
bruhanimate

Ethan Christensen

Aug 24, 2024

BRUHANIMATE:

1	bruhanimate.bruhutil package	3
1.1	Submodules	3
1.2	bruhanimate.bruhutil.bruhffer module	3
1.3	bruhanimate.bruhutil.bruhscreen module	4
1.4	bruhanimate.bruhutil.images module	7
1.5	bruhanimate.bruhutil.utils module	7
1.6	Module contents	7
2	bruhanimate.bruheffect package	13
2.1	Submodules	13
2.2	bruhanimate.bruheffect.audio_effect module	13
2.3	bruhanimate.bruheffect.base_effect module	13
2.4	bruhanimate.bruheffect.chatbot_effect module	14
2.5	bruhanimate.bruheffect.draw_lines_effect module	15
2.6	bruhanimate.bruheffect.game_of_life_effect module	15
2.7	bruhanimate.bruheffect.matrix_effect module	16
2.8	bruhanimate.bruheffect.noise_effect module	16
2.9	bruhanimate.bruheffect.offset_effect module	17
2.10	bruhanimate.bruheffect.plasma_effect module	17
2.11	bruhanimate.bruheffect.rain_effect module	18
2.12	bruhanimate.bruheffect.snow_effect module	18
2.13	bruhanimate.bruheffect.star_effect module	19
2.14	bruhanimate.bruheffect.static_effect module	19
2.15	bruhanimate.bruheffect.twinkle_effect module	19
2.16	Module contents	19
3	bruhanimate.bruhrenderer package	25
3.1	Submodules	25
3.2	bruhanimate.bruhrenderer.background_color_renderer module	25
3.3	bruhanimate.bruhrenderer.base_renderer module	25
3.4	bruhanimate.bruhrenderer.center_renderer module	26
3.5	bruhanimate.bruhrenderer.effect_renderer module	26
3.6	bruhanimate.bruhrenderer.focus_renderer module	27
3.7	bruhanimate.bruhrenderer.pan_renderer module	27
3.8	Module contents	27
	Python Module Index	31
	Index	33

Welcome to bruhanimate's documentation!

BRUHANIMATE.BRUHUTIL PACKAGE

1.1 Submodules

1.2 bruhanimate.bruhutil.bruhffer module

Copyright 2023 Ethan Christensen Copied, Guided, and Adapted from Asciiatics <<https://github.com/peterbrittain/asciiatics/blob/master/asciiatics/screen.py>>

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

class bruhanimate.bruhutil.bruhffer.**Buffer**(*height, width*)

Bases: object

Class for creating and managing a buffer

clear_buffer(*x=0, y=0, w=None, h=None, val=' '*)

Clear a section of this buffer :param x: x position to start the clear :param y: y position to start the clear
:param w: width of the section to be cleared :param h: height of the section to be cleared

get_buffer_changes(*in_buf*)

Return all the differences between this buffer and buffer that was passed in :param in_buf: buffer to compare
this buffer to

get_char(*x, y*)

Return the value at the given location

grab_slice(*x, y, width*)

Grabs a part of a row from this buffer :param x: column position to start grabbing :param y: row position
to start grabbing :param width: number of chracters to grab

height()

put_at(*x, y, text, transparent=False*)

Put text at a given x, y coordinate in the buffer :param x: column position to start placing the text :param
y: row position to start placing the text :param text: the text to be placed

put_at_center(*y, text, transparent=False*)

Puts the given text in the center of the row given by *y*. :param *y*: row to place the text. :param *text*: text to write to the buffers.

put_char(*x, y, val, transparent=False*)

Put the value at the given location

scroll(*shift*)

Scrolls the buffer up or down a number of lines denoted by the shift value. '-' -> scroll down, '+' -> scroll up :param *shift*: amount to shift up or down

shift(*shift*)

Shift the entire buffer to the right by the value denoted by *shift* :param *shift*: amount to shift the row by

shift_line(*y, shift*)

Shift the given line to the right by the value denoted by *shift*. :param *y*: index of the row to shift :param *shift*: amount to shift the row by

sync_over_top(*in_buf*)

Apply non-none values over top this buffer from the *in_buffer* :param *in_buf*: buffer to take non-none values from

sync_with(*in_buf*)

Sync this buffer with the given buffer :param *in_buf*: buffer to be applied to this buffer

width()

1.3 bruhanimate.bruhutil.bruhscreen module

Copyright 2023 Ethan Christensen Copied, Guided, and Adapted from Asciiatics <<https://github.com/peterbrittain/asciiatics/blob/master/asciiatics/screen.py>>

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

class bruhanimate.bruhutil.bruhscreen.**Screen**(*stdout, stdin, old_out, old_in*)

Bases: object

Class for creating and managing a terminal screen in a WINDOWS OS terminal

KEY_ADD = -411

KEY_BACK = -300

KEY_BACK_TAB = -302

KEY_CAPS_LOCK = -500

KEY_CONTROL = -601

KEY_DECIMAL = -413


```
KEY_DELETE = -102
KEY_DIVIDE = -414
KEY_DOWN = -206
KEY_END = -201
KEY_ESCAPE = -1
KEY_F1 = -2
KEY_F10 = -11
KEY_F11 = -12
KEY_F12 = -13
KEY_F13 = -14
KEY_F14 = -15
KEY_F15 = -16
KEY_F16 = -17
KEY_F17 = -18
KEY_F18 = -19
KEY_F19 = -20
KEY_F2 = -3
KEY_F20 = -21
KEY_F21 = -22
KEY_F22 = -23
KEY_F23 = -24
KEY_F24 = -25
KEY_F3 = -4
KEY_F4 = -5
KEY_F5 = -6
KEY_F6 = -7
KEY_F7 = -8
KEY_F8 = -9
KEY_F9 = -10
KEY_HOME = -200
KEY_INSERT = -101
```

```
KEY_LEFT = -203
KEY_MENU = -602
KEY_MULTIPLY = -410
KEY_NUMPAD0 = -400
KEY_NUMPAD1 = -401
KEY_NUMPAD2 = -402
KEY_NUMPAD3 = -403
KEY_NUMPAD4 = -404
KEY_NUMPAD5 = -405
KEY_NUMPAD6 = -406
KEY_NUMPAD7 = -407
KEY_NUMPAD8 = -408
KEY_NUMPAD9 = -409
KEY_NUM_LOCK = -501
KEY_PAGE_DOWN = -208
KEY_PAGE_UP = -207
KEY_PRINT_SCREEN = -100
KEY_RIGHT = -205
KEY_SCROLL_LOCK = -502
KEY_SHIFT = -600
KEY_SUBTRACT = -412
KEY_TAB = -301
KEY_UP = -204

clear()
close(restore=True)
get_event()
    Check for any event without waiting.
has_resized()
classmethod open()
print_at(text, x, y, width)
print_center(text, y, width)
```

set_title(*title: str*) → None

classmethod show(*function, args=None*)

wait_for_input(*timeout: int*) → None

1.4 bruhanimate.bruhutil.images module

Copyright 2023 Ethan Christensen

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

bruhanimate.bruhutil.images.get_image(*name*)

Function to return one of the premade images :param name: name of the image to get

bruhanimate.bruhutil.images.text_to_image(*text, font='standard', padding_top_bottom=0, padding_left_right=0*)

Function to take a piece of text and turn it into an image that can be used. :param text: text to turn into an image :param font: pyfiglet font to use :param padding_top_bottom: padding to apply to the generated image :param padding_left_right: padding to apply to the generate image

1.5 bruhanimate.bruhutil.utils module

bruhanimate.bruhutil.utils.sleep(*s*)

1.6 Module contents

class bruhanimate.bruhutil.Buffer(*height, width*)

Bases: object

Class for creating and managing a buffer

clear_buffer(*x=0, y=0, w=None, h=None, val=' '*)

Clear a section of this buffer :param x: x position to start the clear :param y: y position to start the clear :param w: width of the section to be cleared :param h: height of the section to be cleared

get_buffer_changes(*in_buf*)

Return all the differences between this buffer and buffer that was passed in :param in_buf: buffer to compare this buffer to

get_char(*x, y*)

Return the value at the given location

grab_slice(*x, y, width*)

Grabs a part of a row from this buffer :param x: column position to start grabbing :param y: row position to start grabbing :param width: number of chracters to grab

height()

put_at(*x, y, text, transparent=False*)

Put text at a given x, y coordinate in the buffer :param x: column position to start placing the text :param y: row position to start placing the text :param text: the text to be placed

put_at_center(*y, text, transparent=False*)

Puts the given text in the center of the row given by y. :param y: row to place the text. :param text: text to write to the buffers.

put_char(*x, y, val, transparent=False*)

Put the value at the given location

scroll(*shift*)

Scrolls the buffer up or down a number of lines denoted by the shift value. '-' -> scroll down, '+' -> scroll up :param shift: amount to shift up or down

shift(*shift*)

Shift the entire buffer to the right by the value denoted by shift :param shift: amount to shift the row by

shift_line(*y, shift*)

Shift the given line to the right by the value denoted by shift. :param y: index of the row to shift :param shift: amount to shift the row by

sync_over_top(*in_buf*)

Apply non-none values over top this buffer from the in_buffer :param in_buf: buffer to take non-none values from

sync_with(*in_buf*)

