

## My Project

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Labyrinth</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	Background Class Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Constructor & Destructor Documentation . . . . .	5
3.1.2.1	Background() [1/2] . . . . .	5
3.1.2.2	Background() [2/2] . . . . .	6
3.1.2.3	~Background() . . . . .	6
3.1.3	Member Function Documentation . . . . .	6
3.1.3.1	add_layer() . . . . .	6
3.1.3.2	blank() . . . . .	6
3.1.3.3	draw() . . . . .	6
3.1.3.4	left() . . . . .	7
3.1.3.5	reset() [1/2] . . . . .	7
3.1.3.6	reset() [2/2] . . . . .	7
3.1.3.7	right() . . . . .	7
3.2	Character Class Reference . . . . .	7
3.2.1	Detailed Description . . . . .	8
3.2.2	Constructor & Destructor Documentation . . . . .	8
3.2.2.1	Character() . . . . .	8

3.2.2.2	<code>~Character()</code>	9
3.2.3	Member Function Documentation	9
3.2.3.1	<code>equals()</code>	9
3.2.3.2	<code>follow()</code>	9
3.2.3.3	<code>gasp()</code>	9
3.2.3.4	<code>get_name()</code>	9
3.2.3.5	<code>get_position()</code>	10
3.2.3.6	<code>get_screen_position()</code>	10
3.2.3.7	<code>happy()</code>	10
3.2.3.8	<code>is_recruited()</code>	10
3.2.3.9	<code>jump()</code> [1/2]	10
3.2.3.10	<code>jump()</code> [2/2]	10
3.2.3.11	<code>reset()</code> [1/2]	10
3.2.3.12	<code>reset()</code> [2/2]	11
3.2.3.13	<code>set_position()</code>	11
3.2.3.14	<code>set_recruited()</code>	11
3.2.3.15	<code>set_screen_position()</code>	11
3.2.3.16	<code>set_stage_pos()</code>	11
3.2.3.17	<code>stand()</code>	11
3.2.3.18	<code>update_pos()</code>	11
3.2.3.19	<code>walk_left()</code>	11
3.2.3.20	<code>walk_right()</code>	12
3.3	Conversation Class Reference	12
3.3.1	Detailed Description	12
3.3.2	Constructor & Destructor Documentation	13
3.3.2.1	<code>Conversation()</code> [1/2]	13
3.3.2.2	<code>Conversation()</code> [2/2]	13
3.3.3	Member Function Documentation	13
3.3.3.1	<code>add_dialogue()</code>	13
3.3.3.2	<code>get_angry_response()</code>	13

3.3.3.3	<a href="#">get_dialogue()</a>	13
3.3.3.4	<a href="#">get_length()</a>	14
3.3.3.5	<a href="#">set_angry_response()</a>	14
3.4	<a href="#">Game Class Reference</a>	14
3.4.1	<a href="#">Detailed Description</a>	14
3.4.2	<a href="#">Constructor &amp; Destructor Documentation</a>	14
3.4.2.1	<a href="#">Game()</a>	15
3.4.2.2	<a href="#">~Game()</a>	15
3.4.3	<a href="#">Member Function Documentation</a>	15
3.4.3.1	<a href="#">add_scene()</a>	15
3.4.3.2	<a href="#">join_scenes()</a>	15
3.4.3.3	<a href="#">play()</a>	15
3.4.3.4	<a href="#">set_introduction()</a>	16
3.5	<a href="#">Interaction Class Reference</a>	16
3.5.1	<a href="#">Detailed Description</a>	16
3.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	16
3.5.2.1	<a href="#">Interaction()</a>	16
3.5.3	<a href="#">Member Function Documentation</a>	17
3.5.3.1	<a href="#">get_interaction_length()</a>	17
3.5.3.2	<a href="#">get_message()</a>	17
3.5.3.3	<a href="#">get_scene_position()</a>	17
3.6	<a href="#">Panel Class Reference</a>	18
3.6.1	<a href="#">Detailed Description</a>	18
3.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	18
3.6.2.1	<a href="#">Panel()</a>	18
3.6.3	<a href="#">Member Function Documentation</a>	18
3.6.3.1	<a href="#">display()</a>	18
3.7	<a href="#">Report Struct Reference</a>	19
3.8	<a href="#">Scene Class Reference</a>	19
3.8.1	<a href="#">Detailed Description</a>	19

3.8.2	Constructor & Destructor Documentation	20
3.8.2.1	Scene()	20
3.8.2.2	~Scene()	20
3.8.3	Member Function Documentation	20
3.8.3.1	add_conversation()	20
3.8.3.2	add_follower()	20
3.8.3.3	add_interaction()	21
3.8.3.4	enter()	21
3.8.3.5	get_previous_junction_position()	21
3.8.3.6	interact()	21
3.8.3.7	marionette()	21
3.8.3.8	movement_key_pressed()	21
3.8.3.9	play()	21
3.8.3.10	prompt_interact()	22
3.8.3.11	reset()	22
3.8.3.12	set_junction()	22
3.8.3.13	set_previous_junction_position()	22
3.8.3.14	stage_junction()	22
3.8.3.15	stage_left()	23
3.8.3.16	stage_left_barrier()	23
3.8.3.17	stage_right()	23
3.8.3.18	stage_right_barrier()	23
3.9	SceneJunction Class Reference	23
3.9.1	Detailed Description	24
3.9.2	Constructor & Destructor Documentation	24
3.9.2.1	SceneJunction() [1/2]	24
3.9.2.2	SceneJunction() [2/2]	24
3.9.3	Member Function Documentation	24
3.9.3.1	contains()	25
3.9.3.2	get_next_character_position()	25

