

Python Game Development Course. Module 1

Learning goals are to learn the basics of programming in Python language and to create simple, 2D and 3D games using the Pygame library

Course Syllabus:

Day one

Introduction to Python

- What is a variable?
- Input and output functions
- Data types in Python
- Creating a text-based game

Learning outcome: students learned what a variable is, the print() and input () functions, got acquainted with data types in Python, and learned how to solve problems on the topic they have covered.

Practical task: create a text-based game.

Day two

Python operators

- Arithmetic operators in Python
- Comparison operators
- Logical operators
- Introduction to random numbers
- Create a game based on covered topic

Learning outcome: students learned arithmetic operators, worked with random numbers, as well as comparison operators, and learned how to solve tasks on the topic they have covered.

Practical task: create the Magic Forest game.

Day three

Conditional statements

- Introduction to conditional if-else statements
- Boolean values
- Solving tasks problems with conditional algorithms
- Learning the match-case statement
- Creating flowcharts

Learning outcome: students learned conditional and logical operators, learned how to work with the match-case statement, and how to solve tasks on covered topic.

Practical task: Create Text-based RPG game.

Day four

String variables

- Work with strings
- Introduction to slice notation
- String methods
- Creating the Cities game

Learning outcome: students will learn how to work with strings, methods for finding characters, permutations, finding the length of strings.

Practical task: Creating the Cities game.

Python Game Development Course. Module 2

Learning goals are to learn the basics of programming in Python language and to create simple, 2D and 3D games using the Pygame library

Course Syllabus:

Day one

Loops in Python

- Introduction to loops
- Studying looping algorithms
- Commands related to loops
- Nested loops
- Create a game based on covered topic

Learning outcome: Students learned how to work with loops in Python and the basic commands for working with loops.

Practical task: create Rock-paper-scissors game.

Day two

Arrays in Python

- Introduction to arrays
- Learning commands for working with arrays
- Creating a 2x3 matrix
- Create a game based on covered topic

Learning outcome: students learned how to create arrays and matrices, and mastered the basic commands for working with arrays.

Practical task: create Tic-tac-toe game.

Day three

Dictionaries and sets in Python

- Introduction to dictionaries and sets
- Learn basic commands for dictionaries and sets
- Processing and modifying data from dictionaries and sets
- Create a game based on covered topic

Learning outcome: students learned how to create dictionaries and sets, learn their main methods and commands.

Practical task: Create Quiz game.

Day four

Functions in Python

- Learn how functions work
- Return values
- Introduction to function parameters and arguments
- Create a game based on covered topic

Learning outcome: students learned how to work with functions, learn how to create their own functions, and learn what the return value, parameters, and arguments of functions are.

Practical task: Create Math Quiz game.

Python Game Development Course. Module 3

Learning goals are to learn the basics of programming in Python language and to create simple, 2D and 3D games using the Pygame library

Course Syllabus:

Day one

Classic algorithms

- Learning sorting algorithms
- Introduction to search algorithms
- Study implementation of mathematical algorithms (calculating LCM and GCD)
- Creating a game using algorithms

Learning outcome: students learned the implementation of classical algorithms in Python.

Practical task: create Matching pairs game.

Day two

Working with text files in Python

- Open command
- Processing text files
- Save values and writing to files commands
- Create a game using files

Learning outcome: students learned how to work with files and basic commands and modes of working with files, learned how to save game values into a file.

Practical task: creating Workers game.

Day three

Classes in Python

- Introduction to the concept of class
- Creating objects
- Learning the concepts of field and method
- Creating a game using classes

Learning outcome: students learned how to work with classes, learned how to create objects, and how to create methods.

Practical task: Creating Memo game.

Day four

Object-oriented programming (OOP) concept in Python

- Study of the concept and principles of OOP
- Introduction to inheritance
- Learn the concepts of polymorphism and encapsulation
- Creating a game using the knowledge learned

Learning outcome: students learned how to work with OOP, learned about inheritance, polymorphism, and encapsulation.