Sync this buffer with the given buffer :param in_buf: buffer to be applied to this buffer

width()

class bruhanimate.bruhutil.**Screen**(*stdout, stdin, old_out, old_in*)

Bases: object

Class for creating and managing a terminal screen in a WINDOWS OS terminal

KEY_ADD = -411

KEY_BACK = -300

KEY_BACK_TAB = -302

KEY_CAPS_LOCK = -500

KEY_CONTROL = -601

KEY_DECIMAL = -413

KEY_DELETE = -102

KEY_DIVIDE = -414

KEY_DOWN = -206

```
KEY_END = -201
KEY_ESCAPE = -1
KEY_F1 = -2
KEY_F10 = -11
KEY_F11 = -12
KEY_F12 = -13
KEY_F13 = -14
KEY_F14 = -15
KEY_F15 = -16
KEY_F16 = -17
KEY_F17 = -18
KEY_F18 = -19
KEY_F19 = -20
KEY_F2 = -3
KEY_F20 = -21
KEY_F21 = -22
KEY_F22 = -23
KEY_F23 = -24
KEY_F24 = -25
KEY_F3 = -4
KEY_F4 = -5
KEY_F5 = -6
KEY_F6 = -7
KEY_F7 = -8
KEY_F8 = -9
KEY_F9 = -10
KEY_HOME = -200
KEY_INSERT = -101
KEY_LEFT = -203
KEY_MENU = -602
KEY_MULTIPLY = -410
```

```
KEY_NUMPAD0 = -400
KEY_NUMPAD1 = -401
KEY_NUMPAD2 = -402
KEY_NUMPAD3 = -403
KEY_NUMPAD4 = -404
KEY_NUMPAD5 = -405
KEY_NUMPAD6 = -406
KEY_NUMPAD7 = -407
KEY_NUMPAD8 = -408
KEY_NUMPAD9 = -409
KEY_NUM_LOCK = -501
KEY_PAGE_DOWN = -208
KEY_PAGE_UP = -207
KEY_PRINT_SCREEN = -100
KEY_RIGHT = -205
KEY_SCROLL_LOCK = -502
KEY_SHIFT = -600
KEY_SUBTRACT = -412
KEY_TAB = -301
KEY_UP = -204

clear()

close(restore=True)

get_event()
    Check for any event without waiting.

has_resized()

classmethod open()

print_at(text, x, y, width)

print_center(text, y, width)

set_title(title: str) → None

classmethod show(function, args=None)

wait_for_input(timeout: int) → None
```

`bruhanimate.bruhutil.get_image(name)`

Function to return one of the premade images :param name: name of the image to get

`bruhanimate.bruhutil.sleep(s)`

`bruhanimate.bruhutil.text_to_image(text,font='standard',padding_top_bottom=0,padding_left_right=0)`

Function to take a piece of text and turn it into an image that can be used. :param text: text to turn into an image
:param font: pyfiglet font to use :param padding_top_bottom: padding to apply to the generated image :param
padding_left_right: padding to apply to the generate image

BRUHANIMATE.BRUHEFFECT PACKAGE

2.1 Submodules

2.2 `bruhanimate.bruheffect.audio_effect` module

class `bruhanimate.bruheffect.audio_effect.AudioEffect`(*buffer, background: str, num_bands: int = 24, audio_halt: int = 10*)

Bases: `BaseEffect`

evenly_distribute_original_values(*original_list, desired_width*)

generate_even_ranges(*groups, start, end*)

map_bands_to_range(*N*)

process_audio(*data, frame_count, time_info, status*)

render_frame(*frame_number*)

To be defined by each effect

set_audio_gradient(*gradient=[232, 233, 235, 237, 239, 241, 243, 245, 247, 249, 251, 253, 255], mode='extend'*)

set_audio_properties(*num_bands=24, audio_halt=10, use_gradient=True, non_gradient_color=27*)

set_orientation(*orientation*)

2.3 `bruhanimate.bruheffect.base_effect` module

class `bruhanimate.bruheffect.base_effect.BaseEffect`(*buffer, background*)

Bases: `object`

Class for keeping track of an effect, and updating its buffer

abstract render_frame(*frame_number*)

To be defined by each effect

2.4 bruhanimate.bruheffect.chatbot_effect module

```
class bruhanimate.bruheffect.chatbot_effect.ChatbotEffect(screen: Screen, buffer, back_buffer,
                                                         background: str = '')

    Bases: BaseEffect

    place_all_keys()

    render_frame(frame_number)
        To be defined by each effect

    scroll_keys(shift: int = 1)

    set_avatar_properties(size: int)

    set_chatbot_blink_halt(halt: int)

    set_chatbot_cursor_colors(color_one: int | str, color_two: int | str)

    set_chatbot_print_halt(halt: int)

    set_chatbot_properties(interface: str | None, model: str, user: str | None = None, client: OpenAI |
                          AzureOpenAI | None = None, use_message_history: bool = False,
                          message_history_cap: int = 5)

    set_chatbot_stats(show: bool = False)

    set_chatbot_text_gradient(gradient: list[int | str], mul: int)

    set_chatbot_user_colors(chatbot_text_color: int | str | None = None, chatbot_background_color: int | str
                          | None = None, chatbot_avatar_color: int | str | None = None,
                          chatbot_avatar_text_color: int | str | None = None, user_text_color: int | str |
                          None = None, user_background_color: int | str | None = None,
                          user_avatar_color: int | str | None = None, user_avatar_text_color: int | str |
                          None = None)

    set_divider_flag(divider: bool, divider_character: str = '-')

    set_gradient_noise_halts(char_halt: int | None = None, color_halt: int | None = None)

    set_second_effect(effect: str)

class bruhanimate.bruheffect.chatbot_effect.GradientNoise(x, y, length, char_halt=1, color_halt=1,
                                                         gradient_length=1)

    Bases: object

    generate(frame_number: int)

    mark_done()

    update_gradient(gradient)

class bruhanimate.bruheffect.chatbot_effect.Key(character, representation, value, x, y)

    Bases: object

class bruhanimate.bruheffect.chatbot_effect.Loading(animate_part: GradientNoise)

    Bases: object
```

mark_done()

update(*frame: int*)

```
class bruhanimate.bruheffect.chatbot_effect.OllamaApiCaller(model: str, use_message_history:
                                                         bool = False, message_history_cap:
                                                         int = 5)
```

Bases: object

chat(*message: str, user: str | None, previous_messages: list[str] | None = None*) → str

```
class bruhanimate.bruheffect.chatbot_effect.OpenAiCaller(client: OpenAI | AzureOpenAI, model:
                                                         str, use_message_history: bool = False,
                                                         message_history_cap: int = 5)
```

Bases: object

chat(*message: str, user: str | None*) → str

```
class bruhanimate.bruheffect.chatbot_effect.StringStreamer(x: int, y: int, text: str, start_frame: int,
                                                            halt: int = 1)
```

Bases: object

generate(*frame: int*)

2.5 bruhanimate.bruheffect.draw_lines_effect module

```
class bruhanimate.bruheffect.draw_lines_effect.DrawLinesEffect(buffer, background, char=None,
                                                                thin=False)
```

Bases: *BaseEffect*

add_line(*start_point, end_point*)

render_frame(*frame_number*)

To be defined by each effect

```
class bruhanimate.bruheffect.draw_lines_effect.Line(start_point, end_point)
```

Bases: object

get_points()

update_points(*start_point, end_point*)

2.6 bruhanimate.bruheffect.game_of_life_effect module

```
class bruhanimate.bruheffect.game_of_life_effect.GameOfLifeEffect(buffer, background,
                                                                    decay=False, color=False,
                                                                    color_type=None,
                                                                    scale='random'))
```

Bases: *BaseEffect*

Effect to simulate Conway's Game of Life

render_frame(*frame_number*)

Function to render the next frame of the GOL effect

update_decay(*decay*, *color_type*='GREYSCALE', *scale*='random')

Function to enable to decay and select the color map :param decay: True / False :param color_type: color map for the effect

update_rules(*life_rule*, *death_rule*)

2.7 bruhanimate.bruheffect.matrix_effect module

```
class bruhanimate.bruheffect.matrix_effect.MatrixEffect(buffer, background,
                                                         chracter_halt_range=(1, 2),
                                                         color_halt_range=(1, 2),
                                                         character_randomness_one=0.7,
                                                         character_randomness_two=0.6,
                                                         color_randomness=0.5,
                                                         gradient_length=1)
```

Bases: *BaseEffect*

Effect to mimic the cliche coding backgroud with falling random characters

get_gradient()

render_frame(*frame_number*)

Renders the next frame for the Matrix effect into the effect buffer

set_matrix_gradient(*gradient*)

set_matrix_properties(*chracter_halt_range*=(1, 2), *color_halt_range*=(1, 2),
character_randomness_one=0.7, *character_randomness_two*=0.6,
color_randomness=0.5, *gradient_length*=1)

2.8 bruhanimate.bruheffect.noise_effect module

```
class bruhanimate.bruheffect.noise_effect.NoiseEffect(buffer, background, intensity=200,
                                                         color=False)
```

Bases: *BaseEffect*

Class for generating noise. :param intensity: randomness for the noise, higher the value the slower the effect (due to computation).

