Research in Game Design

By Pranav A. Bhounsule, pranav@uic.edu

Outcomes

At the end of this research, students will be able to:

1. Learn basics of programming, plotting, and animations in Python
2. Learn how to install and run Python packages such as numpy and matplotlib for scientific computing
3. Learn fundamentals of game programming using pygame.

Equipment list

All the resources are free.

1. **Microsoft Visual Studio Code for Python:** Download and install from here. Microsoft Visual Studio Code. Set up Python using this video and test a simple script shown in the video (e.g., YouTube video https://youtu.be/9o4gDQvVkLU).
2. **Install additional python packages:** Packages can be installed by typing the following in the "terminal" in Microsoft Visual Studio Code: python3 -m pip install {package_name}. You need to run this command for each package. For {package name}, type numpy and matplotlib. This minute-long video shows how to install a package. https://youtu.be/sQPgR1Fz67c.
3. **Pygame for creating Python games:** Install pygame (https://www.pygame.org/news) after you have Microsoft Visual Studio Code. In the "terminal" typing: python3 -m pip install -U pygame==2.6.0
4. **AI tools (optional):** Feel free to use AI tools like ChatGPT, Gemini, etc. You can also search and copy-paste code from other sources like GitHub, Stack Exchange. My general advise is to outline a method for solving the problem and use these tools to help you code.

1 Part 1: Introduction to Python

For this section, you will need to install Microsoft Visual Studio Code and the numpy and matplotlib packages (see 1 and 2 in Equipment list).

1.1 Basics of Python (1 week)

1. **Material:** Watch the video and practise coding as you watch it.
   https://www.youtube.com/watch?v=yutMT6NbjOU
   All the code is also available for download here:
   https://pab47.github.io/robotics/f24/Python/Lec01_python_basics.zip

2. **Exercise:** For the problems below send your Python files to pranav@uic.edu

   (a) **If-else conditionals and inputs:** Write a python file that will prompt the user to input a year and your code will print if it is a leap year or not. (You will have to search how to use Python to input a number).

   (b) **Computing π using numerical experiments.** The value of $\pi = 3.141592$ (6 decimals). How do you estimate this number? If you put a square of size $2r$ and a circle of diameter $2r$ in the rain and count the number of times raindrop falls on the circle and square then the ratio of drops falling on either shape will be proportional to the area of the respective shape. Thus

$$\frac{\text{no. of drops on circle}}{\text{no. of drops on square}} = \frac{\pi r^2}{(2r)^2} = \frac{\pi}{4} \tag{1}$$

   You can watch this video for more information
   https://www.youtube.com/watch?v=I-BC_vI4CAE.
   We can setup a numerical raindrop experiment using random number generator in Python to compute the value of π using $r = 1$. Generate N random numbers for x and y between -1 and 1. By default these points lie inside the square of side 2. Now for each number check if the position (x, y) lies inside the circle. If it does then increment the counter (lets call it N_c) that keeps track of points inside the circle. For a large value of N the ratio of $\frac{N_c}{N} = \frac{\pi}{4}$. Thus $\pi = 4\frac{N_c}{N}$.
   Write a computer program that computes the value of π for a given N. Then tune N such that your solution to π is accurate to at least 4 decimal places.

1.2 Basics of Animation (1 week)

1. **Material:** Watch the video and practise coding as you watch it
   https://youtu.be/NYKemmycMm4
   All the code is also available for download here:
   https://pab47.github.io/robotics/f24/Python/Lec02_python_animations.zip.

2. **Exercise:** For the problems below send your Python files to pranav@uic.edu

   (a) **Creating a plot:** Consider the function $f(v)$ where v is the velocity in m/s,

$$f(v) = \frac{0.25v^2}{450 + [log(v)]^{2.5}} - 0.023v - 9.8 \tag{2}$$

   Write a Python file m that generates a plot of $f(v)$ vs v for velocities from 50 to 250 m/s. Be sure to plot the x- and y-labels and put an appropriate title. Use a font of at least 21. Also, ensure that the x- and y-axes have numbers in a font size of 12 or more.

   (b) **Python animations:** Create an animated face of your choice. For example, a smiling face, a crying face. At a minimum, you need to include:

      i. At least one patch object. Use (e.g., *Circle*, *Rectangle*, *Polygon*)
      ii. At least one moving object. Use *pause*() command in a loop.
      iii. At least one line object. Use *Line2D* command.

   Here are some example faces created in my robotics course http://youtu.be/hkML3tLI3IQ

2 Part 2: Game Design with pygame

For this section, you will need to install the package pygame (see 3 in the Equipment list).

2.1 Complete an existing pong game and add features (2 weeks)

The handout for creating a single-player pong game is here:
https://github.com/pab47/robotics/blob/main/f24/project_pong.pdf
A short video explaining the instructions is here: https://youtu.be/eO1Eb2pRAGY. There are two Python submissions for this part, as indicated in the handout. Please send your Python files (in a zipped folder) to pranav@uic.edu.

2.2 Create your own game (2 weeks)

Create your own game using Pygame. You can take some inspiration from pygame community here: https://www.pygame.org/tags/all. Once ready, please send your Python files (in a zipped folder) to pranav@uic.edu.