CTI Charlotte Teachers Institute
Collaborative Teacher Education

Tic-Tac-Toe-BOTS

by Aaron Kollar, 2014 CTI Fellow
Piedmont IB Middle

This curriculum unit is recommended for:
5th to 8th Grade Math

Keywords: Tic-Tac-Toe, Artificial Intelligence, Minimax, Turing, Math, Games, Turing Test, Mathematical Practices, Common Core.

Teaching Standards: See Appendix 1 for teaching standards addressed in this unit. (Insert a hyperlink to Appendix 1 where you've stated your unit's main standards. For directions on how to insert a hyperlink, see Fellows Handbook, p. 29.)

Synopsis: This unit focuses on getting students to logically think their way through a game of tic-tac-toe. Students are put into groups, and they have to create a set of directions that anyone could follow in order to play a game of tic-tac-toe. So, in a sense, they are "writing a program" that can be followed, which is the basis for all artificial intelligence. The students will then play each other in tic-tac-toe, with the caveat being that they have to follow the instructions that they have written exactly. At the end of the unit, the students will analyze some data that was collected while the games were happening in order to determine which program was the most successful.

I plan to teach this unit during the coming year to 120 students in 6th grade Math

I give permission for the Institute to publish my curriculum unit and synopsis in print and online. I understand that I will be credited as the author of my work.

Tic-Tac-Toe BOTS

Aaron Kollar

Background Information

Piedmont IB Middle School

Piedmont IB Middle School is an urban public middle school located in Charlotte, North Carolina. There are approximately 947 students at Piedmont IB Middle School. Our ethnic makeup is 68% African-American, 15% Caucasian, 7% Hispanic, and 7% Asian. Piedmont is a magnet school, meaning that we serve kids from multiple feeder schools that are not necessarily in the same communities. Because of this, we have a very diverse population, both in ethnicity and socioeconomic status. We draw students from some communities that are impoverished, and approximately 44% of our students are on free or reduced lunch.[1] The culture at Piedmont encourages innovation, creativity, and hands on learning. Teachers are comfortable taking risks and conducting lessons that may not be found in everyday textbooks.

I teach 6th grade at Piedmont IB Middle, where we have both Honors and Standard classes. During the 2014-2015 school year, Piedmont has 4 Honors classes and 8 Standard classes. We are expected to teach to the Common Core State standards, on which the students are tested at the end of the school year. This curriculum unit can be used with either level of student, and it is meant to be used after any state end-of-grade testing, because it introduces some of the 7th grade Common Core State Standards. (see appendix).

Background Knowledge of Students

My sixth graders come to me every year with varying levels of math expertise. They do, however, have one thing in common. They are all either 11 or 12 years old, with a few exceptions. Even though they are at different levels mathematically, 11 and 12 year olds have a lot in common. For one, they tend to want to rush through their work. In my experience, when given a problem to solve, the most important thing to them is the answer. Many will do just about anything just to *get the answer*. They do not like to focus on the process. They do not like to take the time to draw out big tables or represent math with pictures and symbols if they do not have to. Many have a downright fear of math, mostly the ones that lack a solid foundation of computational skills. A large portion of my students struggle with open-ended tasks that involve critical thinking, mostly due to the fact that they were not exposed to this type of learning in elementary school. This unit will help to expose these students to such a task, and thus help them move away from "*What's the answer?"*

Artificial Intelligence

Emeritus Professor of Computer Science at Stanford University John McCarthy defined artificial intelligence (AI) as "*the science and engineering of making intelligent machines, especially intelligent computer programs."* If this is to be taken as a good definition of Artificial Intelligence, then it would seem that humans are so close, and yet so far, from developing true artificial intelligence. For example, when a person asks a question of Siri in their IPhone, it *appears* as though Siri has an ample amount of intelligence. She can figure things out, like how many vegan restaurants are within a five mile radius, faster than any human can. This, however, is an example of what is known as *weak* AI.