3.9.3.3	<a href="#">get_next_scene()</a>	25
3.9.3.4	<a href="#">get_next_track()</a>	25
3.9.3.5	<a href="#">link_scenes()</a>	26
3.10	<a href="#">Script Class Reference</a>	26
3.10.1	<a href="#">Detailed Description</a>	26
3.10.2	<a href="#">Constructor &amp; Destructor Documentation</a>	27
3.10.2.1	<a href="#">Script()</a>	27
3.10.3	<a href="#">Member Function Documentation</a>	27
3.10.3.1	<a href="#">insert_conversation()</a>	27
3.10.3.2	<a href="#">insert_recruit_conversation()</a>	27
3.10.3.3	<a href="#">speak_to()</a>	28
3.11	<a href="#">Sprite Class Reference</a>	28
3.11.1	<a href="#">Detailed Description</a>	28
3.11.2	<a href="#">Constructor &amp; Destructor Documentation</a>	29
3.11.2.1	<a href="#">Sprite()</a>	29
3.11.2.2	<a href="#">~Sprite()</a>	29
3.11.3	<a href="#">Member Function Documentation</a>	29
3.11.3.1	<a href="#">draw()</a> [1/2]	29
3.11.3.2	<a href="#">draw()</a> [2/2]	29
3.11.3.3	<a href="#">flip_draw()</a> [1/2]	30
3.11.3.4	<a href="#">flip_draw()</a> [2/2]	30
3.11.3.5	<a href="#">get_height()</a>	30
3.11.3.6	<a href="#">get_width()</a>	30
3.11.3.7	<a href="#">get_x()</a>	31
3.11.3.8	<a href="#">get_y()</a>	31
3.11.3.9	<a href="#">reset_position()</a> [1/2]	31
3.11.3.10	<a href="#">reset_position()</a> [2/2]	31
3.11.3.11	<a href="#">set_destination()</a>	31
3.11.3.12	<a href="#">set_position()</a> [1/2]	31
3.11.3.13	<a href="#">set_position()</a> [2/2]	32

3.11.3.14 <code>set_source()</code> . . . . .	32
3.12 SpriteLayer Class Reference . . . . .	32
3.12.1 Detailed Description . . . . .	33
3.12.2 Constructor & Destructor Documentation . . . . .	33
3.12.2.1 <code>SpriteLayer()</code> . . . . .	33
3.12.2.2 <code>~SpriteLayer()</code> . . . . .	33
3.12.3 Member Function Documentation . . . . .	34
3.12.3.1 <code>draw()</code> . . . . .	34
3.12.3.2 <code>left()</code> . . . . .	34
3.12.3.3 <code>reset()</code> [1/2] . . . . .	34
3.12.3.4 <code>reset()</code> [2/2] . . . . .	34
3.12.3.5 <code>right()</code> . . . . .	35
3.13 Text_box Class Reference . . . . .	35
3.13.1 Detailed Description . . . . .	35
3.13.2 Constructor & Destructor Documentation . . . . .	35
3.13.2.1 <code>Text_box()</code> . . . . .	35
3.13.3 Member Function Documentation . . . . .	36
3.13.3.1 <code>display()</code> . . . . .	36
<b>Index</b>	<b>37</b>



## Chapter 1

# Labyrinth

A rudimentary video game. All sprites and backgrounds courtesy of Hugh Keene.



## Chapter 2

# Class Index

### 2.1 Class List

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

Background	5
Character	7
Conversation	12
Game	14
Interaction	16
Panel	18
Report	19
Scene	19
SceneJunction	23
Script	26
Sprite	28
SpriteLayer	32
Text_box	35



## Chapter 3

# Class Documentation

### 3.1 Background Class Reference

```
#include <background.h>
```

#### Public Member Functions

- [Background](#) (vector< [SpriteLayer](#) \* > layers, SDL\_Renderer \*ren)
- [Background](#) (SDL\_Renderer \*ren)
- [~Background](#) ()
- void [add\\_layer](#) ([SpriteLayer](#) \*layer)
- void [left](#) (uint speed)
- void [right](#) (uint speed)
- void [reset](#) ()
- void [reset](#) (int offset)
- void [draw](#) ()
- void [blank](#) ()

#### 3.1.1 Detailed Description

A background class

##### Author

Jared Allen

##### Version

8 February 2019

#### 3.1.2 Constructor & Destructor Documentation

##### 3.1.2.1 [Background\(\)](#) [1/2]

```
Background::Background (
    vector< SpriteLayer * > layers,
    SDL_Renderer * ren )
```

##### Constructor

**Parameters**

<i>layers</i>	the sprites that layer the background
---------------	---------------------------------------

