
Pycraft Documentation

Release Pycraft-v9.5.0

Mar 01, 2023

CONTENTS

1	Contents:	3
1.1	Introduction	3
1.1.1	About	3
1.1.2	Setup	3
1.1.3	Running The Program	4
1.1.4	Credits	5
1.1.5	Uncompiled Pycraft Dependencies	6
1.1.6	Changes	7
1.1.7	Understanding the release notes	9
1.1.8	Input mapping	9
1.1.9	Our Update Policy	10
1.1.10	Version Naming	11
1.1.11	Releases	11
1.1.12	Other Sources	11
1.1.13	Final Notices	11
1.2	Module Breakdown	11
1.2.1	Nomenclature and programming techniques	11
1.2.2	Notices	13

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.

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 things 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, GPUutil, 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 device, 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 devices, 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's '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
- GPUUtil @ 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 v9.5.0 is now live! Here is a list of all the added features to this major update:

- Feature: This update features the new start-up animation by Dogukan Demir.
- Feature: This update also includes transitional effects between the different GUIs, which can be seen when navigating through different GUIs
- Feature: Basic collisions have been added to Pycraft, with better implementations coming after Pycraft v9.4.0
- Feature: Screen animation/transitions have been added throughout the project. This was showcased in Pycraft v9.4.1 but now works with the Pycraft v9.5.0 update.
- Feature: Wide range support for controllers has been added, you can now use a controller seamlessly throughout the game (currently only PS4 controllers have been tested but that's only due to hardware limitations).
- Performance: The inventory and map sections have had some serious performance optimisations, up to 2x in frame rates there now compared to Pycraft v9.4.0.
- Feature: The adaptive mode AI style performance adjustment utility has been re-written and now performs much better.
- Bug-Fix: The map icon in the map GUI now has transparency and a lightly improved resolution.
- Feature: The game engine has been completely redesigned, now with an additional utility file (“GameEngineUtils.py”) that helps to make the project more modular. This was done as part of making the game engine compatible with shadow mapping, the focus of this update. At the same time, we have improved other game engine functions and improved performance, and fixed numerous bugs some of which are listed below:
- Bug-Fix: You can now move in the direction of the camera with either WASD or the controller.
- Bug-Fix: The camera movement is not restricted to the size of the window.
- Bug-Fix: The screenshot functionality for the inventory now supports multiple monitors (* although for now this functionality is windowed mode only)
- Feature: Accessing the inventory and map functions from in game are now significantly faster and does not take you back to the load screen.
- Feature: All of Pycraft’s functions now have error handling in some form, with the most common method now closing the program and displaying a suitable error message onscreen.
- Feature: The entire adaptive mode functionality has been reprogrammed and is now much better at changing the performance of the game to balance performance and detail.
- Bug-Fix: The home screen had numerous issues with the bottom of the screen, this has now been fixed and the messages function is cleaned up. Also, on the home screen, the name has changed from “Tom Jebbo” to PycraftDev as that is the username that we use for my Pycraft related accounts - avoiding confusion.
- Identified-Bugs: There is an issue with changing to full-screen in game, when pressing any keys or interacting with the window will make an ‘error sound’ although no error is raised and the project runs fine.
- Identified-Bugs: There is an issue with moving backwards and left using the controller or keyboard, this will be fixed before the full release of Pycraft v9.5.0.
- Feature: The game engine now has had a sky change, instead of using 6 unique textures in a cube style, we now use one texture and a sphere to reduce the complexity and number of files Pycraft uses.
- Feature: The game engine has now got a skysphere (that is a spherical skybox) that changes with time. We have an 8-minute day, with a 2-minute sunset, then an 8-minute night with a 2-minute sunrise, this means 1 hour is 50 seconds long in real time.

