New York 1901 and VEX IQ Curriculum

Goals

The main goal for this curriculum is to teach young people about entrepreneurship, especially in the sense of strategy. The student are forced to evaluate and discuss whether to take the short or the long term investments. They will also have to evaluate what a good investment is based on the tools they have at their disposal which in this case it is a clawbot with various sensors.

Method

The students will build small scale educational robots of the brand Vex IQ, more specifically they will build what is called a clawbot. The clawbot is essentially a driving robot with an arm and a claw at the end of that arm. Furthermore, the clawbot will have different sensors connected to enable it to be more or less autonomous. The student will also be responsible for programming the robot.

In building and programming the robot, the student will learn the advantages of the clawbot, but just as importantly its limitations. Knowing this is crucial in making the right decisions and investments.

The decision making and investments will be made while playing a custom version of the board game "New York 1901". The game will be played in similar way to that of the normal game, however, in the custom version the players are using their knowledge of the robot and how it works to obtain extra points.

Target Group

Males and females with age ranging from 14 to 25 who are interested in learning about entrepreneurship and robotics.

Number of Participants

6 to 16 participants. The participants will be divided in 3 to 4 teams with at least 2 people per team. The participant should be evenly spread out across the teams.

Tools

- At least three Vex IQ starter kits with sensors, each kit results in one clawbot.
- The same number of PC's as the number of clawbots
- The PC's must have the ROBOTC software installed so the students can program the robot.
- One New York 1901 board game
- 10 pieces of red, 10 pieces of green paper, 10 pieces of blue paper. All these pieces must have at least have the dimensions 30 cm X 30 cm.
- 24 pieces of A4 paper rolled into 12 paper balls.
- A lot of transparent tape.

Duration

It is expected that this curriculum will have a duration of 9 hours divided into 9 amount of lessons. Each lesson being one hour.

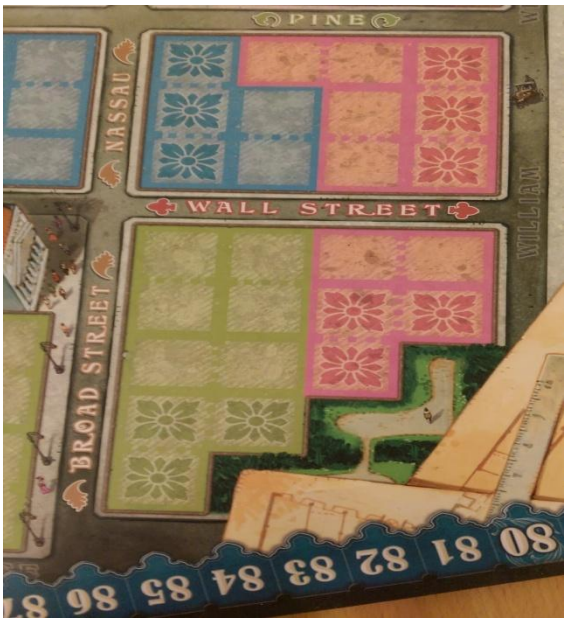
Rules of The Custom Game

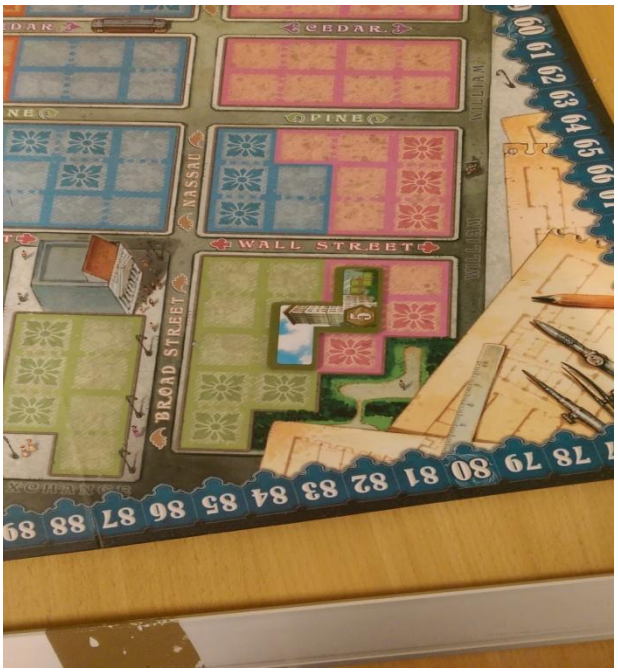
The rules for the custom game are the same as the rules specified in the manual for the normal game unless it is otherwise specified here. So if there is something you are uncertain of because it is not specified in this section read the manual that comes with the game for clarification.