Will be a value 1 - 999

Parameters

color – whether or not to color the noise

render_frame(*frame_number*)

Function to render the next frame of the Noise effect

update_color(*color, characters*)

Function to enable / disable color for the effect :param color: True / False :param character: True / False to make characters visible

update_intensity(*intensity*)

Function to update the intensity of the effect :param intensity: new intensity

2.9 bruhanimate.bruheffect.offset_effect module

class bruhanimate.bruheffect.offset_effect.**OffsetEffect**(*buffer, background, direction='right'*)

Bases: *BaseEffect*

Class for generating an offset-static background. :new-param direction: which way the offset should go.

render_frame(*frame_number*)

Function to render the next frame of the Offset effect

update_direction(*direction*)

Function to update the direction of the offset :param direction: East / West

2.10 bruhanimate.bruheffect.plasma_effect module

class bruhanimate.bruheffect.plasma_effect.**PlasmaEffect**(*buffer, background*)

Bases: *BaseEffect*

Function to generate a plasma like effect

func(*x, y, a, b, n*)

Helper function to calculate the plasma value given the four plasma values

render_frame(*frame_number*)

Function to render the next frame of the Plasma Effect

shuffle_plasma_values()

Function to generate a new-random set of plasma values

update_background(*background*)

Update the background character(s) :param background: the new background

update_color(*colors*)

Function to update the colors used

update_color_properties(*color, characters=True, random_color=False*)

Function to update the color properties. random_color overrules other functions like update greyscale size and update color :param color: True / False to enable color :param characters: True / False to show the characters :param random_color: True / False to generate random colors

update_grey_scale_size(*size*)

Function to change the size of the grey scale

update_info_visibility(*visible*)

Function to enable or disable info about the effect

update_plasma_values(*a=43, b=18, c=19, d=19*)

Function to set the plasma values

2.11 bruhanimate.bruheffect.rain_effect module

```
class bruhanimate.bruheffect.rain_effect.RainEffect(buffer, background, img_start_x=None,
                                                    img_start_y=None, img_width=None,
                                                    img_height=None, collision=False, intensity=1,
                                                    swells=False, wind_direction='none')
```

Bases: *BaseEffect*

Effect to emulate the look of rain

render_frame(frame_number)

Function to render the next frame of the Rain Effect

update_collision(img_start_x, img_start_y, img_width, img_height, collision, smart_transparent=False, image_buffer=None)

Function to set whether or not to visually see the rain collide with the ground or images if they are present
:param img_start_x: where the image starts on the screen :param img_start_y: where the image starts on the screen :param img_width: the width of the image :param img_height: the height of the image
:param collision: update collision variable :param smart_transparent: update smart_transparent :param image_buffer: the buffer that contains the image

update_intensity(intensity)

Function to update the intensity of the rain :param intensity: intensity value

update_multiplier(val)

Update the multiplier value that relates to shift amount :param val: value to set the multiplier to

update_swells(swells)

Function to set whether the intensity should evolve on it's own :param swells: True / False

update_wind_direction(direction)

Update the direction of the rain :param direction: direction for the rain to fall (east, west, none)

2.12 bruhanimate.bruheffect.snow_effect module

```
class bruhanimate.bruheffect.snow_effect.SnowEffect(buffer, background, img_start_x=None,
                                                    img_start_y=None, img_width=None,
                                                    img_height=None, collision=False,
                                                    show_info=False)
```

Bases: *BaseEffect*

render_frame(frame_number)

To be defined by each effect

show_info(show_info: bool)

update_collision(img_start_x, img_start_y, img_width, img_height, collision, image_buffer=None)

Function to set whether or not to visually see the snow collide with the ground or images if they are present
:param img_start_x: where the image starts on the screen :param img_start_y: where the image starts on the screen :param img_width: the width of the image :param img_height: the height of the image :param collision: update collision variable

2.13 bruhanimate.bruheffect.star_effect module

class bruhanimate.bruheffect.star_effect.**StarEffect**(*buffer, background, color_type='GREYSCALE'*)

Bases: *NoiseEffect*

Class for rendering out a blinking star effect. This is just a Noise effect with a predefined intensity. Ideally the background would be ' ' for the best effect, but the choice is yours.

render_frame(*frame_number*)

Function to update the next frame of the Stars effect

update_background(*background*)

Function to update the background of the effect :param background: the new background

update_color_type(*color_type*)

Function to update the color of the stars :param color_type: color map

2.14 bruhanimate.bruheffect.static_effect module

class bruhanimate.bruheffect.static_effect.**StaticEffect**(*buffer, background*)

Bases: *BaseEffect*

Class for generating a static background.

render_frame(*frame_number*)

Renders the background to the screen

2.15 bruhanimate.bruheffect.twinkle_effect module

class bruhanimate.bruheffect.twinkle_effect.**TWINKLE_SPEC**(*char, value*)

Bases: object

copy()

next()

class bruhanimate.bruheffect.twinkle_effect.**TwinkleEffect**(*buffer, background*)

Bases: *BaseEffect*

render_frame(*frame_number*)

To be defined by each effect

2.16 Module contents

class bruhanimate.bruheffect.**AudioEffect**(*buffer, background: str, num_bands: int = 24, audio_halt: int = 10*)

Bases: *BaseEffect*

evenly_distribute_original_values(*original_list, desired_width*)

```
generate_even_ranges(groups, start, end)

map_bands_to_range(N)

process_audio(data, frame_count, time_info, status)

render_frame(frame_number)
    To be defined by each effect

set_audio_gradient(gradient=[232, 233, 235, 237, 239, 241, 243, 245, 247, 249, 251, 253, 255],
                  mode='extend')

set_audio_properties(num_bands=24, audio_halt=10, use_gradient=True, non_gradient_color=27)

set_orientation(orientation)

class bruhanimate.bruheffect.BaseEffect(buffer, background)
    Bases: object
    Class for keeping track of an effect, and updataing it's buffer
    abstract render_frame(frame_number)
        To be defined by each effect

class bruhanimate.bruheffect.ChatbotEffect(screen: Screen, buffer, back_buffer, background: str = '')
    Bases: BaseEffect
    place_all_keys()

    render_frame(frame_number)
        To be defined by each effect

    scroll_keys(shift: int = 1)

    set_avatar_properties(size: int)

    set_chatbot_blink_halt(halt: int)

    set_chatbot_cursor_colors(color_one: int | str, color_two: int | str)

    set_chatbot_print_halt(halt: int)

    set_chatbot_properties(interface: str | None, model: str, user: str | None = None, client: OpenAI |
                          AzureOpenAI | None = None, use_message_history: bool = False,
                          message_history_cap: int = 5)

    set_chatbot_stats(show: bool = False)

    set_chatbot_text_gradient(gradient: list[int | str], mul: int)

    set_chatbot_user_colors(chatbot_text_color: int | str | None = None, chatbot_background_color: int | str
                          | None = None, chatbot_avatar_color: int | str | None = None,
                          chatbot_avatar_text_color: int | str | None = None, user_text_color: int | str |
                          None = None, user_background_color: int | str | None = None,
                          user_avatar_color: int | str | None = None, user_avatar_text_color: int | str |
                          None = None)

    set_divider_flag(divider: bool, divider_character: str = '-')

```



```

set_gradient_noise_halts(char_halt: int | None = None, color_halt: int | None = None)

set_second_effect(effect: str)

class bruhanimate.bruheffect.DrawLinesEffect(buffer, background, char=None, thin=False)
    Bases: BaseEffect
    add_line(start_point, end_point)
    render_frame(frame_number)
        To be defined by each effect
class bruhanimate.bruheffect.GameOfLifeEffect(buffer, background, decay=False, color=False,
                                                color_type=None, scale='random')
    Bases: BaseEffect
    Effect ot simulate Conway's Game of Life
    render_frame(frame_number)
        Function to render the next frame of the GOL effect
    update_decay(decay, color_type='GREYSCALE', scale='random')
        Function to enable to decay and select the color map :param decay: True / False :param color_type: color
        map for the effect
    update_rules(life_rule, death_rule)

class bruhanimate.bruheffect.GradientNoise(x, y, length, char_halt=1, color_halt=1, gradient_length=1)
    Bases: object
    generate(frame_number: int)
    mark_done()
    update_gradient(gradient)

class bruhanimate.bruheffect.Key(character, representation, value, x, y)
    Bases: object

class bruhanimate.bruheffect.Line(start_point, end_point)
    Bases: object
    get_points()
    update_points(start_point, end_point)

class bruhanimate.bruheffect.Loading(animate_part: GradientNoise)
    Bases: object
    mark_done()
    update(frame: int)

class bruhanimate.bruheffect.MatrixEffect(buffer, background, chracter_halt_range=(1, 2),
                                            color_halt_range=(1, 2), character_randomness_one=0.7,
                                            character_randomness_two=0.6, color_randomness=0.5,
                                            gradient_length=1)
    Bases: BaseEffect
    Effect to mimic the cliché coding background with falling random characters

```

get_gradient()

render_frame(*frame_number*)

Renders the next frame for the Matrix effect into the effect buffer

set_matrix_gradient(*gradient*)

set_matrix_properties(*chacter_halt_range*=(1, 2), *color_halt_range*=(1, 2),
character_randomness_one=0.7, *character_randomness_two*=0.6,
color_randomness=0.5, *gradient_length*=1)

class bruhanimate.bruheffect.**NoiseEffect**(*buffer*, *background*, *intensity*=200, *color*=False)

Bases: [*BaseEffect*](#)

Class for generating noise. :param intensity: randomness for the noise, higher the value the slower the effect (due to computation).

