Pycraft Documentation

Release Pycraft-v9.5.1

Jan 15, 2023

CONTENTS

Pycraft is an OpenGL, open world, video game made entirely with Python. This project is a game to shed some light on OpenGL programming in Python as it is a seldom touched area of Python's vast amount of uses. Feel free to give this project a run, and message us if you have any feedback!

Made with Python 3 64-bit and Microsoft Visual Studio Code.

CHAPTER

ONE

CONTENTS:

1.1 Introduction

1.1.1 About

Pycraft is a 3D open-source, open-world video game made in Python. For a long time attempts to make large 3D games in Python have been ignored, we believe there are two reasons: one; People use Python primarily for data handling and processing and not graphics and, two; there is little to no documentation out there to do anything more than make a 3D rotating cube in Python. Making a 3D game in Python for us hasn't been an easy experience, far from it but we have decided to share my project, complete with tutorials, explanations, articles and code explanations in the hope that 3D game development in Python can be seen as a more easily attainable target, and to fill that gap in documentation. Pycraft then is a trial project, as we learn and experiment on what goes best where and how thing go together, this is why development can sometimes appear to have stopped, because we are learning and testing what we have learned, so hopefully for people in the future it will be an easier experience. Also, don't forget there is more to game development than just graphics, there is AI, sound, physics and all the other GUIs that go with it, and as we learn the quality of the overall program will improve. Pycraft is not going to be the final name of the game, however until something better becomes available, we shall stick to it.

1.1.2 Setup

Installing the project from GitHub (Method 1)

The project will download as a (.zip) compressed file. Please make sure you have the project decompressed before use. Next make sure that any folders and files outside of the 'Pycraft' folder are removed and that the 'Pycraft' file is in the intended place for the file to be run from. This file can be freely moved around, transported between drives, computers and folders in this form!

_Just make sure that if you plan to use the installer that you make sure the file location is correct after you have moved the project, to do this simply remove everything in the 'pycraft/Data_Files/InstallerConfig.json' file and re-load the game, it will try to repair the file and write the new path instead, during this process it may appear that Pycraft has crashed as it will likely bring up an error message, a more user-friendly experience is coming **soon__**

When running the program please make sure you have a minimum of 1GB of free space on the drive and also have Python 3.7 or above installed on your device. This can be found here: (www.python.org/downloads). The version of Python isn't too important in this circumstance however the project has been tested in Python 3.7 and above and is known to work. In addition to all this please make sure you have the following modules installed on your device:

Pygame, Numpy, Pillow, PyAutoGUI, Psutil, PyWaveFront, CPUinfo, Ctypes, ModernGL, ModernGL_Window, GPUtil, Pyrr, PyJoystick, Noise and Matplotlib.

For those not familiar they can be found here: (pypi.org).

You can use the following syntax to install, update and remove these modules:

`pip install <module>` `pip uninstall <module>`

Here is a short video tutorial walk you through all this: (https://youtu.be/DG5YbE-umw0)

Installing the project from GitHub (Method 2)

If you are installing the project from the GitHub releases page or through Source Forge, then this will be relevant for you. After you have selected your preferred file type (it'll be either a compiled (.exe) file or a (.zip) file, those that download the (.zip) file will find the information above more relevant.

If you, however, download the (.exe) type file, then this will be more relevant for you. If you locate the file in your file explorer and double click it, then this will run the project. You do not need Python, or any of the projects required modules, as they come built-in with this method. This method does also not install anything extra to your devise, to remove the project, simply delete the (.exe) file in your file explorer. Please note that it can take a few moments for everything in the (.exe) file to load and initialise, so nothing might not appear to happen at first. Also, you can only run one instance of Pycraft at any time (even if you are using another method).

