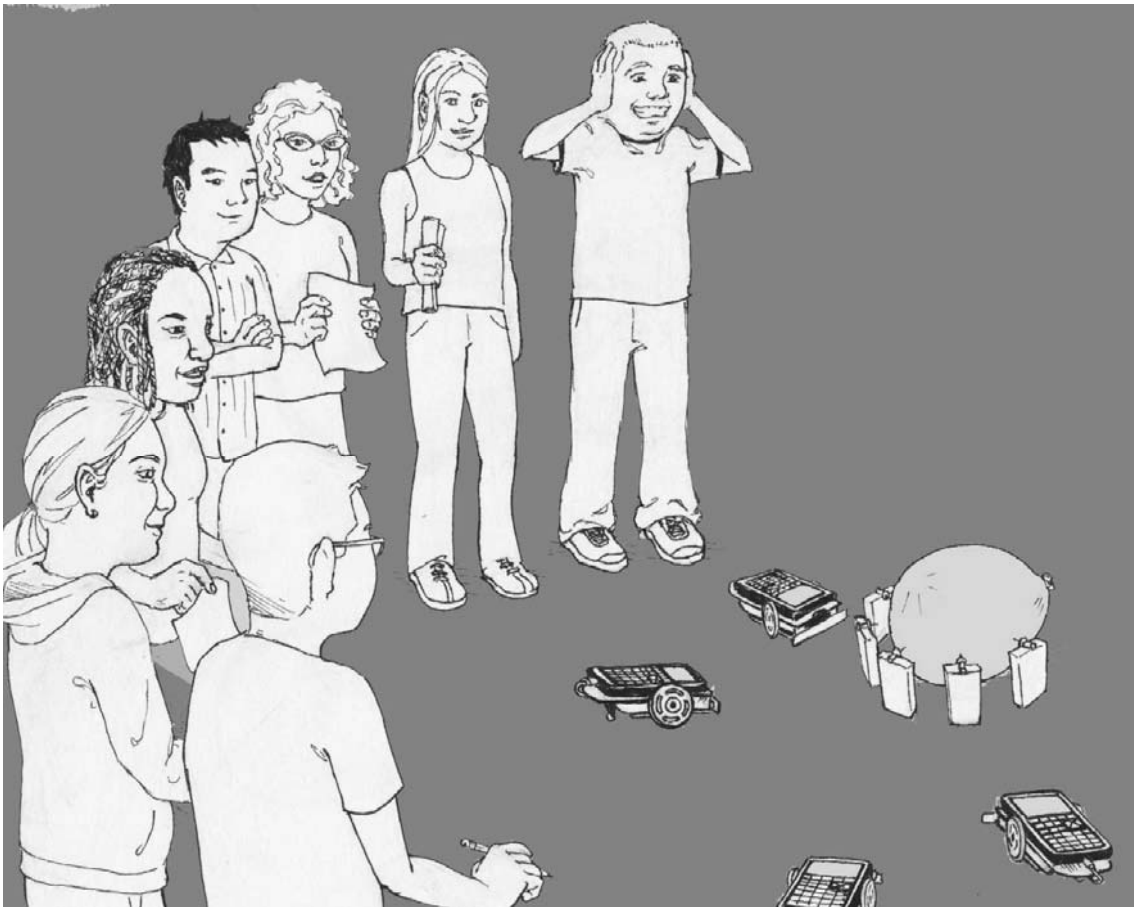
Name _____ Date _____

Graph and Predict Mission 2

Your second mission, should you decide to take it (and you know you will), is to come as close to crashing your robot into an object as possible, but without actually hitting it.