Will be a value 1 - 999

Parameters

color – whether or not to color the noise

render_frame(*frame_number*)

Function to render the next frame of the Noise effect

update_color(*color*, *characters*)

Function to enable / disable color for the effect :param color: True / False :param character: True / False to make characters visible

update_intensity(*intensity*)

Function to update the intensity of the effect :param intensity: new intensity

class bruhanimate.bruheffect.**OffsetEffect**(*buffer*, *background*, *direction*='right')

Bases: [*BaseEffect*](#)

Class for generating an offset-static background. :new-param direction: which way the offset should go.

render_frame(*frame_number*)

Function to render the next frame of the Offset effect

update_direction(*direction*)

Function to update the direction of the offset :param direction: East / West

class bruhanimate.bruheffect.**PlasmaEffect**(*buffer*, *background*)

Bases: [*BaseEffect*](#)

Function to generate a plasma like effect

func(*x*, *y*, *a*, *b*, *n*)

Helper function to calculate the plasma value given the four plasma values

render_frame(*frame_number*)

Function to render the next frame of the Plasma Effect

shuffle_plasma_values()

Function to generate a new-random set of plasma values

update_background(*background*)

Update the background character(s) :param background: the new background

update_color(*colors*)

Function to update the colors used

update_color_properties(*color, characters=True, random_color=False*)

Function to update the color properties. *random_color* overrules other functions like *update greyscale size* and *update color* :param *color*: True / False to enable color :param *characters*: True / False to show the characters :param *random_color*: True / False to generate random colors

update_grey_scale_size(*size*)

Function to change the size of the grey scale

update_info_visibility(*visible*)

Function to enable or disable info about the effect

update_plasma_values(*a=43, b=18, c=19, d=19*)

Function to set the plasma values

class bruhanimate.bruheffect.**RainEffect**(*buffer, background, img_start_x=None, img_start_y=None, img_width=None, img_height=None, collision=False, intensity=1, swells=False, wind_direction='none'*)

Bases: [BaseEffect](#)

Effect to emulate the look of rain

render_frame(*frame_number*)

Function to render the next frame of the Rain Effect

update_collision(*img_start_x, img_start_y, img_width, img_height, collision, smart_transparent=False, image_buffer=None*)

Function to set whether or not to visually see the rain collide with the ground or images if they are present :param *img_start_x*: where the image starts on the screen :param *img_start_y*: where the image starts on the screen :param *img_width*: the width of the image :param *img_height*: the height of the image :param *collision*: update collision variable :param *smart_transparent*: update smart_transparent :param *image_buffer*: the buffer that contains the image

update_intensity(*intensity*)

Function to update the intensity of the rain :param *intensity*: intensity value

update_multiplier(*val*)

Update the multiplier value that relates to shift amount :param *val*: value to set the multiplier to

update_swells(*swells*)

Function to set whether the intensity should evolve on it's own :param *swells*: True / False

update_wind_direction(*direction*)

Update the direction of the rain :param *direction*: direction for the rain to fall (east, west, none)

class bruhanimate.bruheffect.**SnowEffect**(*buffer, background, img_start_x=None, img_start_y=None, img_width=None, img_height=None, collision=False, show_info=False*)

Bases: [BaseEffect](#)

render_frame(*frame_number*)

To be defined by each effect

show_info(*show_info: bool*)

update_collision(*img_start_x, img_start_y, img_width, img_height, collision, image_buffer=None*)

Function to set whether or not to visually see the snow collide with the ground or images if they are present
:param img_start_x: where the image starts on the screen :param img_start_y: where the image starts on the screen :param img_width: the width of the image :param img_height: the height of the image :param collision: update collision variable

class bruhanimate.bruheffect.**StarEffect**(*buffer, background, color_type='GREYSCALE'*)

Bases: *NoiseEffect*

Class for rendering out a blinking star effect. This is just a Noise effect with a predefined intensity. Ideally the background would be ' ' for the best effect, but the choice is yours.

render_frame(*frame_number*)

Function to update the next frame of the Stars effect

update_background(*background*)

Function to update the background of the effect :param background: the new background

update_color_type(*color_type*)

Function to update the color of the stars :param color_type: color map

class bruhanimate.bruheffect.**StaticEffect**(*buffer, background*)

Bases: *BaseEffect*

Class for generating a static background.

render_frame(*frame_number*)

Renders the background to the screen

class bruhanimate.bruheffect.**StringStreamer**(*x: int, y: int, text: str, start_frame: int, halt: int = 1*)

Bases: object

generate(*frame: int*)

class bruhanimate.bruheffect.**TWINKLE_SPEC**(*char, value*)

Bases: object

copy()

next()

class bruhanimate.bruheffect.**TwinkleEffect**(*buffer, background*)

Bases: *BaseEffect*

render_frame(*frame_number*)

To be defined by each effect

BRUHANIMATE.BRUHRENDERER PACKAGE

3.1 Submodules

3.2 bruhanimate.bruhrenderer.background_color_renderer module

```
class bruhanimate.bruhrenderer.background_color_renderer.BackgroundColorRenderer(screen,  
                                                                                frames,  
                                                                                time, img,  
                                                                                on_color_code,  
                                                                                ef-  
                                                                                fect_type='static',  
                                                                                back-  
                                                                                ground=  
                                                                                ', transpar-  
                                                                                ent=False)
```

Bases: *BaseRenderer*

render_img_frame(frame_number)

3.3 bruhanimate.bruhrenderer.base_renderer module

```
class bruhanimate.bruhrenderer.base_renderer.BaseRenderer(screen: Screen, frames: int = 100, time:  
float = 0.1, effect_type: str = 'static',  
background: str = '', transparent: bool  
= False, collision: bool = False)
```

Bases: object

Defines the base methods, abstract methods, and base attributes for the render class, is an Effect Only Renderer

push_front_to_screen()

Pushes changes between the back_buffer and front_buffer and applies them to the screen.

Parameters

None – This method does not take any parameters.

Return None

This method does not return anything.

render_exit()

Renders out the exit prompt to the screen.

abstract render_frame()

To be defined by each renderer

run(end_message=True)

Updates the image_buffer and effect_buffer. Then the image_buffer is applied over top the effect_buffer and stored into the back_buffer. After the front_buffer is rendered to the screen, the front_buffer is synced with the back_buffer. Why? So the effect and image, and there associated calculations can be done independently.

update_collision(collision)

Method for updating the collision for the rain effect

update_exit_stats(msg1=None, msg2=None, wipe=None, x_loc=None, y_loc=None, centered=False)

Set the exit messages for when the animation finishes :param msg1: primary message :param msg2: secondary message :param wipe: whether to clear the buffer :param x_loc: where to put the message along the xaxis :param y_loc: where to put the message along the yaxis :param centered: whether or not the message should be centered

update_smart_transparent(smart_transparent)

Enable / Disable the smart transparency effect :param smart_transparent: True / False

bruhanimate.bruhrenderer.base_renderer.sleep(s)

3.4 bruhanimate.bruhrenderer.center_renderer module

```
class bruhanimate.bruhrenderer.center_renderer.CenterRenderer(screen: Screen, img: list[str],
                                                              frames: int = 100, time: float = 0.1,
                                                              effect_type: str = 'static',
                                                              background: str = '', transparent:
                                                              bool = False)
```

Bases: [BaseRenderer](#)

A renderer to load an image in the center of the screen. Updates the image_buffer only

render_img_frame(frame_number)

Renders out the image to the center of the screen, if there is no image passed into the renderer then the background is rendered on it's own

3.5 bruhanimate.bruhrenderer.effect_renderer module

```
class bruhanimate.bruhrenderer.effect_renderer.EffectRenderer(screen: Screen, frames: int = 100,
                                                              time: float = 0.1, effect_type: str =
                                                              'static', background: str = '',
                                                              transparent: bool = False)
```

Bases: [BaseRenderer](#)

Class for rendering the Effect and only the Effect

render_effect_frame(frame_number: int)

We only need to render the effect, so we just call the effects render frame method to update the effect buffer

run(end_message: bool = True)

Generate the next effect frame and sync it with the back / front buffer

3.6 bruhanimate.bruhrenderer.focus_renderer module

```
class bruhanimate.bruhrenderer.focus_renderer.FocusRenderer(screen, frames, time, img,
                                                         effect_type='static', background='',
                                                         transparent=False, start_frame=0,
                                                         reverse=False, start_reverse=None,
                                                         loop=True)
```

Bases: [BaseRenderer](#)

A Renderer that takes an image and randomly spreads the characters around the screen. The characters are then pulled to the middle of the screen

render_img_frame(frame_number)

Renders the next image frame into the image buffer

solved(end_state)

Function that determines if the image has been moved back to its original shape

update_reverse(reverse, start_reverse)

Function to update whether or not to reverse the Focus :param reverse: True / False

update_start_frame(frame_number)

Updates the frame at which the Focus Effect should start :param frame_number: Frame to start

