# PiEngine

Created by Elliot Codling

# Table of Contents

# Targeted Hardware

PiEngine is a 2D graphics library enhancer. It provides pre-written functions for a developer to use. These functions help accelerate game development as the user does not need to write their own functions to produce similar results that PiEngine can create. PiEnigne is very flexible in what it can do and aims for many platforms and hardware configurations.

Supported platforms:

- Windows 7 or later
- MacOS X 10.9 or later
- Linux

Recommended platforms:

- Windows 10 or later
- MacOS X 10.13 or later
- Linux Kernel version 4.0 or later

As well as supporting the most popular platforms on the market, this Engine is specifically designed to be used for the Raspberry Pi platforms (Raspberry Pi 3b or later). Hence the name “PiEngine”.

# Future Revisions

Later revisions will incorporate new features from other popular Game Engines. I plan to include the following:

- Multicore rendering
- Tilemap support
- Improved sceneview support (For scrolling games)
- Gamepad support
- Predefined UI elements
- Improved Sound effects
- Particle system
- Saving variables with and without encryption
- Network
- Spritesheet support
- Navigation Mesh (using A* algorithm)
- Shaders

- Lighting

Additionally, PiEngine will be compiled to work on MicroPython intended for the RP2040.
Simply drag and drop your game with little changes to the code and the PiEngine will do the rest!
It is recommended to purchase the Pimoroni PicoSystem so games can be easily transferred with little changes to your code.

# Functions of the Engine

Write here

# Version Changes

Not all versions are backwards compatible with the previous versions of the PiEngine. For a guide of what engines are backwards compatible:

- Anything with an increment of 1.x.x will not be backwards compatible with previous versions.
- Versions with an increment of x.1.x will be backwards compatible with additional features to the engine.
- Values with an increment of x.x.1 will be backwards compatible with no additional features, only minor bug fixes to existing code.

## 0.0.0

Changelog:

1. Added player() class
    a. Added left(player_screenx, player_screenx_2, player_x, player_x_2, vel) -> player_screenx, player_x, player_screenx_2, player_x_2
    b. Added right(player_screenx, player_screenx_2, player_x, player_x_2, vel) -> player_screenx, player_x, player_screenx_2, player_x_2
    c. Added up(player_screeny, player_screeny_2, player_y, player_y_2, vel) -> player_screeny, player_y, player_screeny_2, player_y_2
    d. Added down(player_screeny, player_screeny_2, player_y, player_y_2, vel) -> layer_screeny, player_y, player_screeny_2, player_y_2
    e. Added [left, right, top, bottom]_border(x) -> None

Description:

1.a: The X values will decrease with the velocity once the function is called. This is true for as long as the player_screenx is greater than the left_border value.

1.b: The X values will increase with the velocity once the function is called. This is true for as long as the player_screenx is less than the right_border value.

1.c: The Y values will decrease with the velocity once the function is called. This is true for as long as the player_screeny is greater than the top_border value.

1.d: The Y values will increase with the velocity once the function is called. This is true for as long as the player_screeny is less than the bottom_border value.

1.e: Defines the [left, right, top, bottom]_border values. This is used to define the area of the screen that the player can move around.

## 0.1.0

Changelog:

1. Added update() class
    a. Added list_debug(display, display_sprite, foreground, clock) -> None
    b. Added debug(debug, x1, x2, y, y2) -> None
    c. Added define(name, w, h) -> None
    d. Added window(display, display_sprite, foreground, clock) -> None

Description:

1.a: Outputs the number of elements in each list provided, and the frame rate of the application to the terminal.

1.b: Globalises player_x, player_x_2, player_y, player_y_2, debug_state and sets the variable debug_state to the value provided in debug.

1.c: Creates a window with the width and height provided. Then renames the window to the specified variable name.

