void )`

Checks the state of the window and swaps to fullscreen or windowed mode.

## 6.24.6 Member Data Documentation

6.24.6.1 `TTF_Font*` `mediawrap::VideoContext::font` `[private]`

6.24.6.2 `SDL_Renderer*` `mediawrap::VideoContext::renderer` `[private]`

6.24.6.3 `std::unordered_map<TextureID, SDL_Texture*>*` `mediawrap::VideoContext::textures` `[private]`

6.24.6.4 `VideoDisplay*` `mediawrap::VideoContext::video_display` `[private]`

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

- [VideoContext.hpp](#)
- [VideoContext.cpp](#)

## 6.25 mediawrap::VideoDisplay Class Reference

Creates a window and initializes SDL2 and SDL2\_IMG.

```
#include <VideoDisplay.hpp>
```

### Public Member Functions

- [VideoDisplay](#) (const std::string &title, unsigned int width, unsigned int height)  
*Attempts to init SDL2 and SDL2\_IMG and create a window.*
- [~VideoDisplay](#) (void)  
*Destroys the window and renderer.*
- `SDL_Renderer *` [get\\_renderer](#) (void)  
*Creates a renderer attached to this window.*
- void [swapFullscreen](#) (void)

### Private Attributes

- `SDL_Window *` [window](#)

### 6.25.1 Detailed Description

Creates a window and initializes SDL2 and SDL2\_IMG.

Must be destroyed after use.

### 6.25.2 Constructor & Destructor Documentation

6.25.2.1 `mediawrap::VideoDisplay::VideoDisplay ( const std::string & title, unsigned int width, unsigned int height )`

Attempts to init SDL2 and SDL2\_IMG and create a window.

Throws runtime\_error if unable to set up any of these.

#### Parameters

|              |  |
|--------------|--|
| <i>title</i> | The title to display at the top of the window. |
|--------------|--|

|               |                                   |
|---------------|-----------------------------------|
| <i>width</i>  | The width of the window created.  |
| <i>height</i> | The height of the window created. |

#### 6.25.2.2 mediawrap::VideoDisplay::~~VideoDisplay ( void )

Destroys the window and renderer.

Uninitializes SDL and SDL\_Image.

### 6.25.3 Member Function Documentation

#### 6.25.3.1 SDL\_Renderer \* mediawrap::VideoDisplay::get\_renderer ( void )

Creates a renderer attached to this window.

Must be deleted after use.

##### Returns

An SDL2 renderer for this window.

#### 6.25.3.2 void mediawrap::VideoDisplay::swapFullscreen ( void )

### 6.25.4 Member Data Documentation

#### 6.25.4.1 SDL\_Window\* mediawrap::VideoDisplay::window [private]

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

- [VideoDisplay.hpp](#)
- [VideoDisplay.cpp](#)

## 6.26 teamusa::VideoEngine Class Reference

Provides video capabilities that are specific to Legend of the Great Unwashed.

```
#include <VideoEngine.hpp>
```

### Public Member Functions

- [VideoEngine](#) (const std::string &title, unsigned int width, unsigned int height)  
*Creates a new window that provides basic 2D drawing capabilities.*
- [~VideoEngine](#) ()  
*Destroys the video engine after freeing all associated textures.*
- void [loadTexture](#) (const std::string &path, [TextureID](#) id, [ResourceGroup](#) group)  
*Loads the image file from the given path, transforms it into a surface, and pushes it onto the graphics card as a texture.*
- void [render](#) (const [Region](#) &region, const unsigned int layer, const [TextureID](#) id)  
*Renders the texture onto the given layer in the given region.*
- void [renderDebugBox](#) (const [Region](#) &region, const [VideoContext::DebugColor](#)=[VideoContext::DebugColor::BLUE](#))
- void [renderRotate](#) ([Region](#) &region, unsigned int layer, [TextureID](#) id, float angle=0.0)

- Renders the texture onto the given layer in the given region with the given rotation angle.*
  - void [swapFullscreen](#) (void)
    - Calls [swapFullscreen\(\)](#) on [VideoDisplay](#).*
  - bool [isShowingTextbox](#) ()
    - States whether a textbox is currently being displayed or not.*
  - void [showTextbox](#) (const std::string &text)
    - Displays the given text in a textbox.*
  - void [hideTextbox](#) ()
    - Clears the current textbox so it does not appear.*
  - void [deleteTexture](#) ([TextureID](#) id)
    - Removes the current texture from graphics memory.*
  - void [deleteResourceGroup](#) ([ResourceGroup](#) resourceGroup)
    - Deletes all textures associated with the given resource group.*
  - void [display](#) ()
    - Displays all rendered textures on screen.*

### Private Member Functions

- void [clearLayers](#) ()
  - Clears all layers with the default clear color.*

### Private Attributes

- bool [textboxActive](#)
- [TextureID](#) [layers](#) [[NUM\\_LAYERS](#)]
- std::vector< [TextureID](#) > [coreResources](#)
- std::vector< [TextureID](#) > [levelResources](#)
- [VideoContext](#) \* [videoContext](#)
- [Region](#) [textboxPadding](#)
- [Region](#) [textboxRegion](#)

### Static Private Attributes

- static const unsigned int [NUM\\_LAYERS](#) = 7
- static const unsigned int [SHADOW\\_LAYER](#) = 4
- static const [TextureID](#) [TEXT\\_LAYER](#) = 8
- static const [TextureID](#) [MAX\\_RESERVED\\_ID](#) = 1000

#### 6.26.1 Detailed Description

Provides video capabilities that are specific to Legend of the Great Unwashed.

Utilizes [VideoContext](#) to perform rendering.

#### 6.26.2 Constructor & Destructor Documentation

##### 6.26.2.1 [VideoEngine::VideoEngine](#) ( const std::string & *title*, unsigned int *width*, unsigned int *height* )

Creates a new window that provides basic 2D drawing capabilities.

## Parameters

|               |   |
|---------------|---|
| <i>title</i>  | The title to be displayed at the top of the window. |
| <i>width</i>  | The width of the window in pixels.                  |
| <i>height</i> | The height of the window in pixels.                 |

## 6.26.2.2 VideoEngine::~~VideoEngine ( )

Destroys the video engine after freeing all associated textures.

## 6.26.3 Member Function Documentation

## 6.26.3.1 void VideoEngine::clearLayers ( ) [private]

Clears all layers with the default clear color.

Does not modify the textbox layer.

## 6.26.3.2 void VideoEngine::deleteResourceGroup ( ResourceGroup resourceGroup )

Deletes all textures associated with the given resource group.

## Parameters

|                      |  |
|----------------------|--|
| <i>resourceGroup</i> | The group of textures to delete from video memory. |
|----------------------|--|

## 6.26.3.3 void VideoEngine::deleteTexture ( TextureID id )

Removes the current texture from graphics memory.

## Parameters

|           |                                  |
|-----------|----------------------------------|
| <i>id</i> | The id of the texture to delete. |
|-----------|----------------------------------|

## 6.26.3.4 void VideoEngine::display ( void )

Displays all rendered textures on screen.

## 6.26.3.5 void VideoEngine::hideTextbox ( )

Clears the current textbox so it does not appear.

## 6.26.3.6 bool VideoEngine::isShowingTextbox ( )

States whether a textbox is currently being displayed or not.

## Returns

The status of the textbox.

**6.26.3.7** void VideoEngine::loadTexture ( const std::string & *path*, TextureID *id*, ResourceGroup *group* )

Loads the image file from the given path, transforms it into a surface, and pushes it onto the graphics card as a texture.

## Parameters

|                 |   |
|-----------------|---|
| <i>path</i>     | The relative location of the image to load. |
| <i>id</i>       | The id to assign to the loaded texture.     |
| <i>resGroup</i> | The group to load the resource into.        |

## 6.26.3.8 void VideoEngine::render ( const Region &amp; region, const unsigned int layer, const TextureID id )

Renders the texture onto the given layer in the given region.