3.7 bruhanimate.bruhrenderer.pan_renderer module

```
class bruhanimate.bruhrenderer.pan_renderer.PanRenderer(screen: Screen, img: list[str], frames: int,
                                                         time: float, effect_type: str = 'static',
                                                         background: str = '', transparent: bool =
                                                         False, direction: str = 'h', shift_rate: int =
                                                         1, loop: bool = False)
```

Bases: [BaseRenderer](#)

A renderer to pan an image across the screen. Update the image_buffer only.

render_horizontal_frame(frame_number)

Renders the next image frame for a horizontal pan

render_img_frame(frame_number)

Renders out the next frame of the pan animation, if there is no image passed into the renderer then the background is rendered on it's own

3.8 Module contents

```
class bruhanimate.bruhrenderer.BackgroundColorRenderer(screen, frames, time, img, on_color_code,
                                                         effect_type='static', background='',
                                                         transparent=False)
```

Bases: [BaseRenderer](#)

render_img_frame(frame_number)

```
class bruhanimate.bruhrenderer.BaseRenderer(screen: Screen, frames: int = 100, time: float = 0.1,  
                                           effect_type: str = 'static', background: str = '',  
                                           transparent: bool = False, collision: bool = False)
```

Bases: `object`

Defines the base methods, abstract methods, and base attributes for the render class, is an Effect Only Renderer

push_front_to_screen()

Pushes changes between the back_buffer and front_buffer and applies them to the screen.

Parameters

None – This method does not take any parameters.

Return None

This method does not return anything.

render_exit()

Renders out the exit prompt to the screen.

abstract render_frame()

To be defined by each renderer

run(end_message=True)

Updates the image_buffer and effect_buffer. Then the image_buffer is applied over top the effect_buffer and stored into the back_buffer. After the front_buffer is rendered to the screen, the front_buffer is synced with the back_buffer. Why? So the effect and image, and there associated calculations can be done independently.

update_collision(collision)

Method for updating the collision for the rain effect

update_exit_stats(msg1=None, msg2=None, wipe=None, x_loc=None, y_loc=None, centered=False)

Set the exit messages for when the animation finishes :param msg1: primary message :param msg2: secondary message :param wipe: whether to clear the buffer :param x_loc: where to put the message along the xaxis :param y_loc: where to put the message along the yaxis :param centered: whether or not the message should be centered

update_smart_transparent(smart_transparent)

Enable / Disable the smart transparency effect :param smart_transparent: True / False

```
class bruhanimate.bruhrenderer.CenterRenderer(screen: Screen, img: list[str], frames: int = 100, time:  
                                              float = 0.1, effect_type: str = 'static', background: str = '',  
                                              transparent: bool = False)
```

Bases: [BaseRenderer](#)

A renderer to load an image in the center of the screen. Updates the image_buffer only

render_img_frame(frame_number)

Renders out the image to the center of the screen, if there is no image passed into the renderer then the background is rendered on it's own

```
class bruhanimate.bruhrenderer.EffectRenderer(screen: Screen, frames: int = 100, time: float = 0.1,  
                                              effect_type: str = 'static', background: str = '',  
                                              transparent: bool = False)
```

Bases: [BaseRenderer](#)

Class for rendering the Effect and only the Effect

render_effect_frame(frame_number: int)

We only need to render the effect, so we just call the effects render frame method to update the effect buffer

run(end_message: bool = True)

Generate the next effect frame and sync it with the back / front buffer

```
class bruhanimate.bruhrenderer.FocusRenderer(screen, frames, time, img, effect_type='static',
                                             background=' ', transparent=False, start_frame=0,
                                             reverse=False, start_reverse=None, loop=True)
```

Bases: [BaseRenderer](#)

A Renderer that takes an image and randomly spreads the characters around the screen. The characters are then pulled to the middle of the screen

render_img_frame(frame_number)

Renders the next image frame into the image buffer

solved(end_state)

Function that determines if the image has been moved back to its original shape

update_reverse(reverse, start_reverse)

Function to update whether or not to reverse the Focus :param reverse: True / False

update_start_frame(frame_number)

Updates the frame at which the Focus Effect should start :param frame_number: Frame to start

```
class bruhanimate.bruhrenderer.PanRenderer(screen: Screen, img: list[str], frames: int, time: float,
                                             effect_type: str = 'static', background: str = ' ', transparent:
                                             bool = False, direction: str = 'h', shift_rate: int = 1, loop:
                                             bool = False)
```

Bases: [BaseRenderer](#)

A renderer to pan an image across the screen. Update the image_buffer only.

render_horizontal_frame(frame_number)

Renders the next image frame for a horizontal pan

render_img_frame(frame_number)

Renders out the next frame of the pan animation, if there is no image passed into the renderer then the background is rendered on it's own

PYTHON MODULE INDEX

b

- `bruhanimate.bruheffect`, 19
- `bruhanimate.bruheffect.audio_effect`, 13
- `bruhanimate.bruheffect.base_effect`, 13
- `bruhanimate.bruheffect.chatbot_effect`, 14
- `bruhanimate.bruheffect.draw_lines_effect`, 15
- `bruhanimate.bruheffect.game_of_life_effect`, 15
- `bruhanimate.bruheffect.matrix_effect`, 16
- `bruhanimate.bruheffect.noise_effect`, 16
- `bruhanimate.bruheffect.offset_effect`, 17
- `bruhanimate.bruheffect.plasma_effect`, 17
- `bruhanimate.bruheffect.rain_effect`, 18
- `bruhanimate.bruheffect.snow_effect`, 18
- `bruhanimate.bruheffect.star_effect`, 19
- `bruhanimate.bruheffect.static_effect`, 19
- `bruhanimate.bruheffect.twinkle_effect`, 19
- `bruhanimate.bruhrenderer`, 27
- `bruhanimate.bruhrenderer.background_color_renderer`, 25
- `bruhanimate.bruhrenderer.base_renderer`, 25
- `bruhanimate.bruhrenderer.center_renderer`, 26
- `bruhanimate.bruhrenderer.effect_renderer`, 26
- `bruhanimate.bruhrenderer.focus_renderer`, 27
- `bruhanimate.bruhrenderer.pan_renderer`, 27
- `bruhanimate.bruhutil`, 7
- `bruhanimate.bruhutil.bruhffer`, 3
- `bruhanimate.bruhutil.bruhscreen`, 4
- `bruhanimate.bruhutil.images`, 7
- `bruhanimate.bruhutil.utils`, 7

A

`add_line()` (*bruhanimate.bruheffect.draw_lines_effect.DrawLinesEffect* method), 15

`add_line()` (*bruhanimate.bruheffect.DrawLinesEffect* method), 21

`AudioEffect` (class in *bruhanimate.bruheffect*), 19

`AudioEffect` (class in *bruhanimate.bruheffect.audio_effect*), 13

B

`BackgroundColorRenderer` (class in *bruhanimate.bruhrenderer*), 27

`BackgroundColorRenderer` (class in *bruhanimate.bruhrenderer.background_color_renderer*), 25

`BaseEffect` (class in *bruhanimate.bruheffect*), 20

`BaseEffect` (class in *bruhanimate.bruheffect.base_effect*), 13

`BaseRenderer` (class in *bruhanimate.bruhrenderer*), 27

`BaseRenderer` (class in *bruhanimate.bruhrenderer.base_renderer*), 25

bruhanimate.bruheffect
module, 19

bruhanimate.bruheffect.audio_effect
module, 13

bruhanimate.bruheffect.base_effect
module, 13

bruhanimate.bruheffect.chatbot_effect
module, 14

bruhanimate.bruheffect.draw_lines_effect
module, 15

bruhanimate.bruheffect.game_of_life_effect
module, 15

bruhanimate.bruheffect.matrix_effect
module, 16

bruhanimate.bruheffect.noise_effect
module, 16

bruhanimate.bruheffect.offset_effect
module, 17

bruhanimate.bruheffect.plasma_effect
module, 17

bruhanimate.bruheffect.rain_effect

module, 18

bruhanimate.bruheffect.snow_effect
module, 18

bruhanimate.bruheffect.star_effect
module, 19

bruhanimate.bruheffect.static_effect
module, 19

bruhanimate.bruheffect.twinkle_effect
module, 19

bruhanimate.bruhrenderer
module, 27

bruhanimate.bruhrenderer.background_color_renderer
module, 25

bruhanimate.bruhrenderer.base_renderer
module, 25

bruhanimate.bruhrenderer.center_renderer
module, 26

bruhanimate.bruhrenderer.effect_renderer
module, 26

bruhanimate.bruhrenderer.focus_renderer
module, 27

bruhanimate.bruhrenderer.pan_renderer
module, 27

bruhanimate.bruhutil
module, 7

bruhanimate.bruhutil.bruhffer
module, 3

bruhanimate.bruhutil.bruhscreen
module, 4

bruhanimate.bruhutil.images
module, 7

bruhanimate.bruhutil.utils
module, 7

`Buffer` (class in *bruhanimate.bruhutil*), 7

`Buffer` (class in *bruhanimate.bruhutil.bruhffer*), 3

C

`CenterRenderer` (class in *bruhanimate.bruhrenderer*), 28

`CenterRenderer` (class in *bruhanimate.bruhrenderer.center_renderer*), 26

- `chat()` (*bruhanimate.bruheffect.chatbot_effect.OllamaApiClient* method), 15
- `chat()` (*bruhanimate.bruheffect.chatbot_effect.OpenAiClient* method), 15
- `ChatbotEffect` (class in *bruhanimate.bruheffect*), 20
- `ChatbotEffect` (class in *bruhanimate.bruheffect.chatbot_effect*), 14
- `clear()` (*bruhanimate.bruhutil.bruhscreen.Screen* method), 6
- `clear()` (*bruhanimate.bruhutil.Screen* method), 10
- `clear_buffer()` (*bruhanimate.bruhutil.bruhfffer.Buffer* method), 3
- `clear_buffer()` (*bruhanimate.bruhutil.Buffer* method), 7
- `close()` (*bruhanimate.bruhutil.bruhscreen.Screen* method), 6
- `close()` (*bruhanimate.bruhutil.Screen* method), 10
- `copy()` (*bruhanimate.bruheffect.twinkle_effect.TWINKLE_SPEC* method), 19
- `copy()` (*bruhanimate.bruheffect.TWINKLE_SPEC* method), 24
- ## D
- `DrawLinesEffect` (class in *bruhanimate.bruheffect*), 21
- `DrawLinesEffect` (class in *bruhanimate.bruheffect.draw_lines_effect*), 15

