



PYTHON PROGRAMMING

Game Development and Artificial Intelligence

PYTHON & ARTIFICIAL INTELLIGENCE



The Ultimate Programming Curriculum for Future AI

The Python & AI curriculum introduces Python programming through game development before moving on to Artificial Intelligence (AI) courses. Like all Exceed Robotics curriculums, programming concepts and languages are taught through relevant applications. Python is the leading coding language for Big Data, Advanced Robotics, and AI.

Higher-level courses move on to programming for image manipulation, object detection and text-to-speech before integrating these elements into a final AI project. The end goal of this program is to teach Python to create advanced devices with Artificial Intelligence

PROGRAM DETAILS

20-month curriculum (In-centre)

Classes once per week

One-hour in duration

8 students max per class

The Python and Artificial Intelligence curriculum was created by a team of software engineers working in the tech industry, with years of experience in teaching and course development. The program's vision is to create a community of young innovators by developing relevant computer science & engineering skills with industry-related experience to kickstart their careers in STEM

1 GAME PROGRAMMING

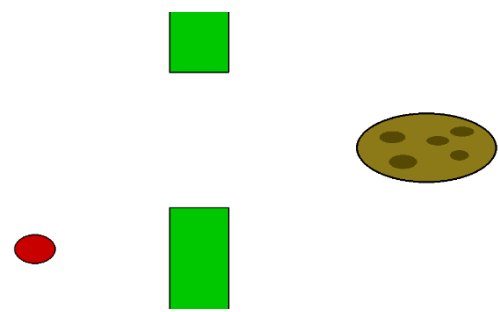
PY-01: INTRODUCTION TO PYTHON GAME PROGRAMMING

Prerequisite: ages 12+ (no experience required)

```

49 #events
50 if top_pipe[0] <=0:
51     top_pipe[0] = 800
52     bottom_pipe[0] = 800
53     score += 1
54
55 if L== 1:
56     bird[1] += -10
57
58 if bird.colliderect(top_pipe) or bird.colliderect(bottom_pipe):
59     play = 0
60
61 #Drawing
62 screen.fill(white)
63 pg.draw.rect(screen, green,top_pipe)
64 pg.draw.rect(screen, black,top_pipe,2)
65 pg.draw.rect(screen, green,bottom_pipe)

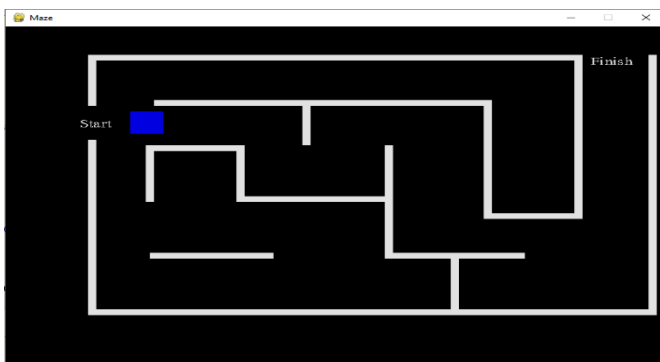
```



This program is designed to introduce students to coding in Python through a fun application - Game Programming. Students learn programming fundamentals using an object-oriented language. Concepts such as variables, if statements, while loops, mouse input, and basic shapes are taught in the first course. Students conclude the course by designing and programming a game in Python, using their newly acquired skills.

PY-02: GAME PROGRAMMING FUNDAMENTALS I

Prerequisite: PY-01



```

#-----DRAWING-----#
screen.fill(black)

for maze in Maze:
    pg.draw.rect(screen,white,maze)
    pg.draw.rect(screen,Blue,shape)

for wall in Maze:
    if shape.colliderect(wall):
        shape = pg.Rect(100,150,40,40)

printToScreen(90,160,"Start",white,20)
printToScreen(710,50,"Finish",white,20)
pg.display.flip()

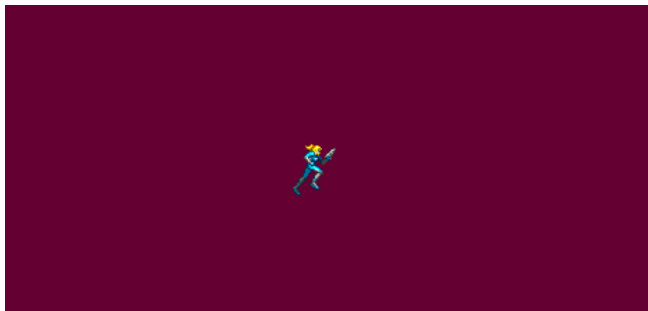
#-----CLOCKS-----#
clock.tick(60)

```

Students are introduced to more advanced concepts including Functions, Lists and Keyboard input. The material covered in this course could be extended to any object-oriented programming language and serve to advance our game development skills. Students will complete the course by creating a new game and applying the learned fundamentals.