AI can currently be broken down into two subcategories, weak AI and strong AI. Why would Siri be considered weak AI? Well, for example, try asking Siri if it is possible for a baby to start at quarterback for the Cleveland Browns? Or ask Siri if a bird can compete in the Olympics? These are questions that a typical 1st grader can answer, yet Siri cannot. In other words, although Siri can produce answers to various factual questions, she cannot hold a real conversation. Another example of weak AI is IBM's supercomputer, Watson. Watson is famous for displaying the ability to destroy any human in a game of Jeopardy, even Ken Jennings, who is considered the ultimate Jeopardy player. But, if you ask Watson what color Alex Trebek's tie is, it would not be able to tell you. Nor would Alex Trebek be able to hang out with Watson and converse about their favorite types of music.

Weak AI is what we have now, and it focuses on doing a narrowly defined task, like information retrieval, really well. Strong AI, which isn't a reality yet, would be similar to what we see as AI in the movies, like machines taking on a truly human form. Alan Turing, also known as the father of theoretical computer science and artificial intelligence, designed a test, which he called the "Imitation Game" to judge whether a machine can truly be considered intelligent. A brief summary of the Imitation Game goes like this: There are three players, Player A, B, and C. One of the players, perhaps Player C, is behind a curtain, and cannot hear or see the other two. Of the remaining two players (A and B), one is a man and one is a woman. Player C's job is to determine which of the other players is a man, and which is a woman, by writing notes back and forth to Players A and B. Player A tries to fool Player C, while Player B tells the truth. Turing proposed that, if we could replace Player A with a machine, and the machine can manipulate Player C so that Player C guesses the wrong gender of Player A over half the time, then that machine can be seen as intelligent.[2]

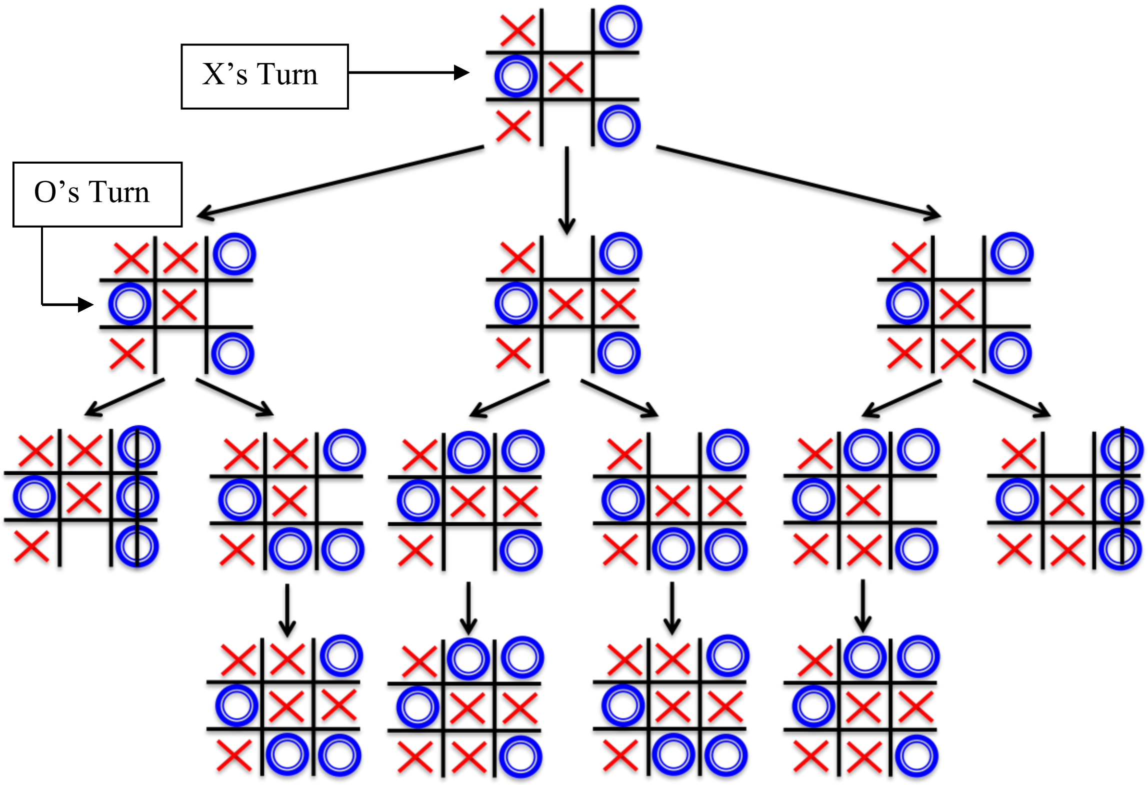
Over time, this test has evolved to become known as "The Turing Test". A more modern version of the Turing Test is conducted without Player C caring about the gender of Players A and B, but rather just focusing on which of the other players is human, and which is a computer. Within the field of AI, there is an annual contest in which computer programmers from all over the world try to create a machine that fools Player C into believing that it is human. The prize, known as The Loebner Prize, is awarded to the machine that fools the most humans (or Player C's, if you will), into believing that it is a human too. The creator of this contest, Hugh Loebner, has pledged a $100,000 prize to any programmer that can design a machine that the judges *cannot* tell whether it is a machine or a human. To date, the $100,000 prize has not been awarded to anyone.[3]