E

`EffectRenderer` (class in *bruhanimate.bruhrenderer*), 28

`EffectRenderer` (class in *bruhanimate.bruhrenderer.effect_renderer*), 26

`evenly_distribute_original_values()` (*bruhanimate.bruheffect.audio_effect.AudioEffect* method), 13

`evenly_distribute_original_values()` (*bruhanimate.bruheffect.AudioEffect* method), 19

F

`FocusRenderer` (class in *bruhanimate.bruhrenderer*), 29

`FocusRenderer` (class in *bruhanimate.bruhrenderer.focus_renderer*), 27

`func()` (*bruhanimate.bruheffect.plasma_effect.PlasmaEffect* method), 17

`func()` (*bruhanimate.bruheffect.PlasmaEffect* method), 22

G

`GameOfLifeEffect` (class in *bruhanimate.bruheffect*), 21

`GameOfLifeEffect` (class in *bruhanimate.bruheffect.game_of_life_effect*), 15

`generate()` (*bruhanimate.bruheffect.chatbot_effect.GradientNoise* method), 14

`generate()` (*bruhanimate.bruheffect.chatbot_effect.StringStreamer* method), 15

`generate()` (*bruhanimate.bruheffect.GradientNoise* method), 21

`generate()` (*bruhanimate.bruheffect.StringStreamer* method), 24

`generate_even_ranges()` (*bruhanimate.bruheffect.audio_effect.AudioEffect* method), 13

`generate_even_ranges()` (*bruhanimate.bruheffect.AudioEffect* method), 19

`get_buffer_changes()` (*bruhanimate.bruhutil.bruhfffer.Buffer* method), 3

`get_buffer_changes()` (*bruhanimate.bruhutil.Buffer* method), 7

`get_char()` (*bruhanimate.bruhutil.bruhfffer.Buffer* method), 3

`get_char()` (*bruhanimate.bruhutil.Buffer* method), 7

`get_event()` (*bruhanimate.bruhutil.bruhscreen.Screen* method), 6

`get_event()` (*bruhanimate.bruhutil.Screen* method), 10

`get_gradient()` (*bruhanimate.bruheffect.matrix_effect.MatrixEffect* method), 16

`get_gradient()` (*bruhanimate.bruheffect.MatrixEffect* method), 21

`get_image()` (in module *bruhanimate.bruhutil*), 10

`get_image()` (in module *bruhanimate.bruhutil.images*), 7

`get_points()` (*bruhanimate.bruheffect.draw_lines_effect.Line* method), 15

`get_points()` (*bruhanimate.bruheffect.Line* method), 21

`grab_slice()` (*bruhanimate.bruhutil.bruhfffer.Buffer* method), 3

`grab_slice()` (*bruhanimate.bruhutil.Buffer* method), 7

`GradientNoise` (class in *bruhanimate.bruheffect*), 21

`GradientNoise` (class in *bruhanimate.bruheffect.chatbot_effect*), 14

H

`has_resized()` (*bruhanimate.bruhutil.bruhscreen.Screen* method), 6

`has_resized()` (*bruhanimate.bruhutil.Screen* method), 10

`height()` (*bruhanimate.bruhutil.bruhfffer.Buffer* method), 3

`height()` (*bruhanimate.bruhutil.Buffer* method), 8

K

`KeyNoise` (class in *bruhanimate.bruheffect*), 21

`Key` (class in *bruhanimate.bruheffect.chatbot_effect*), 14

[illegible]

KEY_F9 (*bruhanimate.bruhutil.Screen* attribute), 9
KEY_HOME (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 5
KEY_HOME (*bruhanimate.bruhutil.Screen* attribute), 9
KEY_INSERT (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 5
KEY_INSERT (*bruhanimate.bruhutil.Screen* attribute), 9
KEY_LEFT (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 5
KEY_LEFT (*bruhanimate.bruhutil.Screen* attribute), 9
KEY_MENU (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_MENU (*bruhanimate.bruhutil.Screen* attribute), 9
KEY_MULTIPLY (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_MULTIPLY (*bruhanimate.bruhutil.Screen* attribute), 9
KEY_NUM_LOCK (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUM_LOCK (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_NUMPAD0 (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUMPAD0 (*bruhanimate.bruhutil.Screen* attribute), 9
KEY_NUMPAD1 (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUMPAD1 (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_NUMPAD2 (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUMPAD2 (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_NUMPAD3 (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUMPAD3 (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_NUMPAD4 (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUMPAD4 (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_NUMPAD5 (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUMPAD5 (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_NUMPAD6 (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUMPAD6 (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_NUMPAD7 (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUMPAD7 (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_NUMPAD8 (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUMPAD8 (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_NUMPAD9 (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_NUMPAD9 (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_PAGE_DOWN (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_PAGE_DOWN (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_PAGE_UP (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_PAGE_UP (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_PRINT_SCREEN (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_PRINT_SCREEN (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_RIGHT (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_RIGHT (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_SCROLL_LOCK (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_SCROLL_LOCK (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_SHIFT (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_SHIFT (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_SUBTRACT (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_SUBTRACT (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_TAB (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_TAB (*bruhanimate.bruhutil.Screen* attribute), 10
KEY_UP (*bruhanimate.bruhutil.bruhscreen.Screen* attribute), 6
KEY_UP (*bruhanimate.bruhutil.Screen* attribute), 10

L
Line (class in *bruhanimate.bruheffect*), 21
Line (class in *bruhanimate.bruheffect.draw_lines_effect*), 15
Loading (class in *bruhanimate.bruheffect*), 21
Loading (class in *bruhanimate.bruheffect.chatbot_effect*), 14

M

[map_bands_to_range\(\)](#) (*bruhanimate.bruheffect.audio_effect.AudioEffect method*), 13
[map_bands_to_range\(\)](#) (*bruhanimate.bruheffect.AudioEffect method*), 20
[mark_done\(\)](#) (*bruhanimate.bruheffect.chatbot_effect.GradientNoise method*), 14
[mark_done\(\)](#) (*bruhanimate.bruheffect.chatbot_effect.Loading method*), 14
[mark_done\(\)](#) (*bruhanimate.bruheffect.GradientNoise method*), 21
[mark_done\(\)](#) (*bruhanimate.bruheffect.Loading method*), 21
[MatrixEffect](#) (*class in bruhanimate.bruheffect*), 21
[MatrixEffect](#) (*class in bruhanimate.bruheffect.matrix_effect*), 16
[module](#)
[bruhanimate.bruheffect](#), 19
[bruhanimate.bruheffect.audio_effect](#), 13
[bruhanimate.bruheffect.base_effect](#), 13
[bruhanimate.bruheffect.chatbot_effect](#), 14
[bruhanimate.bruheffect.draw_lines_effect](#), 15
[bruhanimate.bruheffect.game_of_life_effect](#), 15
[bruhanimate.bruheffect.matrix_effect](#), 16
[bruhanimate.bruheffect.noise_effect](#), 16
[bruhanimate.bruheffect.offset_effect](#), 17
[bruhanimate.bruheffect.plasma_effect](#), 17
[bruhanimate.bruheffect.rain_effect](#), 18
[bruhanimate.bruheffect.snow_effect](#), 18
[bruhanimate.bruheffect.star_effect](#), 19
[bruhanimate.bruheffect.static_effect](#), 19
[bruhanimate.bruheffect.twinkle_effect](#), 19
[bruhanimate.bruhrenderer](#), 27
[bruhanimate.bruhrenderer.background_color_renderer](#), 25
[bruhanimate.bruhrenderer.base_renderer](#), 25
[bruhanimate.bruhrenderer.center_renderer](#), 26
[bruhanimate.bruhrenderer.effect_renderer](#), 26
[bruhanimate.bruhrenderer.focus_renderer](#), 27
[bruhanimate.bruhrenderer.pan_renderer](#), 27
[bruhanimate.bruhutil](#), 7
[bruhanimate.bruhutil.bruhffer](#), 3
[bruhanimate.bruhutil.bruhscreen](#), 4
[bruhanimate.bruhutil.images](#), 7
[bruhanimate.bruhutil.utils](#), 7