## Parameters

|               |   |
|---------------|---|
| <i>region</i> | The region to draw the texture into.                |
| <i>layer</i>  | The layer to render the image onto (0-6) are valid. |
| <i>id</i>     | The id of the texture to draw.                      |

## 6.26.3.9 void VideoEngine::renderDebugBox ( const Region &amp; region, const VideoContext::DebugColor color = VideoContext::DebugColor::BLUE )

## 6.26.3.10 void VideoEngine::renderRotate ( Region &amp; region, unsigned int layer, TextureID id, float angle = 0.0 )

Renders the texture onto the given layer in the given region with the given rotation angle.

## Parameters

|               |  |
|---------------|--|
| <i>region</i> | The region to draw the texture into.                     |
| <i>layer</i>  | The layer to render the image onto (0-6) are valid.      |
| <i>id</i>     | The id of the texture to draw.                           |
| <i>angle</i>  | The angle in degrees to rotate the image. Defaults to 0. |

## 6.26.3.11 void VideoEngine::showTextbox ( const std::string &amp; text )

Displays the given text in a textbox.

## Parameters

|             |                                |
|-------------|--------------------------------|
| <i>text</i> | The text to display on screen. |
|-------------|--------------------------------|

## 6.26.3.12 void VideoEngine::swapFullscreen ( void )

Calls [swapFullscreen\(\)](#) on VideoDisplay.

## 6.26.4 Member Data Documentation

6.26.4.1 std::vector<TextureID> teamusa::VideoEngine::coreResources [private]

6.26.4.2 TextureID teamusa::VideoEngine::layers[NUM\_LAYERS] [private]

6.26.4.3 std::vector<TextureID> teamusa::VideoEngine::levelResources [private]

6.26.4.4 const TextureID teamusa::VideoEngine::MAX\_RESERVED\_ID = 1000 [static], [private]

6.26.4.5 const unsigned int teamusa::VideoEngine::NUM\_LAYERS = 7 [static], [private]

6.26.4.6 `const unsigned int teamusa::VideoEngine::SHADOW_LAYER = 4` `[static], [private]`

6.26.4.7 `const TextureID teamusa::VideoEngine::TEXT_LAYER = 8` `[static], [private]`

6.26.4.8 `bool teamusa::VideoEngine::textboxActive` `[private]`

6.26.4.9 `Region teamusa::VideoEngine::textboxPadding` `[private]`

6.26.4.10 `Region teamusa::VideoEngine::textboxRegion` `[private]`

6.26.4.11 `VideoContext* teamusa::VideoEngine::videoContext` `[private]`

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

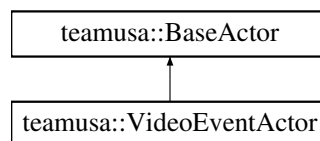
- [VideoEngine.hpp](#)
- [VideoEngine.cpp](#)

## 6.27 teamusa::VideoEventActor Class Reference

Will display a texture and perform no action until clicked.

```
#include <VideoEventActor.h>
```

Inheritance diagram for teamusa::VideoEventActor:



### Public Member Functions

- [VideoEventActor](#) ([Region](#) region, int textureID, [ActorEventType](#) eventType, int eventValue, int layer)
- virtual `~VideoEventActor` (void) override
- virtual const [ActorEvent](#) `onClick` ([Player](#) &player) override  
*Generates an [ActorEvent](#) when the actor's region is clicked.*
- virtual const [ActorEvent](#) `onHover` ([Player](#) &player) override  
*Generates an [ActorEvent](#) when the actor's region is hovered over.*
- virtual const [ActorEvent](#) `step` ([Player](#) &player)  
*Advances the actor one frame.*

### Private Attributes

- [ActorEvent](#) actorEvent

### Additional Inherited Members

#### 6.27.1 Detailed Description

Will display a texture and perform no action until clicked.

The TextureID can be set to an invalid value during construction if no image needs to be displayed.



## 6.27.2 Constructor & Destructor Documentation

6.27.2.1 `VideoEventActor::VideoEventActor ( Region region, int textureID, ActorEventType eventType, int eventValue, int layer )` `[explicit]`

6.27.2.2 `VideoEventActor::~VideoEventActor ( void )` `[override]`,`[virtual]`

## 6.27.3 Member Function Documentation

6.27.3.1 `const ActorEvent VideoEventActor::onClick ( Player & player )` `[override]`,`[virtual]`

Generates an [ActorEvent](#) when the actor's region is clicked.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.27.3.2 `const ActorEvent VideoEventActor::onHover ( Player & player )` `[override]`,`[virtual]`

Generates an [ActorEvent](#) when the actor's region is hovered over.

Parameters

|                        |             |
|------------------------|-------------|
| <a href="#">Player</a> | The player. |
|------------------------|-------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Reimplemented from [teamusa::BaseActor](#).

6.27.3.3 `const ActorEvent VideoEventActor::step ( Player & player )` `[virtual]`

Advances the actor one frame.

Parameters

|                        |            |
|------------------------|------------|
| <a href="#">Player</a> | The player |
|------------------------|------------|

Returns

Returns an [ActorEvent](#) that triggers an actor to perform an action.

Implements [teamusa::BaseActor](#).

## 6.27.4 Member Data Documentation

6.27.4.1 `ActorEvent teamusa::VideoEventActor::actorEvent` `[private]`

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

- [VideoEventActor.h](#)
- [VideoEventActor.cpp](#)



# Chapter 7

## File Documentation

### 7.1 ActorEvent.h File Reference

Declares ActorEvent struct.

```
#include "Headers.h"
```

#### Classes

- class [teamusa::ActorEvent](#)

*Event data generated by Actors, handled by [Engine](#).*

#### Namespaces

- [teamusa](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

#### Enumerations

- enum [teamusa::ActorEventType](#) {  
    [teamusa::Nil](#) = -1, [teamusa::ChangeScene](#), [teamusa::LoadLevel](#), [teamusa::PlayAudio](#),  
    [teamusa::NewGame](#), [teamusa::LoadGame](#), [teamusa::DisplayText](#), [teamusa::ExitGame](#),  
    [teamusa::StreamAudio](#) }

*Events that actors can trigger.*

#### 7.1.1 Detailed Description

Declares ActorEvent struct.

### 7.2 Assert.h File Reference

Declares custom Assert macro.

## Namespaces

- [teamusa](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

## Macros

- `#define Assert(exp) ;`

### 7.2.1 Detailed Description

Declares custom Assert macro.

### 7.2.2 Macro Definition Documentation

#### 7.2.2.1 `#define Assert( exp );`

## 7.3 AudioEngine.cpp File Reference

```
#include "AudioEngine.hpp"
```

## 7.4 AudioEngine.hpp File Reference

```
#include <string>
#include <vector>
#include "AudioPlayer.hpp"
#include "Engine/ResourceGroup.hpp"
```

## Classes

- class [teamusa::AudioEngine](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

## Namespaces

- [teamusa](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

## Typedefs

- typedef [mediawrap::AudioPlayer::AudioID](#) [teamusa::AudioID](#)

## 7.5 AudioPlayer.cpp File Reference

```
#include "AudioPlayer.hpp"
```

## 7.6 AudioPlayer.hpp File Reference

```
#include <stdexcept>
#include <string>
#include <unordered_map>
#include "SDL2/SDL.h"
#include "SDL2/SDL_mixer.h"
```

### Classes

- class [mediawrap::AudioPlayer](#)  
*Provides basic audio playing capabilities with WAV files.*

### Namespaces

- [mediawrap](#)  
*Provides basic audio playing capabilities with WAV files.*

## 7.7 AudioStreamActor.cpp File Reference

Implements AudioStreamActor class.

```
#include "AudioStreamActor.h"
```

### 7.7.1 Detailed Description

Implements AudioStreamActor class.

## 7.8 AudioStreamActor.h File Reference

Declares AudioStreamActor class.

```
#include "BaseActor.h"
```

### Classes

- class [teamusa::AudioStreamActor](#)  
*If this actor is not activated, it will emit a StreamAudio event and set its status to activated when the step method is called.*

### Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.8.1 Detailed Description

Declares AudioStreamActor class.