So far, humans have not developed machines that we can actually consider "intelligent", but we have come a long way since Alan Turing first proposed the Imitation Game, in 1950. Although machines such as Watson and Siri come about as close to strong AI as we can get, they cannot be classified as strong AI.

Game Playing

The area of AI that this unit focuses on is the area of game playing. Programming a machine to play a "perfect" version of a game is definitely a form of weak AI. A key component of AI as it relates to game play is *looking ahead*. The best chess players in the world have the ability to quickly look ahead while playing the game, and thus analyze which moves will result in a positive outcome and which moves will result in a negative outcome. When programming a computer to play a perfect game of chess, the challenge to the programmer is to give the computer the ability to "see" further into the game than the human. Amazingly, in chess, there are 10^{120} different possible states that the game could be in at any given moment. To put that in perspective, there are *only* 10^{80} atoms in the entire universe! Interestingly, when computers are programmed to play chess at a superhuman level, such as Deep Blue, the supercomputer that famously beat Garry Kasparov in 1997, they are not able to "see" all 10^{120} game states, for that would take up too much memory. It is sufficient to program these computers with the ability to "see" further than a human, while not "seeing" every single game state.

Since many 6th graders do not know the rules to Chess, and since it thinking through a game of chess is so difficult, my unit focuses on the simpler (although not simple) game of Tic-Tac-Toe. In a game of Tic-Tac-Toe, there is still a lot of analyzing to be done to create the "perfect" game-playing algorithm. To analyze each move of Tic-Tac-Toe (or Chess), a programmer can use a game tree, like the one below, for assistance. In this game tree, the players are on the 6th move of the game, and it is Player X's turn:



If we look at the game tree above, we see that, from the top board, this game will end in one of two ways, in a win for O or a tie. If a programmer was trying to create a computer to play perfectly as the "X" player, it would seek to follow the tree down the middle, as opposed to the moves on the left and right in the second row of the tree, as they could potentially result in a loss. To design a perfect Tic-Tac-Toe player, a programmer would make a tree similar to the one above, with *every* possible move. Then, the programmer would score each move from the bottom up, probably giving a score of "0" to any move that can result in a loss. This type of programming in game play is also known as the Minimax Algorithm.

This unit *will not* seek to create the first machine to possess strong AI. As crazy as I may be for teaching middle school for over ten years, I am not that crazy. I would like to think that this unit, however, may inspire a student to try and be the first human to develop strong AI. In this unit, the students will work to create machines that follow a game play algorithm, just like Deep Blue did in 1997. Of course, our machines will not be nearly as complex as Deep Blue, but our machines will be able to accomplish a specific task, and hopefully do it very well.