N

[next\(\)](#) (*bruhanimate.bruheffect.twinkle_effect.TWINKLE_SPEC method*), 19
[next\(\)](#) (*bruhanimate.bruheffect.TWINKLE_SPEC method*), 24
[NoiseEffect](#) (*class in bruhanimate.bruheffect*), 22
[NoiseEffect](#) (*class in bruhanimate.bruheffect.noise_effect*), 16

O

[OffsetEffect](#) (*class in bruhanimate.bruheffect*), 22
[OffsetEffect](#) (*class in bruhanimate.bruheffect.offset_effect*), 17
[OllamaApiCaller](#) (*class in bruhanimate.bruheffect.chatbot_effect*), 15
[open\(\)](#) (*bruhanimate.bruhutil.bruhscreen.Screen class method*), 6
[open\(\)](#) (*bruhanimate.bruhutil.Screen class method*), 10
[OpenAiCaller](#) (*class in bruhanimate.bruheffect.chatbot_effect*), 15

P

[PanRenderer](#) (*class in bruhanimate.bruhrenderer*), 29
[PanRenderer](#) (*class in bruhanimate.bruhrenderer.pan_renderer*), 27
[place_all_keys\(\)](#) (*bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method*), 14
[place_all_keys\(\)](#) (*bruhanimate.bruheffect.ChatbotEffect method*), 20
[PlasmaEffect](#) (*class in bruhanimate.bruheffect*), 22
[PlasmaEffect](#) (*class in bruhanimate.bruheffect.plasma_effect*), 17
[print_at\(\)](#) (*bruhanimate.bruhutil.bruhscreen.Screen method*), 6
[print_at\(\)](#) (*bruhanimate.bruhutil.Screen method*), 10
[print_center\(\)](#) (*bruhanimate.bruhutil.bruhscreen.Screen method*),
[print_center\(\)](#) (*bruhanimate.bruhutil.Screen method*), 10
[process_audio\(\)](#) (*bruhanimate.bruheffect.audio_effect.AudioEffect method*), 13
[process_audio\(\)](#) (*bruhanimate.bruheffect.AudioEffect method*), 20
[push_front_to_screen\(\)](#) (*bruhanimate.bruhrenderer.base_renderer.BaseRenderer method*), 25
[push_front_to_screen\(\)](#) (*bruhanimate.bruhrenderer.BaseRenderer method*), 28
[put_at\(\)](#) (*bruhanimate.bruhutil.bruhffer.Buffer method*), 3

put_at() (*bruhanimate.bruhutil.Buffer* method), 8
 put_at_center() (*bruhanimate.bruhutil.bruhfffer.Buffer* method), 3
 put_at_center() (*bruhanimate.bruhutil.Buffer* method), 8
 put_char() (*bruhanimate.bruhutil.bruhfffer.Buffer* method), 4
 put_char() (*bruhanimate.bruhutil.Buffer* method), 8

R

RainEffect (class in *bruhanimate.bruheffect*), 23
 RainEffect (class in *bruhanimate.bruheffect.rain_effect*), 18
 render_effect_frame() (*bruhanimate.bruhrenderer.effect_renderer.EffectRenderer* method), 26
 render_effect_frame() (*bruhanimate.bruhrenderer.EffectRenderer* method), 28
 render_exit() (*bruhanimate.bruhrenderer.base_renderer.BaseRenderer* method), 25
 render_exit() (*bruhanimate.bruhrenderer.BaseRenderer* method), 28
 render_frame() (*bruhanimate.bruheffect.audio_effect.AudioEffect* method), 13
 render_frame() (*bruhanimate.bruheffect.AudioEffect* method), 20
 render_frame() (*bruhanimate.bruheffect.base_effect.BaseEffect* method), 13
 render_frame() (*bruhanimate.bruheffect.BaseEffect* method), 20
 render_frame() (*bruhanimate.bruheffect.chatbot_effect.ChatbotEffect* method), 14
 render_frame() (*bruhanimate.bruheffect.ChatbotEffect* method), 20
 render_frame() (*bruhanimate.bruheffect.draw_lines_effect.DrawLinesEffect* method), 15
 render_frame() (*bruhanimate.bruheffect.DrawLinesEffect* method), 21
 render_frame() (*bruhanimate.bruheffect.game_of_life_effect.GameOfLifeEffect* method), 15
 render_frame() (*bruhanimate.bruheffect.GameOfLifeEffect* method), 21
 render_frame() (*bruhanimate.bruheffect.matrix_effect.MatrixEffect*

method), 16
 render_frame() (*bruhanimate.bruheffect.MatrixEffect* method), 22
 render_frame() (*bruhanimate.bruheffect.noise_effect.NoiseEffect* method), 16
 render_frame() (*bruhanimate.bruheffect.NoiseEffect* method), 22
 render_frame() (*bruhanimate.bruheffect.offset_effect.OffsetEffect* method), 17
 render_frame() (*bruhanimate.bruheffect.OffsetEffect* method), 22
 render_frame() (*bruhanimate.bruheffect.plasma_effect.PlasmaEffect* method), 17
 render_frame() (*bruhanimate.bruheffect.PlasmaEffect* method), 22
 render_frame() (*bruhanimate.bruheffect.rain_effect.RainEffect* method), 18
 render_frame() (*bruhanimate.bruheffect.RainEffect* method), 23
 render_frame() (*bruhanimate.bruheffect.snow_effect.SnowEffect* method), 18
 render_frame() (*bruhanimate.bruheffect.SnowEffect* method), 23
 render_frame() (*bruhanimate.bruheffect.star_effect.StarEffect* method), 19
 render_frame() (*bruhanimate.bruheffect.StarEffect* method), 24
 render_frame() (*bruhanimate.bruheffect.static_effect.StaticEffect* method), 19
 render_frame() (*bruhanimate.bruheffect.StaticEffect* method), 24
 render_frame() (*bruhanimate.bruheffect.twinkle_effect.TwinkleEffect* method), 19
 render_frame() (*bruhanimate.bruheffect.TwinkleEffect* method), 24
 render_frame() (*bruhanimate.bruhrenderer.base_renderer.BaseRenderer* method), 25
 render_frame() (*bruhanimate.bruhrenderer.BaseRenderer* method), 28
 render_horizontal_frame() (*bruhanimate.bruhrenderer.pan_renderer.PanRenderer* method), 27
 render_horizontal_frame() (*bruhanimate.bruhrenderer.PanRenderer* method),

29		set_audio_properties()	(bruhanimate.bruheffect.AudioEffect method), 20
render_img_frame()	(bruhanimate.bruhrenderer.background_color_renderer.BackgroundColorRenderer method), 25	set_avatar_properties()	(bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14
render_img_frame()	(bruhanimate.bruhrenderer.BackgroundColorRenderer method), 27	set_avatar_properties()	(bruhanimate.bruheffect.ChatbotEffect method), 20
render_img_frame()	(bruhanimate.bruhrenderer.center_renderer.CenterRenderer method), 26	set_chatbot_blink_halt()	(bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14
render_img_frame()	(bruhanimate.bruhrenderer.CenterRenderer method), 28	set_chatbot_blink_halt()	(bruhanimate.bruheffect.ChatbotEffect method), 20
render_img_frame()	(bruhanimate.bruhrenderer.focus_renderer.FocusRenderer method), 27	set_chatbot_cursor_colors()	(bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14
render_img_frame()	(bruhanimate.bruhrenderer.FocusRenderer method), 29	set_chatbot_cursor_colors()	(bruhanimate.bruheffect.ChatbotEffect method), 20
render_img_frame()	(bruhanimate.bruhrenderer.pan_renderer.PanRenderer method), 27	set_chatbot_print_halt()	(bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14
render_img_frame()	(bruhanimate.bruhrenderer.PanRenderer method), 29	set_chatbot_print_halt()	(bruhanimate.bruheffect.ChatbotEffect method), 20
run()	(bruhanimate.bruhrenderer.base_renderer.BaseRenderer method), 26	set_chatbot_properties()	(bruhanimate.bruheffect.ChatbotEffect method), 20
run()	(bruhanimate.bruhrenderer.BaseRenderer method), 28	set_chatbot_stats()	(bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14
run()	(bruhanimate.bruhrenderer.effect_renderer.EffectRenderer method), 26	set_chatbot_stats()	(bruhanimate.bruheffect.ChatbotEffect method), 20
run()	(bruhanimate.bruhrenderer.EffectRenderer method), 29	set_chatbot_text_gradient()	(bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14
S		set_chatbot_text_gradient()	(bruhanimate.bruheffect.ChatbotEffect method), 20
Screen	(class in bruhanimate.bruhutil), 8	set_chatbot_user_colors()	(bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14
Screen	(class in bruhanimate.bruhutil.bruhscreen), 4	set_chatbot_user_colors()	(bruhanimate.bruheffect.ChatbotEffect method), 20
scroll()	(bruhanimate.bruhutil.bruhbuffer.Buffer method), 4	set_divider_flag()	(bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14
scroll()	(bruhanimate.bruhutil.Buffer method), 8	set_divider_flag()	(bruhanimate.bruheffect.ChatbotEffect method), 20
scroll_keys()	(bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14	set_gradient_noise_halts()	(bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14
scroll_keys()	(bruhanimate.bruheffect.ChatbotEffect method), 20	set_gradient_noise_halts()	(bruhanimate.bruheffect.ChatbotEffect method), 20
set_audio_gradient()	(bruhanimate.bruheffect.audio_effect.AudioEffect method), 13	set_matrix_gradient()	(bruhanimate.bruheffect.matrix_effect.MatrixEffect method), 13
set_audio_gradient()	(bruhanimate.bruheffect.AudioEffect method), 20		
set_audio_properties()	(bruhanimate.bruheffect.audio_effect.AudioEffect method), 13		

