DIVISION C

MAY 18, 2019

2019 RoboPlay

Racing & Automotive

UC DAVIS
C–STEM CENTER
c-stem.ucdavis.edu

C-STEM is a UC approved Educational Preparation Program for Undergraduate Admission to all UC Campuses

Message From the Director

Dear C-STEM Teachers and Students,

Welcome to the 2019 RoboPlay Challenge Competition!

As always, our C-STEM team has organized an extraordinary day for our C-STEM students. Our goal is for all our C-STEM students show off their teamwork, critical thinking, and problem solving skills in a fun environment. Today will be full of excitement as students overcome the racing and automotive challenges we have laid out. For the first time ever we are welcoming students from 5th and 6th grade into their own specialized division!

We are very proud to be a UC Approved Educational Preparation Program for undergraduate admission to all UC campuses. We are particularly proud of our C-STEM Math-ICT Curriculum which provides students with up to 13 years of computer science education through hands-on integrated learning of math and computer science.

As the program grows and expands, so does our wealth of curriculum and educational technologies. We are excited to announce our upcoming release of C-STEM Studio version 6.0 which overhauls the user experience to add more features for Linkbots with Arduino, Raspberry Pi, and RoboBlockly. In addition, this version will provide support for controlling hardware Linkbots from RoboBlockly and Chromebooks! As always, C-STEM Studio continues to be a freely available resource for all students and teachers.

We would like to extend a warm welcome to our new participants this year and welcome back those who are returning. We have an extraordinary group of students with us and are operating at maximum capacity of the UC Davis Pavilion. We have an impressive showing of over 150 teams between our two sites.

Excitement is also growing as we get closer to our Girls in Robotics Leadership (GIRL) and GIRL+ camps this summer where we will have more participants than ever before both in California and, for the first time, internationally.

We are proud of all of you.
Good luck in the competition!

Dr. Harry H. Cheng
C-STEM Center Director and Professor

Organized by

UCDAVIS
C–STEM Center

COASTLINE COLLEGE

RoboPlay Challenge Competition Schedule - May 18, 2019

TIME	EVENT
7:30 – 8:30 AM	Registration and Setup for RoboPlay Challenge Competition
8:30 – 8:40 AM	Welcome and Introduction
8:40 – 9:00 AM	RoboPlay Challenge Competition Introduction
9:00 – 12:00 PM	RoboPlay Challenge Competition Problem Solving
12:00 – 12:45 PM	Lunch Break
12:45 – 3:45 PM	RoboPlay Challenge Competition
3:45 – 4:00 PM	Break Time
4:00 – 5:00 PM	Awards Ceremony: • C-STEM Awards of Achievement • GIRL's Leadership Award • C-STEM Awards of Excellence • C-STEM Scholarship • RoboPlay Video Competition Winners • RoboPlay Challenge Competition Winners

Contact Information

C-STEM Center Director:

Harry Cheng
Email: hhcheng@ucdavis.edu
Phone: (530) 752-5020

Education Service Manager:

Daniel Ryan
Email: djryan@ucdavis.edu
Phone: (530) 752-8788

C-STEM Regional Organizer
Orange County Site:

Merry Kim
Email: mkim170@coastline.edu
Phone: (714) 714-7008

Share photographs taken during the RoboPlay Competition with the UC Davis C-STEM Center at roboplay@c-stem.ucdavis.edu for a chance to be featured on the C-STEM website and social media!

Table Of Contents

Competition Details

Challenges

Competition Details

General Information

Each Division has 10 challenges to complete in any order. Challenges provide explicit instructions for receiving points. The goal is to get as many points as possible. Most challenges have partial points available, so teams may attempt portions of challenges as well.

The day is broken into two parts, unscored practice and scored competition. Each part is three hours long. Students may check their nametags or the schedule brochure for their practice and competition time slots. If they are late, they will not be allowed to make up any time.

Unscored Practice Information

- All teams have a designated practice area (pit) that gives them space to practice with their own 2019 RoboPlay practice mat.
- Each team receives two 17-minute practice periods to practice on their official 2019 RoboPlay Competition Board between 10am and noon that is located in the competition area.

Scored Competition Information

- Each team is assigned an official 2019 RoboPlay Competition Board in the Competition Area that is monitored by one or more RoboPlay Judges.
- Each team receives three 17-minute competition periods to compete on their official 2019 RoboPlay Competition Board between 12:45pm and 3:45pm.

Competition Rules

In The Pits (Practice)

- Teams may use as many laptops as they have students.

At The Competition Table (Competition Time)

General Rules

- Teams may only bring one laptop into the Competition Area at a time.
- Teams may not interact with their running program unless explicitly allowed in the challenge text. Some challenges will require user interaction at startup.
- Teams are responsible for setting up the Competition Board for each run of each challenge as specified in the challenge text unless otherwise stated.