Rationale and Teaching Strategies

Quite recently, a veteran teacher of 15 years decided to shadow a 10th grade student for a day, and then a 12th grade student for a day. The teacher, whose name is Alexis Wiggins, was shocked at what she discovered, and she immediately wanted to go back in time and teach differently. The reason that she wanted to go back and teach differently was because she couldn't believe how much of her day shadowing was spent sitting and listening. She estimated that up to 90 percent of a student's day is spent sitting quietly and listening. Wiggins also could not believe how tired she was at the end of the day, because in her words, "sitting is exhausting."[4]

It is my hope that this unit reduces that 90 percent, and that it gets the students up, moving, collaborating, and thinking. This unit is meant to motivate students to be excited about learning something. When students are progressing through this unit, I don't want them to see it as learning math. I want them to see it as playing and analyzing a game. This unit will not involve lecturing or note taking, but rather strategizing and analyzing.

Tic-Tac-Toe is a game that just about everyone has played. It is fun, simple, fast, and deceivingly strategic. By using the game of Tic-Tac-Toe as a backdrop, I feel that I can hook most students into the unit. Sometimes as a middle school math teacher, I feel like a salesman that continuously has to sell the notion that learning math is important, and can also be fun, to my students. Many times, they are not interested in buying. But a lesson about playing Tic-Tac-Toe? That should be an easy sell, even to the most disinterested students.

In the Common Core State Standards for math, there are eight overriding "Mathematical Practices" that are common throughout all grade levels, from kindergarten through high school. These principals are: *1) Students will make sense of problems and persevere in solving them. 2) Students will reason abstractly and quantitatively. 3) Students will construct viable arguments and critique the reasoning of others. 4) Students will model with mathematics. 5) Students will use appropriate tools strategically. 6) Students will attend to precision. 7) Students will look for and make use of structure. 8) Students will look for and express regularity in repeated reasoning.*[5]

Although my unit does not really focus on specific 6th grade Common Core objectives, the greater goal is to have students practice most or all of the eight mathematical practices listed above as they move through the unit. Since this unit focuses more on the eight mathematical practices than it does more specific 6th grade objectives, it may be best used after any state testing is finished. The topic of predicted outcomes, which this unit focuses on, relate more to 7th grade Common Core Standards. In addition, because the eight mathematical practices are common to all grade levels, this unit can be done on a variety of grade levels, and I believe the best range would be students in 6th grade to students in any high school math class.

Since so much of the materials that are provided to math teachers involve test prep and multiple choice answers, I think that problem-based units like this are essential to develop higher order thinking skills. Problem based learning is an approach where you give the students a real life problem, or task, and use that problem to teach content. If we use Bloom's Taxonomy as a backdrop, most of the time when students are answering multiple choice questions or taking notes, they are focused on remembering, understanding, and applying knowledge (the bottom three levels in Bloom).[6] In my unit, I want to focus more on the higher order thinking skills, which are to synthesize, analyze, evaluate, and create. In my opinion, these critical thinking skills are massively under-taught in math classes, as teachers feel more and more pressure to produce students that can perform really well on multiple choice math tests.

Research, however, supports the fact that problem-based learning and open-ended projects can produce students that are good at both critically thinking, and performing well on standardized tests. Problem based learning also gets students excited about learning in a way that seated, direct instruction does not. Students are given a real life problem to solve, or a task to be completed, and they feel more in control of the outcome, as opposed to copying notes, where they have no control.[7]

Another focus of this unit is teamwork. In my 14 years as a teacher, I have realized that students have to be taught how to work in a team. You cannot just put four random students in a group, and say "Go get 'em!", *especially* with middle school students. It has been reported that the two most desirable skills among employers in the United States are critical thinking, and the ability to work on a team.[8] If we are really to prepare our students for success beyond high school and college, we have to take the time to focus on these skills. So, although it is not a Common Core State Standard, this unit will help the students understand what it means to be a part of a team.

Before beginning this unit, I think it is essential for students to learn about the history of Tic-Tac-Toe, and just in case there is a student that has never played before, the class should take a few minutes to play a practice game or two and go over the rules. It appears that a variant of Tic-Tac-Toe was played by the Romans in as early as the 1st century BC. In the Roman version, each player had only 3 pieces, which they could move around the board continuously until a player got three in a row. (This version actually sounds pretty fun!) Some scholars argue that Tic-Tac-Toe can be traced to the Egyptians.[9] Since 6th graders in North Carolina study both the ancient Roman and Egyptian empires in their humanities class, this provides an excellent opportunity for some cross-curricular themes as well. In a perfect world, I would have my humanities teacher do a lesson on how Tic-Tac-Toe fit into the Roman or Egyptian lifestyle before I start the unit, so that students can have a connection between their humanities class and their math class.

