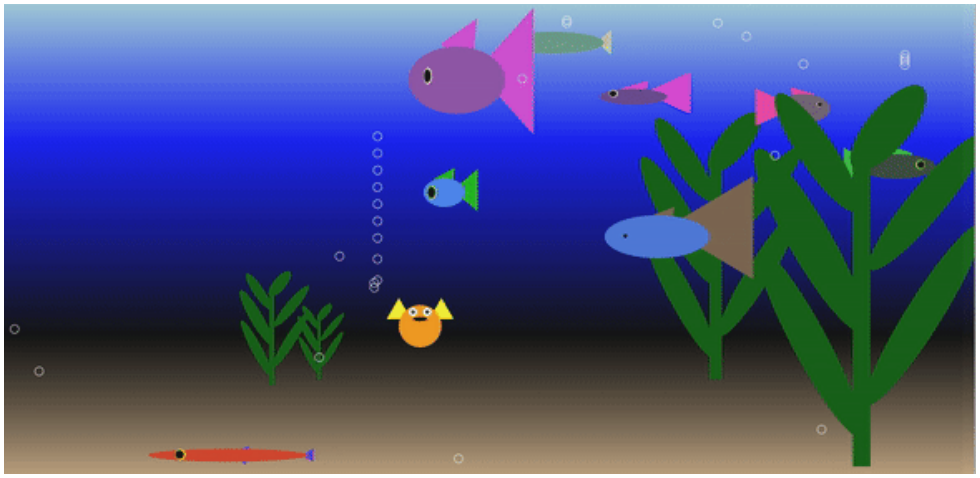
EECS 110: Project 1 (40 Points)

Graphics Track

Due Thursday, 5/16

This assignment asks you to apply what you've learned to create an animated, interactive terrarium or aquarium. To get an idea of what you might make, please see the sample videos (as well as the animated gif below). Some of these samples have gone above and beyond the assignment specifications, so no pressure. All of these submissions got full credit. Please calibrate your implementation according to your experience level with programming (i.e. do more if you're more experienced): you'll get out of it what you put into it. See the rubric below to get a sense of how you will be evaluated on this assignment. Then, download the starter files and navigate to the option1_graphics folder.



Created by Rachel Smith (Berkeley City College)

1. Create a make_creature function (7 points)

In **helpers.py**, create a custom function called ***make_creature*** that makes a creature of your choosing. It should have at least 3 parameters that enable the calling function to specify arguments for:

1. ***canvas*** *(positional)*: a reference to the canvas object.
2. ***center*** *(positional)*: a tuple representing the (x, y) coordinate of the centerpoint of the creature.
3. ***width*** *(keyword):* an integer that controls the width of the creature.

Feel free to use/modify/improve upon the function you already made in Homework 3. You may also add any additional ***keyword*** parameters (i.e., they must be optional, not required) of your choosing to make your creature even more customizable (e.g.: primary_color='red',

has_sunglasses=True, etc.). Or, feel free to use the random module to create creatures with 'surprise' features.

You are also welcome to create additional creature functions (if you want different kinds of creatures to be present in your terrarium / aquarium). That said, please ensure that your main creature function is called ***make_creature*** (for our grading pipeline). Any additional creature functions can be named anything that you want and work in any way you want. Up to 2 points will be given for fantastic beasts.

2. Create a make_landscape_object function (7 points)

You have already experimented with stars and bubbles in Homework 4. Here, you will make an object of your own choosing (e.g. a tree, a building, a rock, grass, coral, etc.). To do this, you will create a custom function called ***make_landscape_object,*** in **helpers.py**, that draws a landscape feature of your choosing. It should have at least 3 parameters that enable the calling function to specify arguments for:

1. ***canvas*** *(positional)*: a reference to the canvas object.
2. ***center*** *(positional)*: a tuple representing the (x, y) coordinate of the centerpoint of the background object.
3. ***width*** *(keyword):* an integer that controls the width of the background object.

You may also add any additional ***keyword*** parameters of your choosing to make your even more customizable (e.g.: color='red', texture='rough').

You are also welcome to create several different landscape feature functions to make different landscape features. Note that while your main landscape function must be called ***make_landscape_object***, any additional landscape feature functions can be named anything that you want. Up to 2 points will be given for fantastic landscape features.

3. Render your aquarium / terrarium (7 points)

In **main.py**, you will render your landscape. This involves (a) instantiating some background creatures (using your make_creature function), and (b) creating some landscape features (using your make_landscape_object function).

a. Background creatures

Instantiate at least 5 different "background" creatures in your landscape, located at different positions, and with varying sizes (and colors and features — optional). You may use a loop (and perhaps a random function) to position your background creatures, or you can intentionally place your background creatures in specific places — or some combination of the two.

b. Landscape features

Instantiate at least 5 instances of your landscape feature (in the same manner as you did with your background creatures).