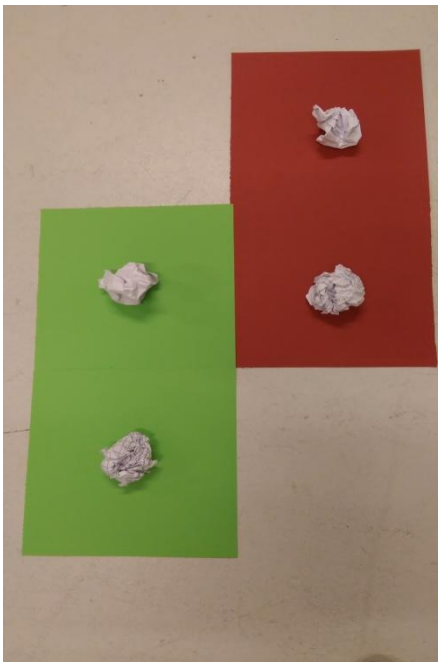
Before you read this this section with the custom rules you should read the manual with the rules for the standard game. That way this section will make the most sense.

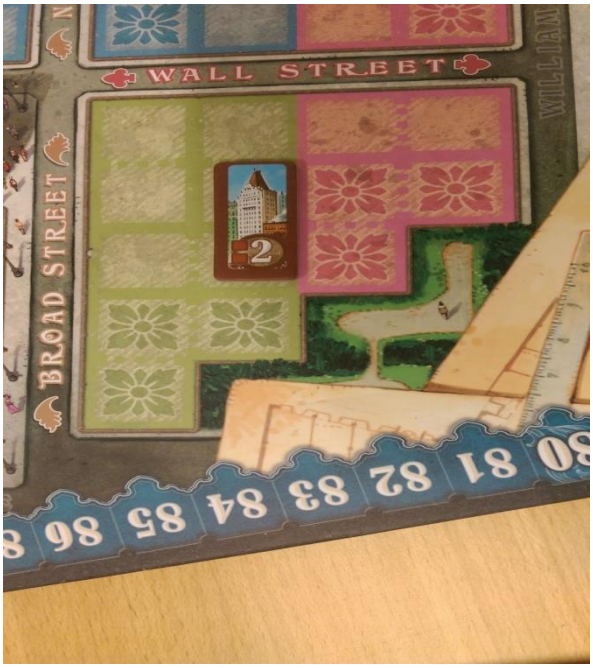
Tear Down

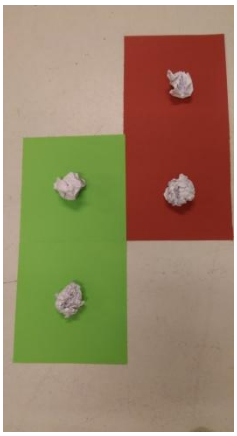
In the standard version of game you have to tear down old buildings in order to upgrade to a new building. The same goes for the custom game, however now the players have to use the robot to tear down these buildings (Robot will tear down one building at a time and not all of the ones necessary at once). Instead of having the robots drive on the board game, you create a shape that is the same as the one of buildings that are to be torn down out of the colored pieces of paper. Not only does the shape have to be the same, the colors of the grid of the colored paper has to change according to that of the board game. Here are some examples:

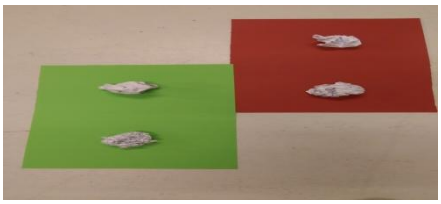



CEDAR
PINE
WALL STREET
BROAD STREET
78 79 80 81 82 83 84 85 86 87 88




WALL STREET
BROAD STREET
WILLIAM
2
80 81 82 83 84 85 86





All of the participants from all the teams help make this new grid on the ground if it does not already exist. The colored pieces of paper should be fixed to the ground and to each other using transparent tape (it is important to use transparent tape, so the color sensor does not detect it). The pieces of paper should be fixed to the ground so they do not move when the robot drives on them. When the grid has been properly made, place one ball of paper directly in the middle of each of the colored pieces of paper. See the examples above.

