PyEngine4

Release 4.0.0-indev

LavaPower

Apr 04, 2021

INFORMATIONS

Welcome to documentation of PyEngine4.

PyEngine4 is constantly under development, the documentation is therefore subject to change. Feel free to come back to it as soon as you have a problem.

Note: It is important to remember that PyEngine is an OpenSource project developed by non-professionals. You can also participate with Github.

Summary:

CHAPTER ONE

INTRODUCTION

PyEngine was created by LavaPower.

PyEngine relies on PyGame to work. It was made to be used in 2D games of all types: Platformer, Pong, Casse bricks…

You can find tutorials, examples and documentation of the different classes.

Note: PyEngine is still very young and still very limited.

CHAPTER TWO

DOWNLOAD AND INSTALLATION

- Have Python, PyGame and PyQt installed
- Download and decompress GitHub files (http://github.com/Lycos-Novation/PyEngine4)
- PyEngine is downloaded and installed

CHAPTER THREE

FAQ

3.1 What is PyEngine?

PyEngine is a game engine for creating games 2D videos more easily.

3.2 Why create PyEngine?

To create a video game in python, there is already the very good PyGame.

But when I created my game, I had to create systems (such as the entity system) that are useful for all. So I chose to create PyEngine (which uses PyGame itself) (And then it allows a good training in Python).

3.3 What are the dependencies of PyEngine?

Apart from Python, PyEngine uses PyGame and PyQt. But games built with PyEngine only use PyGame.

3.4 What are the platforms where PyEngine can be used?

If you can use PyGame, PyQt and Python, you can use PyEngine.

3.5 I would like to participate in the development of PyEngine, how do I do it?

Send me a message by Discord (LavaPower / Lyos#2480) to see what you can do or go to GitHub

CHAPTER

FOUR

CHANGELOG

4.1 V 4.0.0 - XX/XX/XX (INDEV)

- Complete rework on PyEngine

CHAPTER

FIVE

PROJECTWINDOW

The first window you will see when you launch PyEngine4 is the ProjectWindow.

This window list all the projects of PyEngine4. You can launch a project with a double click.

Note: A project made with other version of PyEngine will not be launched.

You can also delete an existed project or create a new project with a name, an author and an icon (the last is not obligatory).

CHAPTER SIX

EDITOR

One of principal components of PyEngine is the Editor. He's composed of four other components.

6.1 Viewport

In the center of the screen, there is the viewport.

You can see the current selected scene here.

Warning: The viewport haven't got any logic or game loop.

6.2 ProjectExplorer

At the bottom, there is the project explorer.

You can choose if you want see scripts, prefabs, textures or scenes. If you right click on the right part, you can create new assets or delete an old one.

6.2.1 Scripts

A Script in PyEngine4 is a python file where you create a logic for an object.

On creation, PyEngine4 ask a name for the script. If you double click on it, you will open the script.

6.2.2 Prefabs

Coming Soon

6.2.3 Textures

A texture is a .PNG or .JPG file which can be used for a sprite.

On creation, PyEngine4 ask a name and a path to the file. If you double click on it, you will be able to change the path.

6.2.4 Scenes

A scene is a 2D world where you can put objects. You must have a scene for any game.

On creation, PyEngine4 ask a name for the scene.

6.3 SceneTree

On the left, you can see the Scene Tree.

It shows the objects which are in the current selected scene.

You can create a new game object with selected an already existed game object and make a right click. You will choose a name and the new game object will be the child of the existed.

You can also delete a game object with a right click.

6.4 Inspector

On the right, there is the inspector.

This panel will show the properties and components of the selected game object (or asset in the case of texture).

You can add a new component in a normal game object with the button on the bottom or with right click. You can also delete a component of a normal game object with right click.

CHAPTER

SEVEN

GAME

Class which represents the core of the game.

7.1 Variables

- engine: Utility class to control the game (Engine)
- name: Title of the Game (string)
- width: Width of Window (integer)
- height: Height of Window (integer)
- scenes: List of scenes (list of Scene)
- current_scene: Index of current displayed scene (integer)
- clock: Pygame's clock to control time of Game (Clock)
- screen: Window of Game (Surface)
- is_running: True if game is currently running (boolean)

Note: Changing name, width and height will have any effect.

Warning: Most of these variables mustn't be manipulated. Only current_scene, is_running and getting engine is safe.

7.2 Functions

- stop(): Stop Game
- get_fps(): Return current FPS

CHAPTER

EIGHT

SCENE

Class which represents a world for the game.

8.1 Variables

- bg_color: Color of background (list of four integers : [Red, Green, Blue, Alpha])
- game_objects: List of Game Objects own by the scene (List of GameObject)

Warning: game_objects mustn't be manipulated, use functions.

8.2 Functions

- add_game_objects(game_objects): Adding Game Objects to the scene (game_objects -> List of GameObjects)
- add_game_object(game_object): Add a Game Object to the scene (game_object -> GameObject)
- get_game_object(id): Getting Game Object by id or None if any Game Object is found (id -> integer)

CHAPTER

NINE

GAMEOBJECT

Class which represents an object of the game.

9.1 Variables

- parent: Parent of current object or None (GameObject)
- childs: List of childs game objects of current object (list of GameObject)
- components: List of components of current object (list of Components)
- id_: Id of object (integer)

Warning: Most of these variables mustn't be manipulated. Only id_ and parents are safe on reading.

9.2 Functions

- add_child(child): Add a child to object (child -> GameObject)
- add_component(comp): Add a component to object (comp -> Component)
- get_component(name): Return a component by his name or None if any component is found (name -> string)

CHAPTER

TEN

ENGINE

Utility class which can onlly be used in script.