**3.1.2.2 Background()** [2/2]

```
Background::Background (
    SDL_Renderer * ren )
```

**Constructor****3.1.2.3 ~Background()**

```
Background::~~Background ( )
```

**Destructor****3.1.3 Member Function Documentation****3.1.3.1 add\_layer()**

```
void Background::add_layer (
    SpriteLayer * layer )
```

add a layer to background

**Parameters**

<i>layer</i>	
--------------	--

**3.1.3.2 blank()**

```
void Background::blank ( )
```

draw a blank background

**3.1.3.3 draw()**

```
void Background::draw ( )
```

draw the background

## Parameters

<i>renderer</i>	the renderer
-----------------	--------------

## 3.1.3.4 left()

```
void Background::left (
    uint speed )
```

move background left

## 3.1.3.5 reset() [1/2]

```
void Background::reset ( )
```

reset the background

## 3.1.3.6 reset() [2/2]

```
void Background::reset (
    int offset )
```

reset the background with offset

## 3.1.3.7 right()

```
void Background::right (
    uint speed )
```

move background right

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

- background.h
- background.cpp

## 3.2 Character Class Reference

```
#include <character.h>
```

## Public Member Functions

- [Character](#) (string name, [Sprite](#) \*full\_body, [Sprite](#) \*torso, uint force, uint diversion, uint stealth, uint num\_walking\_sprites, uint num\_talking\_sprites)
- [~Character](#) ()
- void [happy](#) ()
- void [gasp](#) ()
- void [walk\\_left](#) (uint speed, uint count)
- void [walk\\_right](#) (uint speed, uint count)
- void [jump](#) (int y\_velocity)
- void [jump](#) (int x\_velocity, int y\_velocity)
- void [stand](#) ()
- void [set\\_position](#) (int x, int y)
- void [set\\_screen\\_position](#) (int x, int y)
- vector< int > [get\\_screen\\_position](#) ()
- vector< int > [get\\_position](#) ()
- void [follow](#) ([Character](#) \*leader, uint speed, uint count)
- void [update\\_pos](#) (bool left, uint speed)
- void [reset](#) ()
- void [reset](#) (uint offset)
- bool [equals](#) ([Character](#) \*character)
- string [get\\_name](#) ()
- void [set\\_stage\\_pos](#) (int screen\_pos, int pos)
- void [set\\_recruited](#) ()
- bool [is\\_recruited](#) ()

### 3.2.1 Detailed Description

A character class

#### Author

Jared Allen

#### Version

9 February 2019

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 [Character\(\)](#)

```
Character::Character (
    string name,
    Sprite * full_body,
    Sprite * torso,
    uint force,
    uint diversion,
    uint stealth,
    uint num_walking_sprites,
    uint num_talking_sprites )
```

#### Constructor



## Parameters

<i>full_body</i>	the full body sprite
<i>torso</i>	the torso sprite

**3.2.2.2 ~Character()**

```
Character::~~Character ( )
```

Destructor

**3.2.3 Member Function Documentation****3.2.3.1 equals()**

```
bool Character::equals (
    Character * character )
```

determine if two characters are equal

**3.2.3.2 follow()**

```
void Character::follow (
    Character * leader,
    uint speed,
    uint count )
```

make NPCs follow the character

**3.2.3.3 gasp()**

```
void Character::gasp ( )
```

set character gasp expression

**3.2.3.4 get\_name()**

```
string Character::get_name ( )
```

gets the name of the character

**3.2.3.5 get\_position()**

```
vector< int > Character::get_position ( )
```

get the characters coordinates

**3.2.3.6 get\_screen\_position()**

```
vector< int > Character::get_screen_position ( )
```

get the characters coordinates

**3.2.3.7 happy()**

```
void Character::happy ( )
```

set character happy expression

**3.2.3.8 is\_recruited()**

```
bool Character::is_recruited ( )
```

determine if character has been recruited

**3.2.3.9 jump()** [1/2]

```
void Character::jump (
    int y_velocity )
```

make the character jump

**3.2.3.10 jump()** [2/2]

```
void Character::jump (
    int x_velocity,
    int y_velocity )
```

make the character jump laterally

**3.2.3.11 reset()** [1/2]

```
void Character::reset ( )
```

reset characters position

**3.2.3.12 reset()** [2/2]

```
void Character::reset (
    uint offset )
```

reset characters position with offset

**3.2.3.13 set\_position()**

```
void Character::set_position (
    int x,
    int y )
```

set the characters coordinates

**3.2.3.14 set\_recruited()**

```
void Character::set_recruited ( )
```

recruit a character

**3.2.3.15 set\_screen\_position()**

```
void Character::set_screen_position (
    int x,
    int y )
```

set the characters screen coordinates

**3.2.3.16 set\_stage\_pos()**

```
void Character::set_stage_pos (
    int screen_pos,
    int pos )
```

set stage position

**3.2.3.17 stand()**

```
void Character::stand ( )
```

make the character stand

**3.2.3.18 update\_pos()**

```
void Character::update_pos (
    bool left,
    uint speed )
```

update characters position

**3.2.3.19 walk\_left()**

```
void Character::walk_left (
    uint speed,
    uint count )
```

make the character walk left

**Parameters**

<i>speed</i>	the speed that the character walks
<i>count</i>	the characters cycle count

**3.2.3.20 walk\_right()**

```
void Character::walk_right (
    uint speed,
    uint count )
```

make the character walk right

**Parameters**

<i>speed</i>	the speed that the character walks
<i>count</i>	the characters walking cycle count

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

- character.h
- character.cpp

## 3.3 Conversation Class Reference

```
#include <conversation.h>
```

**Public Member Functions**

- [Conversation](#) (string filepath, SDL\_Renderer \*renderer)
- [Conversation](#) ()
- string [get\\_dialogue](#) (uint index)
- void [add\\_dialogue](#) (char \*words, SDL\_Renderer \*renderer)
- uint [get\\_length](#) ()
- void [set\\_angry\\_response](#) (char \*words, SDL\_Renderer \*renderer)
- string [get\\_angry\\_response](#) ()

**3.3.1 Detailed Description**

A conversation class

**Author**

Jared Allen

**Version**

30 March 2019

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 Conversation() [1/2]

```
Conversation::Conversation (
    string filepath,
    SDL_Renderer * renderer )
```