Installing from PyPi (preferred)

If you are installing the project from PyPi, then you will need an up-to-date build of Python (3.7 or greater ideally) and also permission to install additional files to your device. Then you need to open a command-line interface (or CLI), we recommend Terminal on Apple based devises, and Command Prompt on Windows based machines. You install the latest version of Pycraft, and all its needed files though this command:

```
pip install Python-Pycraft
```

and you can also uninstall the project using the command:

```
pip uninstall Python-Pycraft
```

And now you can run the project as normal. Please note that at present it can be a bit tricky to locate the files that have downloaded, you can import the project into another python file using:

```
import Pycraft
```

Installing using Pipenv

You can alternatively run these commands in the directory containing a file called *Pipfile*:

`pip install pipenv` then: `pipenv install python-pycraft`

And to start the game: `pipenv run python <PATH to 'main.py'>`

1.1.3 Running The Program

When running the program, you will either have a (.exe) file, downloaded from the releases page, or you will have the developer preview, if you have the developer preview, which can be found in the files section of this repository then this is how you run that program.

Now you have the program properly installed hopefully (you'll find out if you haven't promptly!) you need to locate and run the file "main.py" if it crashes on your first run then chances are you haven't installed the program correctly, if it still doesn't work then you can contact us. We do hope however that it works alright for you and you have a pleasant experience. This program has been developed on a Windows 64-bit computer however should run fine on a 32-bit Windows machine (uncompiled) or through MacOS although they remain untested for now.

We recommend creating a shortcut for the "main.py" file too so it's easier to locate.

1.1.4 Credits

With thanks to

![Python](https://img.shields.io/badge/python-3670A0?style=for-the-badge&logo=python&logoColor=ffdd54)
![OpenGL](https://img.shields.io/badge/OpenGL-%23FFFFFF.svg?style=for-the-badge&logo=opengl)
![OpenCV](https://img.shields.io/badge/opencv-%23white.svg?style=for-the-badge&logo=opencv&logoColor=white) ![Blender](https://img.shields.io/badge/blender-%23F5792A.svg?style=for-the-badge&logo=blender&logoColor=white) ![Gimp Gnu Image Manipulation Program](https://img.shields.io/badge/Gimp-657D8B?style=for-the-badge&logo=gimp&logoColor=FFFFFF) ![Inkscape](https://img.shields.io/badge/Inkscape-e0e0e0?style=for-the-badge&logo=inkscape&logoColor=080A13) ![Visual Studio Code](https://img.shields.io/badge/Visual%20Studio%20Code-0078d7.svg?style=for-the-badge&logo=visual-studio-code&logoColor=white) ![Visual Studio](https://img.shields.io/badge/Visual%20Studio-5C2D91.svg?style=for-the-badge&logo=visual-studio&logoColor=white) ![GitHub](https://img.shields.io/badge/github-%23121011.svg?style=for-the-badge&logo=github&logoColor=white) ![Stack Overflow](https://img.shields.io/badge/-Stackoverflow-FE7A16?style=for-the-badge&logo=stack-overflow&logoColor=white) ![NumPy](https://img.shields.io/badge/numpy-%23013243.svg?style=for-the-badge&logo=numpy&logoColor=white) ![Windows](https://img.shields.io/badge/Windows-0078D6?style=for-the-badge&logo=windows&logoColor=white) ![Edge](https://img.shields.io/badge/Edge-0078D7?style=for-the-badge&logo=Microsoft-edge&logoColor=white)

- Tom Jebbo (PycraftDeveloper) @ www.github.com/PycraftDeveloper
- Count of Freshness Traversal @ www.twitter.com/DmitryChunikhin
- Dogukan Demir (demirdogukan) @ www.github.com/demirdogukan
- Henri Post (HenryFBP) @ www.github.com/HenryFBP
- PyPi @ www.pypi.org
- PIL (Pillow or Python Imaging Library) @ www.github.com/python-pillow/Pillow
- Pygame @ www.github.com/pygame/pygame
- Numpy @ www.github.com/numpy/numpy
- PyAutoGUI @ www.github.com/asweigart/pyautogui
- Psutil @ www.github.com/giampaolo/psutil
- PyWaveFront @ www.github.com/pywavefront/PyWavefront
- Py-CPUinfo @ www.github.com/pytorch/cpuinfo
- GPUtil @ www.github.com/anderskm/gputil
- Tabulate @ www.github.com/p-ranav/tabulate
- Moderngl @ www.github.com/moderngl/moderngl
- Moderngl_window @ www.github.com/moderngl/moderngl-window
- PyJoystick @ www.github.com/justengel/pyjoystick
- Noise @ www.github.com/caseman/noise
- Matplotlib @ www.github.com/matplotlib/matplotlib
- FreeSound: - Erokia's "ambient wave compilation" @ www.freesound.org/s/473545
- FreeSound: - Soundholder's "ambient meadow near forest" @ www.freesound.org/s/425368

