

CS 115, Autumn 2021

Programming Project #4: Pong (20 points)

Due Tuesday, October 19th, 2021, 11:30 PM

This assignment tests your understanding of global variables and `if/else` statements. Submit two Python programs in files named `pong.py` and `animation.py`. (Use exactly these file names, including identical capitalization.)

Part 1: `pong.py`

In the first part of this programming assignment, you will write code to allow the user to play a one-player version of the game Pong. Pong is an old arcade game meant to resemble Ping Pong. Each player gets controls one paddle that can move up or down. If the player places the paddle in the path of the ball, the ball will bounce off it. If the user does not manage to place the paddle in front of the ball it will go off the screen behind the paddle and the other player will earn a point.



You will be writing a simplified version of this game. In your version, there will only be one player and one paddle. We will move the paddle along the bottom of the window so that it is centered horizontally on the user's mouse position. Its `y` position should always be at the bottom of the screen.

When the program starts the ball should start at the top of the window at a random `x` location. It should move downward until it reaches the bottom of the screen. When it reaches the bottom of the screen it should turn around and rise until it hits the top if the paddle is under it. If the paddle is not under it, a losing message should appear. The same thing should happen if it hits the top of the screen.

If the ball only moves in the `y` direction it will bounce straight up and down and so our game will be no fun. We need to make it move a little bit horizontally. When the game starts, the ball should move straight down, no horizontal movement at all. Every time it hits the bottom or the top of the window your program should randomly choose between, `-1`, `0`, and `+1` and make the ball move that amount plus the old `x` change amount until the next bounce.

For example, initially the ball has an `x` change amount of `0` so every time it is drawn it appears directly below the last drawing. This continues until it reaches the ground. Then, for example, your program might generate `-1`. As the ball moves to the top of the screen each time it moves up it should also move one pixel to the left. Then, if when it hit the top, it happened to pick `-1` again when it started moving down again it would move `2` to the left every time.

This program should keep track of how many points the player has earned. Instead of giving points to a player when their opponent misses, we will give points to a player when they successfully hit the ball. This means you will need to add a point to the user's score every time the ball reaches the bottom and the paddle is underneath it. This score should be drawn somewhere on the screen.

You may choose the colors and sizes of your shapes. The only appearance requirements are:

- The ball must use at least one shape and be smaller than the paddle
- The paddle must use at least one shape and be bigger than the ball
- The points must be displayed somewhere on the `drawing_panel` at all times.
- The ball must travel up and down instead of side to side as the original Pong did.

Part 2: `animation.py`

For this part of the assignment, you will create an animation similar to the car animation from class. It must meet the following requirements:

- The panel must be between 200 by 200 pixels and 1000 by 1000 pixels.
- The figure must be made of two or more different shapes.
- The figure should be able to move on its own without any user input (no mouse dependent movement).
- The figure must do something when it reaches the edge of the screen. It can keep going the same direction and wrap around the screen as we did with the car. It can turn around or bounce of the edge.

You are welcome to extend this portion of the project however you wish. You may add an additional figure, you may make figures interact, etc. You will receive full credit as long as you meet the requirements listed above.

Hints:

We have not drawn much text on the `drawing_panel` so you may not be sure how. It is very similar to drawing any shape:

```
panel.draw_string(text, x, y)
```

For example,

```
panel.draw_string("this will appear on screen", 0, 200)
```

will place the text this will appear on screen 200 pixels from the top on the left side of the screen.

Style Guidelines:

For full credit on this project, you must have *constants* for the following values:

- Panel width
- Panel height
- Ball size
- Paddle width

These constants should be used throughout your code so that if their values are changed and your program is rerun it adjusts accordingly.

This program will require you to have some *global* variables. In general, it is considered good style to have as **few** global variables as possible. Therefore, you should declare variables locally when you can.

`if/else` statements are a major focus of this project and so will be a focus when grading. We will be looking for well factored `if/else` statements with no lines of code repeated in multiple branches. We will also be looking for good choice of `if/else` structure.

As in previous assignments, 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 each of your files 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 ...
```

You should also include a comment on each section of your code describing what it does.

For `pong.py`, you should limit yourself to the Python features covered in the first four weeks of class. Though we will cover more material while you work on this assignment, please do not use any of it in this program, such as `while` loops. You may use advanced material in `animation.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. 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`, `constants`, `if/else`, `expressions` and `variables` (both global and local) 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**.