Constructor

#### 3.3.2.2 Conversation() [2/2]

```
Conversation::Conversation ( )
```

default constructor

### 3.3.3 Member Function Documentation

#### 3.3.3.1 add\_dialogue()

```
void Conversation::add_dialogue (
    char * words,
    SDL_Renderer * renderer )
```

add dialogue to conversation

#### 3.3.3.2 get\_angry\_response()

```
string Conversation::get_angry_response ( )
```

get angry response

#### 3.3.3.3 get\_dialogue()

```
string Conversation::get_dialogue (
    uint index )
```

get dialogue at the index provided

#### 3.3.3.4 get\_length()

```
uint Conversation::get_length ( )
```

get length of the dialogue vector

#### 3.3.3.5 set\_angry\_response()

```
void Conversation::set_angry_response (
    char * words,
    SDL_Renderer * renderer )
```

set an angry response

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

- conversation.h
- conversation.cpp

## 3.4 Game Class Reference

```
#include <game.h>
```

### Public Member Functions

- [Game](#) ()
- [~Game](#) ()
- void [play](#) ()
- void [set\\_introduction](#) ([Panel](#) \*panel)
- void [add\\_scene](#) ([Scene](#) \*scene, uint scene\_track)
- void [join\\_scenes](#) (uint track1\_index, uint track2\_index, uint scene1\_pos, uint scene2\_pos, int scene1\_↔ junction\_pos, int scene2\_junction\_pos)

#### 3.4.1 Detailed Description

A game class

##### Author

Jared Allen

##### Version

8 February 2019

#### 3.4.2 Constructor & Destructor Documentation

### 3.4.2.1 Game()

```
Game::Game ( )
```

Constructor

### 3.4.2.2 ~Game()

```
Game::~~Game ( )
```

Destructor

## 3.4.3 Member Function Documentation

### 3.4.3.1 add\_scene()

```
void Game::add_scene (
    Scene * scene,
    uint scene_track )
```

add scene to game

### 3.4.3.2 join\_scenes()

```
void Game::join_scenes (
    uint track1_index,
    uint track2_index,
    uint scene1_pos,
    uint scene2_pos,
    int scene1_junction_pos,
    int scene2_junction_pos )
```

join two scenes

#### Parameters

<i>scene1</i>	the first scene
<i>scene2</i>	the second scene

### 3.4.3.3 play()

```
void Game::play ( )
```

plays the game

#### 3.4.3.4 set\_introduction()

```
void Game::set_introduction (
    Panel * panel )
```

sets the introduction panel of the game

##### Parameters

<i>panel</i>	the introduction panel
--------------	------------------------

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

- game.h
- game.cpp

## 3.5 Interaction Class Reference

```
#include <interaction.h>
```

### Public Member Functions

- [Interaction](#) (vector< string > messages, int param\_scene\_position, SDL\_Renderer \*renderer)
- int [get\\_scene\\_position](#) ()
- uint [get\\_interaction\\_length](#) ()
- string [get\\_message](#) (uint index)

#### 3.5.1 Detailed Description

An interaction class Jared Allen

##### Version

20 July 2019

#### 3.5.2 Constructor & Destructor Documentation

##### 3.5.2.1 Interaction()

```
Interaction::Interaction (
    vector< string > messages,
    int param_scene_position,
    SDL_Renderer * renderer )
```

##### Constructor



### 3.5.3 Member Function Documentation

#### 3.5.3.1 `get_interaction_length()`

```
uint Interaction::get_interaction_length ( )
```

get the length of the interaction

##### Returns

the length of the interaction

#### 3.5.3.2 `get_message()`

```
string Interaction::get_message (
    uint index )
```

get the message at the given position

##### Parameters

<i>index</i>	the index of the message
--------------	--------------------------

##### Returns

the message

#### 3.5.3.3 `get_scene_position()`

```
int Interaction::get_scene_position ( )
```

get the scene position

##### Returns

the scene position

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

- `interaction.h`
- `interaction.cpp`

## 3.6 Panel Class Reference

```
#include <panel.h>
```

### Public Member Functions

- [Panel](#) ([Background](#) \*background, SDL\_Renderer \*renderer)
- bool [display](#) ()

### 3.6.1 Detailed Description

A panel class

#### Author

Jared Allen

#### Version

5 July 2019

### 3.6.2 Constructor & Destructor Documentation

#### 3.6.2.1 Panel()

```
Panel::Panel (
    Background * background,
    SDL_Renderer * renderer )
```

Constructor

### 3.6.3 Member Function Documentation

#### 3.6.3.1 display()

```
bool Panel::display ( )
```

displays the panel

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

- panel.h
- panel.cpp

## 3.7 Report Struct Reference

### Public Attributes

- Scene\_States **status**
- int **character\_position**

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

- scene.h

## 3.8 Scene Class Reference

```
#include <scene.h>
```

### Public Member Functions

- void [add\\_interaction](#) (vector< string > messages, int scene\_position, SDL\_Renderer \*renderer)
- void [prompt\\_interact](#) ()
- void [interact](#) ()
- void [stage\\_junction](#) (int junction\_position)
- [Report play](#) ()
- bool [movement\\_key\\_pressed](#) (const Uint8 \*state)
- void [marionette](#) (int x\_velocity, int y\_velocity, uint cadence)
- void [set\\_previous\\_junction\\_position](#) (int position)
- int [get\\_previous\\_junction\\_position](#) ()
- void [reset](#) ()
- void [add\\_follower](#) (Character \*character)
- void [stage\\_left](#) ()
- void [stage\\_left\\_barrier](#) ()
- void [stage\\_right](#) ()
- void [stage\\_right\\_barrier](#) ()
- void [add\\_conversation](#) (Conversation \*conversation)
- [Scene](#) (SDL\_Renderer \*renderer, [Background](#) \*background, vector< [Character](#) \* > characters, [Character](#) \*main\_character, [Script](#) \*scene\_dialogue, uint speed, uint maximum\_stage\_displacement)
- [~Scene](#) ()
- void [set\\_junction](#) (int position)
- bool [enter](#) ()