1.d: Renders elements from the lists to the window. Items in the list must have the order -> [texture, [x, y]]

## 1.0.0

Changelog:

1. Added music() class
    a. Added fade_out(volume, end_volume, wait) -> None
2. Updated update() class
    a. Added pygame_debug() -> None
    b. Updated list_debug(display, display_sprite, foreground, text_foreground, clock) -> None
    c. Updated debug(debug, x, y, width, height) -> None
    d. Updated window(display, display_sprite, foreground, text_foreground, clock) -> None
3. Updated player() class
    a. Added collisions(player_x, player_y, player_width, player_height, image_list, v) -> image_collided | None
    b. Updated left(player_screenx, player_x, vel) -> player_screenx, player_x
    c. Updated right(player_screenx, player_x, vel) -> player_screenx, player_x
    d. Updated up(player_screeny, player_y, vel) -> player_screeny, player_y
    e. Updated down(player_screeny, player_y, vel) -> player_screeny, player_y

1.a: Used to fade out music songs or sounds. “volume” and “end volume” take values between 0 - 1. Wait is the number of seconds between the initial volume and the end volume. Large wait values will cause your application to freeze.

2.a:
2.b:
2.c:
2.d:
3.a:

3.b:
3.c:
3.d:
3.e:

## 2.0.0

Changelog:

1. Added properties_object() class
   1. Added __init__(self, name, texture, x, y, width, height, alpha) -> None
2. Added properties_text() class
   1. Added __init__(self, name, texture, x, y) -> None
3. Updated update() class
   1. Removed debug(debug, x, y, width, height) function
   2. Updated define(name, w, h) -> window
   3. Updated window(window, display, display_sprite, foreground, text_foreground, clock, debug) -> None
4. Updated player() class
   1. Updated collisions(player_x, player_y, player_width, player_height, list, index) -> list[index].name | None
   2. Updated [left, right, up, down](player, vel, border) -> player
   3. Removed [left, right, top, bottom]_border(x) function

1.a:
2.a:
3.a:
3.b:
3.c:
4.a:
4.b:
4.c:

## 2.1.0

Changelog:

1. Added properties() class
2. Updated update() class
   1. Updated define(name, w, h) -> None
   2. Updated window(display, display_sprite, foreground, text_foreground, clock, debug) -> None
3. Updated player() class
   1. Updated [left, right, up, down](player, vel) -> player
   2. Added [left, right, top, bottom]_border(x) -> None

1.
2.a:
2.b:
3.a:
3.b:

## 3.0.0

Changelog:

1. Removed properties() class
2. Updated properties_object() class

    a. Updated __init__(self, name, file_location, x, y, width, height, alpha, animationTime = 0, animationStage = 0) - > None
3. Updated properties_text() class
    a. Updated __init__(self, name, text, color, x, y, font_size) -> None
4. Added mouse() class
    a. Added collision(mouseX, mouseY, boxName, foreground) -> bool
5. Updated update() class
    a. Updated define(name, w, h) -> window
    b. Updated window(window, display, display_sprite, foreground, text_foreground, clock, debug) -> None
6. Updated player() class
    a. Updated [left, right, up, down](player, vel, border) -> player
    b. Removed [left, right, top, bottom]_border(x) function

1.
2.a:
3.a:
4.a:
5.a:
5.b:
6.a:
6.b:

## 3.0.1

Changelog:

1. Updated properties_object() class
    a. Updated __init__(self, name, file_location, x, y, width, height, alpha, animationTime = 0, animationStage = 0) -> None

1.a:

## 3.1.0

Changelog:

1. Updated properties_object() class
    a. Updated __init__(self, name, loaded_texture, x, y, width, height, alpha, animationTime = 0, animationStage = 0) -> None
2. Updated properties_text() class
    a. Updated __init__(self, name, text, color, x, y, font_size, snapCentre = False) -> None
3. Renamed update() class -> window() class
    a. Updated window() class -> update class
4. Updated player() class
    a. Updated collisions(player, box_list, index) -> box_list[index].name | None

1.a:
2.a:
3.a:
4.a:

## 4.0.0

Changelog:

1. Updated music() class
    a. Added stop() -> None

2. Updated properties_text() class
    a. Added reload_text(text, color, font_size) -> texture
3. Updated mouse() class
    a. Updated collision(boxName, foreground, mouseX = 0, mouseY = 0, ) -> bool
4. Updated window() class
    a. Added music_debug() class
    b. Updated define(name, w, h, flags=0, vsync=0) -> window
    c. Updated update(window, display, display_sprite, foreground, text_foreground, clock, debug) -> None
5. Added Pygame Module test
6. Added activate_music: bool
7. Added Music Module test

1.a:

2.a:

3.a:

4.a:

4.b:

4.c:

5:

6:

7:

## 4.1.0

Changelog:

1. Added counter() class
    a. Added update() -> None
2. Updated properties_object() class
    a. Added reload_texture(loaded_texture, width, height) -> texture
3. Updated window(0) class
    a. Updated update(window, display = None, display_sprite = None, foreground = None, text_foreground = None, clock = 0, debug = 0) -> None
4. Added camera() class
    a. Added init() -> None
    b. Added moveCamera(xOffSet, yOffSet) -> None
5. Updated player() class
    a. Added animate(self, animationList, frameWait) -> None
    b. Updated [left, right, up, down](player, vel, border) -> bool
6. Added frames: int
7. Added camera initialisation

1.a:

2.a:

3.a:

4.a:

4.b:

5.a:

5.b:

6:

7:

## 5.0.0

Changelog:

1. Updated properties_object() class
    a. Updated __init__(self, name, loaded_texture, x, y, width, height, alpha, rotation = 0) -> None
2. Updated player() class
    a. Added setAngle(self, angle) -> None
    b. Updated animate(self, animationList, frameWait, flipX = 0, flipY = 0) -> None

1.a:
2.a:
2.b:

## 6.0.0

Changelog:

1. Updated properties_object() class
    a. Updated reload_texture(self, loaded_texture, width, height) -> None
2. Updated window() class
    a. Added __init__(self, name, w, h, color = (0, 0, 0), flags=0, vsync=0) -> None
    b. Removed define() function
    c. Added writeDebug_file(file, display, display_sprite, foreground, text_foreground, clock) -> None
    d. Updated update(window, display = None, display_sprite = None, foreground = None, text_foreground = None, clock = 0) -> None
3. Removed camera() class
4. Added canvas() class
    a. Added __init__(self) -> None
5. Renamed player() class to object()
    a. Updated collisions(player, box_list, index) to collision_rect(self, box_list, index) -> None
    b. Added collision_mask(self, box_list) -> object.name | None
6. Removed camera initialisation
7. Added Canvas = canvas()

1.a:
2.a:
2.b:
2.c:
2.d:
3:
4.a:
5.a:
5.b:
6:
7:

## 7.0.0

Changelog:

1. Updated properties_object() class
    a. Updated __init__(self, name, loaded_texture, x, y, width, height, alpha, layer=0, rotation = 0) -> None

2. Updated window() class
    a. Updated writeDebug_file(file, display, clock) -> None
    b. Updated update(window, display = None, clock = 0, layer = 0) -> None
    c. Added delete_layer(display, layer) -> display

1.a:
2.a:
2.b:
2.c:

## 8.0.0

Changelog:

1. Added sfx() class
    a. Added __init__(self, soundDirectory, channel) -> None
    b. Added playSound(self, playIfBusy=False) -> None
2. Updated music() class
    a. Removed fade_out() function
    b. Added __init__(self, track, channel=0) -> None
    c. Added loop(self, shuffle) -> None
    d. Added play(self, trackNumber, playIfBusy=False) -> None
    e. Updated stop(self) -> None
3. Added event() class
    a. Added update() -> events
    b. Added getKeydown(events, pygameEvent) -> bool
    c. Added getKeyUp(events, pygameEvent) -> bool
4. Updated object() class
    a. Updated setAngle(self, angle) -> None
    b. Updated collision_rect() parameters: box_list -> display
    c. Updated collision_mask(self, display, collidableLayers=[]) -> object.id | None
    d. Updated [left, right, up, down](self, vel, border) -> bool
5. Updated properties_object(object) class
    a. Updated __init__ parameters: name, alpha -> id, alpha=False
6. Updated properties_text() class
    a. Updated __init__(self, id, text, color, x, y, font_size, snapCentre = False, layer=0) -> None
    b. Updated reload_text(text, color, font_size) -> None
7. Updated mouse() class
    a. Updated collision() parameters: boxName -> objectId
8. Updated window() class
    a. Removed writeDebug_file() function
    b. Updated update(window, display = None, debug=[layer, clock, color]) -> None
    c. Updated delete_layer(display, layer) -> display
9. Removed canvas() class
10. Added initDebug() function
11. Added run: bool
12. Added directory: string