## 7.9 BaseActor.cpp File Reference

```
#include "BaseActor.h"  
#include "Engine/Assert.h"  
#include "Engine/Point.h"
```

## 7.10 BaseActor.h File Reference

Implements BaseActor class.

```
#include "ActorEvent.h"  
#include "Audio/AudioEngine.hpp"  
#include "Video/VideoEngine.hpp"
```

### Classes

- struct [teamusa::ActorVideo](#)
- class [teamusa::BaseActor](#)

*Abstract class which all actors must derive from.*

### Namespaces

- [teamusa](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.10.1 Detailed Description

Implements BaseActor class.

Declares BaseActor class.

## 7.11 CursorStyle.h File Reference

Declares CursorStyle enumerations.

### Namespaces

- [teamusa](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

## Enumerations

- enum [teamusa::CursorStyle](#) {  
    [teamusa::CursorStyle::CURSOR\\_DEFAULT](#),    [teamusa::CursorStyle::CURSOR\\_SELECT](#),    [teamusa::CursorStyle::CURSOR\\_LEFT](#), [teamusa::CursorStyle::CURSOR\\_RIGHT](#),  
    [teamusa::CursorStyle::CURSOR\\_UP](#), [teamusa::CursorStyle::CURSOR\\_DOWN](#) }

*The possible styles for the mouse cursor.*

### 7.11.1 Detailed Description

Declares CursorStyle enumerations.

## 7.12 DelayedAudioActor.cpp File Reference

Implements DelayedAudioActor class.

```
#include "DelayedAudioActor.h"
```

### 7.12.1 Detailed Description

Implements DelayedAudioActor class.

## 7.13 DelayedAudioActor.h File Reference

Declares DelayedAudioActor class.

```
#include "BaseActor.h"
```

## Classes

- class [teamusa::DelayedAudioActor](#)  
*Will increment a counter every time the step method is called.*

## Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.13.1 Detailed Description

Declares DelayedAudioActor class.

## 7.14 DelayedVideoActor.cpp File Reference

Declares DelayedVideoActor class.

```
#include "DelayedVideoActor.h"
#include <iostream>
```

### 7.14.1 Detailed Description

Declares DelayedVideoActor class.

## 7.15 DelayedVideoActor.h File Reference

Declares DelayedVideoActor class.

```
#include "BaseActor.h"
```

### Classes

- class [teamusa::DelayedVideoActor](#)  
*Will increment a counter every time the step method is called.*

### Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.15.1 Detailed Description

Declares DelayedVideoActor class.

## 7.16 Engine.cpp File Reference

Implements Engine class.

```
#include "Engine.h"
#include "Actor/AudioStreamActor.h"
#include "Actor/SceneLink.h"
#include "Actor/TextboxSpawnActor.h"
#include "Actor/VideoActor.h"
#include "Audio/AudioEngine.hpp"
#include "Engine/Assert.h"
#include "Engine/ResourceGroup.hpp"
#include "Engine/Timer.h"
#include "Video/VideoEngine.hpp"
```

### Macros

- #define [BIND](#)(function) ( std::bind( function, this, std::placeholders::\_1, std::placeholders::\_2 ) )



## Variables

- static const double `FRAME_TIME` = 16.67

### 7.16.1 Detailed Description

Implements Engine class.

### 7.16.2 Macro Definition Documentation

7.16.2.1 `#define BIND( function ) ( std::bind( function, this, std::placeholders::_1, std::placeholders::_2 ) )`

### 7.16.3 Variable Documentation

7.16.3.1 `const double FRAME_TIME = 16.67` [static]

## 7.17 Engine.h File Reference

Declares Engine class.

```
#include "Headers.h"
#include "Engine/Level.h"
#include "GameSaveSerializer/GameSaveSerializer.h"
#include "Player/Player.h"
```

## Classes

- class `teamusa::Engine`  
*Processes all components of the game each frame.*

## Namespaces

- `teamusa`  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.17.1 Detailed Description

Declares Engine class.

## 7.18 GameSaveSerializer.cpp File Reference

```
#include "GameSaveSerializer.h"
#include "Engine/Assert.h"
```

## Namespaces

- `teamusa`  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

## 7.19 GameSaveSerializer.h File Reference

Declares save file serializer class.

```
#include <vector>
#include <fstream>
#include <mutex>
#include <string>
#include <thread>
#include "Player/Player.h"
```

### Classes

- class [teamusa::GameSaveSerializer](#)  
*Provides multithreaded save, single-thread load of save files.*

### Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

#### 7.19.1 Detailed Description

Declares save file serializer class.

## 7.20 Headers.h File Reference

Easy way to include all headers needed.

```
#include <exception>
#include <fstream>
#include <functional>
#include <iostream>
#include <map>
#include <memory>
#include <stack>
#include <string>
#include <vector>
#include <stdint.h>
```

#### 7.20.1 Detailed Description

Easy way to include all headers needed.

## 7.21 InventoryItemActor.cpp File Reference

Implements InventoryItemActor class.

```
#include "InventoryItemActor.h"
#include "Player/Player.h"
```

### 7.21.1 Detailed Description

Implements InventoryItemActor class.

## 7.22 InventoryItemActor.h File Reference

Declares InventoryItemActor class.

```
#include "BaseActor.h"
```

### Classes

- class [teamusa::InventoryItemActor](#)  
*InventoryItemActor creates a collectible item in the game environment.*

### Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.22.1 Detailed Description

Declares InventoryItemActor class.

## 7.23 Level.cpp File Reference

Implements Level class.

```
#include "Assert.h"
#include "Level.h"
#include "Actor/ActorEvent.h"
#include "Actor/AudioStreamActor.h"
#include "Actor/DelayedAudioActor.h"
#include "Actor/DelayedVideoActor.h"
#include "Actor/InventoryItemActor.h"
#include "Actor/LevelLink.h"
#include "Actor/MovingActor.h"
#include "Actor/ResponsiveAudioActor.h"
#include "Actor/ResponsiveVideoActor.h"
#include "Actor/SceneLink.h"
#include "Actor/TextboxSpawnActor.h"
#include "Actor/VideoActor.h"
#include "Actor/VideoEventActor.h"
#include "Audio/AudioEngine.hpp"
#include "Video/VideoEngine.hpp"
```

### Functions

- static std::istream & [operator>>](#) (std::istream &fs, [Region](#) &dst)
- static std::istream & [operator>>](#) (std::istream &fs, [ActorEventType](#) &dst)
- static void [loadError](#) (const std::string &msg)

### 7.23.1 Detailed Description

Implements Level class.

### 7.23.2 Function Documentation

7.23.2.1 `static void loadError ( const std::string & msg ) [static]`

7.23.2.2 `static std::istream& operator>> ( std::istream & fs, Region & dst ) [inline],[static]`

7.23.2.3 `static std::istream& operator>> ( std::istream & fs, ActorEventType & dst ) [inline],[static]`

## 7.24 Level.h File Reference

Declares Level class.

```
#include <unordered_map>
#include "Headers.h"
```

### Classes

- class [teamusa::Level](#)  
A [Level](#) is a container of Scenes and Actors corresponding to those scenes.
- class [teamusa::Level::Scene](#)  
A scene is a collection of images (Actors) that is displayed on the screen.

### Namespaces

- [teamusa](#)  
Provides project-specific audio functionality for Legend of the Great Unwashed.

### Typedefs

- typedef std::shared\_ptr< BaseActor > [teamusa::BaseActorPtr](#)
- typedef std::vector< BaseActorPtr > [teamusa::ActorList](#)

### 7.24.1 Detailed Description

Declares Level class.

## 7.25 LevelLink.cpp File Reference

Implements LevelLink class.

```
#include "LevelLink.h"
#include "Player/Player.h"
```

### 7.25.1 Detailed Description

Implements LevelLink class.

