The 2013-14 CS2N VEX Programming Challenge

Overview

This section describes the Robotics Academy's Computer Science Student Network (CS2N) VEX Robot Programming Challenge. It also lists the game definitions and game rules.

Game Description

Matches are played by loading code into your computer. The CS2N VEX Programming Challenge is played by one *Player*. The *Player* is challenged to program their autonomous virtual robot to score the highest score possible within the shortest amount of time. In this challenge *Players* can score by *Scoring* any colored *BuckyBalls* and *Large Balls* into the *Middle Zone* and *Scoring Zone*, by *Stashing* any colored *BuckyBalls* and *Large Balls* into the *Goals*, and by *Low Hanging, Hanging,* and *Hanging with a Ball* off either colored *Bar* at the end of the match. The RVW challenge is two minutes long and is played in *Autonomous* Mode only.

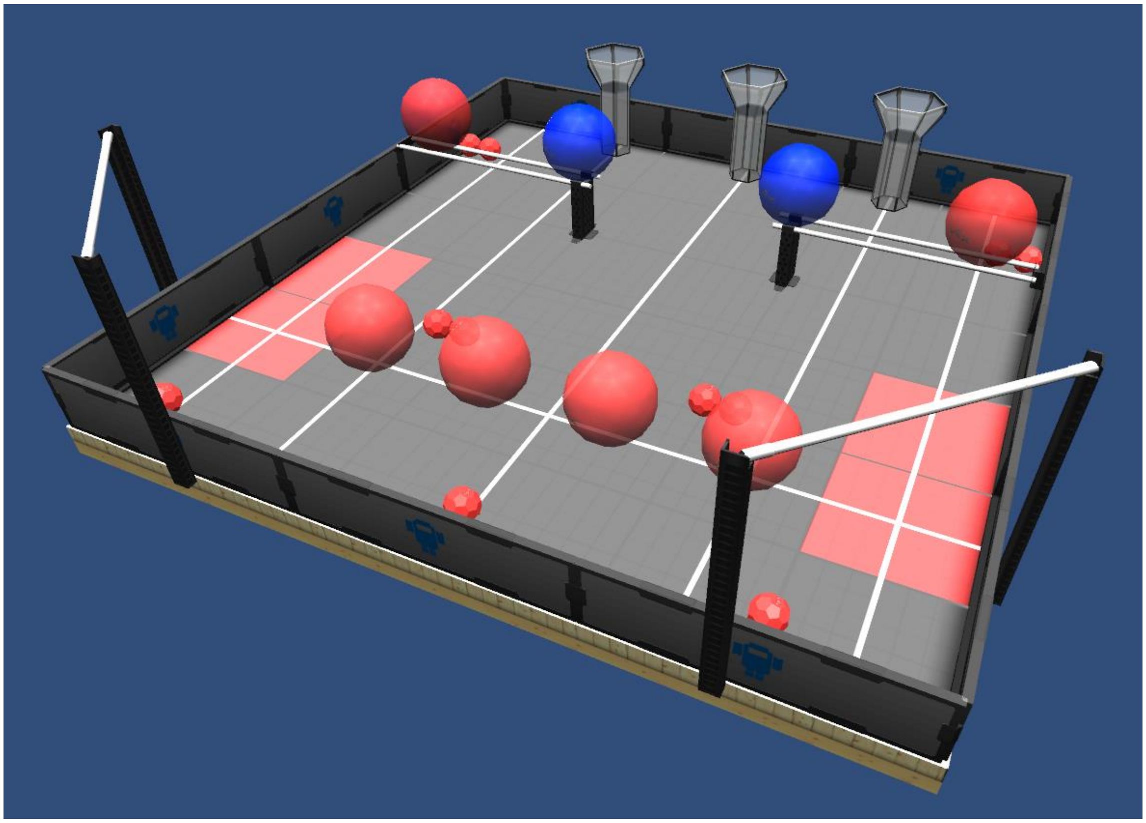


Figure 1: *Isometric view of the field*

Note: *The illustrations in this section of the manual are only provided to give a general visual understanding of the game.*

