VEX IQ Teacher Facilitation Guide
Castle Crasher Coding Activity

This Facilitation Guide offers a step-by-step guide for how to facilitate as your students complete the Castle Crasher Coding Activity with VEX IQ. You know your students best, so tailor your teaching and implementation to best suit your students. The activity is designed to be flexible so that you can meet students where they are, giving them the time, space, and instruction necessary to make the most of their learning.

Overview of Universal Design for Learning (UDL)

The UDL Guidelines, created by CAST, provide a comprehensive framework for Universal Design for Learning—an approach that enhances teaching and learning for everyone. Incorporating UDL into all lesson plans and learning experiences is a powerful way to support learning for all.

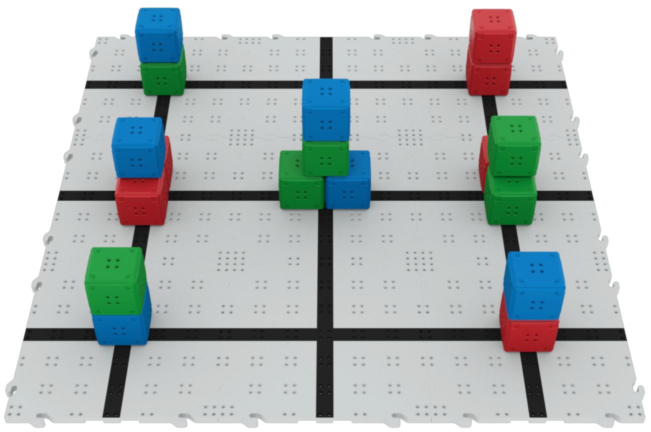
Throughout this document, you'll find "**UDL Move**" callouts that highlight specific areas where UDL principles can be applied to increase student agency and help provide an equitable learning experience. While many UDL Moves are integrated into the Activity itself, these UDL Move callouts offer specific actions you can take as a teacher to improve all students' learning as you implement this Activity.

For deeper exploration of specific UDL principles, you can access detailed resources through embedded links (such as "Engagement - Optimize choice and autonomy (7.1)").

Overview of Castle Crasher

Code the robot to knock over castles and clear all the Cubes from the Field!

- Castle Crasher is played on a 3x3 Field without walls.
- There are 18 Cubes used to build "castles" on the Field in the orientation shown below. The color of the Cubes does not matter.
- To learn more about the challenges for the Castle Crasher activity, see this reference sheet.



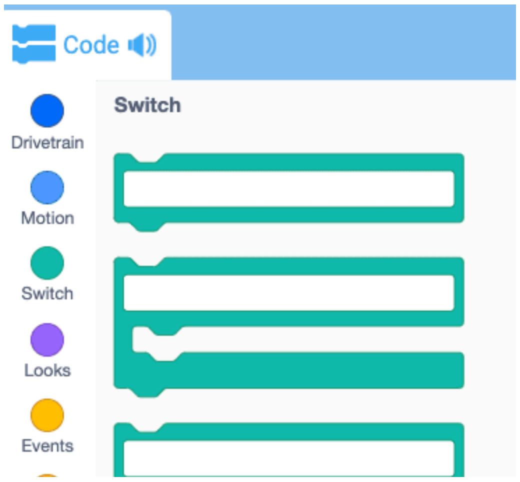
Standard: CSTA 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs

Prepare for the Activity

- Be sure you have access to a printer so that students can print their certificates at the completion of the activity.
- Build and lay out the Field (shown above).
    - When thinking about placement in your classroom, be sure to leave approximately 6 inches of space around each side of the Field. This will allow space for the Cubes to be pushed off of the Field.
- **Try the activity yourself!** A brief experience with the activity will allow you to field questions and support students.
- Decide how your students should be grouped for this activity. If the groups will be predetermined, organize this ahead of time.

UDL Move:
Engagement - Foster collaboration, interdependence, and collective learning (8.3): Encourage students to work in groups to complete the Castle Crasher activity in order to build their collaborative problem solving and teamwork skills, and to help students recognize how individuals' unique contributions can be combined to create stronger solutions.