## 7.26 LevelLink.h File Reference

Declares LevelLink class.

```
#include "BaseActor.h"
```

### Classes

- class [teamusa::LevelLink](#)

### Namespaces

- [teamusa](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.26.1 Detailed Description

Declares LevelLink class.

## 7.27 main.cpp File Reference

Entry point of program.

```
#include "Headers.h"  
#include "Engine/Engine.h"
```

### Namespaces

- [MainNS](#)

### Functions

- static void [MainNS::logError](#) (const std::string &desc)  
*Writes an error message to the log file.*
- int [main](#) (int argc, char \*\*argv)

### 7.27.1 Detailed Description

Entry point of program.

## 7.27.2 Function Documentation

7.27.2.1 `int main ( int argc, char ** argv )`

## 7.28 MovingActor.cpp File Reference

```
#include "MovingActor.h"  
#include "Player/Player.h"
```

## 7.29 MovingActor.h File Reference

Declares movingActor class.

```
#include "BaseActor.h"
```

### Classes

- class [teamusa::MovingActor](#)

*Will transition from one region to the next by calculating the distance to move each frame for a set number of frames.*

### Namespaces

- [teamusa](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.29.1 Detailed Description

Declares movingActor class.

## 7.30 Player.cpp File Reference

Implements Player class.

```
#include "Player.h"  
#include "Engine/Assert.h"
```

### 7.30.1 Detailed Description

Implements Player class.

## 7.31 Player.h File Reference

Declares Player class.

```
#include "Headers.h"
#include "CursorStyle.h"
#include "Engine/Point.h"
#include "Video/VideoEngine.hpp"
```

## Classes

- class [teamusa::Player](#)  
*Handles all data relevant to the player engaging the game.*

## Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.31.1 Detailed Description

Declares Player class.

## 7.32 Point.h File Reference

Declares Point struct.

```
#include <stdint.h>
```

## Classes

- class [teamusa::Point](#)  
*An (x,y) coordinate within the rendering window.*

## Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.32.1 Detailed Description

Declares Point struct.

## 7.33 ResourceGroup.hpp File Reference

## Enumerations

- enum [ResourceGroup](#) { [CORE\\_RESOURCE](#), [LEVEL\\_RESOURCE](#) }

### 7.33.1 Enumeration Type Documentation

#### 7.33.1.1 enum ResourceGroup

Enumerator

**CORE\_RESOURCE**  
**LEVEL\_RESOURCE**

## 7.34 ResponsiveAudioActor.cpp File Reference

Implements ResponsiveAudioActor class.

```
#include "ResponsiveAudioActor.h"
```

### 7.34.1 Detailed Description

Implements ResponsiveAudioActor class.

## 7.35 ResponsiveAudioActor.h File Reference

Declares ResponsiveAudioActor class.

```
#include "BaseActor.h"
```

### Classes

- class [teamusa::ResponsiveAudioActor](#)  
  *\ Brief: Will increment the value of stepCount until it is equal to durationSteps for each call to the step method.*

### Namespaces

- [teamusa](#)  
  *Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.35.1 Detailed Description

Declares ResponsiveAudioActor class.

## 7.36 ResponsiveVideoActor.cpp File Reference

Will display the default TextureID.

```
#include "ResponsiveVideoActor.h"
```

### 7.36.1 Detailed Description

Will display the default TextureID.



## 7.37 ResponsiveVideoActor.h File Reference

Declares ResponsivevideoActor class.

```
#include "BaseActor.h"
```

### Classes

- class [teamusa::ResponsiveVideoActor](#)

### Namespaces

- [teamusa](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

#### 7.37.1 Detailed Description

Declares ResponsivevideoActor class.

## 7.38 SceneLink.cpp File Reference

Implements SceneLink class.

```
#include "SceneLink.h"  
#include "Player/Player.h"
```

#### 7.38.1 Detailed Description

Implements SceneLink class.

## 7.39 SceneLink.h File Reference

Declares SceneLink class.

```
#include "BaseActor.h"  
#include "Player/CursorStyle.h"
```

### Classes

- class [teamusa::SceneLink](#)

*Allows the player to transition between scenes.*

### Namespaces

- [teamusa](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.39.1 Detailed Description

Declares SceneLink class.

## 7.40 TextboxSpawnActor.cpp File Reference

Declares TextboxSpawnActor class.

```
#include "TextboxSpawnActor.h"
```

### 7.40.1 Detailed Description

Declares TextboxSpawnActor class.

## 7.41 TextboxSpawnActor.h File Reference

Declares TextboxSpawnActor class.

```
#include "BaseActor.h"  
#include <string>
```

## Classes

- class [teamusa::TextboxSpawnActor](#)  
*Spawns a textbox that gives the player written information.*

## Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.41.1 Detailed Description

Declares TextboxSpawnActor class.

## 7.42 Timer.cpp File Reference

Implements Timer class.

```
#include "Engine/Timer.h"  
#include <SDL2/SDL.h>
```

### 7.42.1 Detailed Description

Implements Timer class.

## 7.43 Timer.h File Reference

Declares Timer class.

```
#include "Headers.h"
```

### Classes

- class [teamusa::Timer](#)  
*A timer that counts up from zero in milliseconds.*

### Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

#### 7.43.1 Detailed Description

Declares Timer class.

## 7.44 VideoActor.cpp File Reference

Implements VideoActor class.

```
#include "VideoActor.h"
```

#### 7.44.1 Detailed Description

Implements VideoActor class.

## 7.45 VideoActor.h File Reference

This module makes sure An actor that will only display a texture at a given region.

```
#include "BaseActor.h"
```

### Classes

- class [teamusa::VideoActor](#)

### Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.45.1 Detailed Description

This module makes sure An actor that will only display a texture at a given region.

This actor will have no interaction with the player.

## 7.46 VideoContext.cpp File Reference

```
#include "VideoContext.hpp"
```

## 7.47 VideoContext.hpp File Reference

```
#include <unordered_map>
#include <string>
#include "SDL2/SDL.h"
#include "SDL2/SDL_image.h"
#include "SDL2/SDL_ttf.h"
#include "VideoDisplay.hpp"
```

### Classes

- class [mediawrap::VideoContext](#)  
*Provides basic 2D rendering capabilities.*

### Namespaces

- [mediawrap](#)  
*Provides basic audio playing capabilities with WAV files.*

## 7.48 VideoDisplay.cpp File Reference

```
#include "VideoDisplay.hpp"
```

## 7.49 VideoDisplay.hpp File Reference

```
#include <stdexcept>
#include "SDL2/SDL.h"
#include "SDL2/SDL_image.h"
#include "SDL2/SDL_ttf.h"
```

### Classes

- class [mediawrap::VideoDisplay](#)  
*Creates a window and initializes SDL2 and SDL2\_IMG.*

## Namespaces

- [mediawrap](#)

*Provides basic audio playing capabilities with WAV files.*

## 7.50 VideoEngine.cpp File Reference

```
#include "VideoEngine.hpp"
```

## 7.51 VideoEngine.hpp File Reference

```
#include <stdexcept>
#include <string>
#include <vector>
#include "VideoContext.hpp"
#include "Engine/ResourceGroup.hpp"
```

## Classes

- class [teamusa::VideoEngine](#)

*Provides video capabilities that are specific to Legend of the Great Unwashed.*

## Namespaces

- [teamusa](#)

*Provides project-specific audio functionality for Legend of the Great Unwashed.*

## Typedefs

- typedef [mediawrap::VideoContext::TextureID](#) [teamusa::TextureID](#)
- typedef [mediawrap::VideoContext::Region](#) [teamusa::Region](#)

## 7.52 VideoEventActor.cpp File Reference

declares VideoEventActor class

```
#include "VideoEventActor.h"
#include "Player/Player.h"
```

### 7.52.1 Detailed Description

declares VideoEventActor class

## 7.53 VideoEventActor.h File Reference

Declares VideoEventActor class.

```
#include "BaseActor.h"
```

### Classes

- class [teamusa::VideoEventActor](#)  
*Will display a texture and perform no action until clicked.*

### Namespaces

- [teamusa](#)  
*Provides project-specific audio functionality for Legend of the Great Unwashed.*

### 7.53.1 Detailed Description

Declares VideoEventActor class.

