

CS 115, Autumn 2021

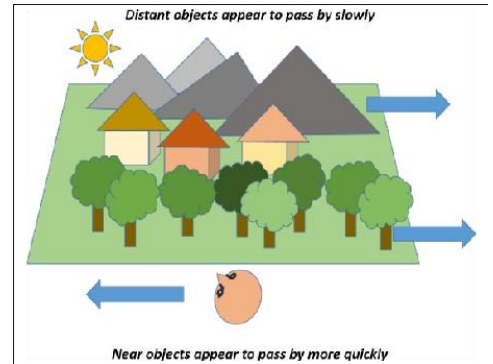
Programming Project #2: Motion Parallax (16 points)

Due Tuesday, October 5th 2021, 11:30 PM

This assignment tests your understanding of drawing shapes, expressions and variable. Write two Python programs in files named `creative_parallax.py` and `motion_parallax.py`. (Use exactly these file names, including identical capitalization.)

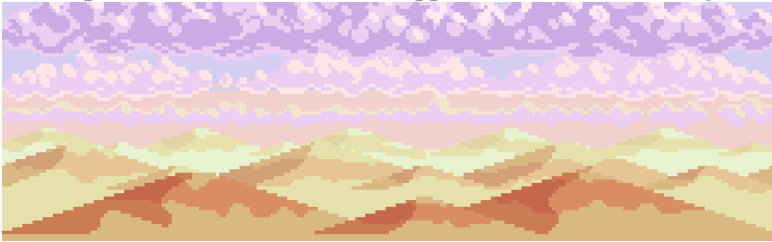
Description:

In this programming assignment, you will be building a program that displays a landscape and allows the user to control the perspective of the landscape by moving the mouse. The perspective seems to change due to an effect called motion parallax, which you should implement in this program. The picture linked from the course website shows an example of the user interacting with a completed version of the program.



Motion Parallax:

Motion parallax is one of the ways humans are able to distinguish how far away objects are. When objects go past at the same speed, those that are closer appear to move more. Objects that are far away, like mountains, barely appear to move while close objects, like a car parked on the side of a street you are driving down appear to move much more.



Notice that the mountains closest to the bottom in the scrolling image to the left appear closest. This is partly because they are on top of the mountains behind but also because they are moving faster than the mountains behind.

The above picture shows parallax when it scrolls. However, we will be making our parallax appear when the user moves their mouse. You can get the current x and y position of the mouse by using `panel.get_mouse_x()` and `panel.get_mouse_y()`. To create the parallax effect, add a percentage of the mouse x to the x position of an object and a percentage of the mouse y to the y position of an object. Add a larger percentage to closer objects so they will appear to move more and a smaller percentage to a further away object. For example, using `starting_location + panel.get_mouse_x() * 0.1` for the x position would make an object move more and look closer than using `starting_location + panel.get_mouse_x() * 0.05` for the x position. You should alter both the x and y coordinates of the shapes you draw in order to achieve a good parallax effect.

Part 1: `creative_parallax.py`

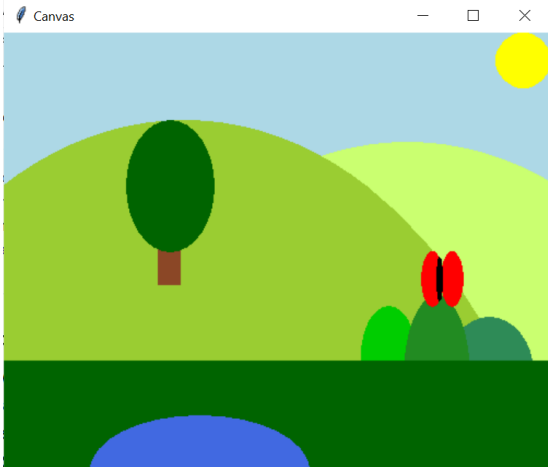
For this part of the assignment, write a program that draws a picture of your choice and uses the mouse x and y location to make at least one part of the image appear closer to the viewer than another. You will receive full credit if your program meets the following requirements and the requirements listed below in the style section:

1. Include at least 8 shapes
2. Include at least 4 colors
3. Be drawn on a drawing panel sized between 200 x 200 and 1000 x 1000
4. Have at least two layers of parallax movement.

Part 2: `motion_parallax.py`

For this part of the assignment, output the picture below and implement the motion parallax. The objects in the picture are listed below in the order of closeness to the viewer. Remember that the closer the object the more it should move.

All objects should be drawn on a `drawing_panel` 500 pixels wide and 400 pixels tall.



1. The closest object is the butterfly. It wings are 20 wide and 50 tall. The body is 5 wide and 40 tall. The wings are located at 300, 200 and 320, 200. The body is located at 315, 205.
2. The next closest is the lake, located 20 from the left side and 350 from the top. It should be 100 tall and twice as wide.
3. The bushes are next closest. The furthest left is located at 300, 250 and 50 wide, 90 tall. The middle one is 340 from the left, 240 from the top, 60 wide and 150 tall. The last is 380 from the left, 260 from the top, 80 wide and 100 tall.
4. The tree is next closest. The leaves are located at 85 from the left, 80 from the top, 80 wide and 100 tall. The trunk is 180 from the top and centered under the leaves. It is 20 wide and 50 tall.
5. The hills are the furthest back piece of the image that should move.

They are ovals that are so big (700 wide, 1000 tall) that most of them are drawn off the screen. The first starts on the left side of the screen and 100 from the top. The second is located at -200 x (starts 200 off the left side of the screen) and 80 from the top.

6. The last two pieces of the landscape do not move.
 - a. The sun, 50 wide and 50 tall, located at the top of the panel, 450 from the left side.
 - b. The grass, as wide as the window, covering the lowest 100 pixels of the window.

You should output the same shapes located in the same places as the ones described and pictured above. You will find it necessary to think carefully about which order you draw them in so that the right shape is drawn on top.

You are welcome to choose any colors you wish. You may also choose the percentages the different shapes move. However, make sure closer ones move more.

Style Guidelines:

All of your code must be in a file that is runnable. You should **not** type each of your statements into the interpreter and run them one at a time.

Use variables to avoid repeating the same computations more than once. Make sure to pick good variable names that describe what the values the variables store represent. Names like `a`, `b`, `c`, `variable1`, `variable2`, `variable3`, `pmt`, `prple`, `ipp` and `tp` are not good variable names. If your names include multiple words separate them with an underscore. All names should be all lowercase.

Include a comment at the beginning of your program with some basic information and a description of the program in your own words. For example:

```
# Suzy Student, CS 115, Autumn 2049
# Programming Project #1, 06/07/49
#
# This program's behavior is ...
```

For `motion_parallax.py`, you should limit yourself to the Python features covered in the first nine lectures. Though we will cover more material while you work on this assignment, please do not use any of it in this program, such as `if/else` statements. You may use advanced material in `creative_parallax.py`.

Submission and Grading:

Turn in your Python source code files electronically from the Project page on the course web site.

Part of your program's score will come from its "external correctness." External correctness measures whether the output matches exactly what is expected. We are very picky about the output matching exactly and expect every character and space to match. Use the **output comparison tool** to help you make sure your output is perfect. Programs that do not run will receive no external correctness points.

The rest of your program's score will come from its "internal correctness." Internal correctness measures whether your source code follows the stylistic guidelines specified in this document. This includes having an adequate comment header and using `drawing_panel`, expressions and variables well. You should make sure to name your variables with descriptive names in all lowercase and underscores between words. You should also limit the lengths of all lines in your program to **fewer than 80 characters**.