- Ensure students will have access to their VEX IQ Kits and a device that supports VEXcode IQ.
    - Ensure Batteries are charged before the activity.
    - For more information about accessing VEXcode IQ, view this page.
- Decide how you would like to distribute the reference sheet for this activity. If you are providing it digitally, decide how you will distribute the link. If you are printing them out and handing them to students, have them printed ahead of time.
- **Decide how students will code in VEXcode IQ. Students can use Blocks, C++, Python, or Switch to transition between blocks and text coding.**
    - Switch blocks enable students to incorporate Python commands into their VEXcode IQ Blocks projects, creating a bridge that eases the transition from block-based to text-based coding with Python. Switch blocks have the same shapes as other blocks in the VEXcode IQ Toolbox, but have a text field in which students can enter Python commands.
    - **Do the following to set the stage for using Switch with your students:**
        - Read this article to learn more about how Switch can be used to aid students
        - Watch this following video with your students to learn what Switch is and why it is helpful
            - Why Use Switch? video link
        - Watch this video with your students to learn how to use Switch in VEXcode IQ
            - How to Use Switch Blocks video link



UDL Move:
Engagement - Optimize choice and autonomy (7.1): Students can explore Python through Switch blocks and decide how to transition from block-based to text-based coding at their own pace.

Representation - Connect prior knowledge to new learning (3.1): Switch blocks act as a bridge to help students connect their prior knowledge of block-based coding to learning new Python commands.

- Check out the IQ Leaderboard! Think about how you can use the leaderboard to motivate your students. They can see the highest scores of their class and challenge themselves to try to make it into the top. You may want to project the leaderboard to motivate students as they play the game.

Teaching the Activity

This section will guide you through the steps to implement this activity in your classroom. Step by step instructions are provided for how to introduce the activity to your students and what direct instruction is recommended before having students complete the challenges within the activity.

1) Introduce the activity to your students.
    a) Let them know that they will be coding a VEX IQ robot to knock down castles! First, students will build the robot, then they will learn basic commands to make the robot move, and then code it to drive and knock down the castles.
    b) Assign students to their groups and have them construct the Speed Build.

Note: Using VEX IQ (1st generation)? Have students build the Standard Drive Base (steps 1-19).

2) Have students launch VEXcode IQ. They can access it using the application or via a chrome browser at codeiq.vex.com.

3) Tell students that they must first understand how the robot moves before they can participate in the activity. Guide them through the following instructions to give students a basic understanding of how to use the Drivetrain commands:
    a) In VEXcode IQ, have students open the BaseBot (Drivetrain 2-motor) template.
    b) Drag in a [Drive for] block and connect it to the {When started} block.
        - Explain that this block is used to drive the robot forward or in reverse a specified distance.
    c) Connect the robot to your device (app based or web based).
    d) Download and run the project.
    e) Show students that you can change the parameters of the [Drive for] block (forward/reverse, distance, and units of measurement).
    f) Drag in a [Turn for] block and attach it after the [Drive for] block.
        - Explain that the [Turn for] block is used to turn the robot left or right a specified distance.
    g) Reconnect the robot to your device, and re-download and run the project to test.
    h) Drag in a [Set drive velocity] block and attach it after the {When started} block.
        - Change the parameter from 50% to 100%.
        - Explain that the [Set drive velocity] block sets the speed of the Drivetrain.
    i) Reconnect the robot to your device, and re-download and run the project to test.

4) Next, explain the details of the activity to the students. Provide them with this reference sheet that they can use while participating in the activity.
    a) Tell students to start with the first challenge, which is to knock over all of the castles on the Field. The castles (Cubes) can be pushed off of the Field, but do not have to be, they just have to be knocked down (not stacked).

        - Use the Field and Cubes to show students what it looks like to knock a castle over.
    b) Share the additional rules they should follow while completing the challenge:
        - The robot can start from any location on the Field where there is not currently a castle.
        - If the robot falls off of the Field, stop the project and try again.
        - Reset the castles each time the project is run.
    c) Once students have their robot built, understand the basic Drivetrain commands to make it move, and are familiar with the activity, allow them to work on the activity in their groups.
    d) Tell students to start by sketching out the Castle Crasher Field on a sheet of paper and planning the path of their robot.
        - Students can complete the path planning via sketching or through digital methods like creating a graphical representation of the Field or describing the path of the robot in written directions.

UDL Moves:
Action & Expression - Use multiple tools for construction, composition, and creativity (5.2): Use different media methods to help students as they plan the path for their robot on the raised Field. This could include drawn versions, graphical representations with computer-aided software, or through written descriptions of the path of the robot.
Action & Expression - Set meaningful goals (6.1): While reviewing the activity with students, be sure they understand the goal of the activity and the clear, measurable objectives that are outlined on the activity sheet.