### 3.8.1 Detailed Description

A scene class

#### Author

Jared Allen

#### Version

8 February 2019

## 3.8.2 Constructor & Destructor Documentation

### 3.8.2.1 Scene()

```
Scene::Scene (
    SDL_Renderer * renderer,
    Background * background,
    vector< Character * > characters,
    Character * main_character,
    Script * scene_dialogue,
    uint speed,
    uint maximum_stage_displacement )
```

#### Constructor

##### Parameters

<i>renderer</i>	the renderer
<i>background</i>	the background
<i>characters</i>	the characters
<i>main_character</i>	the main character
<i>speed</i>	the speed

### 3.8.2.2 ~Scene()

```
Scene::~Scene ( )
```

#### Destructor

## 3.8.3 Member Function Documentation

### 3.8.3.1 add\_conversation()

```
void Scene::add_conversation (
    Conversation * conversation )
```

adds conversation to scene

### 3.8.3.2 add\_follower()

```
void Scene::add_follower (
    Character * character )
```

adds a following character

### 3.8.3.3 add\_interaction()

```
void Scene::add_interaction (
    vector< string > messages,
    int scene_position,
    SDL_Renderer * renderer )
```

add an interaction

### 3.8.3.4 enter()

```
bool Scene::enter ( )
```

determine entry to linked scene

### 3.8.3.5 get\_previous\_junction\_position()

```
int Scene::get_previous_junction_position ( )
```

gets the previous scene junction position

### 3.8.3.6 interact()

```
void Scene::interact ( )
```

perform interaction if within proximity

### 3.8.3.7 marionette()

```
void Scene::marionette (
    int x_velocity,
    int y_velocity,
    uint cadence )
```

moves the main character

### 3.8.3.8 movement\_key\_pressed()

```
bool Scene::movement_key_pressed (
    const Uint8 * state )
```

determines whether a movement key is being held down

### 3.8.3.9 play()

```
Report Scene::play ( )
```

plays the scene

### 3.8.3.10 prompt\_interact()

```
void Scene::prompt_interact ( )
```

prompt interaction

### 3.8.3.11 reset()

```
void Scene::reset ( )
```

resets the scene

### 3.8.3.12 set\_junction()

```
void Scene::set_junction (
    int position )
```

set scene junction

#### Parameters

<i>pos</i>	the junction position
------------	-----------------------

### 3.8.3.13 set\_previous\_junction\_position()

```
void Scene::set_previous_junction_position (
    int position )
```

sets previous scene junction position

### 3.8.3.14 stage\_junction()

```
void Scene::stage_junction (
    int junction_position )
```

Set the stage to a position based on the junction position

#### Parameters

<i>junction</i>	position the position to set the stage to
-----------------	-------------------------------------------

**3.8.3.15 stage\_left()**

```
void Scene::stage_left ( )
```

sets the scene to the left

**3.8.3.16 stage\_left\_barrier()**

```
void Scene::stage_left_barrier ( )
```

sets the scene to the left barrier

**3.8.3.17 stage\_right()**

```
void Scene::stage_right ( )
```

sets the scene to the right

**3.8.3.18 stage\_right\_barrier()**

```
void Scene::stage_right_barrier ( )
```

sets the scene to the right barrier

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

- scene.h
- scene.cpp

## 3.9 SceneJunction Class Reference

```
#include <sceneJunction.h>
```

### Public Member Functions

- [SceneJunction](#) (uint track1\_index, uint track2\_index, uint scene1\_index, uint scene2\_index, int scene1\_link↔\_position, int scene2\_link\_position)
- [SceneJunction](#) ()
- void [link\\_scenes](#) (uint track1\_index, uint track2\_index, uint scene1\_index, uint scene2\_index, int scene1↔\_link\_position, int scene2\_link\_position)
- uint [get\\_next\\_scene](#) (uint scene, uint track)
- uint [get\\_next\\_track](#) (uint scene, uint track)
- int [get\\_next\\_character\\_position](#) (uint scene, uint track)
- bool [contains](#) (uint scene\_pos, uint track\_pos, int link\_position)

### 3.9.1 Detailed Description

A scene junction class

#### Author

Jared Allen

#### Version

17 May 2019

### 3.9.2 Constructor & Destructor Documentation

#### 3.9.2.1 SceneJunction() [1/2]

```
SceneJunction::SceneJunction (
    uint track1_index,
    uint track2_index,
    uint scene1_index,
    uint scene2_index,
    int scene1_link_position,
    int scene2_link_position )
```

#### Constructor

##### Parameters

<i>position</i>	the position of the junction
-----------------	------------------------------

#### 3.9.2.2 SceneJunction() [2/2]

```
SceneJunction::SceneJunction ( )
```

default constructor

### 3.9.3 Member Function Documentation



### 3.9.3.1 contains()

```
bool SceneJunction::contains (
    uint scene_pos,
    uint track_pos,
    int link_position )
```

determine whether sceneJunction contains a scene

### 3.9.3.2 get\_next\_character\_position()

```
int SceneJunction::get_next_character_position (
    uint scene,
    uint track )
```

get next character position updates the character position for the next scene

#### Parameters

<i>scene</i>	the scene of the current track
<i>track</i>	the track of the current scene