- method), 16
- set_matrix_gradient() (bruhanimate.bruheffect.MatrixEffect method), 22
- set_matrix_properties() (bruhanimate.bruheffect.matrix_effect.MatrixEffect method), 16
- set_matrix_properties() (bruhanimate.bruheffect.MatrixEffect method), 22
- set_orientation() (bruhanimate.bruheffect.audio_effect.AudioEffect method), 13
- set_orientation() (bruhanimate.bruheffect.AudioEffect method), 20
- set_second_effect() (bruhanimate.bruheffect.chatbot_effect.ChatbotEffect method), 14
- set_second_effect() (bruhanimate.bruheffect.ChatbotEffect method), 21
- set_title() (bruhanimate.bruhutil.bruhscreen.Screen method), 6
- set_title() (bruhanimate.bruhutil.Screen method), 10
- shift() (bruhanimate.bruhutil.bruhfffer.Buffer method), 4
- shift() (bruhanimate.bruhutil.Buffer method), 8
- shift_line() (bruhanimate.bruhutil.bruhfffer.Buffer method), 4
- shift_line() (bruhanimate.bruhutil.Buffer method), 8
- show() (bruhanimate.bruhutil.bruhscreen.Screen class method), 7
- show() (bruhanimate.bruhutil.Screen class method), 10
- show_info() (bruhanimate.bruheffect.snow_effect.SnowEffect method), 18
- show_info() (bruhanimate.bruheffect.SnowEffect method), 23
- shuffle_plasma_values() (bruhanimate.bruheffect.plasma_effect.PlasmaEffect method), 17
- shuffle_plasma_values() (bruhanimate.bruheffect.PlasmaEffect method), 22
- sleep() (in module bruhanimate.bruhutil), 11
- sleep() (in module bruhanimate.bruhutil.utils), 7
- SnowEffect (class in bruhanimate.bruheffect), 23
- SnowEffect (class in bruhanimate.bruheffect.snow_effect), 18
- solved() (bruhanimate.bruhrenderer.focus_renderer.FocusRenderer method), 27
- solved() (bruhanimate.bruhrenderer.FocusRenderer method), 29
- StarEffect (class in bruhanimate.bruheffect), 24
- StarEffect (class in bruhanimate.bruheffect.star_effect), 19
- StaticEffect (class in bruhanimate.bruheffect), 24
- StaticEffect (class in bruhanimate.bruheffect.static_effect), 19
- StringStreamer (class in bruhanimate.bruheffect), 24
- StringStreamer (class in bruhanimate.bruheffect.chatbot_effect), 15
- sync_over_top() (bruhanimate.bruhutil.bruhfffer.Buffer method), 4
- sync_over_top() (bruhanimate.bruhutil.Buffer method), 8
- sync_with() (bruhanimate.bruhutil.bruhfffer.Buffer method), 4
- sync_with() (bruhanimate.bruhutil.Buffer method), 8
- ## T
- text_to_image() (in module bruhanimate.bruhutil), 11
- text_to_image() (in module bruhanimate.bruhutil.images), 7
- TWINKLE_SPEC (class in bruhanimate.bruheffect), 24
- TWINKLE_SPEC (class in bruhanimate.bruheffect.twinkle_effect), 19
- TwinkleEffect (class in bruhanimate.bruheffect), 24
- TwinkleEffect (class in bruhanimate.bruheffect.twinkle_effect), 19
- ## U
- update() (bruhanimate.bruheffect.chatbot_effect.Loading method), 15
- update() (bruhanimate.bruheffect.Loading method), 21
- update_background() (bruhanimate.bruheffect.plasma_effect.PlasmaEffect method), 17
- update_background() (bruhanimate.bruheffect.PlasmaEffect method), 22
- update_background() (bruhanimate.bruheffect.star_effect.StarEffect method), 19
- update_background() (bruhanimate.bruheffect.StarEffect method), 24
- update_collision() (bruhanimate.bruheffect.rain_effect.RainEffect method), 18
- update_collision() (bruhanimate.bruheffect.RainEffect method), 23
- update_collision() (bruhanimate.bruheffect.snow_effect.SnowEffect method), 18
- update_collision() (bruhanimate.bruheffect.SnowEffect method), 23
- update_collision() (bruhanimate.bruhrenderer.base_renderer.BaseRenderer method), 26
- update_collision() (bruhanimate.bruhrenderer.BaseRenderer method),

28			
update_color()	(bruhanimate.bruheffect.noise_effect.NoiseEffect method), 16	update_intensity()	(bruhanimate.bruheffect.noise_effect.NoiseEffect method), 17
update_color()	(bruhanimate.bruheffect.NoiseEffect method), 22	update_intensity()	(bruhanimate.bruheffect.NoiseEffect method), 22
update_color()	(bruhanimate.bruheffect.plasma_effect.PlasmaEffect method), 17	update_intensity()	(bruhanimate.bruheffect.rain_effect.RainEffect method), 18
update_color()	(bruhanimate.bruheffect.PlasmaEffect method), 22	update_intensity()	(bruhanimate.bruheffect.RainEffect method), 23
update_color_properties()	(bruhanimate.bruheffect.plasma_effect.PlasmaEffect method), 17	update_multiplier()	(bruhanimate.bruheffect.rain_effect.RainEffect method), 18
update_color_properties()	(bruhanimate.bruheffect.PlasmaEffect method), 23	update_multiplier()	(bruhanimate.bruheffect.RainEffect method), 23
update_color_type()	(bruhanimate.bruheffect.star_effect.StarEffect method), 19	update_plasma_values()	(bruhanimate.bruheffect.plasma_effect.PlasmaEffect method), 17
update_color_type()	(bruhanimate.bruheffect.StarEffect method), 24	update_plasma_values()	(bruhanimate.bruheffect.PlasmaEffect method), 23
update_decay()	(bruhanimate.bruheffect.game_of_life_effect.GameOfLifeEffect method), 16	update_points()	(bruhanimate.bruheffect.draw_lines_effect.Line method), 15
update_decay()	(bruhanimate.bruheffect.GameOfLifeEffect method), 21	update_points()	(bruhanimate.bruheffect.Line method), 21
update_direction()	(bruhanimate.bruheffect.offset_effect.OffsetEffect method), 17	update_reverse()	(bruhanimate.bruhrenderer.focus_renderer.FocusRenderer method), 27
update_direction()	(bruhanimate.bruheffect.OffsetEffect method), 22	update_reverse()	(bruhanimate.bruhrenderer.FocusRenderer method), 29
update_exit_stats()	(bruhanimate.bruhrenderer.base_renderer.BaseRenderer method), 26	update_rules()	(bruhanimate.bruheffect.game_of_life_effect.GameOfLifeEffect method), 16
update_exit_stats()	(bruhanimate.bruhrenderer.BaseRenderer method), 28	update_rules()	(bruhanimate.bruheffect.GameOfLifeEffect method), 21
update_gradient()	(bruhanimate.bruheffect.chatbot_effect.GradientNoise method), 14	update_smart_transparent()	(bruhanimate.bruhrenderer.base_renderer.BaseRenderer method), 26
update_gradient()	(bruhanimate.bruheffect.GradientNoise method), 21	update_smart_transparent()	(bruhanimate.bruhrenderer.BaseRenderer method), 28
update_grey_scale_size()	(bruhanimate.bruheffect.plasma_effect.PlasmaEffect method), 17	update_start_frame()	(bruhanimate.bruhrenderer.focus_renderer.FocusRenderer method), 27
update_grey_scale_size()	(bruhanimate.bruheffect.PlasmaEffect method), 23	update_start_frame()	(bruhanimate.bruhrenderer.FocusRenderer method), 29
update_info_visibility()	(bruhanimate.bruheffect.plasma_effect.PlasmaEffect method), 17	update_swells()	(bruhanimate.bruheffect.rain_effect.RainEffect method), 18
update_info_visibility()	(bruhanimate.bruheffect.PlasmaEffect method), 23	update_swells()	(bruhanimate.bruheffect.RainEffect method), 23

`update_wind_direction()` (*bruhanimate.bruheffect.rain_effect.RainEffect* method),
18

`update_wind_direction()` (*bruhanimate.bruheffect.RainEffect* method), 23

W

`wait_for_input()` (*bruhanimate.bruhutil.bruhscreen.Screen* method),
7

`wait_for_input()` (*bruhanimate.bruhutil.Screen* method), 10

`width()` (*bruhanimate.bruhutil.bruhfffer.Buffer* method),
4

`width()` (*bruhanimate.bruhutil.Buffer* method), 8