Coding Rules

- All challenge tasks must be completed using a computer program written in Ch and run in ChIDE. TiltDrive and Copycat modes are prohibited unless stated otherwise in the challenge.

Scoring Rules

- Any challenge that is ongoing when a team's 17-minute time slot ends will be immediately stopped and points will be calculated based on the rules for a partial call.
- Students may attempt each challenge as many times as they like within their allotted competition time. If a challenge is attempted multiple times, only the points from the highest scoring run will be kept.
- Challenges may not be "chained together" meaning that a single program cannot receive points for more than one challenge at a time.
- Each challenge attempt, regardless of outcome, counts as a run. In the case of two teams with identical scores, the number of runs will be used as a tie-breaker, with the lowest number of attempts winning the tie.
- Teams abort a run at any time by touching a running Linkbot or calling "abort." Aborted runs still count as attempts and score zero points.
- While a program is still executing but no penalty points are possible, teams may ask the judge for a "partial call" in order to abort the run but still receive partial points. The judge must agree to the partial call before teams touch any Linkbots or the run will be scored as an abort.
- At the end of each run the judge will show teams their run number and run score prior to submission. If a team wishes to contest the score for a run, they must call for a Head or Lead Judge at that time.

Random Values

- Input random values into the program at the beginning of each run using the scanf() function.
- Random values change at the start of every run. Refer to the Table Judge, who will display and announce the relevant values for each run.
- Enter your random values only after pressing “Run”. Step away from the computer after inputting the values.
- Do not strategically abort your challenge to get better random values. Judges may ban teams that abort challenges from participating in the remainder of the competition period.

General Rules

- Teams may not share laptops or use more materials than are specified in the Equipment section at any time in any location.
- Use of electronics other than the allowed laptops is strictly prohibited. This includes other computers, calculators, cell phones, tablets, or any other computing device.
- There will be no internet access during the competition. Any team caught using the internet will be disqualified.
- Teams may not share the computer programs they create with any other team. This will be considered cheating and both teams will be disqualified.
- Teams may speak to the judges or the Support Team for clarification, but students may not solicit help with challenges or Linkbots from students outside their team, any teachers, or any parents or observers.

Challenge Competition Awards

Regional Awards

Regional awards are given to the first, second, and third place winners for each division at each of the RoboPlay Locations. Regional awards are not issued in divisions with fewer than 4 competing teams.

Statewide Awards

Statewide awards are given to the first, second, and third place winners for each division across the state.

Judges Awards

The judges decide three additional awards for each division at each RoboPlay Location:

- Perseverance Award - This award goes to the team that improvises and overcomes a difficult situation while still maintaining a high level of performance.
- Spirit Award - This award celebrates a team that displays extraordinary enthusiasm and spirit
- Teamwork Award - This award recognizes a team that fluidly works together with strong communication, tasks delegation, and excellent time management.

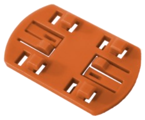
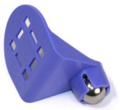
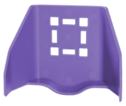
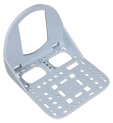
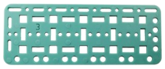
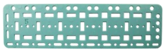
Technology Requirements

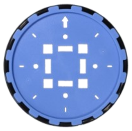
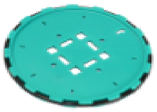
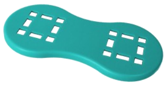
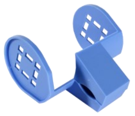
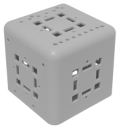
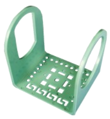
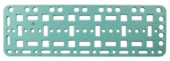
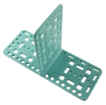
- In order to receive technical support from our RoboPlay Challenge Competition Staff, please check that your systems meet the necessary specifications before the day of the competition.
- Software: C-STEM Studio v5.5 or above, Ch 8.0, Linkbot Labs 1.1.1
- Hardware: Windows XP or above, Mac OS X 10.6.8 or above

Equipment

Each team must bring their own Laptops, Linkbots, and accessories for the competition. Teams may also bring a protractor, writing utensils, a compass, string, USB flash drives, measuring tapes (8 feet), extension cords, and multi-port USB chargers for the Linkbots.

In the interest of fairness, each team brings the same Linkbots and accessories. Backup Linkbots and accessories are allowed but may not be used in the pit or competition areas in excess of the quantities listed below. Note that for 2019, there is only one acceptable version of each accessory. Teams may not use parts other than those listed and pictured below. The Linkbots with the opaque white chasse are acceptable as well, as are blue snap connectors.