Practical task: Creating Sea World game.

Python Game Development Course. Module 4

Learning goals are to learn the basics of programming in Python language and to create simple, 2D and 3D games using the Pygame library

Course Syllabus:

Day one

Pygame Basics

- Introduction to the Pygame library
- Creating a screen
- Drawing geometric shapes
- Display text on the screen
- Creating a game using Pygame

Learning outcome: students learned what Pygame is, its functions and commands, and created the first game with a graphical interface.

Practical task: create an application with a graphical interface.

Day two

Creating a sandbox RPG game. Part 1

- Processing commands from the keyboard
- Output images in Pygame
- Creating the RPG game environment

Learning outcome: Students created their first game using Pygame and learned how to process keyboard keystrokes.

Practical task: create the RPG game environment.

Day three

Creating a sandbox RPG game. Part 2

- Handling colliding objects in Pygame
- Creating custom items to collect
- Implementation of "You win" and "Game over" situations

Learning outcome: Students will learn how to handle element touches and create collectable elements in the game.

Practical task: adding collected items and winning and losing conditions to the roam game.

Day four

Creating a sandbox RPG game. Part 3

- Learning how to work with time
- Adding boosters (potions) to the game
- Implementation of the life system

Learning outcome: Students learned how to work with time in Pygame, created potions, and implemented the life system to the game.

Practical task: final improvement of the RPG game.

Python Game Development Course. Module 5

Learning goals are to learn the basics of programming in Python language and to create simple, 2D and 3D games using the Pygame library

Course Syllabus:

Day one

Creating the Flappy Bird game. Part 1

- Creating new game environment
- Processing the display of the bird sprite, its animation
- Implementation of bird movement

Learning outcome: students created Flappy Bird game environment.

Practical task: create the basis of the game Flappy Bird: debug the movement of the platform, finalize the movement of the bird.

Day two

Creating the Flappy Bird game. Part 2

- Creating dynamic elements
- Collision handling with dynamic elements (pipes)
- Implementation of a collision and game over situation
- Implementation of the points system

Learning outcome: students improved the Flappy Bird game, added the pipes reappearance and interaction with them.

Practical task: tune parameters and improve the Flappy Bird game.

Day three

Creating the Flappy Bird game. Part 3

- Creating a level system
- Saving points to a file
- Creating a high score table
- Implementation of the start menu

Learning outcome: students finished Flappy Bird game, implemented saving points and the start menu.

Practical task: final improvement of Flappy Bird game.

Day four

Games Compilation and publication

- Introduction to the Pygbag library
- Compilation of games created using Pygame
- Publishing the game on the Internet

Learning outcome: students learned how to compile games and publish them on the Internet.

Practical task: Publish previously created games.

Python Game Development Course. Module 6

Learning goals are to learn the basics of programming in Python language and to create simple, 2D and 3D games using the Pygame library

Course Syllabus:

Day one

Creating a Platformer game. Part 1

- Creating Platformer game environment
- Implementation of the character and his movements
- Creating solid objects and platforms

Learning outcome: students learned the basics of physics in Pygame, created a character with gravity and solid objects.

Practical task: create the basis of the Platformer game.

Day two

Creating a Platformer game. Part 2

- Implementation of level movement
- Creating obstacles
- Implementation of various types of platforms
- Implementation of game over and game reload situations

Learning outcome: students learned the implementation of various types of platforms and implementation several game situations.

Practical task: implement the level system in the Platformer game.

Day three

Creating a Platformer game. Part 3

- Create patrolling enemies
- Implement character attack (shooting)
- Create moving enemies

Learning outcome: students learned how to create mobs (patrolling enemies), implemented character attacks.

Practical task: add enemies and attacks in the Platformer game.

Day four

Creating a Platformer game. Part 4

- Improvements implementation
- Points system creation
- Adding music and game soundtrack
- Writing game modifications and publishing the game

Learning outcome: students finished the game and publish it.

Practical task: Improvements to the Platformer game.