YOU NEED:

1 Norland Calculator Robot (Wheels)
1 TI-83 or TI-83 Plus Graphing Device (Brains)
1 Meter Stick
Graph Paper
Program: **MISSION2**



INSTRUCTIONS:

Use a meter stick with your robot and the **GO** program from Mission 1 to obtain data for the table below:

Table 1

Time (In seconds)	**Distance** (In meters)
	1
	1.5
	2
	2.5
	3

Graph the data as points on graph paper with TIME on the horizontal or x-axis and DISTANCE on the vertical or y-axis. Draw the best-fitting line that most closely follows the pattern shown by your data points.

HOW GOOD IS YOUR GRAPH?

Write a simple program (see **PROGRAMMING INSTRUCTIONS** if needed) for your robot on a TI-83/TI-83 Plus graphing device named **MISSION2**:

PROGRAM: MISSION2
: randInt (1,15)->X

: Disp X
: Pause
: X*100->T
: Send ({122,T})
: Get (R)
: Stop

This program will randomly pick a number from 1 to 15. This number represents the time in seconds the robot will be instructed to travel forward. The program will pause while you use your graph to predict the distance the robot will travel. Record your prediction in the table below, then press **ENTER** on the graphing device and measure the actual distance. Record your degree of error.

Table 2

Time (In seconds)	Prediction (In meters)	Actual	Error

ADVANCED:

Reading from your graph can help you predict how far the robot will travel forward for a given time. What if a time is given that exceeds the limitations of your graph? Is there a more accurate way to predict or calculate the expected distance?

Find the slope of your best-fitting line on the graph. What does the slope represent?

Write an equation for your line using the slope-intercept form y=mx+b.
What does b equal?

Graph this equation on your TI-83/TI-83 Plus graphing device. Press the **TRACE** button and confirm your values from table one. For any given Y value (distance) there will be an X value (time).

Rewrite your slope-intercept equation above, substituting d for y, r for m, and t for x. Does it look familiar? This equation can be used to predict the time needed for the robot to travel any given distance.

CHALLENGE:

The mission is to instruct your robot to move as close as possible to a teacher designated object without actually hitting it. You may measure from the starting point to the object and then you must predict the time that your robot will need to complete the task. Using the **GO** program from Mission 1 change code line 1 from ":Send ({222})" TO ":Send ({122,xxx})" where xxx represents the time in centiseconds you want your robot to travel forward, i.e. 850=8.5 seconds. (See **EDIT INSTRUCTIONS** if needed.)

This is like sending a $125 million satellite to Mars and not a trial and error exercise. You have one shot. Make sure your estimates are accurate and that you have accounted for all variables.

Accuracy of Prediction Grading Scale:

Within 0-.1 m A
.11-.2 m B
.21-.3 m C

If you hit the object or you are off by more than .3 m (or clowning around): Write out the first 500 digits of pi on graph paper, one digit per square!

RESULTS:

1. In meters, how close did you get?

2. What could you do to improve your results?

3. What other designated object would be interesting to use in this mission besides the one given by your teacher? (Perhaps one that would show a definite reaction if a robot hit it.)

4. How did you predict the travel time needed for the robot?

EXTENSION:

GAME1 is a competitive, but sometimes frustrating game. The object of **GAME1** is to set your robot on a starting line and reach a wall or designated object ahead of everyone else. It's a race. The program randomly picks numbers from 1 to 10. If you get a 2, your robot moves 2 seconds forward towards your goal. However if get a 3, your robot sits for 3 seconds while others move ahead. Some numbers move you forward (good) others move you back or spin your robot in circles (not good).

Everyone starts at the same time. **GAME1** can be downloaded off the Internet. After you've tried the program a few times, "EDIT" it and put in your own obstacles and backward moves for other racers.

Graph and Predict Mission 2

PROGRAMMING INSTRUCTIONS:

Turn on TI graphing handheld. Press the **PRGM** button, then use the arrow to highlight "NEW". Press the **ENTER** button, then spell out **MISSION2** by pressing the appropriate keys. (Press the **ALPHA** button to switch from letters back to numbers for the **2**.) Press the **ENTER** button and you're ready to enter the first command for the program.