- Feature: The game engine now includes celestial entities, in the form of the sun and moon, which are in sync with the day and night periods and are the light source through the day/night cycles.
- Feature: The game engine now includes a fog effect, this is not noticeable unless you travel really far at the moment, with fog starting at 1200m and ending at 1600m, this is going to be much more noticeable when the final map for the game is released, which will be much larger than it is right now and take advantage of this effect. Additionally, this will be edited in the next update to change based on the weather conditions.
- Feature: The game engine now has dynamic shadows that adjust based on the time of day, with more harsher shadows coming in the night to reflect the change in lighting.
- Bug-Fix: The benchmark function now works completely after issues with the OpenGL section and trying to use deprecated functions.
- Bug-Fix: Many issues with the installer are being fixed in the next few updates, if you identify one, please feel free to let us know so we can fix it!
- Bug-Fix: Many improvements have been made to the game engine to improve the quality of the shadows, which are now much less glitchy.
- Performance: Yet more performance improvements have been made to Pycraft, in particular in the game engine section.
- Feature: The game now includes the start of a weather implementation, including procedurally generated clouds that move around the player. The clouds and weather will change randomly through the day/night, based on the rough relative probabilities of that respective weather in the UK. They also employ the fog effect we added to the terrain last time and have variable height based on the weather. Additionally, the fog throughout the game now varies based on the current weather, so on a rainy day there will be more fog.
- Feature: The entire project has started to undergo restructuring to bring it in line with some of the PEP-8 standards.
- Bug-Fix: Numerous issues and bugs have been fixed regarding the benchmark functionality and exiting the benchmark now is much friendlier and any errors that used to be raised have now been fixed.
- Bug-Fix: More work has been done on the installer to improve its functionality, expect more improvements and eventual Linux compatibility there.
- Identified-Bugs: After toggling between full-screen in the 2D engine of Pycraft (which uses Pygame) the window cannot be resized on some systems, even with the correct flags.
- Identified-Bugs: The installer will render incorrectly if the user has chosen to scale their displays.
- Identified-Bugs: The cloud noise file for Pycraft (Rand_noise.png) is only generated once and the file is never re-written unless it's deleted.
- Bug-Fix: we are improving compatibility with non-Nvidia GPUs as the GLSL programmes can occasionally incur errors due to minor changes to how different GPUs accept syntaxes.
- Feature: The day and night has had some work done, the orbital path of the sun and moon is altered, simplified and along with many other formulae for the GameEngine.py and GameEngineUtils.py modules been simplified and tidied up a bit.
- Feature: Introducing storms, thunder and lightning has been added to the storm section of Pycraft under the 'rain.heavy.thunder' tag.
- Feature: The probabilities of different weather events (as well as their duration) has been tweaked in Pycraft.
- Feature: Added a rain sound effect that plays when the weather tag is not 'sunny'.
- Feature: Adjusted how audio (sound and music) is handed in Pycraft slightly.
- Performance: We have made many more performance improvements to Pycraft since Pycraft v9.4.4, including the introduction of mip-maps for textures in OpenGL

- Feature: There have been tweaks to the shadow mapping function so now there is a greater variation between day and night and different weather events.
- Feature: All of the commonly used functions in Pycraft's 2D engine have been added to one function in DisplayUtils.py to improve upgradability and to simplify other areas of the project.
- Identified-Bugs: After toggling between full-screen in the 2D engine of Pycraft (which uses Pygame) the window cannot be resized on some systems, even with the correct flags.
- Identified-Bugs: The installer will render incorrectly if the user has chosen to scale their displays.
- Bug-Fix: The cloud noise file for Pycraft (Rand_noise.png) is only generated once and the file is never re-written unless it's deleted.
- Feature: We have started work on converting most of the audio files to the same format (.ogg) and suitably giving accreditation in the credits GUI and later here.
- Bug-Fix: Improved the joystick implementation and fixed issues that occurred when applying pep8 to Pycraft.
- Feature: Introducing rain particles to Pycraft, they are randomly coloured and are the final distinctive feature of the weather implementation in Pycraft!
- Feature: Introducing the concept of multi-texture terrain into Pycraft, for now there is a grass and rock texture with more complex use cases for this coming soon.
- Feature: We now blend (linearly interpolate) between the different weather events in Pycraft, including the colour and height of the clouds and transparency of the skysphere based on different weather events.
- Feature: We have improved the installer for Pycraft and work on adding Linux compatibility is continuing. We have updated the installer to be compatible with Pycraft v9.5.0

Again, feedback would be much appreciated this update was released on; 19/07/2022 (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 revision 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 (That's 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 preceded 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