The game will now continue without the team that decided to tear down buildings. And this team can only re-enter the game when the teardown has been successfully completed. By the robot tearing down buildings is meant that the robot is picking up balls of paper in a grid.

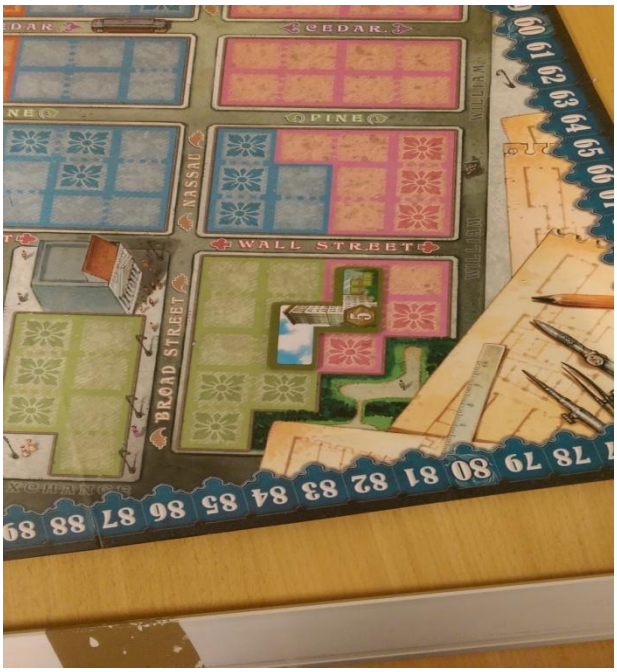
In order for the team to tear down the buildings using the robot, they have to look at shape of the grid and the pattern of colors, and then program the robot to pick up all the balls of paper in the grid, and putting it on the back of the robot. The student are awarded extra points for each sensor they utilize.

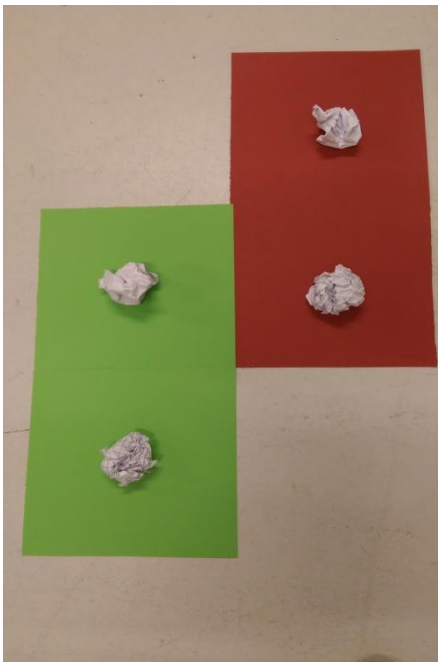
Extra Points

Ten extra points are awarded for each sensor used when tearing down buildings. However, the touch sensor will give negative ten points as this sensor can make the tear down process easier for the teams. Here are a few examples of how to utilize some of the sensors.

Motor Sensor

If you want to tear down this building:





you have to turn at a specific points. If you turn when a paper ball is picked up you turn to early. Instead you could have the clawbot drive forward by turning the wheels a specific number of degrees after the paper ball is picked up. Going based on degrees means using the motor sensors. Here is a coding example:

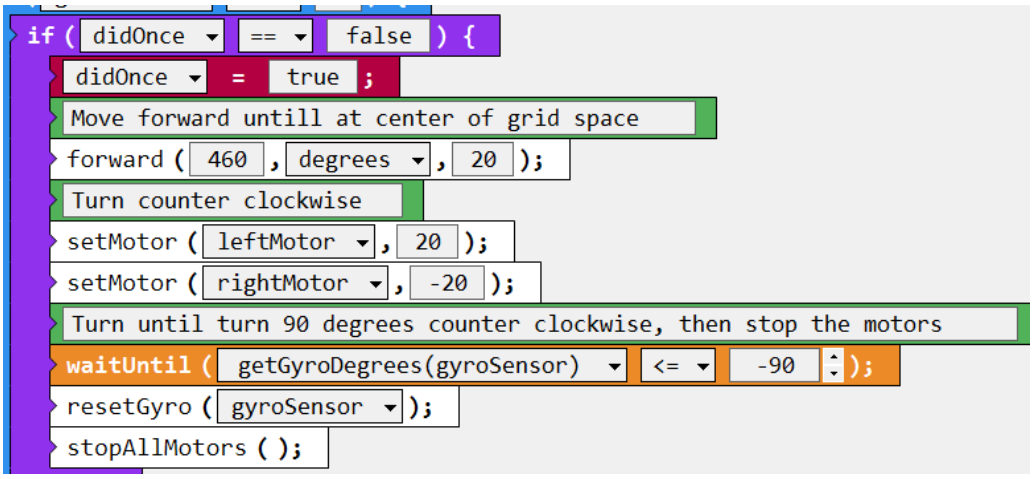
```
if ( didOnce == false ) {
    didOnce = true ;
    Move forward untill at center of grid space
    forward ( 460 , degrees , 20 );
```

When the clawbot's wheels has turned a certain number of degrees it is at the center of the the grid space and thereby ready to turn. Please note that 460 rotations is a placeholder number and you have to experiment to find the correct number.

The variable didOnce is used to make sure that it only happens once and not until the next paper ball is picked up. The didOnce is reset to false every time the clawbot picks up a new paper ball. This means that the didOnce can be used in every grab-loop.

Gyro Sensor

Looking at the scenario from the motor sensor example you still have turn. From the example you have to turn 90 degrees. This is made simple with the use of the gyro sensor.

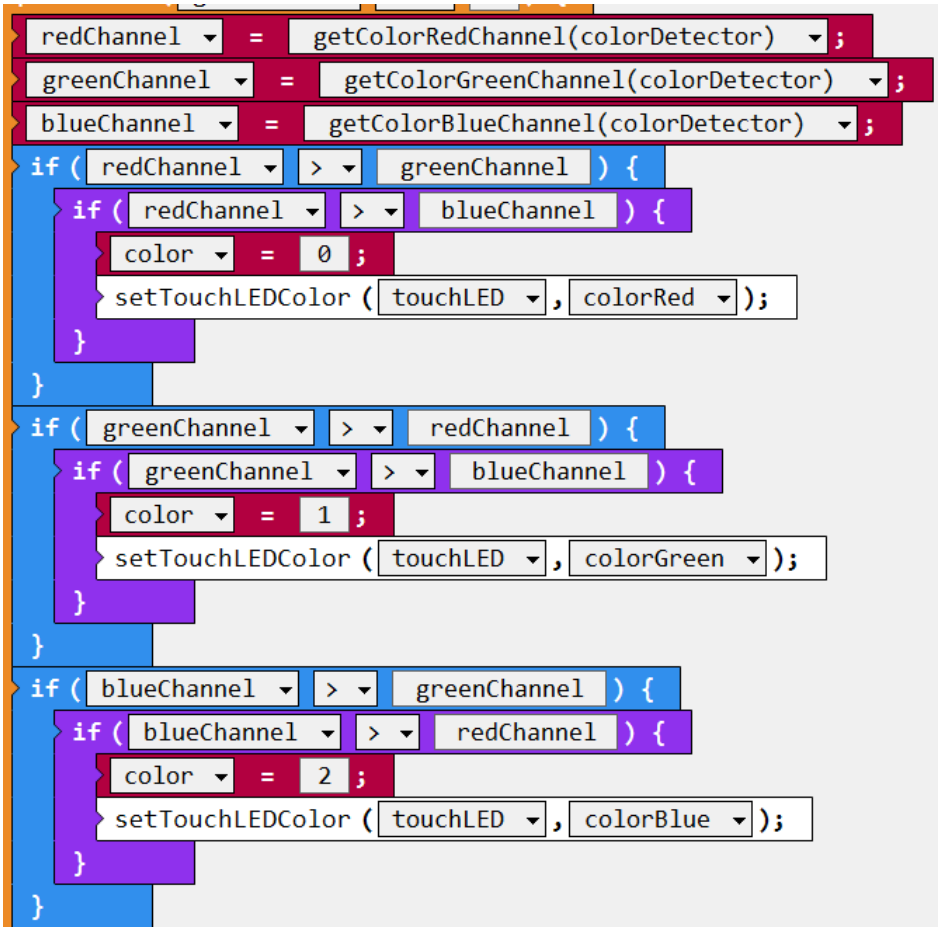


As you have noticed this includes the part from the motor sensor example. After the forward motion the clawbot turns until it has turned 90 degrees clockwise, then it reset the gyro sensor and stops the turning.

