Assignment 4: Custom Obstacle Course Challenge

Objective

Design and build a custom obstacle course for the BaseBot and program the robot to navigate through the course using Python. **This semester, you will also use GitHub to organize and submit your project.**

Instructions

Part 1: Obstacle Course Design and Construction

Design Your Course:

- Sketch out a layout for your obstacle course **on paper or digitally**.
- The course must cover an area of at least **4x3 tiles**.
- Include **at least 5 different obstacles** that your BaseBot must interact with or navigate around.

Build the Course:

- Using the provided materials or making your own, construct your obstacle course based on your design.
- Ensure that **each obstacle is securely placed** and follows your planned layout.

Document Your Setup:

- Take a **clear picture** of your completed obstacle course.
- **Label each obstacle** and describe its **challenge and purpose** in your project documentation.
- **Upload these images and descriptions to GitHub** as part of your project repository.

Part 2: Programming the BaseBot

Code Development:

- Visit the **VEX Coding Platform** and select **Python** as your programming language.

- Write a program that allows your BaseBot to **navigate through the entire course**, interacting with all the obstacles as designed.

Code Documentation and Testing:

- **Include comments** in your code to explain the purpose of each section.
- **Test your program** to ensure that the BaseBot can complete the course as intended.
- **Capture a screenshot** of your complete code **and upload it to your GitHub repository**.
- Highlight key sections of the code where specific **maneuvers or interactions with obstacles occur** in your documentation.

Part 3: GitHub Repository Setup & Submission

Setting Up Your Repository:

1. **Create a new repository** on GitHub named **"BaseBot_Obstacle_Course"**.
2. **Organize your repository** with the following folders/files:
    a. 📁 **design/** → Contains your **initial sketch** and **course description**.
    b. 📁 **images/** → Includes a **photo of your completed obstacle course**.
    c. 📁 **code/** → Stores your **Python program** (main.py) and any additional files.
    d. 📄 **README.md** → Provides an overview of your project, including:
        i. Course **description** and **obstacle details**.
        ii. How your **BaseBot interacts with each obstacle**.
        iii. How to **run your Python script**.
        iv. Reflection on **what worked well** and **what challenges you faced**.

Final Submission:

- Submit the **link to your GitHub repository** on **the class submission portal**.
- Ensure that **all files are properly uploaded and organized** before submitting.

Evaluation Rubric

Criteria	Excellent (4)	Good (3)	Satisfactory (2)	Needs Improvement (1)	Unacceptable (0)
Course Design & Construction	Course is creative, well-constructed,	Course is well-constructed with a	Course meets minimum size and obstacle requirements	Course setup is incomplete or poorly constructed.	Does not meet minimum requirements

	and includes diverse and challenging obstacles.	few creative obstacles.	but lacks creativity.		for size or number of obstacles.
Programming Execution	Code effectively navigates all obstacles with efficient and clear programming.	Code navigates most obstacles with minor issues in efficiency or clarity.	Code completes the course but lacks sophistication or has errors.	Code has significant errors, making course completion sporadic or incomplete.	BaseBot does not complete the course due to poor programming.
GitHub Repository & Documentation	Repository is well-organized with clear folder structure, README, and annotations.	Repository is mostly organized with minor missing details.	Basic organization with README but lacks detail.	Repository is incomplete or missing key components.	Repository is not set up or missing all required files.