# Index

- ~AudioPlayer
  - mediawrap::AudioPlayer, [17](#)
- ~AudioStreamActor
  - teamusa::AudioStreamActor, [19](#)
- ~BaseActor
  - teamusa::BaseActor, [22](#)
- ~DelayedAudioActor
  - teamusa::DelayedAudioActor, [25](#)
- ~DelayedVideoActor
  - teamusa::DelayedVideoActor, [26](#)
- ~Engine
  - teamusa::Engine, [28](#)
- ~GameSaveSerializer
  - teamusa::GameSaveSerializer, [31](#)
- ~InventoryItemActor
  - teamusa::InventoryItemActor, [33](#)
- ~LevelLink
  - teamusa::LevelLink, [37](#)
- ~MovingActor
  - teamusa::MovingActor, [40](#)
- ~Player
  - teamusa::Player, [42](#)
- ~ResponsiveAudioActor
  - teamusa::ResponsiveAudioActor, [46](#)
- ~ResponsiveVideoActor
  - teamusa::ResponsiveVideoActor, [47](#)
- ~SceneLink
  - teamusa::SceneLink, [51](#)
- ~TextboxSpawnActor
  - teamusa::TextboxSpawnActor, [54](#)
- ~Timer
  - teamusa::Timer, [55](#)
- ~VideoActor
  - teamusa::VideoActor, [57](#)
- ~VideoContext
  - mediawrap::VideoContext, [61](#)
- ~VideoDisplay
  - mediawrap::VideoDisplay, [65](#)
- ~VideoEngine
  - teamusa::VideoEngine, [67](#)
- ~VideoEventActor
  - teamusa::VideoEventActor, [71](#)
- activated
  - teamusa::AudioStreamActor, [20](#)
  - teamusa::TextboxSpawnActor, [54](#)
- activeScene
  - teamusa::Level, [36](#)
- ActorEvent
  - teamusa::ActorEvent, [13](#)
- actorEvent
  - teamusa::VideoEventActor, [71](#)
- ActorEvent.h, [73](#)
- ActorEventHandler
  - teamusa::Engine, [28](#)
- ActorEventType
  - teamusa, [11](#)
- ActorList
  - teamusa, [11](#)
- ActorVideo
  - teamusa::ActorVideo, [14](#)
- actors
  - teamusa::Level::Scene, [50](#)
- addItem
  - teamusa::Player, [42](#)
- Assert
  - Assert.h, [74](#)
- Assert.h, [73](#)
- Assert, [74](#)
- audio\_buffer
  - mediawrap::AudioPlayer, [18](#)
- audio\_channels
  - mediawrap::AudioPlayer, [18](#)
- audio\_format
  - mediawrap::AudioPlayer, [18](#)
- audio\_rate
  - mediawrap::AudioPlayer, [18](#)
- audio\_samples
  - mediawrap::AudioPlayer, [18](#)
- audio\_stream
  - mediawrap::AudioPlayer, [18](#)
- AudioEngine.cpp, [74](#)
- AudioEngine.hpp, [74](#)
- AudioID
  - mediawrap::AudioPlayer, [17](#)
  - teamusa, [11](#)
- audiold
  - teamusa::DelayedAudioActor, [25](#)
- AudioPlayer
  - mediawrap::AudioPlayer, [17](#)
- audioPlayer
  - teamusa::AudioEngine, [15](#)
- AudioPlayer.cpp, [74](#)
- AudioPlayer.hpp, [75](#)
- AudioStreamActor
  - teamusa::AudioStreamActor, [19](#)
- AudioStreamActor.cpp, [75](#)
- AudioStreamActor.h, [75](#)
- BIND

- Engine.cpp, 79
- BLENDMODE\_ADD
  - mediawrap::VideoContext, 59
- BLENDMODE\_BLEND
  - mediawrap::VideoContext, 59
- BLENDMODE\_MOD
  - mediawrap::VideoContext, 59
- BLENDMODE\_NONE
  - mediawrap::VideoContext, 59
- BLUE
  - mediawrap::VideoContext, 59
- BaseActor
  - teamusa::BaseActor, 22
- BaseActor.cpp, 76
- BaseActor.h, 76
- BaseActorPtr
  - teamusa, 11
- bglImageID
  - teamusa::Level::Scene, 50
- BlendMode
  - mediawrap::VideoContext, 59
- CORE\_RESOURCE
  - ResourceGroup.hpp, 86
- CURSOR\_DEFAULT
  - teamusa, 12
- CURSOR\_DEFAULT\_ID
  - teamusa::Player, 44
- CURSOR\_DOWN
  - teamusa, 12
- CURSOR\_DOWN\_ID
  - teamusa::Player, 44
- CURSOR\_LEFT
  - teamusa, 12
- CURSOR\_LEFT\_ID
  - teamusa::Player, 44
- CURSOR\_RIGHT
  - teamusa, 12
- CURSOR\_RIGHT\_ID
  - teamusa::Player, 44
- CURSOR\_SELECT
  - teamusa, 12
- CURSOR\_SELECT\_ID
  - teamusa::Player, 44
- CURSOR\_UP
  - teamusa, 12
- CURSOR\_UP\_ID
  - teamusa::Player, 44
- ChangeScene
  - teamusa, 11
- changeScene
  - teamusa::Level, 35
- clear\_samples
  - mediawrap::AudioPlayer, 17
- clearAll
  - teamusa::Level, 35
- clearLayers
  - teamusa::VideoEngine, 67
- clickAudioId
  - teamusa::ResponsiveAudioActor, 47
- clickTexture
  - teamusa::ResponsiveVideoActor, 49
- coreResources
  - teamusa::AudioEngine, 15
  - teamusa::VideoEngine, 69
- create\_texture
  - mediawrap::VideoContext, 61
- currentStep
  - teamusa::DelayedAudioActor, 25
  - teamusa::DelayedVideoActor, 27
  - teamusa::MovingActor, 40
- CursorStyle
  - teamusa, 12
- cursorStyle
  - teamusa::SceneLink, 53
- CursorStyle.h, 76
- DebugColor
  - mediawrap::VideoContext, 59
- defaultTextureId
  - teamusa::ResponsiveVideoActor, 49
- delaySteps
  - teamusa::DelayedAudioActor, 25
  - teamusa::DelayedVideoActor, 27
- DelayedAudioActor
  - teamusa::DelayedAudioActor, 25
- DelayedAudioActor.cpp, 77
- DelayedAudioActor.h, 77
- DelayedVideoActor
  - teamusa::DelayedVideoActor, 26
- DelayedVideoActor.cpp, 77
- DelayedVideoActor.h, 78
- delete\_sample
  - mediawrap::AudioPlayer, 17
- delete\_texture
  - mediawrap::VideoContext, 61
- deleteResourceGroup
  - teamusa::VideoEngine, 67
- deleteSound
  - teamusa::AudioEngine, 15
- deleteSoundGroup
  - teamusa::AudioEngine, 15
- deleteTexture
  - teamusa::VideoEngine, 67
- disappear
  - teamusa::DelayedVideoActor, 27
- display
  - mediawrap::VideoContext, 61
  - teamusa::VideoEngine, 67
- DisplayText
  - teamusa, 11
- endRegion
  - teamusa::MovingActor, 40
- Engine
  - teamusa::Engine, 28
- Engine.cpp, 78
- BIND, 79