The overall goal of this unit is for the students to create machines (I like to call them Tic-Tac-ToeBots) that can intelligently play a game of Tic-Tac-Toe. I use the word "machine" very loosely, because they are basically just going to write a set of rules that will dictate each move in a game of Tic-Tac-Toe. In order to generate this set of rules, the students will have to analyze a plethora of different situations and scenarios. As a class, we will then see whose set of instructions, or programs, are the best.

Activities

Lesson 1 – A Mini-Lesson on Giving Specific Directions. (1/2 of a class period)

The first lesson of this unit, which could be considered a mini-lesson, could also be used in combination with lesson 2. The goal of this lesson is to show students how literal computer language has to be in order for a machine to understand them. This lesson will also require the teacher to do some acting. In this lesson, the teacher should bring in a loaf of bread, some peanut butter, a plate, and a knife. The teacher then tells the students that he or she is a robot. The students' task is to tell the "robot" how to make a peanut butter sandwich. As the students give their directions, the teacher must follow them as literally as possible. For example, if a student says "Put the peanut butter on the bread", the teacher should set the jar of peanut butter on the unopened loaf of bread. If a student says "Open the jar of peanut butter", the teacher could grab the knife and pretend to saw open the jar of peanut butter. This lesson should not only be extremely amusing, it should show the students that machines need to be programmed with very specific directions.

Lesson 2 – Introduce the Unit and Tic-Tac-Toe Language (1 to 2 class periods):

After the students have seen how specific they must be in order to program a machine, it is time to tell them that they are going to create an artificially intelligent robot that can play Tic-Tac-Toe. Briefly discuss what artificial intelligence is, and how we see it in our daily lives (Google, Siri, Chess Supercomputers, Pandora Radio, etc). Although tempting to get into the science of the Minimax Algorithm and game trees, these topics may be a little too heavy for 6th graders, and are best left out.

