Riders

DEMOCRATIZING ROBOTICS EDUCATION

INTRODUCTION TO ROBOTICS COURSE

AWS EdStart Member

F1 TENTH

ROME Maker Faire THE EUROPEAN EDITION

Riders

Introduction to Robotics

Lesson 1: Introduction to Riders with “ScoutRider”

We will learn about making decisions in robotics coding.
Basic Python Commands: Simple Python commands to perform a task.
Algorithms: Connect multiple commands to form a sequence.
Problem Solving: Solve logic puzzles of increasing difficulty.
Simple Translation: Move from one point to another on a grid in unit steps.
Simple Rotation: Right and left turns as part of an algorithm.

Lesson 2: Programming Loops with “DirtRider”

We will learn about while loops.
While Loops: Implement while loops in Python .
Patterns: Recognition and implementation of patterns.
Algorithms: Connect multiple commands to form a sequence.
Simple Translation: Move from one point to another on a grid in unit steps.
Simple Rotation: Right and left turns as part of an algorithm.

Lesson 3: Conditionals with “HotRider”

We will learn to use conditional expressions in robotics coding.
Conditionals: Implement if and elif statements in Python.
Algorithms: Implement adaptive algorithms which respond to current conditions.
Simple Translation: Move from one point to another on a grid in unit steps.
Simple Rotation: Right and left turns as part of an algorithm.

Lesson 4: For Loops with “GarbageRider”

We will learn about for loops.
For Loops: Iterating over a sequence a certain number of times
Refactoring: Improving code efficiency.
Patterns: Recognition and implementation of patterns for algorithm development.
Translation: Move from one point to another on a grid using fractional steps.
Coordinates: Recognize the plotted points on an obstacle course.
Rotation: Rotate using radians to implement simple left/right turns.

Lesson 5: Pathfinding with “LabRider”

We will learn about path finding and flood-fill algorithms.
Double For Loops: Implement nested for loops.
While Loops: Implement while loops in python.
Path Finding: Implement algorithms to find the shortest path on a grid.
Algorithms: Implement adaptive algorithms which respond to current conditions.
2D Coordinates: Work with data assigned on a 2D grid.

Riders

Introduction to Robotics

Lesson 6: Feedback with "WallRider"

We will learn about feedback and continuous-time commands.
Translation: Control robot velocity using meters/second.
Rotation: Control robot angular velocity using radians/second.
Feedback Algorithms: Implement feedback to create a stable control algorithm.
Sensors: Use a distance sensor as an input to an algorithm.
Optimization: Tune an algorithm to improve a result.

Lesson 7: Image Processing with "LineRider"

We will learn about arrays and continue to improve our skills with feedback algorithms.
Translation: Control robot velocity using meters/second.
Rotation: Control robot angular velocity using radians/second.
Arrays: Work with a 1D array in python.
Image Processing: Read a 1D camera image and interpret the pixel data.
Feedback Algorithms: Implement feedback to create a stable control algorithm.

Lesson 8: Introduction to Maze Problems with RodentRider

We will discover new algorithms to see how we can escape from a maze, by trying different strategies from random movements to following walls and pledge algorithm.
Random Numbers: Implement a random search algorithm.
Algorithms: Implement adaptive algorithms which respond to current conditions.
Left Wall Following Algorithm: Implement an algorithm to solve a maze.
Pledge Algorithm: Implement an algorithm to solve a maze which has internal islands.

Lesson 9: Force and Motion with NewtonRider

We will apply Newton's physics rules into robotics and use our knowledge to create an algorithm with a high efficiency.
Arrays: Work with a 2D array in Python.
Color Image Processing: Interpret red, green, blue (RGB) pixel data.
2D Image Processing: Interpret the pixels of a 2D image.
Algorithms: Implement adaptive algorithms which respond to current conditions.

Lesson 10: Recycling with GreenRider

We will use image processing to separate different products and create an algorithm to execute actions according to the product type.
Kinematics: Understand how rotation of a joint influences other joints down the chain.
Algorithms: Connect multiple commands to form a sequence.
Radians: Choose specific angles in radians to accomplish goals.
Geometry: Solves problems of triangles and angles using geometry.
Sin and Cos: (Optional) Use sin and cos to solve advanced problems.

Riders

Introduction to Robotics

Lesson 11: Coordinate Systems with EcoRider

We will use coordinate system and make strategic decisions to complete tasks in planting sites.
We will also work with angles to determine the most efficient path to target positions.
F = ma: Solve physics problems using Newton's second law.
Acceleration Equation: Solve a problem using the relationship between acceleration and position.
Graphing: Make graphs of data to get insight into equation behaviors.
Sin and Cos: Use sin and cos to solve advanced problems.

Lesson 12: Introduction to Kinematics with JointRider

We will combine our knowledge about angles and use kinematic equations in order to calibrate a 4-axis robot arm with multiple joints and a base.
Queues: Interpret a list of data in python which changes over time.
Feedback Algorithms: Implement feedback to create a stable control algorithm.
3D Coordinates: Motion in XYZ space.
Angles: Determine the angle of a vector to solve more advanced problems. Calculate the length of a vector.

Lesson 13: Control in 3D Space with ZeroGRider

We will work with image feedbacks to guide our satellite into a dock. We will use image processing to place the target, and move around 3D Space to successfully place our satellite.
Sensors: Use a distance sensor as an input to an algorithm.
Feedback Algorithms and Damping: Stable feedback in a zero friction environment.
Color Image Processing: Interpret red, green, blue (RGB) pixel data.
3D Coordinates: Motion in XYZ space.

Lesson 14: Angular Motion with ZeroGRider

We will learn the angular motion in 3D space and try to dock our robot while controlling our robot with yaw and pitch.
Color Image Processing: Interpret red, green, blue (RGB) pixel data.
Feedback Algorithms and Damping: Stable feedback in a zero friction environment.
Sensors: Use a distance sensor as an input to an algorithm.
3D Rotation: Rotation in 3D space using yaw and pitch.
Rotation: Control robot angular velocity using radians/second.

Lesson 15: Autonomous Driving with AutoRider

We will try to drive around autonomously in a road where there are other vehicles. We will develop autonomous driving algorithms to avoid any collisions.
Synthesis of Learning: This lesson is not designed to teach a specific new concept. It is designed to synthesize many of the elements introduced in prior lessons.
Feedback Algorithms: Implement feedback to create a stable control algorithm.
2D Image Processing: Interpret the pixels of a 2D image.