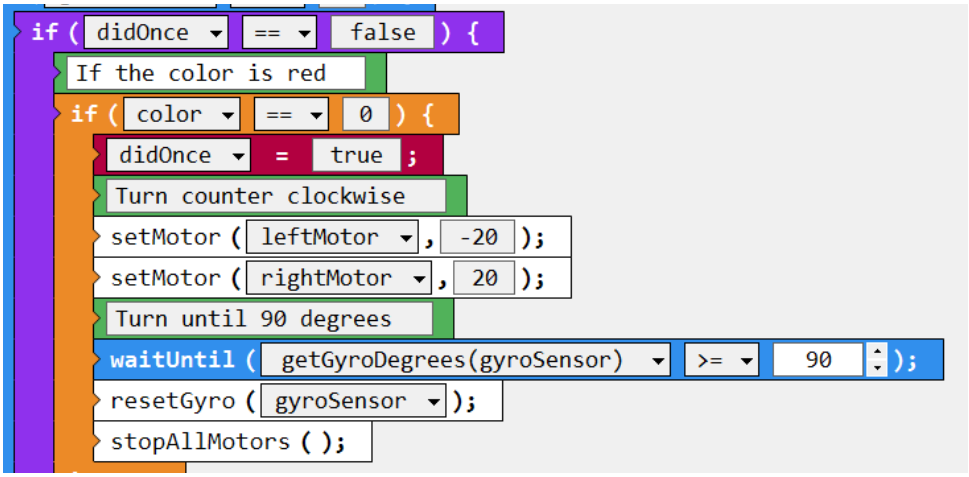
Color Sensor

The color sensor can be used instead of the motor sensors to determine when to turn because of its placement. However, this only works when the dimensions of the colored paper is 30 cm x 30 cm. If your paper (grid spaces) have different dimensions you have to adjust the placement of the color sensor in order for it to function as intended.

The way to use the color sensor is by having the clawbot start turning when it detects a new color. Here is the code example:



The above code illustrates one way of checking the color of the ground. It looks at which color channel has the highest value. It shows it on the touch sensor (which is not necessary) and the information is also stored in a variable called *color* (0 = red, 1 = green, 2 = blue). The color information is then used to determine when to turn.



In this example the color variable value corresponds to the color of the paper it is supposed to turn on. Please note that this only works when there are different colors present.

Touch LED Sensor

Please remember that the use of this sensor gives negative ten points. However, it can be useful for avoiding time loss. The negative ten points are only awarded when it is touched not if it is present in the code. So it could be a good idea to have it as a kind of safety.

It can be programmed to pick up paper balls in case the distance sensor does not detect one. Example:

```
if ( getTouchLEDValue(touchLED) == true ) {
  stopMultipleMotors ( leftMotor , rightMotor , noMotor , noMotor );
  moveMotor ( clawMotor , 70 , degrees , 20 );
  moveMotor ( armMotor , 1070 , degrees , 50 );
  moveMotor ( clawMotor , -70 , degrees , 20 );
  grabCount = grabCount + 1 ;
  setMotor ( armMotor , -50 );
  waitUntil ( getBumperValue(bumpSwitch) == true );
  stopMotor ( armMotor );
  didOnce = false ;
}
```

It can also be used to make turning easier. You could make the robot turn while it is pressed for example:

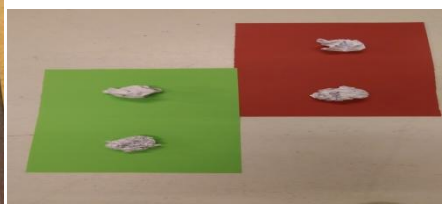
```
if ( getTouchLEDValue(touchLED) == true ) {
  setMotor ( leftMotor , 20 );
  setMotor ( rightMotor , -20 );
}
```

Additional Information About Extra Points

In order to make the participants think more about the building placements, there are restrictions as to which sensors can be used with certain pattern shapes.

