

Chase Experiment Manual

An experiment made with PyGame.

Installation

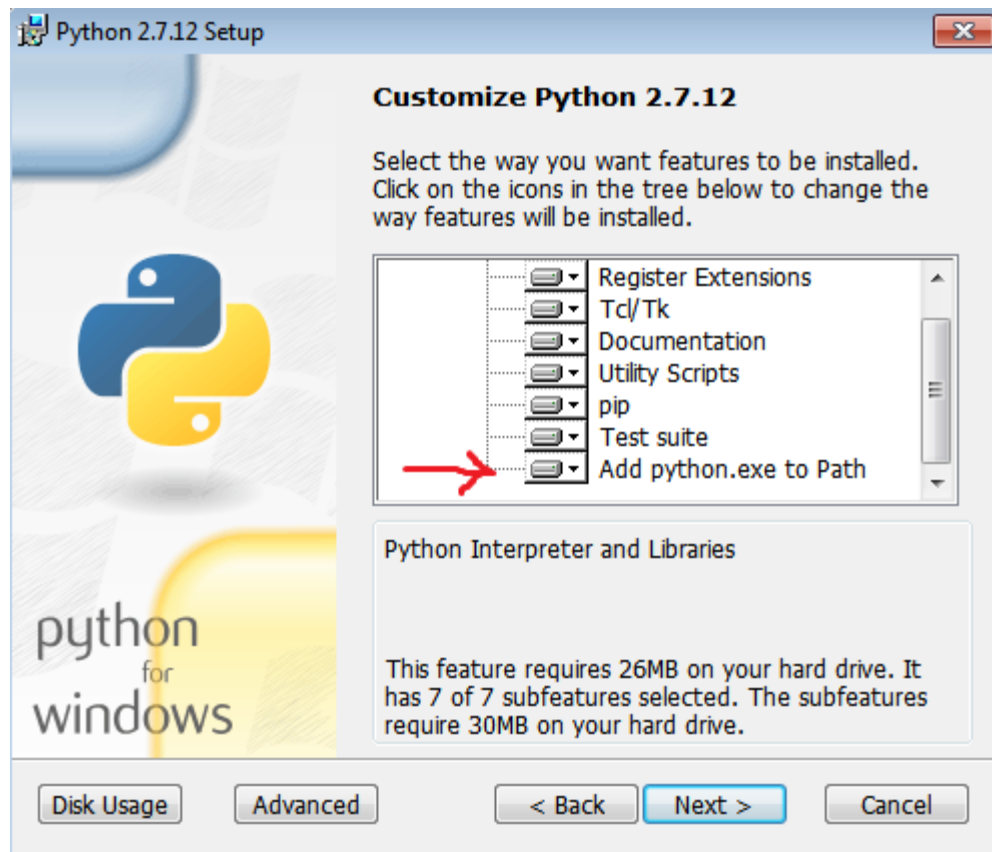
you need the following software packages:

- [Python 2.7 32bit](#)
- [PyGame 1.9.1](#)
- [requests](#)
- [numpy](#)

If you are a complete newbie, installing the packages required for running this experiment will keep you busy for a little while. I try to explain it "as basic as possible", assuming you are a biggener user.

First of all, you **want to make sure that you install Python 32bit 2.7.12 or higher**. Notice that I said **32bit**, even if your machine is **64bit** you still must install the **32bit** Python 2.7.12 or higher. You must not install Python 3!

Next, you want to make sure that **python is defined in your environment**. This means that anytime you type **python** in your command line, python should be executed. However, if python is not in the Path, the command line will not recognize it. This is particularly a problem for Microsoft Windows users (*as usual, whenever we talk about problems, Microsoft shines like a diamond!*). There is two ways you can add python to the environment. The simplest way **is to pay attention** when you install Python 2.7. As shown in the image below, make sure `add python.exe to path` is selected. This option makes your life much easier but is not selected by default in the installation.



The second way is trickier. Microsoft users should open the **System Properties** and go to **Advanced** tab, and click on **Environment Variables**. A new windows will open, in the **system variables** click on **Path** and add the path to your Python directory which is something like:

```
C:\Python27
```

[This tutorial might be helpful!](#). Then, also add the path to the `Scripts` Python directory which is something like:

```
C:\Python27\Scripts
```

Next, you will need to install a few required packages. The easiest way to install packages is using `pip`, which itself, needs an installation. [See this tutorial about how to install pip script](#). Since you have previously added the `Scripts` directory to the path environment, now you can use the `pip` command to install other packages. If you have installed Python 2.7.12, it already includes `pip` but you should update it. To do so, run the `cmd` and paste the following code in it:

```
python -m pip install --upgrade pip
```

Now, you can install requests from the command line:

```
pip install requests
```

You also will need `pygame`. Visit <http://www.pygame.org/download.shtml> and download the correct version of python for your OS.