Python Game Development Course. Module 7

Learning goals are to learn the basics of programming in Python language and to create simple, 2D and 3D games using the Pygame library

Course Syllabus:

Day one

Creating Tower Defense game. Part 1

- Creating the Tower Defense game environment
- Implementation of shooting control
- Processing of mouse clicks and movement on the screen

Learning outcome: students learned how to process the position of the mouse on the screen and implement shooting control by clicking.

Practical task: create the Tower Defense game environment.

Day two

Creating Tower Defense game. Part 2

- Implementation of character movement and attacks
- Castle life system
- Implementation of multiple types of enemies
- Creating a health purchase system

Learning outcome: students implemented the attack and life system, as well as improvements in the game.

Practical task: add new mechanics to the game.

Day three

Creating Tower Defense game. Part 3

- Creating enemy shooter-archers
- Creating castle defenders
- Implementation of the level system in the game

Learning outcome: students implemented archers, added defenders, and implemented a level system.

Practical task: improve defenders and archers, as well as the level system.

Day four

Creating Tower Defense game. Part 4

- Improving the gameplay appearance
- Adding the start menu
- Adding sounds and music
- Compilation and publishing of the game

Learning outcome: students completed the game and published it on the Internet.

Practical task: publish the Tower Defense game on the Internet.

Python Game Development Course. Module 8

Learning goals are to learn the basics of programming in Python language and to create simple, 2D and 3D games using the Pygame library

Course Syllabus:

Day one

Creating 2D Shooter game. Part 1

- Creating 2D Shooter game environment
- Implementation of the character and its movement on the level
- Introduction to switching states of sprite animations
- Approaches to improving your code

Learning outcome: students created the basis for the 2D Shooter game.

Practical task: add animations of the character's movement and jump, adjusting the speed of his movement.

Day two

Creating 2D Shooter game. Part 2

- Adding shooting
- Implementation of grenade throws
- Creating an item collection system

Learning outcome: implemented shooting, throwing grenades, and collecting items.

Practical task: create new game mechanics in the game.

Day three

Creating 2D Shooter game. Part 3

- Implementation of life system
- Adding enemies
- Creating the basis for the game world

Learning outcome: added enemies and life systems, added a basis for the game.

Practical task: adding different types of enemies and improving the gameplay.

Day four

Creating 2D Shooter game. Part 4

- Implementation of the level map
- Setting up game world collisions
- Adding music and sounds
- Publishing the game

Learning outcome: 2D Shooter game publication.

Practical task: set up a game map with different types of blocks, add different sounds to 2D Shooter game.

Python Game Development Course. Module 9

Learning goals are to learn the basics of programming in Python language and to create simple, 2D and 3D games using the Pygame library

Course Syllabus:

Day one

Creating a 3D game in Pygame. Part 1

- Learning commands to create a 3D game in Pygame
- Exploring the basics of the level
- Writing commands for character movement
- Choosing of the final project idea

Learning outcome: Students learned how to create 3D games in Pygame and create the basis for a 3D game.

Practical task: create the basis and character of a 3D game.

Day two

Creating a 3D game in Pygame. Part 2

- Adding textures to a 3D game
- Implementing enemies
- Adding environments, life systems, and dialogs
- Creating a selection of implemented games for your portfolio

Learning outcome: students created a full-fledged 3D level and completed 3D game.

Practical task: create a full-fledged 3D game level in Pygame.

Day three

Preparing for project presentation

- Fixing previously studied topics
- Selecting the final project format
- Discussion and implementation of the final project
- Preparation of materials for the project presentation

Learning outcome: prepared final project and materials for project presentation.

Practical task: finish the selected game, complete the presentation, prepare the speech for the final project demonstration.

Day four

Project presentation

- Final preparation for the project presentation
- Presentations of the final project according to plan
- Discussion of the results of the work, questions and students answers
- Summing up the course results, outlining the study prospects

Learning outcome: the results are summed up and the course is completed

Practical task: presentations and discussion of created projects, projects analysis and feedback on the work on the course.