Straight Patterns

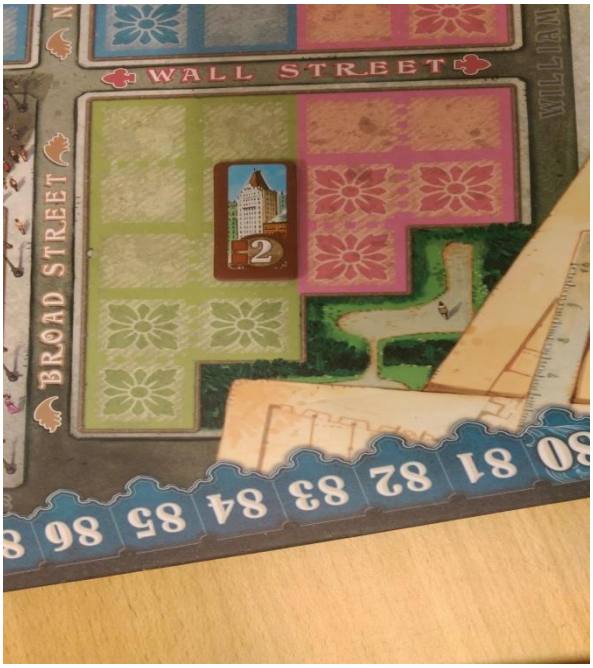
For the straight patterns such as these:



it is not allowed to use the gyro sensor or motor sensors.

Single Colored Patterns

For the single colored patterns such as this:



it is not allowed to use the color sensor.

Lessons:

<u>First Lesson</u>

- General introduction.
    - What are the goals of the course?
    - How is the curriculum structured?
- The teacher trainer involves the participants in the idea of strategies

    - Strategy is to be understood along the lines of: The act of creating a plan to achieve specific goals, where resources and other matters that influence achieving the goals are taken into account. (Such as competitors and their strategies and yours and competitors' strengths and weaknesses).

- The teachers introduces the ideas of the game.

Second Lesson

- The teacher/trainer teaches the participants about the Vex IQ clawbot
    - The teacher/trainer explains the functionality of each sensor and actuator and the basics of how they work.
    - While also describing how they can be utilized and combined to solve different tasks.

Third Lesson

- Formation of teams
    - The number of teams should match the number of clawbots.

- The teams build the clawbot using the booklet that comes with the Vex IQ kit booklet and the video provided for this project.
    - Each team collaborates in building the teams clawbot

Fourth Lesson

- The teacher/trainer introduces the participants to the "Graphical RobotC". program that is used for programming the robot.

- The teacher/trainer teaches the participants the about the relevant parts of programming the robot in "Graphical RobotC".
    - This includes teaching about variables, relevant predefined functions etc..

Fifth Lesson

- Formation of teams
    - The number of teams should match the number of clawbots
- The teacher/trainer will teach the participants about the "curriculum code" which is used as the basis for playing the custom version of "New York 1901".

- Afterwards the teams will be applying their knowledge and experiment with the clawbot and the code, so they can get a better understanding of the clawbot and the programming
    - This also leads to a better understanding of the weaknesses and strengths which is essential.

Sixth Lesson

- The teacher/trainer teaches the participants the rules of both the standard and the custom game of "New York 1901".
    - It is recommended that the teacher/trainer asks the participants to think about strategies while the rules are explained.

- The teams form their strategies in secret of the other team(s).
    - The strategies must take into account the weaknesses and strength of the robot according to tasks it solves. It must also take into account how to get the highest score at the end of the game.
    - However, the instructor has to know each team's strategy and write it down.
    - The teams cannot alter their strategy during the game

Seventh Lesson

- It is now time for the teams to play the custom game.
    - The game should last until ten minutes before the end of the lesson even though the game has not ended according to the rules.

- The teacher trainer asks each team to discuss the strategy of the other team(s) and write down the strategies they think the other team(s) used.

Eighth Lesson

- The teams will explain the strategy they believe the other team(s) used and why they came to that conclusion.

- The class will have a discussion based on the strategies of the teams.

- The class will have a discussion about how they could have altered their strategies to have had an advantage over the other team(s).

Ninth Lesson

- The teams will play the game once more, however, this time the teams can alter their strategy during the game. However they can only do so, if they guess the strategy of another team.
    - The teacher/trainer will keep track of this