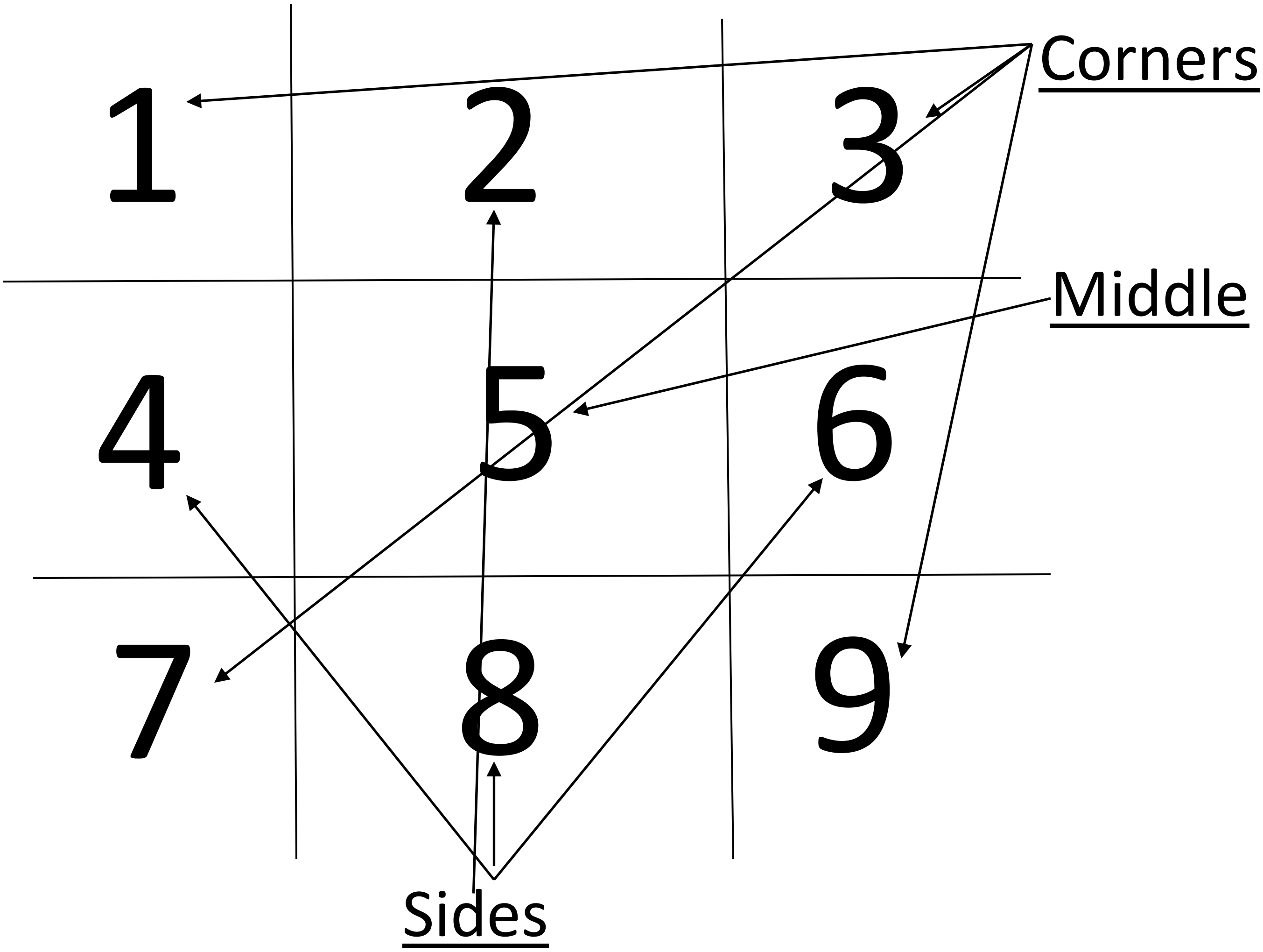
To start, divide the class in half so that you can have a class game of Tic-Tac-Toe, which the teacher will draw on the board. The teacher will still pretend to be a robot, so the class will have to be very specific when telling the teacher where to place the X's and O's. After the game is concluded, the teacher should ask the students if there were different words that were commonly used to tell the teacher where to move. Create a list on the board of words like "corner", "side", "middle", "opposite", "block", etc., that were useful in telling the teacher what to do.

Next, the students should be put into groups. How the groups are set up should be at the teacher's discretion, with the optimal grouping being partners or groups of three. In groups of four or five, it is too easy for some of the less boisterous students to fade away into the background. As mentioned earlier, working in a team is not something that is completely natural for middle school students. Something has to be done to build their camaraderie as a team. To accomplish this, the teacher should tell each team to think of a team name. Each group should then be given a piece of construction paper, so that they can create a name and picture of their Tic-Tac-ToeBot. The act of naming their team and designing their robot together should naturally relax the students and build cohesiveness among the group.

The next thing the teacher needs to teach the teams is how to disagree with each other. When middle school students are given the opportunity to compete, sometimes they naturally feel pressure. When people are under pressure, they tend to have a shorter fuse. The teacher should take the time to discuss certain ways to show someone that you disagree, without a shouting match ensuing. The keys are to stay calm, not to raise your voice, and put yourself in the other person's shoes.