In the virtual game there are a total of fourteen (14) red *BuckyBalls,* six (6) red *Large Balls*, and two (2) blue *Large Balls.* Fourteen (14) *BuckyBalls* and all eight (8) *Large Balls* will start at designated locations on the field*.*

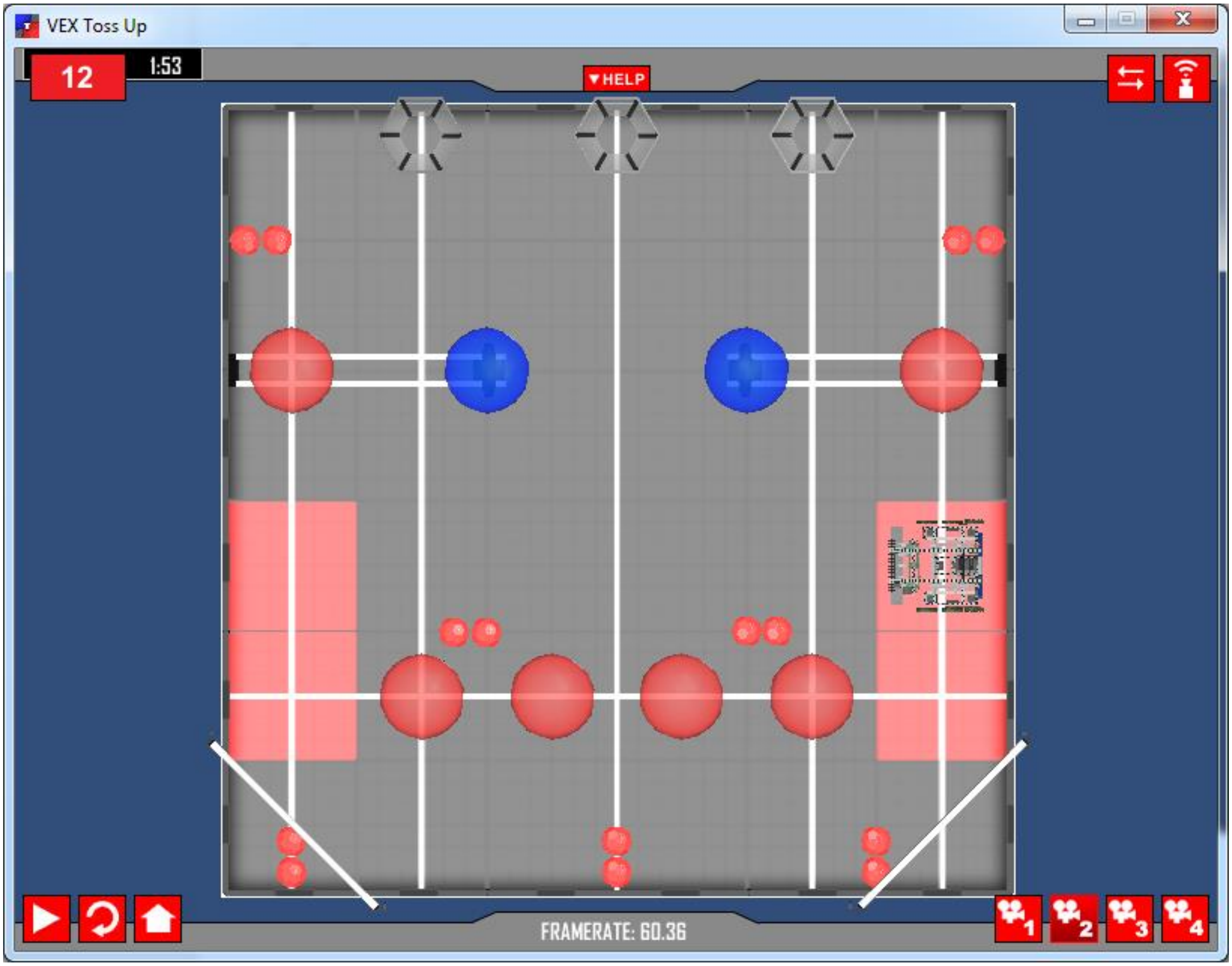


Figure 2: *Overhead view of the field*

This CS2N VEX Programming Challenge is designed to promote Computer Programming and could not have happened without the generous support of the REC Foundation, VEX Robotics Inc., and Robomatter Inc.