- FreeSound: - monte32's "Footsteps_6_Dirt_shoe" @ www.freesound.org/people/monte32/sounds/353799
- Freesound: - Straget's 'Thunder' @ www.freesound.org/people/straget/sounds/527664/
- Freesound: - FlatHill's 'Rain and Thunder 4' @ www.freesound.org/people/FlatHill/sounds/237729/
- Freesound: - BlueDelta's 'Heavy Thunder Strike - no Rain - QUADRO' @ - www.freesound.org/people/BlueDelta/sounds/446753/
- Freesound: - Justkiddink's 'Thunder » Dry thunder1' @ www.freesound.org/people/juskiddink/sounds/101933/
- Freesound: - Netaj's 'Thunder' @ www.freesound.org/people/netaj/sounds/193170/
- Freesound: - Nimlos' 'Thunders » Rain Thunder' @ www.freesound.org/people/Nimlos/sounds/359151/
- Freesound: - Kangaroovindaloo's 'Thunder Clap' @ www.freesound.org/people/kangaroovindaloo/sounds/585077/
- Freesound: - Laribum's 'Thunder » thunder_01' @ www.freesound.org/people/laribum/sounds/353025/
- Freesound: - Jmbphilmes's 'Rain » Rain light 2 (rural)' @ www.freesound.org/people/jmbphilmes/sounds/200273/

1.1.5 Uncompiled Pycraft Dependencies

When you're installing the uncompiled Pycraft variant from here you need to install the following 'modules', which can be done through your Control Panel in Windows (First; press <windows key + r> then type "cmd" then run the below syntax) or on Apple systems in Terminal.

` `pip install <module>` `pip uninstall <module>` ` pip is usually installed by default when installing Python with most versions.

- PIL (Pillow or Python Imaging Library) @ www.github.com/python-pillow/Pillow
- Pygame @ www.github.com/pygame/pygame
- Numpy @ www.github.com/numpy/numpy
- PyAutoGUI @ www.github.com/asweigart/pyautogui
- Psutil @ www.github.com/giampaolo/psutil
- PyWaveFront @ www.github.com/pywavefront/PyWavefront
- Py-CPUinfo @ www.github.com/pytorch/cpuinfo
- GPUtil @ www.github.com/anderskm/gputil
- Tabulate @ www.github.com/p-ranav/tabulate
- Moderngl @ www.github.com/moderngl/moderngl
- Moderngl_window @ www.github.com/moderngl/moderngl-window
- PyJoystick @ www.github.com/justengel/pyjoystick
- Noise @ www.github.com/caseman/noise
- Matplotlib @ www.github.com/matplotlib/matplotlib

Disclaimer; unfortunately, lots of these python modules (first and third party) can require some external modules that will be installed during the installing process of the above modules, unfortunately this makes it really difficult to give credit to those modules, if you have any recommendations, please contact me appropriately.

1.1.6 Changes

Pycraft v9.5.1 is now live! Here is a list of all the added features to this minor update:

- Feature: There have been some major changes and improvements to how Pycraft checks to make sure all the resources it needs are present and in the correct location, the process is now faster and it's easier to add more files later on and has been improved to include all the required files in Pycraft (some were not checked before).
- Feature & Performance: The 'noise' module for generating Perlin noise in Pycraft has now been removed, opting for a solution that is faster, more efficient and also supported on newer hardware and more operating systems.
- Bug-Fix: Numerous improvements and other small bug-fixes that were quickly patched in Pycraft v9.5.0 have been properly patched now.
- Feature: Some of the folders used by Pycraft have been renamed to make them more user friendly.
- Feature: Some of the core variables for Pycraft have been renamed to bring them in-line with the PEP8 specification, this is a small part of the gradual transition to supporting most of the PEP8 standards in Pycraft.

Again, feedback would be much appreciated this update was released on; 10/08/2022 (UK date; DD/MM/YYYY). As always, we hope you enjoy this new release and feel free to leave feedback.