10.1 Variables

- game: Game object (Game)
- pg_constants: Constants from PyGame (module)
- down_keys: List of keys which are currently down (list)
- down_mousebuttons: List of mouse buttons which are currently down (list)

10.2 Functions

- get_game_object(id): Return game object which have specific id (id -> integer)
- get_current_scene(): Return scene which is currently selected
- stop_game(): Stop game

CHAPTER

ELEVEN

AUTOCOMPONENT

This component create automatic movement and rotation to game object.

Warning: This component use the TransformComponent and it can't work without it.

11.1 Editor

In the editor, you can change the movement and the rotation with differents spins. There is also a checkbox to activate/desactivate the component.

11.2 Script

11.2.1 Variables

- game_object: Object which own this component (GameObject)
- engine: Utility class to control the game (Engine)
- move: Movement of component (list of two integers)
- rotation: Rotation of component in degrees (integers)
- active: True if component is active

11.2.2 Functions

No public function in this component.

CHAPTER

TWELVE

BASICPHYSICCOMPONENT

This component add gravity to game object.

Warning: This component use the TransformComponent and it can't work without it.

12.1 Editor

In the editor, you can change gravity with a spin.

12.2 Script

12.2.1 Variables

- game_object: Object which own this component (GameObject)
- engine: Utility class to control the game (Engine)
- gravity: Current gravity of component (integer)
- max_gravity: Maximum gravity of component (integer)
- grounded: True if object can't be move by gravity (boolean)
- time: Time must be elipsed beform apply gravity (float)

Note: You may not change grounded and time.

Warning: grounded only work with the CollisionComponent

12.2.2 Functions

No public function in this component.

CHAPTER

THIRTEEN

COLLISIONCOMPONENT

This component add collision and collision callback to game object.

Warning: This component use the TransformComponent and a component with a render and it can't work without them.

13.1 Editor

In the editor, you can change if object is solid with a checkbox and specify a callback by giving the name of component and the function will be called.

13.2 Script

13.2.1 Variables

- game_object: Object which own this component (GameObject)
- engine: Utility class to control the game (Engine)
- solid: True if object is solid (boolean)
- callback: Component and its function which will be call (string)

Note: callback use a specific format : "NameComponent - NameFunction". For example: "myScript - collide" is a valid callback.

Note: A function which will be used as a callback must accept two arguments : the object which collide with and the cause of collision.

13.2.2 Functions

- can_go(position, cause=”UNKNOWN”): Return if object can go to a position. (position -> list of two integers, cause -> string)

CHAPTER

FOURTEEN

SPRITECOMPONENT

This component add sprite to game object.

Warning: This component use the TransformComponent and it can't work without it.

14.1 Editor

In the editor, you can change sprite by selecting json file.

14.2 Script

14.2.1 Variables

- game_object: Object which own this component (GameObject)
- engine: Utility class to control the game (Engine)
- sprite: Resource file. Will search in resources folder. (string)
- render: Final render.

Warning: You mustn't change render directly but use update_render function.

14.2.2 Functions

- update_render(): Update render of component

CHAPTER

FIFTEEN

SPRITESHEETCOMPONENT

This component add spritesheet to game object.

Warning: This component use the TransformComponent and it can't work without it.

15.1 Editor

In the editor, you can change spritesheet by selecting json file but also the numbers of sprite and the current index of sprite.

15.2 Script

15.2.1 Variables

- game_object: Object which own this component (GameObject)
- engine: Utility class to control the game (Engine)
- sprite: Resource file. Will search in resources folder. (string)
- sprite_number: Number of sprite in width and height. (list of two integers)
- current_sprite: Index of current displayed sprite (integer)
- render: Final render.

Note: Current_sprite start at 0.

Warning: You mustn't change render directly but use update_render function.

15.2.2 Functions

- update_render(): Update render of component

CHAPTER

SIXTEEN

TEXTCOMPONENT

This component add text to game object.

Warning: This component use the TransformComponent and it can't work without it.

16.1 Editor

In the editor, you can change text and informations from font (name, size, bold, italic, underline, color, antialias).

16.2 Script

16.2.1 Variables

- game_object: Object which own this component (GameObject)
- engine: Utility class to control the game (Engine)
- text: Text will be displayed (string)
- font_name: Name of font used (string)
- font_size: Size of font used (integer)
- font_bold: True if text is in bold (boolean)
- font_italic: True if text is in italic (boolean)
- font_underline: True if text is underline (boolean)
- font_color: Color of font used (list of four integers -> [Red, Green, Blue, Alpha])
- font_antialias: True if antialias is active (boolean)
- transformed_font: Final font
- render: Final render.

Warning: You mustn't change transformed_font directly but use update_font function.

Warning: You mustn't change render directly but use update_render function.

16.2.2 Functions

- rendered_size(text): Return size of text which be rendered with the font of component (text -> string)
- update_font(): Update font of component
- update_render(): Update render of component

CHAPTER

SEVENTEEN

TRANSFORMCOMPONENT

This component add position, rotation and scale to game object.

17.1 Editor

In the editor, you can change position, rotation or scale with differents spins.

17.2 Script

17.2.1 Variables

- game_object: Object which own this component (GameObject)
- engine: Utility class to control the game (Engine)
- rotation : Rotation of object in degrees (integer)
- position : Position X and Y of object (list of two integers)
- scale : Scale X and Y of object (list of two integers)

17.2.2 Functions

No public function in this component.