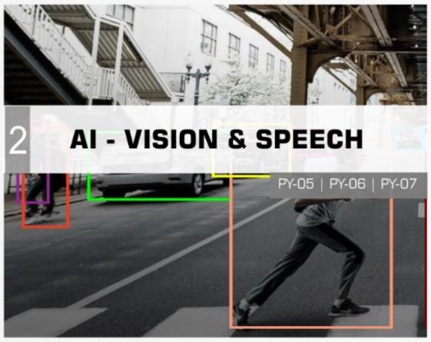
EXCEED
ROBOTICS

PYTHON PROGRAMMING

Game Development

AGES 12+

PYTHON & ARTIFICIAL INTELLIGENCE



The Ultimate Programming Curriculum for Future AI

The Python & AI curriculum introduces Python programming through game development before moving onto 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 Artificial Intelligence.

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

Game Programming courses (PY-01 to PY-04) are offered on the Virtual Classroom which will provide students with a solid introduction to Python. AI courses are available in-centre only.

PROGRAM DETAILS

8-month curriculum (online)

Classes once per week

One-hour in duration

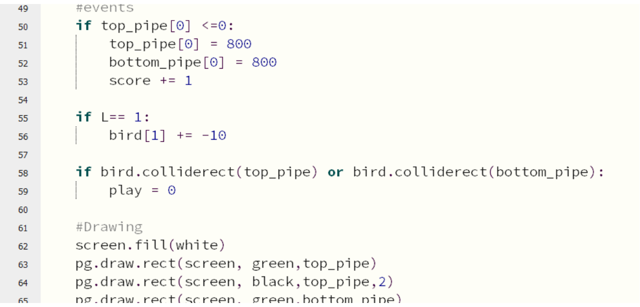
8 students max per class

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 career in STEM

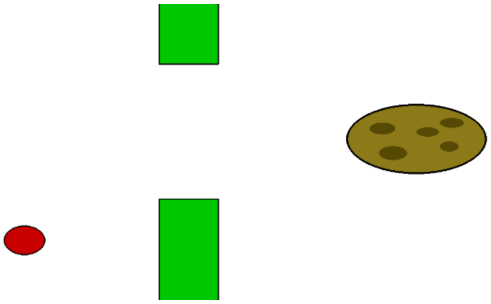
PYTHON GAME PROGRAMMING

PY-01: INTRODUCTION TO PYTHON GAME PROGRAMMING

Prerequisite: ages 12+ (no experience required)



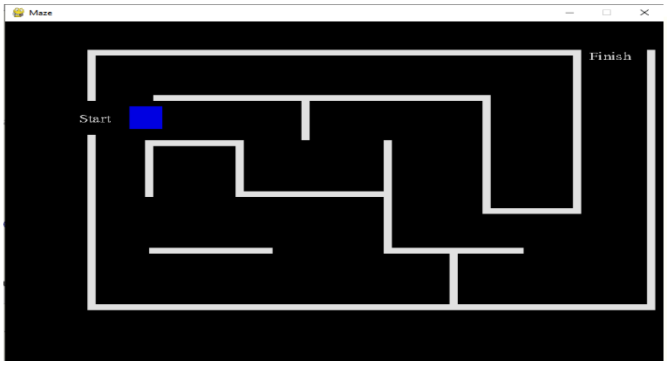
```
#events
if top_pipe[0] <=0:
    top_pipe[0] = 800
    bottom_pipe[0] = 800
    score += 1

if L== 1:
    bird[1] += -10

if bird.colliderect(top_pipe) or bird.colliderect(bottom_pipe):
    play = 0

#Drawing
screen.fill(white)
pg.draw.rect(screen, green,top_pipe)
pg.draw.rect(screen, black,top_pipe,2)
pg.draw.rect(screen, green,bottom_pipe)
```

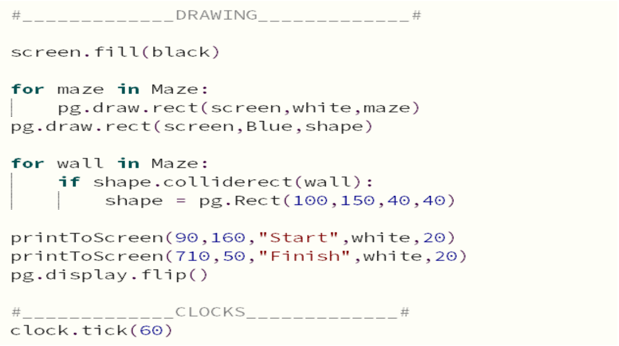


Designed to introduce student to coding in Python through a fun application - Game Programming. Students learn programming fundamentals using an object-oriented language. Concepts such as variables, if statements, while loops, mouse input, and basic shapes are taught in the first course. Students complete the course by creating a simple game with Python.