Part	Image	Quantity
Linkbot-I		4
Linkbot-L		1
Linkbot-I, Linkbot-L, or Dongle		1
Snap Connector		55
Caster		4
Push Scoop		2
L Connector		4
Rectangle Connector 3 inches		6
Rectangle Connector 5 inches		6
Snap Connector Cap		12

Part	Image	Quantity
3.5" Wheel		8
4" Wheel		4
Bridge Connector		8
Gripper Pair		2
Cube Connector		10
Hacky Sack		1
U Connector		4
Rectangle Connector 4 inches		6
T Connector		6
1" RGBY Foam Cubes		4 of each color

Sample scanf() Code

```
string_t name;
int distance;
double radius;

printf("Please enter your name: ");
scanf("%s", &name);
printf("Hi %s\n", name);

printf("Please enter a whole number distance: ");
scanf("%d", &distance);

printf("Please enter a decimal wheel radius: ");
scanf("%lf", &radius);

printf("Okay, %s, driving %d inches.\n", name, distance);

robot.driveDistance(distance, radius);
```

Output:

```
        Please enter your name: RoboPlay
        Hi RoboPlay
        Please enter a whole number distance: 10
        Please enter a decimal wheel radius: 1.75
        Okay, RoboPlay, driving 10 inches.
```

Quick Reference

```
// The first two lines are required in each program
#include <linkbot.h>                          // Include Linkbot functions
CLinkbotI robot1, robot2;                     // Create two Linkbot-I instances
double r = 1.75;                              // Wheel Radius
double tw = 3.69;                             // Trackwidth

// Setting Speed
robot2.setSpeed(4, r);                        // Inches per second, Wheel Radius
robot1.setJointSpeeds(120, NaN, 120);         // Degrees per second for each Joint

// Basic Movements
robot1.driveDistance(10, r);                  // Inches, Wheel Radius
robot2.driveAngle(360);                       // Degrees
robot1.driveTime(5);                          // Seconds
robot2.turnLeft(90, r, tw);                   // Degrees, Wheel Radius, Trackwidth
robot1.turnRight(90, r, tw);                  // Degrees, Wheel Radius, Trackwidth
robot2.moveJoint(JOINT1, 360);                // Joint, Degrees
robot1.moveJointTo(JOINT3, 90);               // Joint, Degrees from 0-pos.

// Special Functions
robot1.setLEDColor("blue");                   // Color
robot2.setBuzzerFrequency(NOTE_C4, 1);        // Note name, Seconds
robot2.setBuzzerFrequency(261.6, 1);          // Frequency (Hz), Seconds

// Using NB (Non-Blocking) to drive two Linkbots at once
robot1.driveDistanceNB(20, r);
robot2.driveDistance(10, r);
robot1.moveWait();                            // Wait until robot1 finishes moving
```

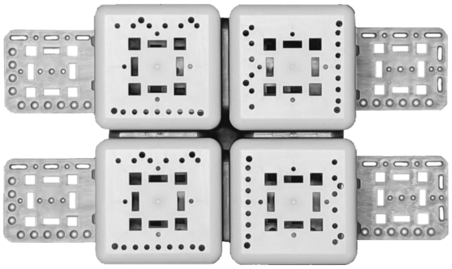
Definitions

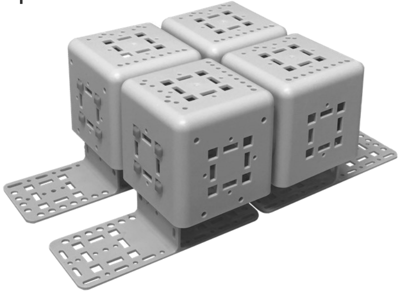
Bot Placement

- **"Bot begins at the Starting Line"** - Body of the Bot as seen from above (not the wheels) is directly to the left of the left edge of the Starting Line within a quarter inch of the line.
- **"Bot crosses the Finish Line"** - The rightmost (front) part of the body of the Bot as seen from above (not the wheels) is past the right edge of the Finish Line
- **"Bot completes one circuit of the track"/"Bot goes around the track"** - Bot begins at the Starting Line and crosses the Finish Line, traveling counterclockwise around the track
- **"Bot is inside/occupies a Parking Space"** - Over 50% of the body of the Bot as seen from above (not the wheels or caster) is inside the outer edge of the rectangle around the Parking Space
- **"Bot is above a Parking Space"** - At least a quarter of the body of the Bot as seen from above is inside the outer edge of the rectangle surrounding the Parking Space
- **"Bot stays inside its lane"** - The ball of the caster (including its purple plastic covering) on the Bot is entirely inside the outer edge of the lines designating its lane