1.1.7 Understanding the release notes

This section will hopefully provide additional information on helping to read the release notes. Points detailed after the "Feature" tag are what was focused on in the update and will likely always be present in each update, often this is the most significant area of the update. Points detailed after the "Bug-Fix" tag are likely to be the most frequent, they outline the most major bugs that have been fixed in this update, although they are not the only bugs that have been fixed. Points detailed after the "Performance" tag are used where there have been significant performance improvements to the project. Points detailed after the "Identified-Bugs" tag are bugs that have been identified in the project and that haven't been fixed as of writing the release notes, these are significant issues and will be fixed as soon as possible. Points detailed after the final "Documentation" tag are indicators of significant improvements to the documentation.

1.1.8 Input mapping

This section will be replaced with a dedicated file for keymapping as well as an in-game guide. The controller keys are labelled differently between controllers but have the same mapping in game.

Keyboard

- Use W, A, S, D in game to move around, and use these keys in the map GUI to move that around.
- Use SPACE to jump in game, reset your zoom in the map GUI, start the benchmark section, or press 10 times to enter Devmode.
- Use E in game to access your inventory
- Use R in game to access the map
- Use F11 to toggle full-screen
- Use Q to access a resource value screen
- Use L in game to toggle locking your mouse (forcing it to stay in the window or not)
- Use X to exit Devmode

Mouse

- SCROLL in the map to zoom in/out, or to scroll the settings menu
- LEFT CLICK to select

Controller

- Use the HAT keys (or the 4 buttons typically on the left of the controller in a '+' shape) to navigate between menu options
- Use the JOYSTICKs for camera panning and in game movement
- Use the 'Options' button to enter your inventory
- Use the 'Share' buttons to enter the map
- Use the Y or TRIANGLE button to jump in game or exit a GUI (not in game)
- Use the X or A button to start the benchmark or to reset your view in the map
- Use the X or SQUARE button to zoom in on the map GUI
- Use the O or B button to zoom out on the map

A detailed map of inputs for keyboard and mouse or controller combinations is coming; for now, see the section below, toggling between full-screen is currently not bound to a button on the controller because we will need all the different buttons for **gameplay**

1.1.9 Our Update Policy

New releases will be introduced regularly, it is likely that there will be some form of error or bug, therefore unless you intend to use this project for development and feedback purposes (Thank you all!) we recommend you use the latest stable release; below is how to identify the stable releases.

1.1.10 Version Naming

Pycraft's versions will always now follow the structure; "vA.B.C" * Where "A" is the major revision number. * Where "B" is the minor revsision number. * Where "C" is the patch and developer preview numbers (combined).

Every version of Pycraft as of the 27/10/2022 (DD/MM/YYYY) must feature all 3 values. Updates also now go sequentially, so Pycraft v9.6.4 is newer than Pycraft v9.5.7. If either of the "A" or "B" version numbers is incremented in a release, documentation MUST be suitably updated, in addition Pycraft MUST be released on PyPi, SourceForge and as a release on GitHub.

1.1.11 Releases

All past versions of Pycraft are available under the releases section of Pycraft, this is a new change, but just as before, major releases like Pycraft v0.9 and Pycraft v0.8 will have (.exe) releases, but smaller sub-releases will not, this is in light of a change coming to Pycraft, this should help with the confusion behind releases, and be more accommodating to the installer that's being worked on as a part of Pycraft v9.4.0. This brings me on to another point, all past updates to Pycraft will be located at the releases page (Thats all versions), and the previous section on the home-page with branches will change. The default branch will be the most recent release, then there will be branches for all the sub-releases to Pycraft there too; and the sister program; Pycraft-Insider-Preview will be deprecated and all data moved to relevant places in this repository, this should hopefully cut down on the confusion and make the project more user-friendly.

1.1.12 Other Sources

We now post a roughly monthly article about Pycraft, showing behind the scenes, tips and tricks and additional information, this is shared to both Medium (medium.com/@PycraftDev) and Dev (dev.to/PycraftDev) and builds on the regular posts we share to Twitter (twitter.com/PycraftDev) and Dev (dev.to/PycraftDev).