Once the teacher has discussed how to, and how not to, disagree, it is time for a quick classroom discussion. Refer to the list on the board of some of the key words that were used to tell the teacher what to do during the class game of Tic-Tac-Toe. Instruct the students that, in order to program our robots, we have to use specific language that the robot can understand. Point out to the students that if you tell the robot to go in a "side", that there are four different spaces on the board considered a side, and likewise with the corners. Draw out of the students that each space on the board should be numbered, in case the robot has more than one side or corner to move into. (Figure 1)

Figure 1



Now, the students will get a chance to briefly try and develop specific instructions for their robots, based on the language and numbers that have been developed as a class. The teacher should instruct the groups of students to come up with as many instructions as they can think of, and write them down on paper. Start this process by writing down a couple of the more common instructions on the board, such as: *Go in a random corner* and *Go in the middle.* As a group, the students should think through a game of Tic-Tac-Toe, and write down all of the specific instructions that they can think of. It is this type of open-ended, logical reasoning that will help foster the students' critical thinking skills, as well as their ability to work in a team. If groups are stuck, remind them that they may need to get even more specific, and use instructions such as *Go in the opposite side of the opponent's last side move.* Once the groups are finished, collect all of the instructions and tell the class that you are going to combine every group's set of instruction, and come up with the best possible instructions from the whole class.

Lesson 2 (1 to 2 class periods):

Prior to lesson 2, the teacher will need to use a set of cards with all of the different instructions on them. (Figure 2). They should be cut out and separated by move, and clearly labeled. It is critical that the cards do not get mixed up. In between lesson one and two the teacher will "pretend" to have chosen the best instructions that the students wrote down in the first lesson. In reality, the teacher will use the pre-determined instructions from figure 2. I am not a fan of being dishonest with my students, but I think that this little fib will enhance the authenticity of this unit for the students. The teacher can refresh the students' memory of the previous lesson by having a discussion about the first move of the game, and how we would tell a robot how to execute that move. Remind students about the language of "center", "middle", and "side". The teacher can write the possibilities of the first move of the game on the board. They are:

1. Go in the center

2. Go in a random corner

3. Go to a random side

Explain to the students that anytime the word "random" appears as a direction, it means that we will pick numbers out of a hat to determine which side or corner the robot will move into. (Eventually, during game play, one hat will represent the *sides* with numbers 2, 4, 6, and 8 in it, and one hat will represent the *corners*, with numbers 1, 3, 7, and 9 in it). The teacher will then pass out sets of cards to the students that were chosen as the best instructions for our robots that were generated by the class *wink, wink.*

The next step is to instruct the students that they are going to "program" their robot as if their robot was Player 1 (or the X's) in a game of Tic-Tac-Toe. Tell them that first, they should go through ONLY the cards labeled ***Player 1's First Move***. As a team, the students should put the cards in order from "best first move" to "worst first move", and then record those moves in order either on paper or on a computer or tablet. Basically, the top card will represent the optimal instructions for a given move, and the bottom card will represent the move that is least likely to succeed. It is important that at least one team program their robot with optimal game play, as described in lesson 4 of this unit. If the teacher circulates the room and notices that no team is starting Player 1's first move by going in a corner, the teacher should pick a group of students to "coach", and help those students organize their instructions for optimal game play. The reason for this is that, at the end of the unit, the students are going to analyze the records of wins/losses/ties for each robot, to determine which one is the best. If no robot has optimal game play instructions, it is possible that every game will end in a tie. The assumption here is that most students *will not* be able to create an optimal set of instructions.

It must be stressed that the instructions ***must*** be recorded, and that they will be collected by the teacher. By collecting and copying each team's set of instructions, the teacher can ensure that there is no cheating once the actual game play starts. Some groups may need some extra motivation to record their robot's moves in order, so the teacher should assist some teams, while letting others work independently. Then, tell them to repeat this process for Player 1's second move. Instruct them ONLY to use the cards labeled ***Player 1's Second Move.*** Have each team do this for each set of cards, which will represent Player 1's first five moves. The reason that we will only represent the first five moves for each player is because Tic-Tac-Toe does not go past five moves. It is a given that the students will have a lot of questions during this part of the lesson. The teacher needs to quickly circulate the room and assist each team. As mentioned earlier, middle school students often struggle with open ended tasks like this, and we do not want frustration to set in. If conflicts arise, the students should also be reminded about *how* to disagree with someone.