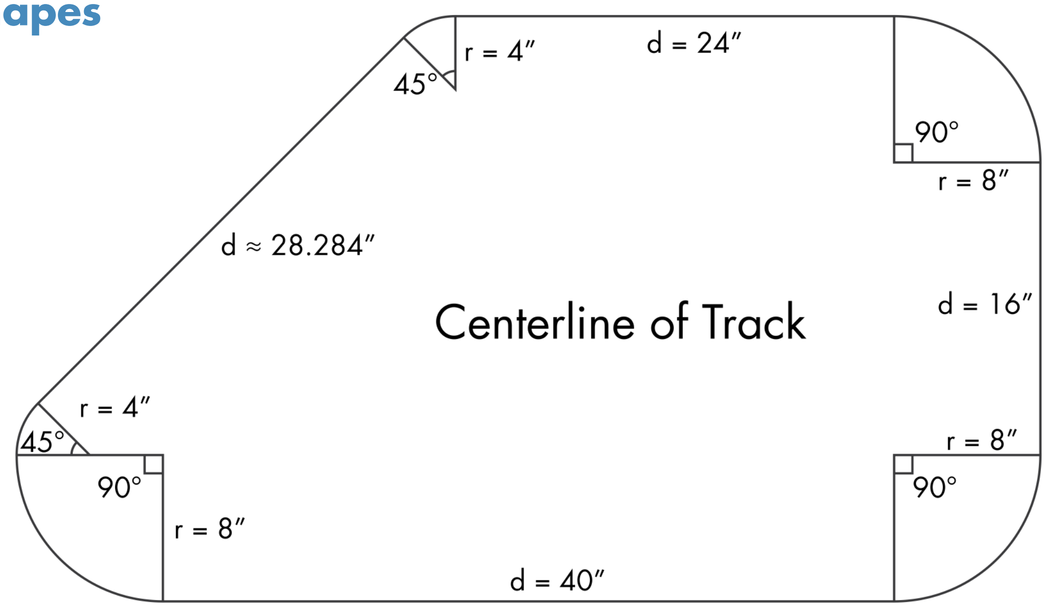
Platform

Put together the Platform using the parts you brought! You need 4 Cube Connectors, 4 T Connectors, and 8 Snap Connectors. The Platform lives on the matching graphic in the Lookout Area