Game Definitions

Adult – Anyone not meeting the definition of *Student*.

Autonomous Mode – The *Robot* operates and reacts only to sensor inputs and to commands pre-programmed by the player into the onboard *Robot* control system. Human interaction via a joystick or keyboard is not allowed during *Autonomous Mode*.

Barrier – The 12" high sheet metal and pipe structure that separates the *Middle Zone* and the *Goal Zone*.

BuckyBall – A red truncated icosahedron shaped plastic *Scoring Object* with an overall diameter of 5".

Driver – A *Player* responsible for operating and controlling the *Robot* in the Robot Skills Challenge.

Scoring Zone – The foam tiles located between the *Barrier* and the far field wall.

Field Element – The foam field tiles, field perimeter, *Bars, Bump*, *Fence, Goals* and all structures found in the RVW *VEX Toss Up* Programming Challenge.

Game Time – The combination of the amount of time that the *Match* has been running which is measured by the *Internal Timer* in milliseconds.

Goal – One of the two (2) tube shaped, *24"* tall field structures, where teams can *Lock Up BuckyBalls* or *Large Balls*

Internal Timer – The internal timer is a clock built into the RVW software that tracks the time in milliseconds that the *Player* has been playing the *Match*.

Hanging – A *Robot* is considered to be *Hanging* if it is touching the *Bar* of any color and not touching any foam field tile.

Hanging with a Ball – A *Robot* is considered to be *Hanging with a Ball* if it is *Hanging* and touching a *Large Ball* that is not touching a foam field tile and is not touching another *Robot*

Hanging Zone- The foam tiles located between the field wall adjacent to the Alliance Stations and the *Bump.*

Large Balls -- A red or blue spherical plastic *Scoring Object* with an overall diameter of 15".

Lock Up - BuckyBalls or *Large Balls* are considered *Locked Up* when the *BuckyBalls* are placed in the Goal or when the *Large Balls* are placed on the *Goal*.

Low Hanging -- A *Robot* is considered to be *Low Hanging* if it is touching the *Bar* of any color and not touching any foam field tile.

Match – A *Match* consists of a two minute or less *Autonomous Mode* programming challenge*.* The first tie-breaker for a *Match* is the amount of time score. For instance, if two *Players* score 15 points each and one *Player* scores them in 1:50and the other *Player* scores them in 2:00, then the team that scored in a shorter amount of time is considered the winner.

Middle Zone – The foam tiles located in between the *Bump* and the *Barrier*

Player – The person that wrote the code and is playing the *Match*. The *Player* must meet all eligibility requirements the RVW Challenge in order to compete. Each *Player* may only have one CS2N account; multiple accounts by the same *Player* will make all of that *Player's* accounts become ineligible for prizes.

Returned to the Starting Tile – A robot is considered to *Returned to a Starting Tile* when any part of the *Robot's* wheel touches the *Starting Tile*.

Robot – A programmable object that a *player* can load into the virtual world.

Robot Reset Button – A button on the RVW Game interface that allows the *Player* to stop the *Match* and reset the *Robot*. The button allows the *Player* to select another Robot to load code onto and run.

RVW – Robot Virtual World

Scored – A *Scoring Object* is *Scored* in a *Zone* if it meets the following criteria.

1. A *Scoring Object* is touching a *Zone* and is not being touched or *Supported* by a *Robot*.
   a. A *BuckyBall Scored* in two *Zones* will only count for the *Scoring Zone*
   b. A *Large Ball Scored* in two *Zones* will only count for the *Middle Zone*

Scoring Object – A *BuckyBall* or a *Large Ball*

Starting Tile – A colored tile (red or blue), which designates the location where your *Robot* starts the match.