### 3.9.3.3 get\_next\_scene()

```
uint SceneJunction::get_next_scene (
    uint scene,
    uint track )
```

get next scene

#### Parameters

<i>scene</i>	the scene of the current track
<i>track</i>	the track of the current scene

### 3.9.3.4 get\_next\_track()

```
uint SceneJunction::get_next_track (
    uint scene,
    uint track )
```

get next track

## Parameters

<i>scene</i>	the scene of the current track
<i>track</i>	the track of the current scene

## 3.9.3.5 link\_scenes()

```
void SceneJunction::link_scenes (
    uint track1_index,
    uint track2_index,
    uint scene1_index,
    uint scene2_index,
    int scene1_link_position,
    int scene2_link_position )
```

set the junction position

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

- sceneJunction.h
- sceneJunction.cpp

## 3.10 Script Class Reference

```
#include <script.h>
```

## Public Member Functions

- [Script](#) (vector< [Character](#) \* > characters)
- [Conversation](#) \* [speak\\_to](#) ([Character](#) \*character)
- void [insert\\_conversation](#) ([Character](#) \*character, [Conversation](#) \*conversation)
- void [insert\\_recruit\\_conversation](#) ([Character](#) \*character, [Conversation](#) \*conversation)

## 3.10.1 Detailed Description

A script class

## Author

Jared Allen

## Version

8 February 2019

## 3.10.2 Constructor & Destructor Documentation

### 3.10.2.1 Script()

```
Script::Script (
    vector< Character * > characters )
```

Constructor

## 3.10.3 Member Function Documentation

### 3.10.3.1 insert\_conversation()

```
void Script::insert_conversation (
    Character * character,
    Conversation * conversation )
```

insert new conversation into a character's dialogue

Parameters

<i>character</i>	the character pointer
<i>conversation</i>	the conversation pointer

### 3.10.3.2 insert\_recruit\_conversation()

```
void Script::insert_recruit_conversation (
    Character * character,
    Conversation * conversation )
```

insert recruitment conversation into a character's dialogue

Parameters

<i>character</i>	the character pointer
<i>conversation</i>	the conversation pointer

### 3.10.3.3 speak\_to()

```
Conversation * Script::speak_to (
    Character * character )
```

talk to this character

#### Parameters

<i>character</i>	the character to speak to
------------------	---------------------------

#### Returns

the conversation to be had

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

- script.h
- script.cpp

## 3.11 Sprite Class Reference

```
#include <sprite.h>
```

### Public Member Functions

- [Sprite](#) (std::string image\_path, SDL\_Renderer \*renderer, int x, int y)
- [~Sprite](#) ()
- void [set\\_source](#) (SDL\_Rect \*source)
- void [set\\_destination](#) (SDL\_Rect \*destination)
- void [draw](#) ()
- void [draw](#) (SDL\_Rect \*destination)
- void [flip\\_draw](#) ()
- void [flip\\_draw](#) (SDL\_Rect \*destination)
- void [set\\_position](#) (int new\_x, int new\_y)
- void [set\\_position](#) (int new\_x)
- void [reset\\_position](#) ()
- void [reset\\_position](#) (int offset)
- int [get\\_x](#) ()
- int [get\\_y](#) ()
- int [get\\_height](#) ()
- int [get\\_width](#) ()

### 3.11.1 Detailed Description

A sprite class

#### Author

Jared Allen

#### Version

8 February 2019

### 3.11.2 Constructor & Destructor Documentation

#### 3.11.2.1 Sprite()

```
Sprite::Sprite (
    std::string image_path,
    SDL_Renderer * renderer,
    int x,
    int y )
```

##### Constructor

##### Parameters

<i>texture_path</i>	the texture file name for the entire texture
<i>x</i>	the x coordinate
<i>y</i>	the y coordinate

#### 3.11.2.2 ~Sprite()

```
Sprite::~Sprite ( )
```

##### Destructor

### 3.11.3 Member Function Documentation

#### 3.11.3.1 draw() [1/2]

```
void Sprite::draw ( )
```

method to draw sprite

##### Parameters

<i>renderer</i>	the pointer to renderer
-----------------	-------------------------

#### 3.11.3.2 draw() [2/2]

```
void Sprite::draw (
```

```
SDL_Rect * destination )
```

method to draw sprite

#### Parameters

<i>renderer</i>	the pointer to renderer
<i>destination</i>	the destination to be drawn to

#### 3.11.3.3 flip\_draw() [1/2]

```
void Sprite::flip_draw ( )
```

method to draw mirror sprite

#### Parameters

<i>renderer</i>	the pointer to the renderer
-----------------	-----------------------------

#### 3.11.3.4 flip\_draw() [2/2]

```
void Sprite::flip_draw (
    SDL_Rect * destination )
```

method to draw mirror sprite

#### Parameters

<i>renderer</i>	the pointer to the renderer
<i>destination</i>	the destination for sprite