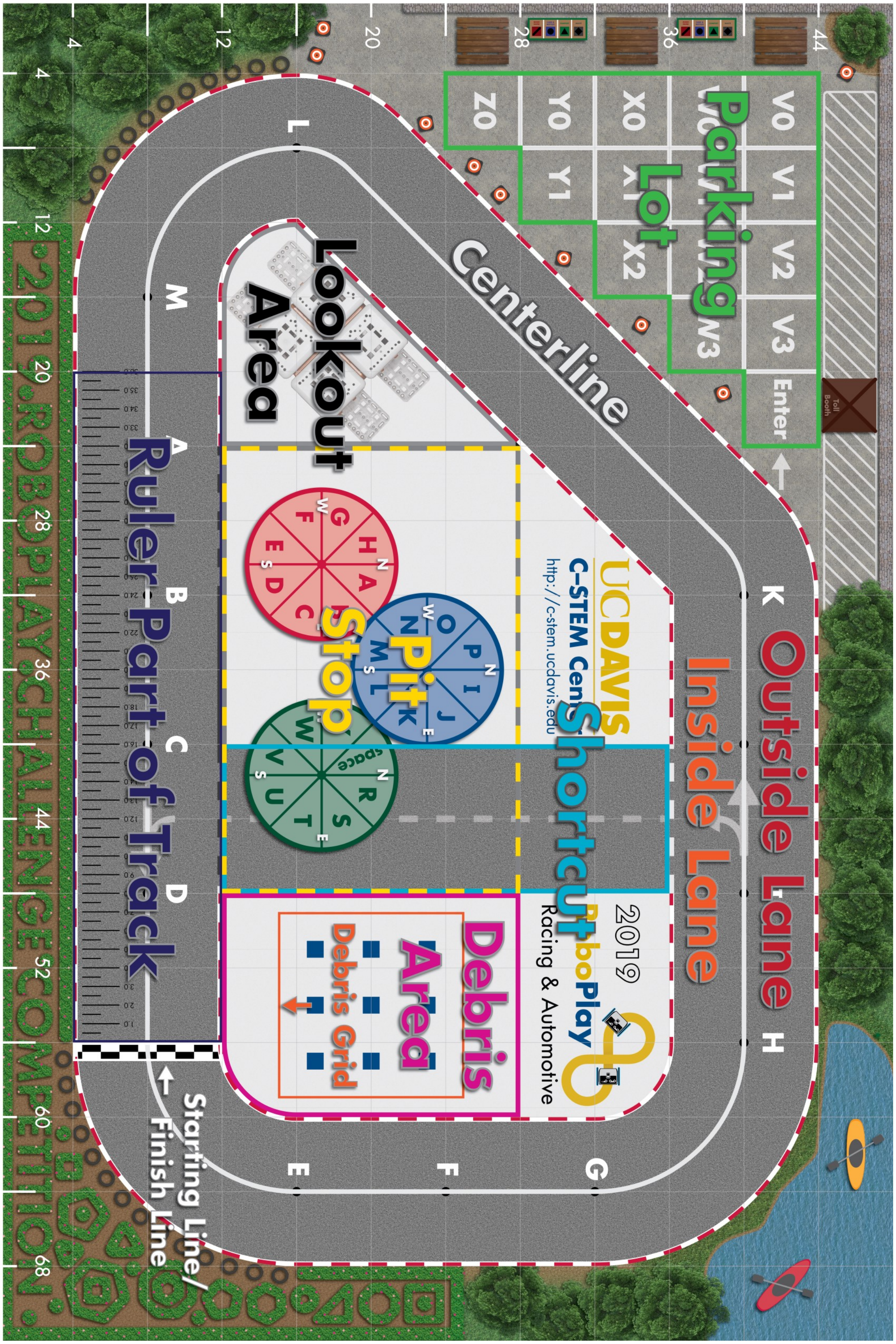




Track Shapes



Board Definitions



1. A Good Start (C) *40 Points*

Background

Start off the day strong by testing your brakes at the Finish Line.

Setup

Receive a whole number X less than 36 and enter the value using scanf
Place Bot X inches away from the Starting Line (using the Ruler markings)

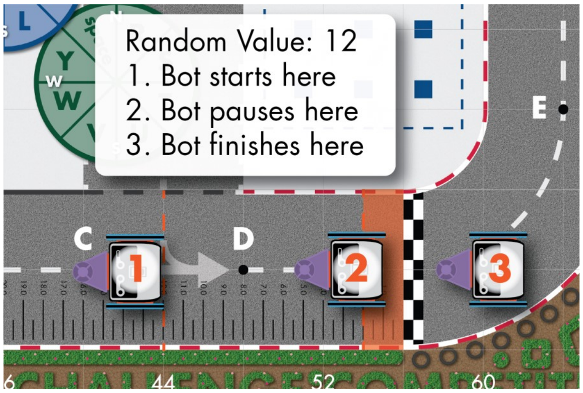
Objective

Bot must cross the Finish Line
Bot pauses within 2 inches of the left edge of the Finish Line before crossing it

Challenge Scoring

#	Description of Scoring Criteria	Points
1	Bot crosses the Finish Line	20
2	Front of Bot pauses within 2 inches of the left edge of the Finish Line for a second	20

Diagram



Random Value: 12
1. Bot starts here
2. Bot pauses here
3. Bot finishes here

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

2. The Wrong Way (C) *50 Points*

Background

Two Bots drive toward each other on the track and stop just after colliding.

Setup

Receive a decimal proportion less than one and enter the value using `scanf`

Two Bots face each other on the Centerline of the Ruler portion of the track

One Bot's front is at the 36-inch mark

The other Bot's front is at the 0-inch mark

Objective

The two Bots drive towards each other on the Centerline

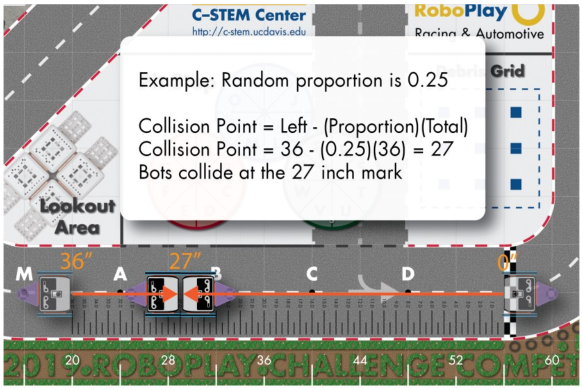
The Bots collide at the random proportion of the space between them from the 36-inch mark

Example: If the random proportion is 0.25, the Bots collide at the 27-inch mark because the Bot on the left has traveled a quarter of the space between them

Challenge Scoring

#	Description of Scoring Criteria	Points
1	The Bots collide within one inch of the correct location based on the random proportion	50
Penalty	Either Bot leaves the Centerline (a wheel crosses the Centerline)	**FAIL**

Diagram



Example: Random proportion is 0.25

Collision Point = Left - (Proportion)(Total)

Collision Point = 36 - (0.25)(36) = 27

Bots collide at the 27 inch mark

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

3. Slow and Steady (C) *60 Points*

Background

HareBot is beating TortoiseBot in a race, but HareBot decides to take a nap in the middle of the race. Help TortoiseBot catch up to HareBot!