Stashed – A Scoring Object is *Stashed* in a *Goal* if some part of the *Scoring Object* is within the two-dimensional space defined by the outer edges of the *Goal*, and not being touched or *Supported* by a *Robot* of the same color as the *Scoring Object*. Note: A goal extends infinitely perpendicular to the playing field surface within the goal boundaries

Student – Anyone enrolled in a pre-college school or home-schooled as part of a pre-college educational curriculum.

Supported – A *Scoring Object* is considered to be *Supported* by a *Robot* if the *Scoring Object* moves along with the *Robot*. i.e. If a referee were to remove a *Robot* from the field and a *Scoring Object* came with it, this *Scoring Object* would be considered to be *Supported*

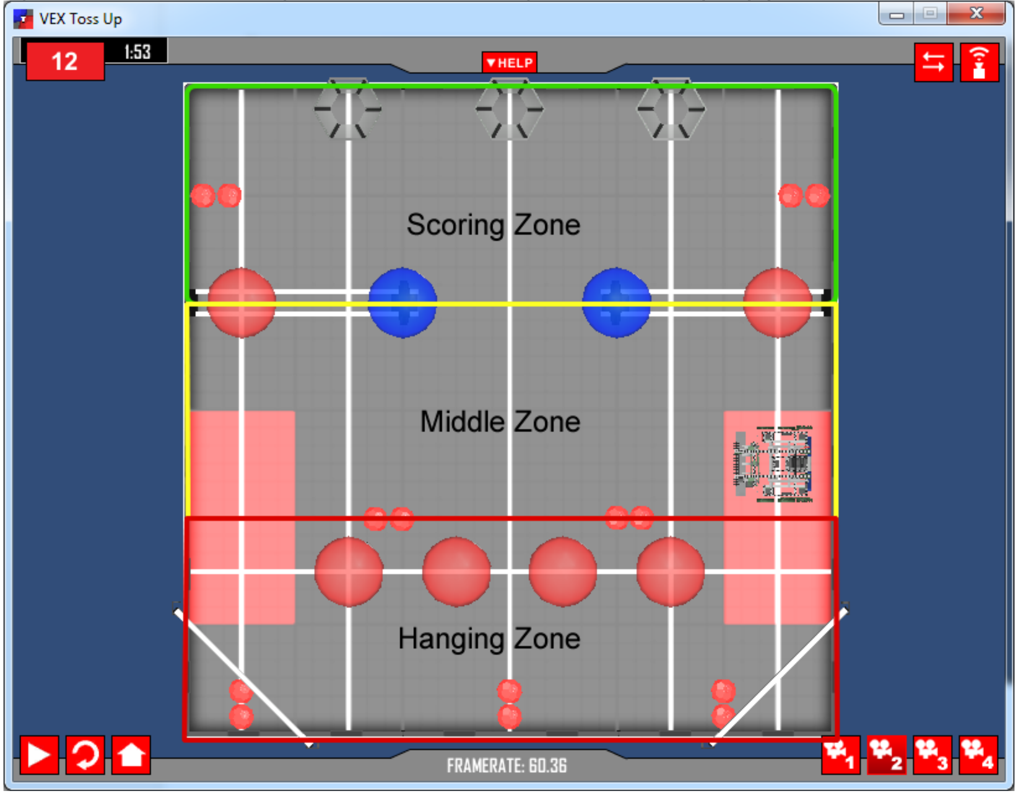
Touch Penalty – A penalty that results from the *Player* switching the *Robot* or loading a new program while the *Robot* is not on a *Starting Tile.* A *Touch Penalty* will cause the one (1) *Buckyball* farthest from the *Hanging Zone* to be returned to the *Hanging Zone* automatically. *Locked Up Buckyballs* are protected from the *Touch Penalty*.