PY-03: GAME PROGRAMMING FUNDAMENTALS II

Prerequisite: PY-02



```

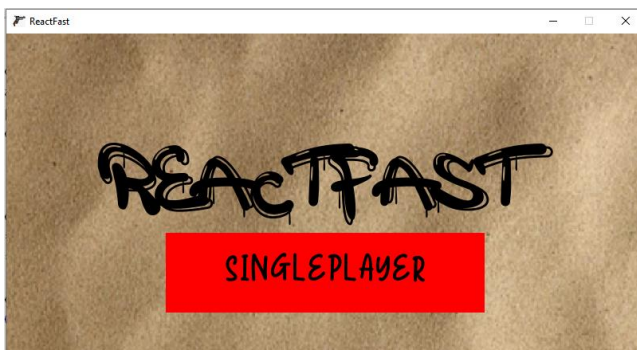
40 def update(self):
41     self.hitbox[0] += self.speed_x
42
43     if self.speed_x != 0:
44         self.frame += 1
45
46     if self.frame > 6:
47         self.frame = 0
48
49
50 def draw(self):
51     x = self.hitbox[0]
52     y = self.hitbox[1]
53
54     screen.blit(self.walk[self.frame], (x,y))
55
56
57 S = Samus(0, 250) #create an instance of samus
58
59
60 while True:
61     #UPDATES
62     S.update()
63
64     #DRAW

```

Our third game programming course focuses on one of the most important concepts of object-oriented programming - classes. Classes play a big role in game programming by simplifying code and have extensive uses in non-game programming applications. Students will create several games throughout this course to better understand and practice this concept.

PY-04: GAME PROGRAMMING APPLICATION

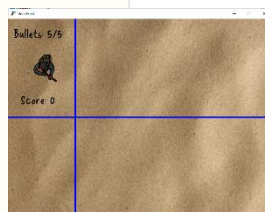
Prerequisite: PY-03



```

0 def game():
1     global state, make, p
2     if make:
3         p = GunGameSingleplayer()
4         lines = []
5         lines.append(pg.Rect(200, 0, 5, 600))
6         lines.append(pg.Rect(0, 295, 800, 5))
7         bg = pg.image.load("images/sand.png")
8         bg = pg.transform.scale(bg, (800, 600))
9         #game loop
10        while state == 1:
11            # Quit game window
12            for event in pg.event.get():
13                if event.type == pg.QUIT:
14                    pg.quit()
15                    exit()
16            # Input
17            pg.event.pump()

```



pip install pygame

 game 2.0.1 (SDL 2.0.14, Python 3.8.5)

 Lo from the pygame community. <https://www.pygame.org/contribute.html>

The final Game Programming course has an open project to provide students with the opportunity to apply the learned concepts for programming a game on their own. Students receive a project outline with game requirements and options to make the game more interesting. At the end of this series of courses, students would have built a solid understanding in programming and coding in Python.

PY-05: IMAGE PROCESSING

Prerequisite: PY-04



```

489 class StaEnabledState extends State {
490     @Override
491     public void enter() {
492         mWifiStateMachine.setSupplicantRunning(true);
493     }
494     @Override
495     public boolean processMessage(Message msg) {
496         switch (msg.what) {
497             case CMD_WIFI_TOGGLED:
498                 if (!mSettingsStore.isWifiToggleEnabled()) {
499                     if (mSettingsStore.isScanAlwaysAvailable()) {
500                         transitionTo(mStaDisabledWithScanState);
501                     } else {
502                         transitionTo(mApStaDisabledState);
503                     }

```

We start the first of the Artificial Intelligence courses with Image Processing. In this first course, students will learn how to manipulate an image in preparation for object detection using various AI tools. Students will practice image manipulation including resizing, rotation, scaling and applying filters to images to highlight areas of interest for image recognition.