Setup

Receive two natural numbers in form "X Y"

Read these numbers into your program, in order, using `scanf`

HareBot and TortoiseBot begin next to each other X inches to the right of the Ruler's 36-inch mark

HareBot travels in the inside lane and TortoiseBot travels in the outside lane

Objective

HareBot and TortoiseBot start at the same time in a straight path to the Finish Line

Bots move at distinct (different) constant speeds

HareBot stops for Y seconds partway through its path to the Finish Line

After stopping, HareBot resumes at its original constant speed

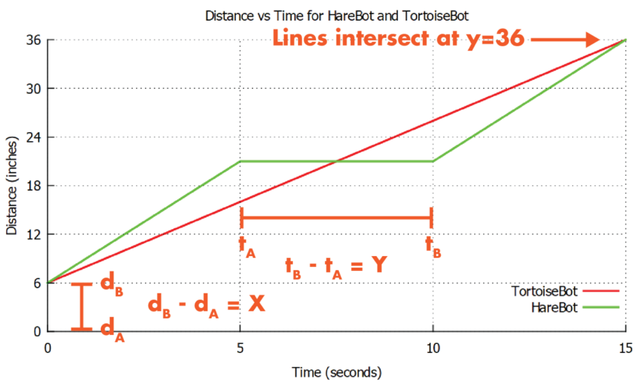
HareBot and TortoiseBot must simultaneously cross the Finish Line

Graph the distances vs. times for both Bots on a single plot and label them as shown in the diagram

Challenge Scoring

#	Description of Scoring Criteria	Points
1	Titles, axis titles, and units on graph match example	15
2	Graph matches provided example	15
3	Bots cross the Finish Line within a Bot's width of each other	30
Penalty	Bots leave their lanes	**FAIL**

Diagram



Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

4. Tape Measure (C) *70 Points*

Background

You need to measure a distance, but you lost your tape measure! Your Bots can help.

Setup

The judge will randomly place two Bots along the Ruler part of the track as shown in the diagram

You MAY NOT type anything on your computer after the judge places the Bots

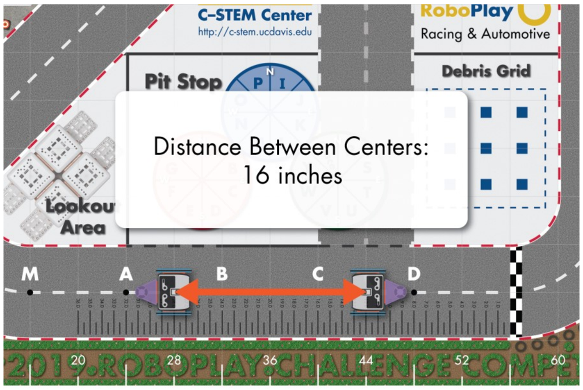
Objective

Determine the original distance between the centers of the Bots

Challenge Scoring

#	Description of Scoring Criteria	Points
1	Program outputs correct distance in inches within 4 inches between Bots	20
2	Program outputs correct distance in inches within 2 inches between Bots	50

Diagram



Distance Between Centers:
16 inches

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

5. Getting Carsick (C) *80 Points*

Background

Your Bot is testing its steering in the Pit Stop.

Setup

Bot starts at the point shown in the diagram
The Bot's wheels should always be on opposite sides of the outline of a circle
Receive a random direction (N, NE, E, SE, S, SW, W, or NW)
Use `scanf` to read the random direction into your program

Objective

Bot drives around the two circles on the right of the Pit Stop two times:

Follow the arrows as shown in the diagram

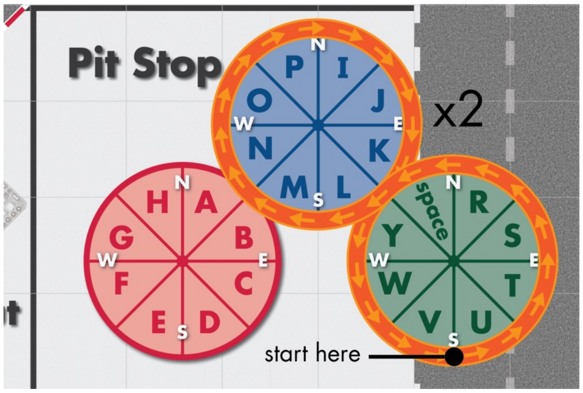
Keep one wheel on either side of the outline of one of the circles at all times

Each time the Bot passes the random direction it must pause for three seconds
Bot may not stop moving at any direction other than the random one

Challenge Scoring

#	Description of Scoring Criteria	Points
1	Bot drives around the two circles two times, following the arrows shown in the diagram	40
2	Bot pauses at the random direction each time it passes that direction on a circle (and full points for Scoring Element 1). Body of the Bot must cover the intersection of the circle's outline and the directional line.	40
Penalty	Bot stops moving at a direction other than the random one	**FAIL**

Diagram



Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

6. Piggy Back (C) *100 Points*

Background

Oh No! PiggybackBot has engine trouble on the Platform and can barely move. CarryBot must take it to a Parking Space.