In addition, you also need `numpy` for doing some numeric manipulations on the trial numbers. The installation is just as before:

```
pip install numpy
```

Settings.cfg

The experiment's settings can be adjusted by updating the `settings.cfg` file. The file can change the following settings:

Configuration Argument	Description
<code>screen_width</code>	takes an integer which defines the width of the screen in pixels
<code>screen_height</code>	takes an integer which defines the height of the screen in pixels
<code>FullScreen</code>	can be <code>True</code> or <code>False</code> . If set to <code>True</code> it makes the game fullscreen. Otherwise, the screen will appear with the specified width and height. If you set the game to full screen without increasing the width and height, the display will stretch.
<code>backgroundColor</code>	takes a vector of 3 integers, separated by comma which represent an RGB values for defining the background color of the game. The default values are <code>100,100,100</code> which is light gray.
<code>fontColor</code>	takes a python color name which can be <code>black</code> , <code>white</code> , or any of the colors mentioned later in the documentation.
<code>mouseCursor</code>	can be <code>True</code> or <code>False</code> . If set to <code>True</code> the mouse cursor appears during the trial. Otherwise it will be hidden during the trial.
<code>showTimer</code>	can be <code>True</code> or <code>False</code> . The experiment has a timer for tracking how long each trial lasts in terms of seconds. The timer can also be shown in the top-right corner of the screen if the value is set to <code>True</code> .
<code>gameover_sound</code>	The value can be <code>True</code> or <code>False</code> . When the trial ends, a buzz sound is played if the value is <code>True</code> .
<code>playerSpeed</code>	takes an integer which defines the movement speed of the player based on number of pixels. The default is 5 pixels

Configuration Argument	Description
wolfSpeed	takes a real number and defines the movement speed of the wolf and sheep.
trialType	a vector of "0" and "1" showing the overall number of trials (counting 0 & 1) and whether a wolf exists or not (1 or 0 respectively). For example, if the experiment has 4 trials and only the first 2 trials have a wolf, the vector should be 1,1,0,0. This option only specifies whether Wolf actually exists or not. It doesn't provide any information about the wolf's strategy.
duration	a vector of number of seconds that each trial lasts. For example, 30,10, ... the integers should be separated by comma.
chaseRate	a vector of chase probability per turnRate. the integer should be between 0 to 100 and separated by comma.
chaseAngle	a vector of chase angles for each trial. the integer can be between 1 ro 180. the integers should be separated by comma. (1 is practically straightline chase, given the size of the game objects and screen)
scapeAngle	a vector of escape angles for each trial, only used when randomPlayer is True. the integer can be between 1 ro 180. the integers should be separated by comma. (1 is practically straightline chase, given the size of the game objects and screen)
killZone	the radius of the death circle, i.e. the distance between the center of the coordinates of the Wolf and the player which ends the game, where the wolf is considered to win. The minimum killZone should be 32 which is the diameter of the circles.

Configuration Argument	Description
<code>sheepNumber</code>	a vector of comma separated integers that specifies the number of sheep including the wolf who appear in the game.
<code>Radius</code>	is an integer which defines the radius of the circle where the sheep and wolf should be located around it. The larger the radius, the more time the wolf will require to approach the player and also, the more time it takes other sheep to get close to the player. If set to anything below <code>killZone</code> there will be a chance of failure right at the start of the trial.
<code>turnRate</code>	is an integer which defines the number of seconds that it takes for sheep and the wolf to make a random turn. The default is 1 turn per second.
<code>randomPlayer</code>	can be <code>True</code> or <code>False</code> . If set to <code>True</code> the player will have a randomwalk similar to the other circles and the mouse will be disabled for moving the player during the trial.

Instructions

To run the experiment in Microsoft Windows click on **RUN.bat** file.

Available colors

For the `footColor` you can pick any of the colors mentioned in the figure below. Write the name without any space in between, for example, `redorange` or `midnightblue`.