PY-06: AI OBJECT DETECTION

Prerequisite: PY-05

```

In [ ]: import cv2
import numpy as np

# path to input image/video
IMAGE = './1.mp4'

# path to yolo config file
# download https://github.com/arunponnusamy/object-detection-opencv/blob/master/yolov3.cfg
CONF_PATH = './yolov3.cfg'

# path to text file containing class names
# download https://github.com/arunponnusamy/object-detection-opencv/blob/master/yolov3.txt
CLASSES_PATH = './yolov3.txt'

# path to yolo pre-trained weights
# wget https://pjreddie.com/media/files/yolov3.weights
WEIGHTS_PATH = './yolov3.weights'

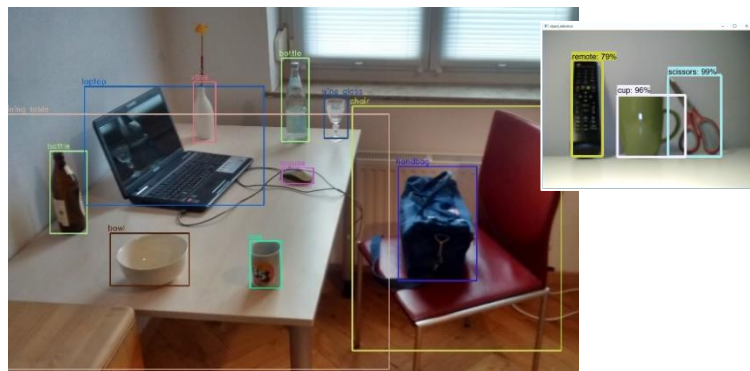
In [ ]: import os
print(os.path.exists(CLASSES_PATH))
print(os.path.exists(CONF_PATH))
print(os.path.exists(WEIGHTS_PATH))
print(os.path.exists(IMAGE))

In [ ]: # read class names from text file
classes = None
with open(CLASSES_PATH, 'r') as f:
    classes = [line.strip() for line in f.readlines()]

scale = 0.00392
conf_threshold = 0.5
nms_threshold = 0.4

# generate different colors for different classes
COLORS = np.random.uniform(0, 255, size=(len(classes), 3))

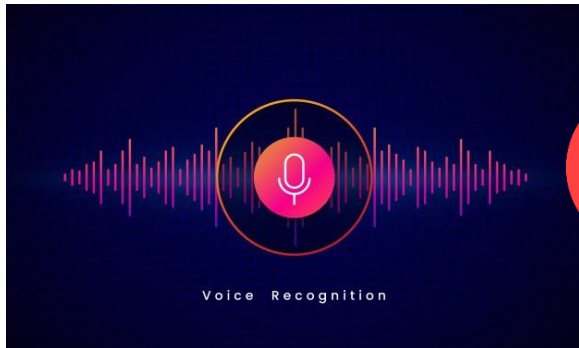
```



Through writing programs to detect colours and recognize traffic signs, students learn how to create their own AI applications. In this course, students will use APIs to create simple object and colour detection programs. Programming skills learned in these two image recognition courses will prove useful in developing AI apps in future courses.

PY-07: AI SPEECH RECOGNITION

Prerequisite: PY-06



```
recContext.Recognition+=new _ISpeechRecoContextEvents_RecognitionEventHandler(
    this.HandleRecognition);
...
public void HandleRecognition(int StreamNumber, System.Object StreamPosition,
    SpeechRecognitionType RecognitionType,
    ISpeechRecoResult Result)
{
    SpeechDisplayAttributes a = Result.PhraseInfo.GetDisplayAttributes(0, 1, false,
    false, Confidence = Result.PhraseInfo.Rule.EngineConfidence);
    int num=0;
    textBoxRecoCmd.Text+=Result.PhraseInfo.GetText(0, -1, true);
    if (Result.PhraseInfo.Properties!=null)
    {
        if (Result.PhraseInfo.Properties.Count > 0)
        {
            foreach (ISpeechPhraseProperty p in Result.PhraseInfo.Properties)
            {
                num += (int) p.Value;
            }
        }
    }
    ...
}
```

Speech recognition and text-to-speech (TTS) are two of the main applications of AI in most smart devices. Students will learn how to create programs to listen to a user and understand words and sentences. They will also learn how to convert written text by users into audio sentences. Combining image and speech recognition, we are ready to proceed to building AI applications!