Setup

Place the Platform as described in the Definitions section in the Lookout Area
PiggybackBot starts anywhere on the Platform
CarryBot consists of a single Bot with any number of parts attached
CarryBot starts anywhere on the board not touching PiggybackBot or the Platform
Receive a random Parking Space name after you press Run
Use `scanf` to read the random Parking Space name

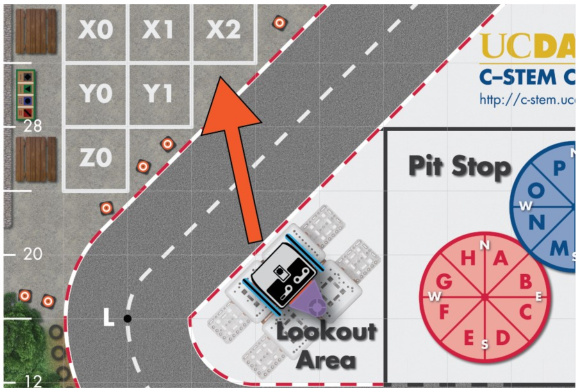
Objective

PiggybackBot must end above the random Parking Space
PiggybackBot never touches the Board and the Platform is not moved

Challenge Scoring

#	Description of Scoring Criteria	Points
1	PiggybackBot moves out of the Lookout Area without touching the Board	40
2	Any part of PiggybackBot ends directly above the random Parking Space without ever touching the Board	60

Diagram



Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

7. Street Sweeper (C) *120 Points*

Background

A crash has left a (strangely regular) array of parts and debris in the middle of the track. You must remove just the parts without disturbing the debris.

Setup

Place CleanupBot(s) at any location on the board outside the Debris Grid

Judge will randomly place Parts and Debris on the squares in the Debris Grid:

Parts: three yellow blocks

Debris: six red or green blocks

After pressing "Run" you may enter text into the console via `scanf` to represent the Parts and Debris

Your Bot may not begin moving until after you are done entering text

Note: The parts will be placed in a straight line or along the diagonal

Objective

Using CleanupBot(s), move just the Parts (yellow blocks) outside of the Debris Area

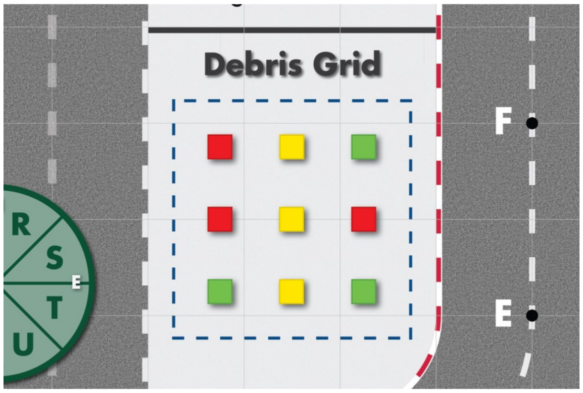
Do not move any of the Debris (red or green blocks)

A Debris block is "moved" if the blue square it is sitting on can be seen

Challenge Scoring

#	Description of Scoring Criteria	Points
1	Move the Parts (yellow blocks) completely outside the Debris Area	40 Points per Part
Penalty	Penalty for each Debris (red or green block) moved such that the blue square can be seen	-20 Points per red or green block moved

Diagram



Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

8. Parking (C) *140 Points*

Background

Your Bots need help finding their spaces in the Parking Lot.

Setup

The Parking Spaces are named with a letter and a number to indicate the row and column

The judge will give you three random Parking Space names to read with `scanf`:

> The first one is an Occupied Space—place a Cube Connector at this space
>
> The last two are Target Parking Spaces

Place two Bots to the right of the Parking Space labeled **Enter**

Objective

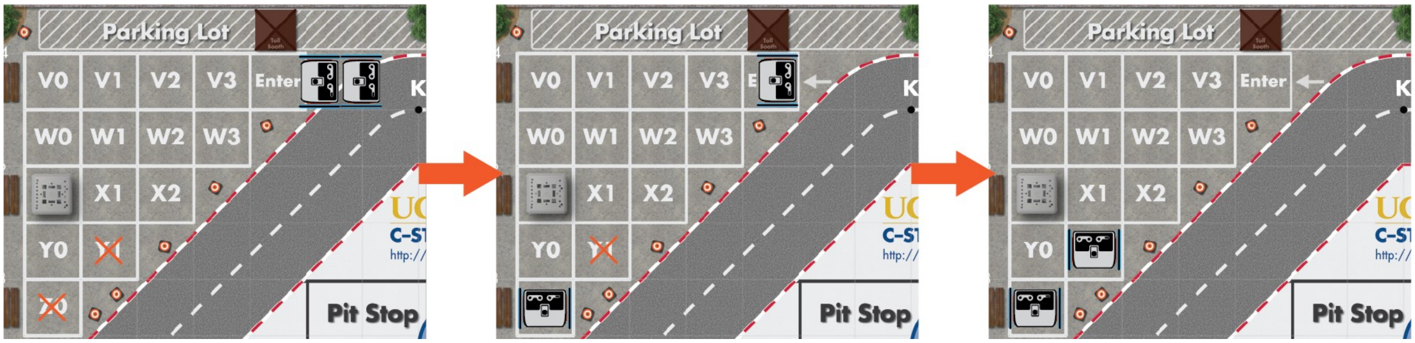
Each Bot, in either order, must enter the Parking Lot at the Parking Space labeled **Enter**

