Python Gym

Python Gym is an advanced course providing teachers and students the opportunity to code in Python, a text-based computer language, using the CoderZ physically-based environment introduced in Cyber Robotics 102. While engaged in a series of activities with increasing complexity students learn the essential principles of text-based coding including syntax and debugging and continue to develop best programming practices for robotics including; motion planning, motion planning with sensors, physics, computational problem solving, and the importance of quality planning. Python Gym provides opportunities to use conditionals, variables, functions, modules and more.

CoderZ and supporting integrated content is sequenced to meet STEAM objectives and is designed to provide students authentic learning opportunities to increase interest and engagement while moving from acquisition of knowledge to transference of learning.

Standards, Skills, and Concepts

CSTA

Computer Systems

2-CS-02	Design projects that combine hardware and software components to collect and exchange data.
3A-CS-01	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

Algorithms and Programming

2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.
2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-14	Create procedures with parameters to organize code and make it easier to use.
2-AP-17	Systematically test and refine programs using a range of test cases.
2-AP-18	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
2-AP-19	Document programs in order to make them easier to follow, test, and debug.

3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-16	Demonstrate code reuse by creating programming solutions using libraries and APIs.
3B-AP-23	Evaluate key qualities of a program through a process such as code review.

Impacts of Computing

2-IC-20	Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.
2-IC-21	Discuss issues of bias and accessibility in the design of existing technologies.
2-IC-23	Describe tradeoffs between allowing information to be public and keeping information private and secure.
3A-IC-24	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
3A-IC-28	Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
3A-IC-30	Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.
3B-IC-27	Predict how computational innovations that have revolutionized aspects of our culture might evolve.

NGSS

MS-ETS1-2	Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.
MS-ETS1-3	Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

K-12 Computer Science Framework

Computing Systems

- Devices – Examine
- e the interaction between humans and computing devices. Evaluate and analyze the advantages, disadvantages, and recognize the unintended consequences.
- Troubleshooting - Deploy a structured process to troubleshoot problems within the system to ensure potential solutions are not overlooked and simple issues are resolved.
- Hardware and Software - Explore the interactions between hardware, software and user of a computing system and evaluate the flow of information.

Algorithms and Programming

- Algorithms - Design, test, and debug algorithms that are readable and easy to follow. Evaluate and select algorithms based on performance, reusability, and ease of implementation.
- Variables - Understand and use variables to represent and process data to produce varying outputs.
  Use data structures to effectively manage program complexity (based on functionality, storage, and performance tradeoffs).
- Control - Use loops, conditions, and other control structures to create more complex programs.
  Evaluate control structures and control structure combinations considering the tradeoffs related to implementation, readability, and program performance.
- Modularity - Use functions to make code easier to reuse and read.
  Use modules to develop and manage complex tasks/programs.

Networks and the Internet

- Cybersecurity - Explain the importance of protecting information sent and received across networks from unauthorized access and modification (encryption and Hypertext Transfer Protocol Secure - HTTPS).

Data and Analysis

- Collection - Discuss how using sensors with a robot to collect data regarding its environment and make decisions about the next steps to take is similar to the human process and how robots and humans interact.

CC Math

Mathematical Practice

MP1 Make sense of problems and persevere in solving them.
MP2 Reason abstractly and quantitatively.
MP3 Construct viable arguments and critique the reasoning of others.
MP4 Model with mathematics.
MP5 Use appropriate tools strategically.
MP6 Attend to precision.
MP7 Look for and make use of structure.
MP8 Look for and express regularity in repeated reasoning.

21st Century Skills

Learning and Innovation Skills

- Creativity and Innovation
- Critical Thinking and Problem Solving
- Communication and Collaboration

Information, Media and Technology Skills

- Information Literacy
- Information and Communications Technology (ICT)

Life and Career Skills

- Flexibility and Adaptability
- Initiative and Self-direction
- Social and Cross-Cultural Skills
- Productivity and Accountability
- Leadership and Responsibility

Computational Thinking Skills

- Decomposition
- Pattern Recognition
- Algorithm Design
- Abstraction

Robotics Concepts and Skills

- Motion Planning
  - Direction
  - Power/Speed
  - Distance
  - Duration
  - Brake
  - Arms
- Sensor-based Motion Planning
- Motors and Optical Encoders
- Sensors and Controllers
  - Touch - collision
  - Gyroscope - rotation and tilts (x, y, z)
  - Ultrasonic - distance
  - Light/Color - surface color/brightness