PY-08: INTRODUCTION TO RASPBERRY PI

Prerequisite: PY-07



Students will be introduced to the Raspberry Pi and 'program & build' applications using Python. Raspberry Pi is a complete computer that costs less than \$100 and can execute Python code. It can be equipped with a camera, microphone and speakers and has built-in Wi-Fi. All this capability is inside a device that fits in your pocket.

PY-09: RASPBERRY PI CAPSTONE PROJECT I

Prerequisite: PY-08



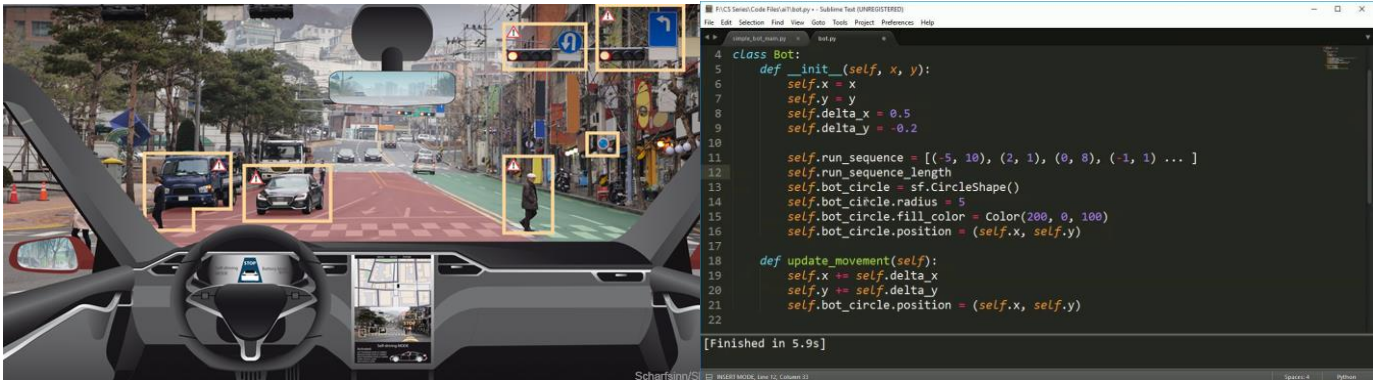
The final two courses in Python curriculum integrate Python with Raspberry Pi to create an AI device. The class will be given a choice of high-tech applications similar to Siri, Google Assistance, Alexa, security camera system with motion detection, etc. Students will work with the instructor and utilize online resources to create and execute a project of their choice.

AI – Raspberry Pi

In-centre only

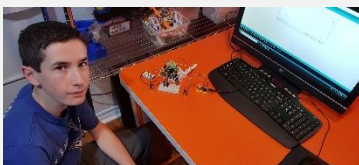
PY-10: RASPBERRY PI CAPSTONE PROJECT II

Prerequisite: PY-09



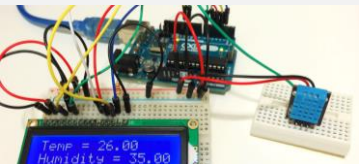
Students continue to work to complete their Capstone Projects which will be submitted at the end of this course for graduation. At this point, students are fully capable of creating simple to medium-complexity AI devices using Python and Raspberry Pi. During graduation, students will receive their diploma in Python & Artificial Intelligence!

What's Next?



ALUMNI WORKSHOP

A project-based environment where students apply their learned skills and work on their creations (products, programs and apps) under the supervision of a workshop manager. Ideal for students who are interested in working on their ideas and projects.



ARDUINO MICROCONTROLLERS

Learn to create electronic circuits and program the Arduino microcontroller to create your robots. Ideal for students who are interested in creating programmable devices and robots.



MECHANICAL DESIGN & 3D PRINTING

Learn mechanical design using Fusion 360 and create your parts using a 3D printer. Ideal for students who are interested in learning how to create custom mechanical parts for their projects.



INTERNET OF THINGS (IOT)

Coming Soon!