- FRAME\_TIME, 79
- Engine.h, 79
- ExitGame
  - teamusa, 12
- FLASHLIGHT\_ID
  - teamusa::Player, 44
- FLIP\_HORIZONTAL
  - mediawrap::VideoContext, 59
- FLIP\_NONE
  - mediawrap::VideoContext, 59
- FLIP\_VERTICAL
  - mediawrap::VideoContext, 59
- FRAME\_TIME
  - Engine.cpp, 79
- fileLock
  - teamusa::GameSaveSerializer, 32
- fill\_texture
  - mediawrap::VideoContext, 61
- Flip
  - mediawrap::VideoContext, 59
- font
  - mediawrap::VideoContext, 63
- freeAndLoadLevel
  - teamusa::Engine, 28
- GREEN
  - mediawrap::VideoContext, 59
- GameSaveSerializer
  - teamusa::GameSaveSerializer, 31
- GameSaveSerializer.cpp, 79
- GameSaveSerializer.h, 80
- get\_renderer
  - mediawrap::VideoDisplay, 65
- getActors
  - teamusa::Level, 35
- getBGImageID
  - teamusa::Level, 35
- getCursorTextureID
  - teamusa::Player, 42
- getInventory
  - teamusa::Player, 42
- getLayer
  - teamusa::BaseActor, 22
- getMouseClickedState
  - teamusa::Engine, 28
- getMouseCoordinates
  - teamusa::Engine, 28
- getPath
  - teamusa::AudioStreamActor, 19
- getPosition
  - teamusa::Player, 42
- getRegion
  - teamusa::BaseActor, 22
- getScene
  - teamusa::Level, 36
- getSceneID
  - teamusa::LevelLink, 37
- getText
  - teamusa::LevelLink, 37
  - teamusa::SceneLink, 51
  - teamusa::TextboxSpawnActor, 54
- getTextureID
  - teamusa::BaseActor, 22
- getTicks
  - teamusa::Timer, 55
- hGrowth
  - teamusa::MovingActor, 40
- handleEvent
  - teamusa::Engine, 29
- hasItem
  - teamusa::Player, 43
- hasVideo
  - teamusa::BaseActor, 22
- Headers.h, 80
- hideTextbox
  - teamusa::VideoEngine, 67
- hoverAudioId
  - teamusa::ResponsiveAudioActor, 47
- hoverTexture
  - teamusa::ResponsiveVideoActor, 49
- Inventory
  - teamusa::Player, 42
- InventoryItemActor
  - teamusa::InventoryItemActor, 33
- InventoryItemActor.cpp, 80
- InventoryItemActor.h, 81
- isActive
  - teamusa::MovingActor, 41
- isInBounds
  - teamusa::BaseActor, 22
- isShowingTextbox
  - teamusa::VideoEngine, 67
- itemId
  - teamusa::InventoryItemActor, 34
- itemRequiredText
  - teamusa::LevelLink, 38
  - teamusa::SceneLink, 53
- LEVEL\_RESOURCE
  - ResourceGroup.hpp, 86
- layer
  - teamusa::ActorVideo, 14
- layers
  - teamusa::VideoEngine, 69
- Level
  - teamusa::Level, 35
- Level.cpp, 81
- loadError, 82
- operator>>, 82
- Level.h, 82
- levelID
  - teamusa::LevelLink, 38
- LevelLink
  - teamusa::LevelLink, 37
- LevelLink.cpp, 82

- LevelLink.h, 83
- levelResources
  - teamusa::AudioEngine, 15
  - teamusa::VideoEngine, 69
- load
  - teamusa::GameSaveSerializer, 31
- load\_font
  - mediawrap::VideoContext, 62
- load\_sample
  - mediawrap::AudioPlayer, 17
- load\_stream
  - mediawrap::AudioPlayer, 17
- load\_texture
  - mediawrap::VideoContext, 62
- loadError
  - Level.cpp, 82
- LoadGame
  - teamusa, 11
- LoadLevel
  - teamusa, 11
- loadLevel
  - teamusa::Level, 36
- loadSound
  - teamusa::AudioEngine, 15
- loadTexture
  - teamusa::VideoEngine, 67
- logError
  - MainNS, 9
- MAX\_RESERVED\_ID
  - teamusa::AudioEngine, 15
  - teamusa::VideoEngine, 69
- mActorEventHandlers
  - teamusa::Engine, 30
- mAudioEngine
  - teamusa::Engine, 30
- mAudioID
  - teamusa::BaseActor, 24
- mCurrentLevelID
  - teamusa::Engine, 30
- mCursorStyle
  - teamusa::Player, 44
- mInventory
  - teamusa::Player, 44
- mIsRunning
  - teamusa::Engine, 30
- mLayer
  - teamusa::Player, 44
- mLevel
  - teamusa::Engine, 30
- mMainMenu
  - teamusa::Engine, 30
- MOUSE\_CLICK\_ID
  - teamusa::Player, 44
- mPauseTicks
  - teamusa::Timer, 56
- mPaused
  - teamusa::Timer, 56
- mPlayer
  - teamusa::Engine, 31
- mPosition
  - teamusa::Player, 44
- mRegion
  - teamusa::BaseActor, 24
  - teamusa::Player, 44
- mSerializer
  - teamusa::Engine, 31
- mStartTicks
  - teamusa::Timer, 56
- mStarted
  - teamusa::Timer, 56
- mTextureID
  - teamusa::Player, 44
- mVideo
  - teamusa::BaseActor, 24
- mVideoEngine
  - teamusa::Engine, 31
- main
  - main.cpp, 84
- main.cpp, 83
  - main, 84
- MainNS, 9
  - logError, 9
- mediawrap, 9
  - mediawrap::AudioPlayer, 16
    - ~AudioPlayer, 17
    - audio\_buffer, 18
    - audio\_channels, 18
    - audio\_format, 18
    - audio\_rate, 18
    - audio\_samples, 18
    - audio\_stream, 18
    - AudioID, 17
    - AudioPlayer, 17
    - clear\_samples, 17
    - delete\_sample, 17
    - load\_sample, 17
    - load\_stream, 17
    - play\_sample, 18
    - stream\_audio, 18
  - mediawrap::VideoContext, 57
    - ~VideoContext, 61
    - BLENDDMODE\_ADD, 59
    - BLENDDMODE\_BLEND, 59
    - BLENDDMODE\_MOD, 59
    - BLENDDMODE\_NONE, 59
    - BLUE, 59
    - BlendMode, 59
    - create\_texture, 61
    - DebugColor, 59
    - delete\_texture, 61
    - display, 61
    - FLIP\_HORIZONTAL, 59
    - FLIP\_NONE, 59
    - FLIP\_VERTICAL, 59
    - fill\_texture, 61
    - Flip, 59