4. Animate your landscape (15 points)

Animate the creatures in your landscape by picking **five** of the following effects to implement:

1. **3 points**: Animate your creatures (and your environmental features if it makes sense to do so). If creatures / environmental features move off of the screen, recreate them on the other side or have them bounce off the side.
2. **3 points**: Experiment with different kinds of motion. Instead of your creatures moving linearly at a constant speed, you can experiment with the math.sin and math.cos functions (or any others techniques) to make your creature oscillate, accelerate, decelerate, etc.
3. **3 points**: Spawn a new creature when the user clicks the screen.
4. **3 points**: Animate each of your creatures so that their movement is slightly different (different speeds, different movement patterns).
5. **3 points**: Enable your user to control one or more of your creatures using keyboard events (pressing the up/down arrow, using the spacebar, etc.)
6. **3 points**: Periodically add or remove creatures to/from your scene.
7. **3 points**: When you click a creature, remove it from the screen
8. **3 points**: Detect creature collisions, and do something interesting if creatures collide
9. **3 points**: Enable the user to reposition a creature or landscape feature by dragging it
10. **3 points**: Create some game mechanics (space bar jumps or shoots, drag 'flings' an object, etc.)

NOTE: If you surpass the **15** points (i.e. 5 features), we will award up to 6 points extra credit for pursuing additional enhancements from the list above. So, a sixth feature would be 3 points extra credit, for example.

5. Create a short video (1 point)

Please create a short (like ~15 seconds) video of your terrarium / aquarium in action. To do this:

1. Take a screencast (QuickTime works, and is free)
2. Upload it to Google Drive
3. Share your video so that anyone with link can view

What to Turn in

1. Your code as a zip file. This should include all of the files that make your code work.

2. A link to your video (paste it into the text area on Canvas)
3. A (very) short paragraph (bullets fine) indicating which 5 features you implemented for part 4, including any extra credit that you completed. This is critical for helping the TAs look out for the features you implemented. Help us help you get the points you deserve.

References Materials

Samples

I have created some sample code (in **demos** folder) to help you to understand how to interact with canvas using the keyboard and mouse. Check out the sample code to get some implementation ideas.

Documentation

You will also want to refer to the tkinter documentation. Here are some useful sites (but feel free to do your own research):

Old Documentation

From Python 2.x (still quite useful):

- tkinter events: http://effbot.org/tkinterbook/tkinter-events-and-bindings.htm
- tkinter canvas: http://effbot.org/tkinterbook/canvas.htm

Newer Tutorials (Python 3.x)

- https://www.tutorialspoint.com/python3/python_gui_programming.htm
- https://dzone.com/articles/python-gui-examples-tkinter-tutorial-like-geeks

POLICY ON CODE SAMPLES AND COLLABORATION

YOU MAY....

- Make use of any sample code that I have provided you at any point during the course
- Consult with and incorporate ideas from Internet sources, so long as you are typing the code for yourself (not copying) and understand how every line of it works
- Help each other debug your code and discuss ideas together

YOU MAY *NOT*...

- Copy anything (except for sample code I have provided you)
- Share code — or look at your neighbor's screen and transcribe their code. We have plagiarism detection software to flag code similarities (e.g. MOSS) — even when whitespace, variable names, and ordering have been changed.
- Use code copied directly from third-party sources; or programming techniques that you don't understand.
- Talk to CS majors (or other more experienced people) and ask them to write code for you.

RUBRIC (40 Possible Points)

Feature	Possible Pts	Scoring Guidelines
Creature function	7	Function enables creature to be customized by size and position. - Function doesn't properly scale creature (1-2 pts) - Function doesn't properly position creature (1-2 pts) - Missing parameters (1-2 pts) - Missing calling arguments (1-2 pts) - Function doesn't run b/c of syntax errors (up to 5 points)
Landscape function	7	Function enables landscape object to be customized by size and position. - Function doesn't properly scale object (1-2 pts) - Function doesn't properly position object (1-2 pts) - Missing parameters (1-2 pts) - Missing calling arguments (1-2 pts) - Function doesn't run b/c of syntax errors (up to 5 points)
Render aquarium / terrarium	7	- Ensure the aquarium program is invoked from **main.py** (up to 1 pt) - At least 5 background creatures have been rendered - At least 5 landscape features have been rendered (with different sizes, positions, colors, etc.)
Animate your landscape	15	- 5 animations/interactions (from the list above) have been implemented. - Each of the animations/interactions follows the specifications given - There is coherence to the design of the interactions
Code Quality	3	- All functions, variables, and file_names have pneumonic names / are snake case (up to -1 points) - Code is organized and without unused or redundant code. Please remove commented out code that isn't running (to help our graders) (up to -2 points)
Video	1	- 15 second video has been submitted
Extra credit	Maximum of 6	For projects that exceed expectations in any of the following ways: - Went above and beyond adding optional parameters and/or additional creature functions. - Went above and beyond adding optional parameters and/or additional landscape object functions. - Creature or landscape function is particularly creative / complex. - The interactions (events) are creative. - The composition is sophisticated (artistically or otherwise). - Extra features listed in animation / interaction section