Zone – The *Hanging Zone, Middle Zone* or the *Scoring Zone*

Game Scoring

- A *BuckyBall Scored* in the *Middle Zone* is worth one (1) point

- A red *Large Ball Scored* in the *Middle Zone* is worth five (5) point
- A *BuckyBall Scored* in the *Scoring Zone* is worth three (3) points
- A red *Large Ball Scored* in the *Scoring Zone* is worth fifteen (15) points
- A *BuckyBall Locked Up* in a *Side Goal* is worth nine (9) points
- A *BuckyBall Locked Up* in the *Center Goal* is worth six (6) points
- A red *Large Ball Locked Up* in a *Goal* is worth a two times (2x) multiplier to the score of all of the *Buckyballs Locked Up* in that Goal
- Three red *Large Balls Locked Up* in the *Goals* is worth a three times (3x) multiplier to all of the *Buckyballs Locked Up* in *Goals*
  - The scoring multipliers do not stack. Two red *Large Balls Locked Up* in two *Goals* will give a 2x multiplier to those *Goals* (not a 4x multiplier). Three red *Large Balls* will give a 3x multiplier to all three *Goals* (not a 6x multiplier).
- A *Robot* that is *Hanging* is worth thirty (30) points
- A *Robot* that is *Hanging with a Ball* is worth fifty (50) points
- A blue *Large Ball* in the *Middle Zone* is worth negative five (-5) points
- A blue *Large Ball* in the *Scoring Zone* is worth negative fifteen (-15) points



Figures 3: *Overhead views of the field.*

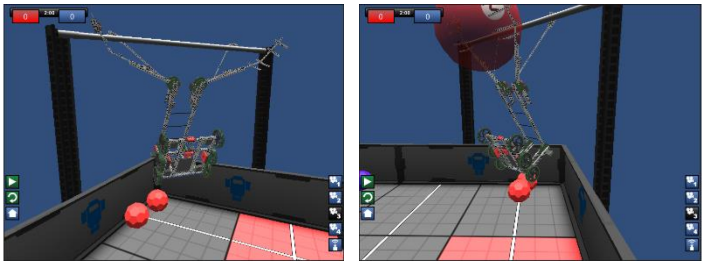


Figure 4: *Pictured above at the left is a Hanging Robot and at the right is a robot Hanging with a Ball.*

General RVW Game Rules

<G1> At the beginning of a *Match*, *Players* can choose to begin in any of the four *Starting Tiles* to start their program from.

<G2> During a *Match*, *Robots* may be operated only by software running in *Autonomous Mode.*

<G3> During a *Match*, a Player may can load new code onto their robot or switch their robot given the following conditions:

- **Condition 1** - A *Robot Autonomously* moves from the *Starting Tile*, does something, and *Returns to the Starting Tile*. The *Internal Timer* stops when the *Player* presses the *Robot Reset Button* and the player can load a new program onto that robot and execute that code or the *Player* can choose another *Robot*, load a new program and execute that program.
- **Condition 2** – A Robot Autonomously moves from the Starting Tile and drives directly to another Starting Tile. The *Internal Timer* stops when the *Player* presses the *Robot Reset Button* and the player can load a new program onto that robot and execute that code or the *Player* can choose another *Robot*, load a new program and execute that program.
- **Condition 3** - A Robot Autonomously moves from the *Starting Tile*, does something and then travels to a different *Starting Tile*. The *Internal Timer* stops when the *Player* presses the *Robot Reset Button* and the player can load a new program onto that robot and execute the code or they can choose another *Robot*, load a new program and execute that program.
- **Condition 4** - A Robot Autonomously moves from the *Starting Tile*, does something, but is not able to *Return to the Starting Tile*. The Player presses the *Reset Button* and the *Internal Timer* stops and the *Robot* is returned to a *Starting Tile* of the *Player's* chosing. The *Player* can then load a new program onto that *Robot* and execute the code or they can choose another *Robot*, load a new program and execute that program. This will cause a *Touch Penalty*.

<G4> *Scoring Objects* that leave the playing field will be returned to the playing field at the closest point on the field border (from the point of exit).

<G5> *Scoring Objects* possessed by *Robots* that *Return to the Starting Tile* when the *Player* presses the *Robot Reset Button* will be loaded onto the Robot when it is redeployed.

<G6> *Scoring Objects* possessed by *Robots* that do not *Return to the Starting Tile* when the *Player* presses the *Robot Reset Button* will be left on the playing field at the position of the *Robot* when the *Player* pressed the *Robot Reset Button*.

