



Python Game Programming

AGES 12+



1 : Introduction to Python

2 : Fundamentals of Game Programming

3 : Game Programming Application

The Ultimate Curriculum for Python Game Programming

The Python Game Programming Curriculum is designed to introduce learners to the exciting world of game development using the Python programming language. This comprehensive curriculum provides a step-by-step guide to creating interactive games, teaching fundamental programming concepts along the way. From basic game mechanics to advanced game design principles, students will acquire the skills needed to develop their own engaging games using Python.

PROGRAM DETAILS

8-month curriculum (online)

Classes once per week classes

One-hour in duration

6 max students 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

PY-01: INTRODUCTION TO PYTHON GAME PROGRAMMING

Prerequisite: ages 12+ (no experience required)

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.

```

"Guess the Number"
Programmed by Zachary Fruhling
Copyright 2020

import random
correctAnswer = random.randint(1, 100)
gameOver = False

while gameOver == False:

    playerGuess = int(input("Guess a number between 1 and 100: "))

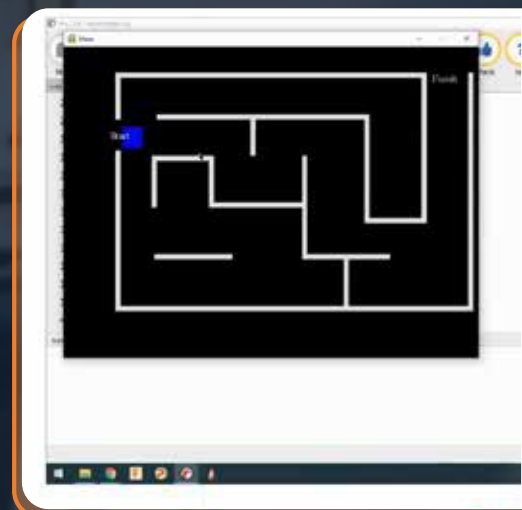
    if playerGuess == correctAnswer:
        compareAnswer = "Right"
        gameOver = True
    elif playerGuess > correctAnswer:
        compareAnswer = "High"
    elif playerGuess < correctAnswer:
        compareAnswer = "Low"

    if compareAnswer == "Right":
        print("Correct! You Win!")
    elif compareAnswer == "High":
        print("Too High! Guess Again!")
    elif compareAnswer == "Low":
        print("Too Low! Guess Again!")
  
```

PY-02: GAME PROGRAMMING FUNDAMENTALS I

Prerequisite: PY-01

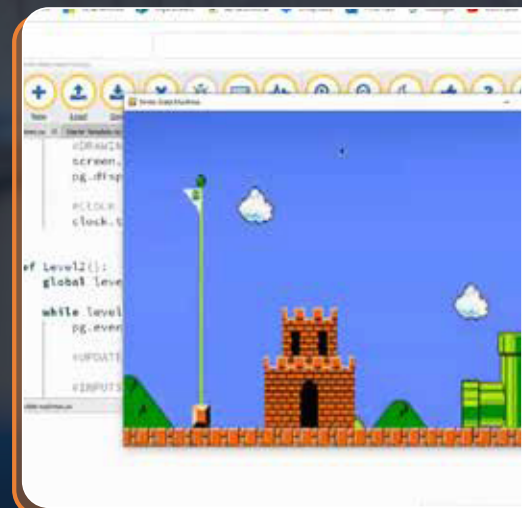
Students are introduced to more advanced concepts including Functions, Lists and Keyboard input. The material covered in this course could be extended to any object-oriented programming language and serve to advance our game development skills. Students will complete the course by creating a new game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



PY-03: GAME PROGRAMMING FUNDAMENTALS II

Prerequisite: PY-03

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: COD-03

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.