PY-02: GAME PROGRAMMING FUNDAMENTALS I

Prerequisite: PY-01





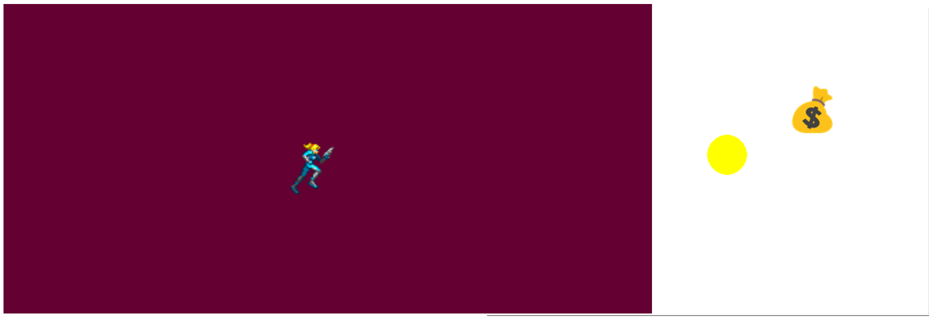
```
#_____________DRAWING_____________#

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)
```

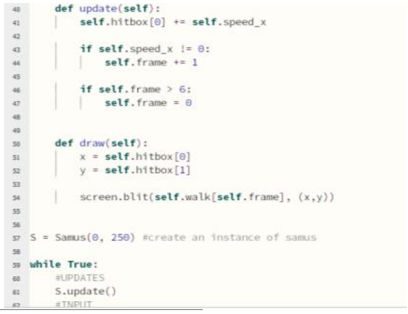
Students are introduced to more advanced concepts including Functions, Lists and Keyboard input. The material covered in this course could be extended to any object-oriented programming language and serve to advance our game development skills. Students will complete the course by creating a new game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

PYTHON GAME PROGRAMMING

PY-03: GAME PROGRAMMING FUNDAMENTALS II

Prerequisite: PY-02



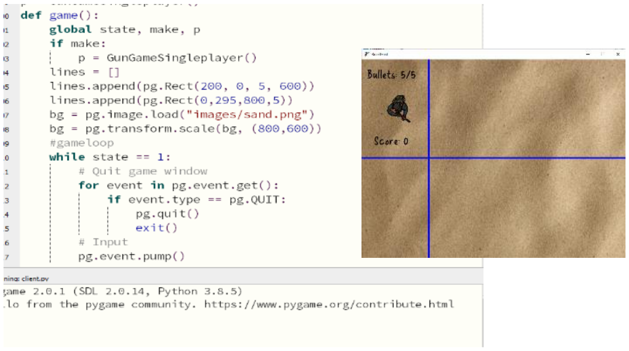


```
    def update(self):
        self.hitbox[0] += self.speed_x

        if self.speed_x != 0:
            self.frame += 1

        if self.frame > 6:
            self.frame = 0


    def draw(self):
        x = self.hitbox[0]
        y = self.hitbox[1]

        screen.blit(self.walk[self.frame], (x,y))


S = Samus(0, 250) #create an instance of samus

while True:
    #UPDATES
    S.update()
    #INPUT
```

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



```
def game():
    global state, make, p
    if make:
        p = GunGameSingleplayer()
    lines = []
    lines.append(pg.Rect(200, 0, 5, 600))
    lines.append(pg.Rect(0,295,800,5))
    bg = pg.image.load("images/sand.png")
    bg = pg.transform.scale(bg, (800,600))
    #gameloop
    while state == 1:
        # Quit game window
        for event in pg.event.get():
            if event.type == pg.QUIT:
                pg.quit()
                exit()
        # Input
        pg.event.pump()
```

ning: client.py

```
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.

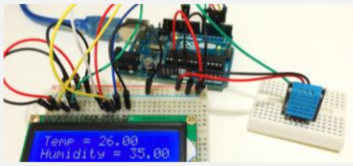
What's Next?

WEB DEVELOPMENT PROGRAMMING

Teaches HTML, CSS and JavaScript programming languages for front-end web development to students ages 11+. These courses were developed in partnership with a team of web developers from the University of Toronto

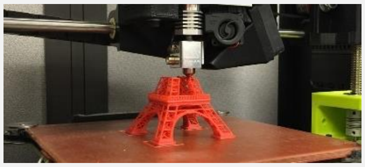
ARTIFICIAL INTELLIGENCE COURSES [In-Centre Only]

The continuation of our Python curriculum where students will start Artificial Intelligence (AI) courses including Image Manipulation, Object Detection, Speech Recognition and use a Raspberry PI to create advanced AI projects



ARDUINO MICROCONTROLLERS

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



MECHANICAL DESIGN & 3D PRINTING

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