Note: Groups may move at different paces. One group may work on the first challenge for an entire hour, while other groups have completed challenge one and are moving on to challenge two. Walk around the room as groups are actively working to gauge their progress and answer any questions they may have.

5) As students complete the first challenge, have them move on to the second challenge. The second challenge is to clear all of the Cubes off of the Field. This means that the robot will push the Cubes out of the 3x3 Field so that no Cube remains in contact with the Tiles.
    a) Use the Field and Cubes to show students what it looks like to clear a Cube completely off of the Field.
    b) For an additional challenge, time students to see how many Cubes they can clear off of the Field in two minutes.

Facilitating Discussions During the Activity

Castle Crasher is designed to be an open-ended coding exploration. To help students stay focused and learn from and with one another, engage them in frequent coding conversations as they are working.

Here are some questions to encourage iteration and problem solving in your students as they progress through the game:

- Where are you starting your robot and why?
- How can you improve your project to knock down more castles?
- What is your strategy to ensure you hit all of the castles?
- How are you using the [Drive for] block in your project?
- How are you using the [Turn for] block in your project?
- What went well with your project? What do you need to improve for your next try? How will you create code to make your project more effective?

- What did not go well for your robot this time? What can you learn from this that you can improve on for the next run?
- Remind students of Pair Programming practices that can help them to work collaboratively on the challenge. View this article to learn more about Pair Programming.

UDL Move:
Engagement - Offer action-oriented feedback (8.5): Use the prompts provided above to provide timely, constructive, and goal-oriented feedback that encourages a growth mindset and supports sustained engagement.

After the Activity

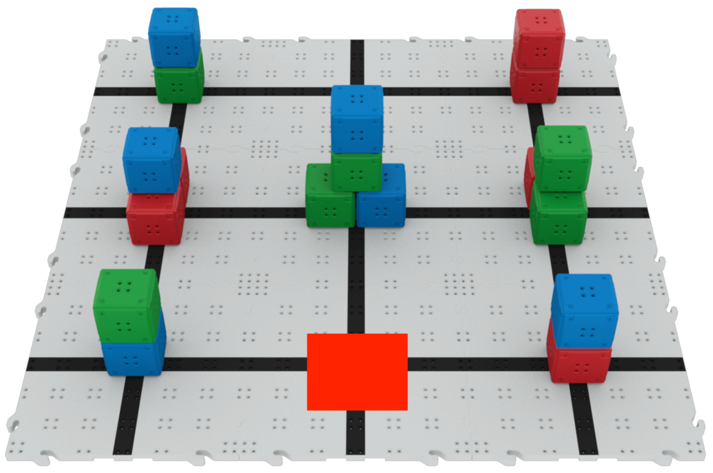
Celebrate success! After the activity has ended, celebrate with your students! Choose one or two of these ideas to share your success with others!

- Print out certificates for each student and display them in your classroom!
- Create a classroom display using the data from the students' certificates, and add up the number of Cubes knocked over or off of the Field. Play the game over multiple days, and create a graph showing student improvement over time.
- Share photos and videos of your classroom event on social media. Tag @VEXRobotics so we can celebrate with you!
- Have students take screenshots of the coding projects they are proud of and write a sentence or two about why they feel good about that particular project. Display them around the classroom.