- font, 63
- GREEN, 59
- load\_font, 62
- load\_texture, 62
- RED, 59
- Region, 59
- render, 62
- render\_clear, 62
- render\_onto, 63
- render\_rotate, 63
- render\_text, 63
- renderDebugBox, 63
- renderer, 64
- swapFullscreen, 63
- texture\_iter, 59
- TextureID, 59
- textures, 64
- video\_display, 64
- VideoContext, 60
- mediawrap::VideoDisplay, 64
  - ~VideoDisplay, 65
  - get\_renderer, 65
  - swapFullscreen, 65
  - VideoDisplay, 64
  - window, 65
- MovingActor
  - teamusa::MovingActor, 40
- MovingActor.cpp, 84
- MovingActor.h, 84
- NUM\_LAYERS
  - teamusa::VideoEngine, 69
- NewGame
  - teamusa, 11
- Nil
  - teamusa, 11
- onChangeScene
  - teamusa::Engine, 29
- onClick
  - teamusa::BaseActor, 23
  - teamusa::InventoryItemActor, 33
  - teamusa::LevelLink, 38
  - teamusa::MovingActor, 40
  - teamusa::ResponsiveAudioActor, 46
  - teamusa::ResponsiveVideoActor, 48
  - teamusa::SceneLink, 51
  - teamusa::TextboxSpawnActor, 54
  - teamusa::VideoEventActor, 71
- onDisplayText
  - teamusa::Engine, 29
- onExitGame
  - teamusa::Engine, 29
- onHover
  - teamusa::BaseActor, 23
  - teamusa::InventoryItemActor, 33
  - teamusa::LevelLink, 38
  - teamusa::MovingActor, 40
  - teamusa::ResponsiveAudioActor, 46
  - teamusa::ResponsiveVideoActor, 49
  - teamusa::SceneLink, 51
  - teamusa::VideoEventActor, 71
- onLoadGame
  - teamusa::Engine, 29
- onLoadLevel
  - teamusa::Engine, 29
- onNewGame
  - teamusa::Engine, 30
- onPlayAudio
  - teamusa::Engine, 30
- onStreamAudio
  - teamusa::Engine, 30
- operator>>
  - Level.cpp, 82
- parseAudioStreamActor
  - teamusa::Level, 36
- parseDelayedAudioActor
  - teamusa::Level, 36
- parseDelayedVideoActor
  - teamusa::Level, 36
- parseInventoryItemActor
  - teamusa::Level, 36
- parseLevelLink
  - teamusa::Level, 36
- parseMovingActor
  - teamusa::Level, 36
- parseResponsiveAudioActor
  - teamusa::Level, 36
- parseResponsiveVideoActor
  - teamusa::Level, 36
- parseSceneLink
  - teamusa::Level, 36
- parseTextboxSpawnActor
  - teamusa::Level, 36
- parseVideoActor
  - teamusa::Level, 36
- parseVideoEventActor
  - teamusa::Level, 36
- path
  - teamusa::AudioStreamActor, 20
- pause
  - teamusa::Timer, 55
- pickedUp
  - teamusa::InventoryItemActor, 34
- play\_sample
  - mediawrap::AudioPlayer, 18
- PlayAudio
  - teamusa, 11
- playSound
  - teamusa::AudioEngine, 15
- playStream
  - teamusa::AudioEngine, 15
- Player
  - teamusa::Player, 42
- Player.cpp, 84
- Player.h, 84
- Point

- teamusa::Point, 45
- Point.h, 85
- RED
  - mediawrap::VideoContext, 59
- Region
  - mediawrap::VideoContext, 59
  - teamusa, 11
- render
  - mediawrap::VideoContext, 62
  - teamusa::Engine, 30
  - teamusa::VideoEngine, 69
- render\_clear
  - mediawrap::VideoContext, 62
- render\_onto
  - mediawrap::VideoContext, 63
- render\_rotate
  - mediawrap::VideoContext, 63
- render\_text
  - mediawrap::VideoContext, 63
- renderDebugBox
  - mediawrap::VideoContext, 63
  - teamusa::VideoEngine, 69
- renderRotate
  - teamusa::VideoEngine, 69
- renderer
  - mediawrap::VideoContext, 64
- requiredItemID
  - teamusa::LevelLink, 38
  - teamusa::SceneLink, 53
- ResourceGroup
  - ResourceGroup.hpp, 86
- ResourceGroup.hpp, 85
  - CORE\_RESOURCE, 86
  - LEVEL\_RESOURCE, 86
  - ResourceGroup, 86
- ResponsiveAudioActor
  - teamusa::ResponsiveAudioActor, 46
- ResponsiveAudioActor.cpp, 86
- ResponsiveAudioActor.h, 86
- ResponsiveVideoActor
  - teamusa::ResponsiveVideoActor, 47
- ResponsiveVideoActor.cpp, 86
- ResponsiveVideoActor.h, 87
- run
  - teamusa::Engine, 30
- SHADOW\_LAYER
  - teamusa::VideoEngine, 69
- save
  - teamusa::GameSaveSerializer, 32
- saveInThread
  - teamusa::GameSaveSerializer, 32
- sceneID
  - teamusa::LevelLink, 38
  - teamusa::SceneLink, 53
- SceneLink
  - teamusa::SceneLink, 51
- SceneLink.cpp, 87
- SceneLink.h, 87
- scenes
  - teamusa::Level, 36
- setCursor
  - teamusa::Player, 43
- setInventory
  - teamusa::Player, 43
- setPosition
  - teamusa::Player, 43
- setRegion
  - teamusa::BaseActor, 23
- setSlot
  - teamusa::GameSaveSerializer, 32
- setTextureId
  - teamusa::ResponsiveVideoActor, 49
- showTextbox
  - teamusa::VideoEngine, 69
- slot
  - teamusa::GameSaveSerializer, 32
- start
  - teamusa::Timer, 56
- startScene
  - teamusa::Level, 36
- step
  - teamusa::AudioStreamActor, 19
  - teamusa::BaseActor, 23
  - teamusa::DelayedAudioActor, 25
  - teamusa::DelayedVideoActor, 26
  - teamusa::InventoryItemActor, 33
  - teamusa::LevelLink, 38
  - teamusa::MovingActor, 40
  - teamusa::ResponsiveAudioActor, 46
  - teamusa::ResponsiveVideoActor, 49
  - teamusa::SceneLink, 51
  - teamusa::TextboxSpawnActor, 54
  - teamusa::VideoActor, 57
  - teamusa::VideoEventActor, 71
- stop
  - teamusa::Timer, 56
- stream\_audio
  - mediawrap::AudioPlayer, 18
- StreamAudio
  - teamusa, 12
- swapFullscreen
  - mediawrap::VideoContext, 63
  - mediawrap::VideoDisplay, 65
  - teamusa::VideoEngine, 69
- TEXT\_LAYER
  - teamusa::VideoEngine, 70
- teamusa, 10
  - ActorEventType, 11
  - ActorList, 11
  - AudioID, 11
  - BaseActorPtr, 11
  - CURSOR\_DEFAULT, 12
  - CURSOR\_DOWN, 12
  - CURSOR\_LEFT, 12
  - CURSOR\_RIGHT, 12

- CURSOR\_SELECT, 12
- CURSOR\_UP, 12
- ChangeScene, 11
- CursorStyle, 12
- DisplayText, 11
- ExitGame, 12
- LoadGame, 11
- LoadLevel, 11
- NewGame, 11
- Nil, 11
- PlayAudio, 11
- Region, 11
- StreamAudio, 12
- TextureID, 11
- teamusa::ActorEvent, 13
  - ActorEvent, 13
  - type, 13
  - value, 13
- teamusa::ActorVideo, 13
  - ActorVideo, 14
  - layer, 14
  - textureID, 14
- teamusa::AudioEngine, 14
  - audioPlayer, 15
  - coreResources, 15
  - deleteSound, 15
  - deleteSoundGroup, 15
  - levelResources, 15
  - loadSound, 15
  - MAX\_RESERVED\_ID, 15
  - playSound, 15
  - playStream, 15
- teamusa::AudioStreamActor, 18
  - ~AudioStreamActor, 19
  - activated, 20
  - AudioStreamActor, 19
  - getPath, 19
  - path, 20
  - step, 19
- teamusa::BaseActor, 20
  - ~BaseActor, 22
  - BaseActor, 22
  - getLayer, 22
  - getRegion, 22
  - getTextureID, 22
  - hasVideo, 22
  - isInBounds, 22
  - mAudioID, 24
  - mRegion, 24
  - mVideo, 24
  - onClick, 23
  - onHover, 23
  - setRegion, 23
  - step, 23
- teamusa::DelayedAudioActor, 24
  - ~DelayedAudioActor, 25
  - audioid, 25
  - currentStep, 25
  - delaySteps, 25
  - DelayedAudioActor, 25
  - step, 25
- teamusa::DelayedVideoActor, 25
  - ~DelayedVideoActor, 26
  - currentStep, 27
  - delaySteps, 27
  - DelayedVideoActor, 26
  - disappear, 27
  - step, 26
  - textureId, 27
- teamusa::Engine, 27
  - ~Engine, 28
  - ActorEventHandler, 28
  - Engine, 28
  - freeAndLoadLevel, 28
  - getMouseClickState, 28
  - getMouseCoordinates, 28
  - handleEvent, 29
  - mActorEventHandlers, 30
  - mAudioEngine, 30
  - mCurrentLevelID, 30
  - mIsRunning, 30
  - mLevel, 30
  - mMainMenu, 30
  - mPlayer, 31
  - mSerializer, 31
  - mVideoEngine, 31
  - onChangeScene, 29
  - onDisplayText, 29
  - onExitGame, 29
  - onLoadGame, 29
  - onLoadLevel, 29
  - onNewGame, 30
  - onPlayAudio, 30
  - onStreamAudio, 30
  - render, 30
  - run, 30
- teamusa::GameSaveSerializer, 31
  - ~GameSaveSerializer, 31
  - fileLock, 32
  - GameSaveSerializer, 31
  - load, 31
  - save, 32
  - saveInThread, 32
  - setSlot, 32
  - slot, 32
- teamusa::InventoryItemActor, 32
  - ~InventoryItemActor, 33
  - InventoryItemActor, 33
  - itemId, 34
  - onClick, 33
  - onHover, 33
  - pickedUp, 34
  - step, 33
- teamusa::Level, 34
  - activeScene, 36
  - changeScene, 35