#### 3.11.3.5 get\_height()

```
int Sprite::get_height ( )
```

get the sprite's height

#### 3.11.3.6 get\_width()

```
int Sprite::get_width ( )
```

get the sprite's width

### 3.11.3.7 `get_x()`

```
int Sprite::get_x ( )
```

method to get sprite x position

### 3.11.3.8 `get_y()`

```
int Sprite::get_y ( )
```

method to get sprite y position

### 3.11.3.9 `reset_position()` [1/2]

```
void Sprite::reset_position ( )
```

method to reset sprite position

### 3.11.3.10 `reset_position()` [2/2]

```
void Sprite::reset_position (
    int offset )
```

method to reset sprite position with offset

#### Parameters

<i>offset</i>	the offset
---------------	------------

### 3.11.3.11 `set_destination()`

```
void Sprite::set_destination (
    SDL_Rect * destination )
```

method to change destination rectangle

#### Parameters

<i>destination</i>	the new destination
--------------------	---------------------

### 3.11.3.12 `set_position()` [1/2]

```
void Sprite::set_position (
```

```
int new_x,
int new_y )
```

method to change sprite position

#### Parameters

<i>new_x</i>	the new x coordinate
<i>new_y</i>	the new y coordinate

#### 3.11.3.13 set\_position() [2/2]

```
void Sprite::set_position (
    int new_x )
```

method to change sprite position

#### Parameters

<i>new_x</i>	the new x coordinate
--------------	----------------------

#### 3.11.3.14 set\_source()

```
void Sprite::set_source (
    SDL_Rect * source )
```

method to change source rectangle

#### Parameters

<i>source</i>	the new source
---------------	----------------

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

- sprite.h
- sprite.cpp

## 3.12 SpriteLayer Class Reference

```
#include <sprite_layer.h>
```



## Public Member Functions

- [SpriteLayer](#) (std::string image\_path, SDL\_Renderer \*renderer, int x, int y, uint distance)
- [~SpriteLayer](#) ()
- void [left](#) (uint speed)
- void [right](#) (uint speed)
- void [reset](#) ()
- void [reset](#) (int offset)
- void [draw](#) ()

### 3.12.1 Detailed Description

A sprite layer class

#### Author

Jared Allen

#### Version

8 February 2019

### 3.12.2 Constructor & Destructor Documentation

#### 3.12.2.1 SpriteLayer()

```
SpriteLayer::SpriteLayer (
    std::string image_path,
    SDL_Renderer * renderer,
    int x,
    int y,
    uint distance )
```

#### Constructor

##### Parameters

<i>sprite</i>	the sprite
<i>distance</i>	the distance

#### 3.12.2.2 ~SpriteLayer()

```
SpriteLayer::~~SpriteLayer ( )
```

#### Destructor

### 3.12.3 Member Function Documentation

#### 3.12.3.1 draw()

```
void SpriteLayer::draw ( )
```

draw the sprite layer

##### Parameters

<i>renderer</i>	the renderer
-----------------	--------------

#### 3.12.3.2 left()

```
void SpriteLayer::left (
    uint speed )
```

move layer to the left

##### Parameters

<i>speed</i>	
--------------	--

#### 3.12.3.3 reset() [1/2]

```
void SpriteLayer::reset ( )
```

reset sprite layer

#### 3.12.3.4 reset() [2/2]

```
void SpriteLayer::reset (
    int offset )
```

reset sprite layer

##### Parameters

<i>offset</i>	the offset
---------------	------------

### 3.12.3.5 right()

```
void SpriteLayer::right (
    uint speed )
```

move layer to the right

#### Parameters

<i>speed</i>	
--------------	--

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

- `sprite_layer.h`
- `sprite_layer.cpp`

## 3.13 Text\_box Class Reference

```
#include <text_box.h>
```

### Public Member Functions

- [Text\\_box](#) (uint param\_x, uint param\_y, uint param\_h, uint param\_w)
- void [display](#) (string message, SDL\_Renderer \*renderer, TTF\_Font \*font, uint letters)

### 3.13.1 Detailed Description

a class to represent a text box

#### Author

Jared Allen

#### Version

1 June 2019

### 3.13.2 Constructor & Destructor Documentation

#### 3.13.2.1 Text\_box()

```
Text_box::Text_box (
    uint param_x,
    uint param_y,
    uint param_h,
    uint param_w )
```

constructor

**Parameters**

<i>param</i> ↔ _x	the x pos
<i>param</i> ↔ _y	the y pos
<i>param</i> ↔ _h	the height
<i>param</i> ↔ _w	the width

**3.13.3 Member Function Documentation****3.13.3.1 display()**

```
void Text_box::display (
    string message,
    SDL_Renderer * renderer,
    TTF_Font * font,
    uint letters )
```

display message on text box

**Parameters**

<i>message</i>	the message to be displayed
----------------	-----------------------------