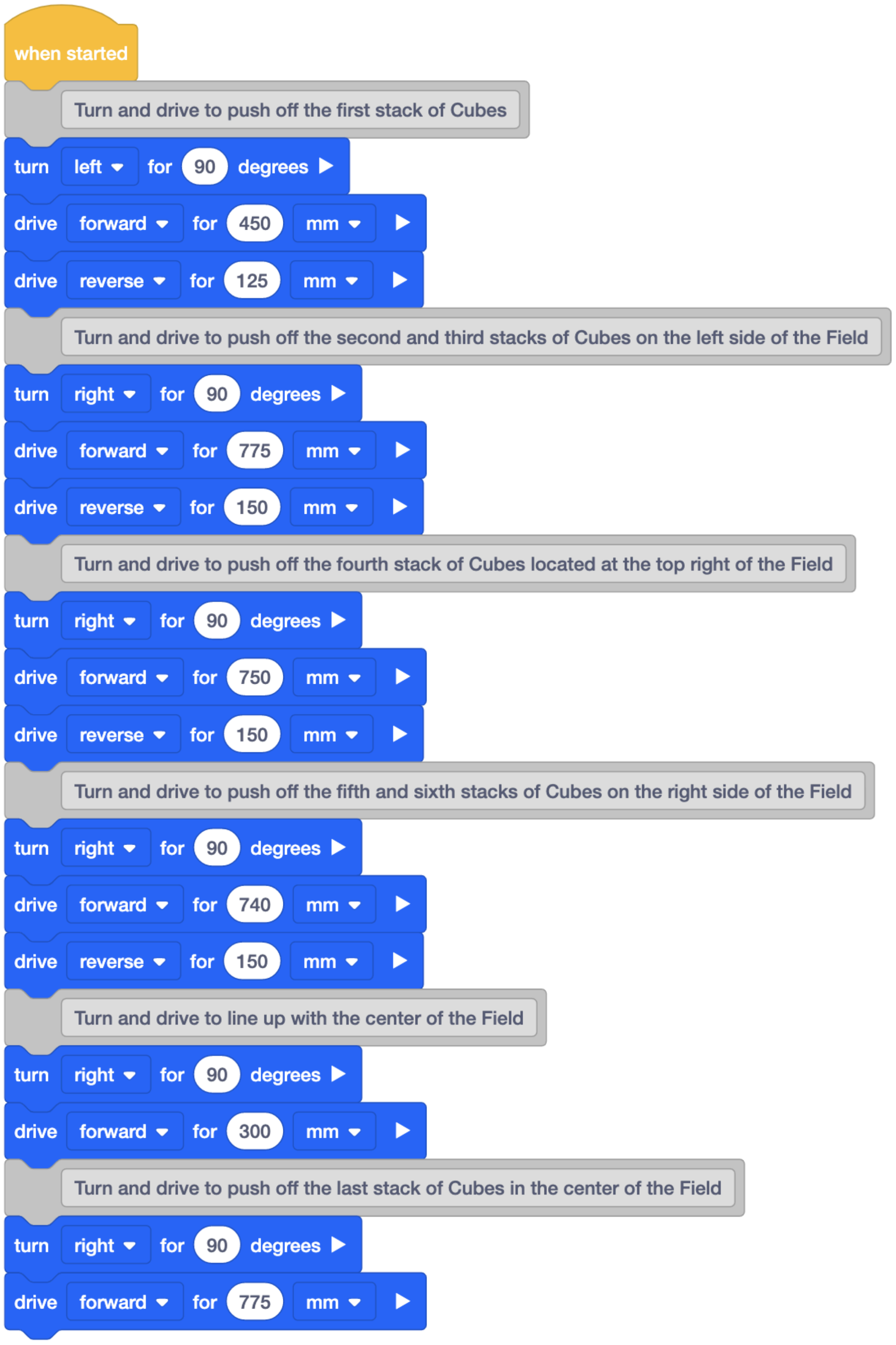
Sample Solution

While there are many different ways to complete this activity, you and your students may need inspiration for where to get started. If needed, share the code below with your students in chunks or use it to help guide them through some of the logic.

The robot's start location for this sample solution is shown here using the red box. Start locations can vary depending on the project.



This code includes [Comment] blocks, which are used to organize the code and plan the path.



```
when started
Turn and drive to push off the first stack of Cubes
turn left for 90 degrees ▶
drive forward for 450 mm ▶
drive reverse for 125 mm ▶
Turn and drive to push off the second and third stacks of Cubes on the left side of the Field
turn right for 90 degrees ▶
drive forward for 775 mm ▶
drive reverse for 150 mm ▶
Turn and drive to push off the fourth stack of Cubes located at the top right of the Field
turn right for 90 degrees ▶
drive forward for 750 mm ▶
drive reverse for 150 mm ▶
Turn and drive to push off the fifth and sixth stacks of Cubes on the right side of the Field
turn right for 90 degrees ▶
drive forward for 740 mm ▶
drive reverse for 150 mm ▶
Turn and drive to line up with the center of the Field
turn right for 90 degrees ▶
drive forward for 300 mm ▶
Turn and drive to push off the last stack of Cubes in the center of the Field
turn right for 90 degrees ▶
drive forward for 775 mm ▶
```