- clearAll, 35
- getActors, 35
- getBGImageID, 35
- getScene, 36
- Level, 35
- loadLevel, 36
- parseAudioStreamActor, 36
- parseDelayedAudioActor, 36
- parseDelayedVideoActor, 36
- parseInventoryItemActor, 36
- parseLevelLink, 36
- parseMovingActor, 36
- parseResponsiveAudioActor, 36
- parseResponsiveVideoActor, 36
- parseSceneLink, 36
- parseTextboxSpawnActor, 36
- parseVideoActor, 36
- parseVideoEventActor, 36
- scenes, 36
- startScene, 36
- teamusa::Level::Scene, 50
  - actors, 50
  - bgImageID, 50
- teamusa::LevelLink, 37
  - ~LevelLink, 37
  - getSceneID, 37
  - getText, 37
  - itemRequiredText, 38
  - levelID, 38
  - LevelLink, 37
  - onClick, 38
  - onHover, 38
  - requiredItemID, 38
  - sceneID, 38
  - step, 38
- teamusa::MovingActor, 39
  - ~MovingActor, 40
  - currentStep, 40
  - endRegion, 40
  - hGrowth, 40
  - isActive, 41
  - MovingActor, 40
  - onClick, 40
  - onHover, 40
  - step, 40
  - transitionSteps, 41
  - wGrowth, 41
  - xSpeed, 41
  - ySpeed, 41
- teamusa::Player, 41
  - ~Player, 42
  - addItem, 42
  - CURSOR\_DEFAULT\_ID, 44
  - CURSOR\_DOWN\_ID, 44
  - CURSOR\_LEFT\_ID, 44
  - CURSOR\_RIGHT\_ID, 44
  - CURSOR\_SELECT\_ID, 44
  - CURSOR\_UP\_ID, 44
  - FLASHLIGHT\_ID, 44
  - getCursorTextureID, 42
  - getInventory, 42
  - getPosition, 42
  - hasItem, 43
  - Inventory, 42
  - mCursorStyle, 44
  - mInventory, 44
  - mLayer, 44
  - MOUSE\_CLICK\_ID, 44
  - mPosition, 44
  - mRegion, 44
  - mTextureID, 44
  - Player, 42
  - setCursor, 43
  - setInventory, 43
  - setPosition, 43
- teamusa::Point, 44
  - Point, 45
  - x, 45
  - y, 45
- teamusa::ResponsiveAudioActor, 45
  - ~ResponsiveAudioActor, 46
  - clickAudioID, 47
  - hoverAudioID, 47
  - onClick, 46
  - onHover, 46
  - ResponsiveAudioActor, 46
  - step, 46
- teamusa::ResponsiveVideoActor, 47
  - ~ResponsiveVideoActor, 47
  - clickTexture, 49
  - defaultTextureID, 49
  - hoverTexture, 49
  - onClick, 48
  - onHover, 49
  - ResponsiveVideoActor, 47
  - setTextureID, 49
  - step, 49
- teamusa::SceneLink, 50
  - ~SceneLink, 51
  - cursorStyle, 53
  - getText, 51
  - itemRequiredText, 53
  - onClick, 51
  - onHover, 51
  - requiredItemID, 53
  - sceneID, 53
  - SceneLink, 51
  - step, 51
- teamusa::TextboxSpawnActor, 53
  - ~TextboxSpawnActor, 54
  - activated, 54
  - getText, 54
  - onClick, 54
  - step, 54
  - text, 54
  - TextboxSpawnActor, 54

- teamusa::Timer, 55
  - ~Timer, 55
  - getTicks, 55
  - mPauseTicks, 56
  - mPaused, 56
  - mStartTicks, 56
  - mStarted, 56
  - pause, 55
  - start, 56
  - stop, 56
  - Timer, 55
  - unpause, 56
- teamusa::VideoActor, 56
  - ~VideoActor, 57
  - step, 57
  - VideoActor, 57
- teamusa::VideoEngine, 65
  - ~VideoEngine, 67
  - clearLayers, 67
  - coreResources, 69
  - deleteResourceGroup, 67
  - deleteTexture, 67
  - display, 67
  - hideTextbox, 67
  - isShowingTextbox, 67
  - layers, 69
  - levelResources, 69
  - loadTexture, 67
  - MAX\_RESERVED\_ID, 69
  - NUM\_LAYERS, 69
  - render, 69
  - renderDebugBox, 69
  - renderRotate, 69
  - SHADOW\_LAYER, 69
  - showTextbox, 69
  - swapFullscreen, 69
  - TEXT\_LAYER, 70
  - textboxActive, 70
  - textboxPadding, 70
  - textboxRegion, 70
  - videoContext, 70
  - VideoEngine, 66
- teamusa::VideoEventActor, 70
  - ~VideoEventActor, 71
  - actorEvent, 71
  - onClick, 71
  - onHover, 71
  - step, 71
  - VideoEventActor, 71
- text
  - teamusa::TextboxSpawnActor, 54
- textboxActive
  - teamusa::VideoEngine, 70
- textboxPadding
  - teamusa::VideoEngine, 70
- textboxRegion
  - teamusa::VideoEngine, 70
- TextboxSpawnActor
  - teamusa::TextboxSpawnActor, 54
  - TextboxSpawnActor.cpp, 88
  - TextboxSpawnActor.h, 88
- texture\_iter
  - mediawrap::VideoContext, 59
- TextureID
  - mediawrap::VideoContext, 59
  - teamusa, 11
- textureID
  - teamusa::ActorVideo, 14
- textureId
  - teamusa::DelayedVideoActor, 27
- textures
  - mediawrap::VideoContext, 64
- Timer
  - teamusa::Timer, 55
- Timer.cpp, 88
- Timer.h, 89
- transitionSteps
  - teamusa::MovingActor, 41
- type
  - teamusa::ActorEvent, 13
- unpause
  - teamusa::Timer, 56
- value
  - teamusa::ActorEvent, 13
- video\_display
  - mediawrap::VideoContext, 64
- VideoActor
  - teamusa::VideoActor, 57
- VideoActor.cpp, 89
- VideoActor.h, 89
- VideoContext
  - mediawrap::VideoContext, 60
- videoContext
  - teamusa::VideoEngine, 70
- VideoContext.cpp, 90
- VideoContext.hpp, 90
- VideoDisplay
  - mediawrap::VideoDisplay, 64
- VideoDisplay.cpp, 90
- VideoDisplay.hpp, 90
- VideoEngine
  - teamusa::VideoEngine, 66
- VideoEngine.cpp, 91
- VideoEngine.hpp, 91
- VideoEventActor
  - teamusa::VideoEventActor, 71
- VideoEventActor.cpp, 91
- VideoEventActor.h, 92
- wGrowth
  - teamusa::MovingActor, 41
- window
  - mediawrap::VideoDisplay, 65
- x

teamusa::Point, [45](#)  
xSpeed  
teamusa::MovingActor, [41](#)  
  
y  
teamusa::Point, [45](#)  
ySpeed  
teamusa::MovingActor, [41](#)