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

- text\_box.h
- text\_box.cpp

# Index

- ~Background
  - Background, 6
- ~Character
  - Character, 9
- ~Game
  - Game, 15
- ~Scene
  - Scene, 20
- ~Sprite
  - Sprite, 29
- ~SpriteLayer
  - SpriteLayer, 33
- add\_conversation
  - Scene, 20
- add\_dialogue
  - Conversation, 13
- add\_follower
  - Scene, 20
- add\_interaction
  - Scene, 20
- add\_layer
  - Background, 6
- add\_scene
  - Game, 15
- Background, 5
  - ~Background, 6
  - add\_layer, 6
  - Background, 5, 6
  - blank, 6
  - draw, 6
  - left, 7
  - reset, 7
  - right, 7
- blank
  - Background, 6
- Character, 7
  - ~Character, 9
  - Character, 8
  - equals, 9
  - follow, 9
  - gasp, 9
  - get\_name, 9
  - get\_position, 9
  - get\_screen\_position, 10
  - happy, 10
  - is\_recruited, 10
  - jump, 10
  - reset, 10
  - set\_position, 11
  - set\_recruited, 11
  - set\_screen\_position, 11
  - set\_stage\_pos, 11
  - stand, 11
  - update\_pos, 11
  - walk\_left, 11
  - walk\_right, 12
- contains
  - SceneJunction, 24
- Conversation, 12
  - add\_dialogue, 13
  - Conversation, 13
  - get\_angry\_response, 13
  - get\_dialogue, 13
  - get\_length, 13
  - set\_angry\_response, 14
- display
  - Panel, 18
  - Text\_box, 36
- draw
  - Background, 6
  - Sprite, 29
  - SpriteLayer, 34
- enter
  - Scene, 21
- equals
  - Character, 9
- flip\_draw
  - Sprite, 30
- follow
  - Character, 9
- Game, 14
  - ~Game, 15
  - add\_scene, 15
  - Game, 14
  - join\_scenes, 15
  - play, 15
  - set\_introduction, 15
- gasp
  - Character, 9
- get\_angry\_response
  - Conversation, 13
- get\_dialogue
  - Conversation, 13

- get\_height
  - Sprite, 30
- get\_interaction\_length
  - Interaction, 17
- get\_length
  - Conversation, 13
- get\_message
  - Interaction, 17
- get\_name
  - Character, 9
- get\_next\_character\_position
  - SceneJunction, 25
- get\_next\_scene
  - SceneJunction, 25
- get\_next\_track
  - SceneJunction, 25
- get\_position
  - Character, 9
- get\_previous\_junction\_position
  - Scene, 21
- get\_scene\_position
  - Interaction, 17
- get\_screen\_position
  - Character, 10
- get\_width
  - Sprite, 30
- get\_x
  - Sprite, 30
- get\_y
  - Sprite, 31
- happy
  - Character, 10
- insert\_conversation
  - Script, 27
- insert\_recruit\_conversation
  - Script, 27
- interact
  - Scene, 21
- Interaction, 16
  - get\_interaction\_length, 17
  - get\_message, 17
  - get\_scene\_position, 17
  - Interaction, 16
- is\_recruited
  - Character, 10
- join\_scenes
  - Game, 15
- jump
  - Character, 10
- left
  - Background, 7
  - SpriteLayer, 34
- link\_scenes
  - SceneJunction, 26
- marionette
  - Scene, 21
- movement\_key\_pressed
  - Scene, 21
- Panel, 18
  - display, 18
  - Panel, 18
- play
  - Game, 15
  - Scene, 21
- prompt\_interact
  - Scene, 21
- Report, 19
- reset
  - Background, 7
  - Character, 10
  - Scene, 22
  - SpriteLayer, 34
- reset\_position
  - Sprite, 31
- right
  - Background, 7
  - SpriteLayer, 35
- Scene, 19
  - ~Scene, 20
  - add\_conversation, 20
  - add\_follower, 20
  - add\_interaction, 20
  - enter, 21
  - get\_previous\_junction\_position, 21
  - interact, 21
  - marionette, 21
  - movement\_key\_pressed, 21
  - play, 21
  - prompt\_interact, 21
  - reset, 22
  - Scene, 20
  - set\_junction, 22
  - set\_previous\_junction\_position, 22
  - stage\_junction, 22
  - stage\_left, 22
  - stage\_left\_barrier, 23
  - stage\_right, 23
  - stage\_right\_barrier, 23
- SceneJunction, 23
  - contains, 24
  - get\_next\_character\_position, 25
  - get\_next\_scene, 25
  - get\_next\_track, 25
  - link\_scenes, 26
  - SceneJunction, 24
- Script, 26
  - insert\_conversation, 27
  - insert\_recruit\_conversation, 27
  - Script, 27
  - speak\_to, 27
- set\_angry\_response

- Conversation, [14](#)
- set\_destination
  - Sprite, [31](#)
- set\_introduction
  - Game, [15](#)
- set\_junction
  - Scene, [22](#)
- set\_position
  - Character, [11](#)
  - Sprite, [31](#), [32](#)
- set\_previous\_junction\_position
  - Scene, [22](#)
- set\_recruited
  - Character, [11](#)
- set\_screen\_position
  - Character, [11](#)
- set\_source
  - Sprite, [32](#)
- set\_stage\_pos
  - Character, [11](#)
- speak\_to
  - Script, [27](#)
- Sprite, [28](#)
  - ~Sprite, [29](#)
  - draw, [29](#)
  - flip\_draw, [30](#)
  - get\_height, [30](#)
  - get\_width, [30](#)
  - get\_x, [30](#)
  - get\_y, [31](#)
  - reset\_position, [31](#)
  - set\_destination, [31](#)
  - set\_position, [31](#), [32](#)
  - set\_source, [32](#)
  - Sprite, [29](#)
- SpriteLayer, [32](#)
  - ~SpriteLayer, [33](#)
  - draw, [34](#)
  - left, [34](#)
  - reset, [34](#)
  - right, [35](#)
  - SpriteLayer, [33](#)
- stage\_junction
  - Scene, [22](#)
- stage\_left
  - Scene, [22](#)
- stage\_left\_barrier
  - Scene, [23](#)
- stage\_right
  - Scene, [23](#)
- stage\_right\_barrier
  - Scene, [23](#)
- stand
  - Character, [11](#)
- Text\_box, [35](#)
  - display, [36](#)
  - Text\_box, [35](#)
- update\_pos
  - Character, [11](#)
- walk\_left
  - Character, [11](#)
- walk\_right
  - Character, [12](#)