Move a Bot to each of the given Target Parking Spaces in any order

Challenge Scoring

#	Description of Scoring Criteria	Points
1	At least one Target Parking Space is occupied by a Bot	60
2	Both Target Parking Spaces are occupied by Bots (and full points for Scoring Element 1)	80
Penalty	A Bot entirely exits the Parking Lot after entering through the Enter Space	**FAIL**
Penalty	A Bot touches a Cube Connector	**FAIL**
Penalty	The center of a Bot crosses the boundary of more than two Parking Spaces at once (Bots may not move diagonally)	**FAIL**

Diagram



Run	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
Score																			

9. Race Against Time (C) *160 Points*

Background

It’s time for your Bot to race! Make sure that RacerBot goes around the track the fastest.

Setup

RacerBot is placed at the Starting Line
RacerBot has one 3.5-inch wheel and one 4-inch wheel

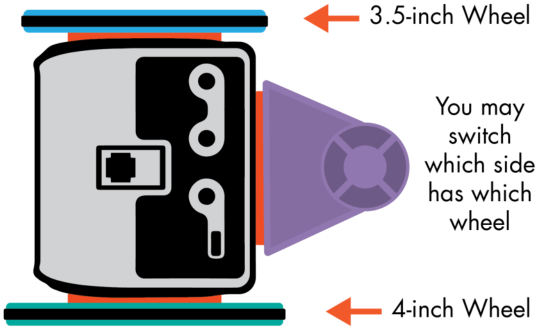
Objective

RacerBot completes a full circuit around the track with one wheel on each side of the Centerline

Challenge Scoring

#	Description of Scoring Criteria	Points
1	RacerBot completes a full circuit around the track with one wheel on each side of the Centerline	80
2	RacerBot finish in X seconds rounded up to the nearest 5 seconds and full points for Scoring Element 1	105 - X where {min=0, max=80}

Diagram



Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

10. RoboJump (C) *180 Points*

Background

A Bot needs to get on top of the Platform to see the race.

Setup

Place the Platform as described in the Definitions on the matching graphic in the Lookout Area
You can use any number of allowed Bots or Parts
At least one of the Bots starts touching the Board (it may have wheels attached)
A hacky sack starts touching the Board

Objective

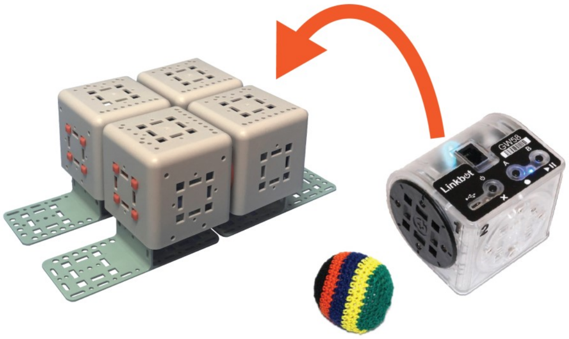
A Bot that starts touching the Board ends on top of the Platform
The hacky sack ends on top of the Platform

Rules: the Bot that ends on top of the Platform may be hanging off the edge of the Platform as long as it is not touched by any other Parts not attached to the Bot or Bots that are not on the Platform

Challenge Scoring

#	Description of Scoring Criteria	Points
1	The hacky sack ends entirely on top of the Platform	50
2	A Bot ends on top of the Platform as described in the Objective	130

Diagram



Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Score																			

Score Tracker

Division C	Practice 1	Practice 2	Run 1	Run 2	Run 3
1. A Good Start (40)					
2. The Wrong Way (50)					
3. Slow and Steady (60)					
4. Tape Measure (70)					
5. Getting Carsick (80)					
6. Piggy Back (100)					
7. Street Sweeper (120)					
8. Parking (140)					
9. Race Against Time (160)					
10. RoboJump (180)					

Notes

UCDAVIS
C–STEM CENTER
c-stem.ucdavis.edu

NSF

California Department of EDUCATION

COASTLINE COLLEGE

BECKMAN COULTER

Barobo

Hewlett Packard Enterprise

SoftIntegration®

UCDAVIS
COLLEGE OF ENGINEERING