<G7> Scores will be calculated for all *Matches* at the highest point value within the 120 second time period of the *Match*.

<G8> There are no possession limits for game objects in the CS2N VEX Toss Up Programming Challenge.

<G9> The *Player* that scores the highest amount of points in the shortest amount of time wins. In the event of a tie the Robot Virtual World team will use the following *Tie Breaking* rules to determine a winner (in this order).

- **Tie Breaker 1 –** The *player* that scores the highest amount of points in the shortest amount of time wins; the *Internal Timer* tracks times in milliseconds.
- **Tie Breaker 2 –** If *Players* are tied and have identical times, then the *Player* that has used the *Robot Reset Button* the least number of times wins.
- **Tie Breaker 3 –** In the case of Tie Breaker 1 and Tie Breaker 2 being unable to determine a winner, the winner will be decided by who submitted their score first.

<G10> All submissions must include the program(s) used to achieve the high score, any include or header files needed to run the program, the comments at the top of the program filled out fully, and a fully filled out Programming Order file. Any submission that lacks these documents will be ineligible for prizes.

- Blank RBC templates can be found in the Toss Up Virtual World, as well as the VEX Toss Up Sample Programs folder. These files contain the comments at the top of the program that must be filled out and included with every program.
- A blank Programming Order text file can also be found in the VEX Toss Up Sample Programs folder. This file must also be completed and included with your file submission.
- Multiple programs should be zipped (with a completed Programming Notebook) for submission. The zipped folder should follow a Lastname_Firstname_Competition.zip naming convention. Example: Smith_John_VEX.zip or Doe_Jane_VEX.zip.

<G11> Any *Player* intentionally using glitches or hacks in the RVW environment to obtain a higher score than would normally be possible may be subject to disqualification from the Competition.

<G12> Each *Player* is eligible for only one prize per *Competition*. In the case of multiple high scores that are eligible for prizes that are submitted by the same *Player*, only the highest score will be used in determining prizes for that *Player*.

<G13> All *Players* must adhere to the Robot Virtual World Competition Rules as they are written, and must abide by the listed intent of the rules. Every *Player* has the opportunity to ask for official rule interpretations in the RVW Robotics Competition Question & Answer Forum at www.robotc.net/forums . Any responses in this Q&A forum should be treated as official rulings from the RVW Robotics Competition Game Design Committee, and represent the correct and official interpretation of the RVW Robotics Competition Rules.

<G14> There may also be periodic game updates. You will receive the updates through the email that is associated with your CS2N account. All updates will also be posted at the RVW blog and at the appropriate RVW Robotics Competition forum in the www.robotc.net/forums. These updates are also “official” parts of the *CS2N VEX Programming Challenge* rules.

Terms of Service

You retain your rights to any Content that you submit, post, or display through the Computer Science Student Network (CS2N) (including but not limited to programs, code, screenshots, pictures, or other digital media). By submitting, posting, or displaying Content through CS2N, you grant us a worldwide, non-exclusive, royalty-free license (with the right to sublicense) to use, copy, reproduce, process, adapt, modify, publish, transmit, display and distribute such Content in any and all media or distribution methods (now known or later developed).

You agree that this license includes the right for CS2N to provide, promote, and improve the Services and to make Content submitted to or through the Services available to other companies, organizations or individuals who partner with CS2N for the syndication, broadcast, distribution or publication of such Content on other media and services, subject to our terms and conditions for such Content use.

We may modify or adapt your Content in order to transmit, display or distribute it over computer networks and in various media and/or make changes to your Content as are necessary to conform and adapt that Content to any requirements or limitations of any networks, devices, services or media.

You are responsible for any Content you provide and for any consequences thereof, including the use of your Content by other users and our third party partners. You understand that your Content may be syndicated, broadcast, distributed, or published by our partners and if you do not have the right to submit Content for such use, it may subject you to liability. CS2N will not be responsible or liable for any use of your Content by CS2N in accordance with these Terms. You represent and warrant that you have all the rights, power and authority necessary to grant the rights granted herein to any Content that you submit.