1.1.13 Final Notices

Thank you greatly for supporting this project simply by running it, we are sorry in advance for any spelling mistakes. The program will be updated frequently and we shall do my best to keep this up to date too. we also want to add that you are welcome to view and change the program and share it with your friends however please may we have some credit, just a name would do and if you find any bugs or errors, please feel free to comment in the comments section any feedback so we can improve my program, it will all be much appreciated and give as much detail as you wish to give out.

1.2 Module Breakdown

1.2.1 Nomenclature and programming techniques

Pycraft maintains a scheme for naming variables and controlling code structure in Pycraft; this section details all the information you will need for understanding the structure for the program, in addition to the nomenclature (a series of rules that determines how objects should be named). This section will also help you understand the comments and documentation attached below; we strongly recommend you read this before getting started!

Some of these rules are NOT yet integrated into Pycraft, but will be accommodated into versions of Pycraft greater than or equal to v9.4.0 (or v9.3.1 pre-release found here: https://github.com/PycraftDeveloper/Pycraft-Insider-Preview).

Variables

- All variables should be named in accordance to its function, or based on a description of the data it stores.
- There is no limit to the length of the name of a variable as at current there is no limit on the length of a of code.
- Here are some good examples of variable names. `StoreRandomNumber` or `StoreMapVertexBuffer`

Subroutines

- Subroutines can be of any length, as there is no limit to the length of a of code in Pycraft at present.
- Subroutines should avoid using global variables as much as possible, as this makes it easier to trace variables and possible bugs. (The exceptions here being the``Class_Startup_variables`` and `self` variables which are referenced throughout the different modules for Pycraft).
- Subroutines should be named according to their function, and not be dependent on other code in a specific module to work. (For example, making a random number generator that relies on global variables created elsewhere in a module)
- Subroutines should only have parameters if they are used within the subroutine.
- If a function returns a value, then this must be implicitly stated in the documentation here.

Modules

- All modules should be preceded by the following code, regardless of function:

```
if not __name__ == "__main__":
    print("Started <Pycraft_<name>>")
    class <name>:
        def __init__(self):
            pass
```

- All modules should also be proceeded by the following code, the 'else' here is important, this connects to the 'if' statement we created above:

```
else:
    print("You need to run this as part of Pycraft")
    import tkinter as tk
    from tkinter import messagebox
    root = tk.Tk()
    root.withdraw()
    messagebox.showerror("Startup Fail", "You need to run this as part of Pycraft,␣
↪please run the 'main.py' file")
    quit()
```

- If a module does not directly have its own GUI (for example the Achievements GUI is made by the 'Achievements.py' file), then it should have 'Utils' attached at the end of the name, this specifies that the program contains code that aids the creation of the game. There may already be a suitable 'Utils' file already. (If the subroutine your creating involves the use of Tkinter, even if it is to create a GUI, and is NOT part of the installer, then place that code under the 'Tkinterutils.py' file).
- Modules that are only ever used in a thread, must be placed into the 'ThreadingUtils.py' file.
- Modules can be broken down into as many classes as needed, but all subroutines must be placed in classes where possible to help speed up locating code if something does go wrong.

Error Handling

- No error should pass silently; errors should be grouped into two categories; 'fatal' and 'recoverable', errors that are deemed to be 'fatal' must immediately lead to the termination of the currently running program, and a message displayed through the crash GUI if possible. Non-'fatal' errors should be appropriately handled in the relevant module, and if expected to pass silently until a fix is available, then must be logged or printed out to the terminal, so other programmers can fix the error later on to stop it potentially causing problems.
- All errors should be -where possible- stored in the variable `message`.

1.2.2 Notices

- This documentation will be updated after a release of Pycraft, but only the necessary parts will be changed, if something is out of date or there is a mistake, then please contact Tom at thomasjebbo@gmail.com or post the issue in the issues tab so we are made aware!
- All indentation will be represented by ¬ in the by breakdowns.
- From here onwards will be the documentation for every in Pycraft, this will be updated regularly. We begin by introducing an overview of what each module and class and subroutine does, then go into a by-analysis, this will

be long and if your looking for something specific then we recommend that you use <control+f> to speed up the process!

- Any other notices will be places here!

Thanks for reading!

Documentation last updated: 29/10/2022 @ 19:18 (DD/MM/YYYY @ GMT using the 24-hour format) | Most recent Pycraft version: v9.5.0 | Most recent developer version of Pycraft: v9.5.4