Line 1: Press the **MATH** button, then use the arrow to highlight " PRB". Use the arrow to scroll down to "5: randInt (". Press the **ENTER** button. Type in 1,15 and close the parentheses by pressing **)**. Press the **STO->** button. Press the [**X,T,2,n**] button. Press the **ENTER** button. The first line should appear as:
:randInt (1, 15) -> X

Line 2: Is blank

Line 3: Press the **PRGM** button, then use the arrow to highlight "I/O". Use the arrow to scroll down to "3: Disp". Press the **ENTER** button. Press the [**X,T,2,n**] button. Press the **ENTER** button. The third line should appear as:
:Disp X

Line 4: Press the **PRGM** button and "CTL" will be highlighted. Use the arrow to scroll down to "8: Pause". Press the **ENTER** button twice. The fourth line should appear as:
:Pause

Line 5: Press the [**X,T,2,n**] button. Press the [**X**] (times) button and then type in 100. Press the **STO->** button. Press the **ALPHA** button, then press **T**. Press the **ENTER** button. The fifth line should appear as:
X*100->T

Line 6: Press the **PRGM** button, then use the arrow to highlight "I/O". Use the arrow to scroll down to "B: Send (". Press the **ENTER** button. Press the **2nd** button and then press **{** for an open brace. Type in 122, then press the coma button. Press the **ALPHA** button, then press **T**. Close the braces and parentheses

by pressing the **2nd** button, the **}** button, and then the **)** button. Press the **ENTER** button. The seventh line should appear as:
:Send ({122,T})

Line 7: Press the **PRGM** button, then use the arrow to highlight "I/O". Use the arrow to scroll down to "A: Get (". Press the **ENTER** button. Press the **ALPHA** button, then press **R**. Press **)** then **ENTER**. The second line should appear as:
:Get (R)

Line 8: Press the **PRGM** button and "CTL" will be highlighted. Use the arrow to scroll down to "F: Stop". Press the **ENTER** button. The fourth line should appear as:
:Stop

Press the **2nd** button, then **QUIT.**

To run the program, attach the TI-83/TI-83 Plus handheld to your robot and connect link cable. Make sure the robot and handheld are both switched on. Press the **PRGM** button and use the arrow to scroll down to ": MISSION2". Press the **ENTER** button. Press the **ENTER** button again and the program will randomly pick a number from 1 to 15. This number represents the time in seconds the robot will be instructed to travel forward. The program will pause while you predict the distance the robot will travel. Place the robot on the floor, then press the **ENTER** button and robot will travel forward for the displayed number of seconds.

EDIT INSTRUCTIONS:

Press the **PRGM** button, then use the arrow to highlight "EDIT". Use the arrow to scroll down to **:GO**. Press the **ENTER** button. Change 222 to 122 then press the coma button. Enter the number of centiseconds you want the robot to run. Close the braces and parentheses by pressing the **2nd** button, the **}** button, and then the **)** button. The new first line (if you want the robot to run 8.5 seconds) should appear as:
:Send ({122,850})

Press the **2nd** button, then **QUIT.** Follow instructions above to run the program except after the second **ENTER** the robot will move forward.

Teacher Notes

Graph and Predict Mission 2

ACTIVITY NOTES:

The graph of TIME verses DISTANCE for this exercise should be a straight line that can be modeled by a linear equation. Students can make predictions by reading off the graph. Algebra 1 students or higher can use the slope formula and the slope-intercept model (y=mx+b) to find an equation for the best-fitting line. The y-intercept point (b) is zero. The slope of the line is the speed or rate of the robot. From their slope-intercept equation, students can discover the formula for distance, d=rt.

The object that students' robots can come close to, but never touch, could be a wall, a house of cards, a balloon surrounded by things that could pop it, a cardboard pi symbol, etc. The program: **GAME1** is available at smallrobot.com.