**VEX ROBOTICS:**

Autonomous: 15 seconds

Major features we will need to implement.

* Robot movements
* Robot avoidance
* Ball detection and ball movement to goal
* Finishing extra tasks

User controlled: 1 minute 45 seconds.

* User controlled Robot movement
* User controlled ball pushing
* Picking up the triball
* Lifting the robot itself

Current resources:

<https://vr.vex.com/>

<https://api.vexcode.cloud/v5/class/classvex_1_1triport_1_1port>

**Naming conventions for git branches:**

‘LastName-Feature’

**Python Standards:**

* All non-private variable and method names will be in snake\_case (e.g., def method\_header\_name():).
* All private variable and method names will follow snake\_case, preceded by a single underscore (e.g., def \_method\_header\_name():).
* Unless using type hints, all arrays will be explicitly typed (e.g., string\_array: List[str]).
* Use try-except blocks to handle most, if not all, exceptions.
* To concatenate strings, use f-strings (e.g., your\_name = f"{first\_name}, {last\_name}").
* Use explicit typing for variables when the type is not obvious from its assigned value.
* Use object initializers when creating new objects to make it clear what values are assigned (e.g., fido = Dog(age\_in\_years=3, favorite\_activity="Barking", favorite\_food="Steak")).
* Use implicit typing in for loops (e.g., for i in range(10):).
* Write comments following standard English conventions with a space after the comment delimiter (e.g., # This is a valid comment.).
* Ensure method and variable names are self-explanatory (e.g., instead of def do\_something(x, y):, use def sum\_of\_integers(num1, num2):).
* Avoid nested if statements and prefer switch statements when possible.