After the teams have "programmed" their robot as if it was Player 1 by recording the order in which they would execute each move based on the cards that they have been given, the students should repeat the process, but this time they should program their robot as if it were Player 2 (O's instead of X's). Some teams will finish this step earlier than others. If they do, tell them to play their robot's Player 1 program vs. their Player 2 program. During this part of the lesson, it is important that the students do not give away any strategies. The teacher has to emphasize the fact that our robots are going to play each other, and in a competition, you don't want to help out your opponent. The teacher will need to collect each team's sets of instructions and make copies of them before the next lesson.

Figure 2 (Adapted from CS4FN.org)

Player 1's First Move

Go in the center	**Go in a random side**	**Go in a random corner**

Player 2's First Move

Go in the center	**Go in a random side**	**Go in a random corner**
Go in the opposite corner of Player 1's first corner move	**Go in the opposite side of Player 1's first side move**	

Player 1’s Second Move

Go in the center	Go in a random side	Go in a random corner
Go in the opposite corner of my last corner move	Go in the corner on the same side as my last corner move	Go to the opposite corner of Player 2’s last corner move

Player 2’s Second Move

Go in the center	Go in a random side	Go in a random corner
Go in the opposite corner of my last corner move	Go in the corner on the same side as my last corner move	Go to the opposite corner of Player 1’s last corner move
Block 3 in a row	If player 1 holds 2 corners, then go in a random side.	

Player 1’s Third Move

Go in the center	Go in a random side	Go in a random corner
Go in the opposite corner of my last corner move	Go in the corner on the same side as my last corner move	Go to the opposite corner of Player 2’s last corner move
Block 3 in a row	If player 2 holds 2 corners, then go in a random side.	Go for 3 in a row

Player 2’s Third Move

Go in the center	Go in a random side	Go in a random corner
Go in the opposite corner of my last corner move	Go in the corner on the same side as my last corner move	Go to the opposite corner of Player 2’s last corner move
Block 3 in a row	If player 2 holds 2 corners, then go in a random side.	Go for 3 in a row

Player 1’s Fourth Move

Go in the center	Go in a random side	Go in a random corner
Go in the opposite corner of my last corner move	Go in the corner on the same side as my last corner move	Go to the opposite corner of Player 2’s last corner move
Block 3 in a row	If player 2 holds 2 corners, then go in a random side.	Go for 3 in a row

Player 2’s Fourth Move

Go in the center	Go in a random side	Go in a random corner
Go in the opposite corner of my last corner move	Go in the corner on the same side as my last corner move	Go to the opposite corner of Player 2’s last corner move
Block 3 in a row	If player 2 holds 2 corners, then go in a random side.	Go for 3 in a row

Player 1’s Fifth Move

Go in the center	**Go in a random side**	**Go in a random corner**
Go in the opposite corner of my last corner move	**Go in the corner on the same side as my last corner move**	**Go to the opposite corner of Player 2’s last corner move**
Block 3 in a row	**If player 2 holds 2 corners, then go in a random side.**	**Go for 3 in a row**

Player 2’s Fifth Move

Go in the center	**Go in a random side**	**Go in a random corner**
Go in the opposite corner of my last corner move	**Go in the corner on the same side as my last corner move**	**Go to the opposite corner of Player 2’s last corner move**
Block 3 in a row	**If player 2 holds 2 corners, then go in a random side.**	**Go for 3 in a row**

Lesson 3 – Let’s play some Tic-Tac-Toe! (1 to 3 days):

Once the students have “written their programs” for both Player 1 and Player 2, by organizing their stacks of instructions with the best moves on top and the worst moves on the bottom, it is time to play some Tic-Tac-Toe! Before students begin play, they need to be given their “programs” back and a “Records Tally” sheet, on which they will need to write two things. They will have to write their first choice for their first two moves (aka their top card for the first two moves) as Player 1, and second, they will have to record their number of wins, ties, and losses as they play (Figure 3). Before the students begin, however, the teacher should pick two teams at random and have them play a game for the entire class to watch, with the teacher facilitating the game. The game is played by each team setting all of their stacks of cards in front of them, in descending order from 1st move to 5th move. Player 1 executes the first move by following the instructions on the top card of Player 1’s First Move. Most likely, it will say “*Go in the center”* or “*Go in a random* corner” If a team’s instructions tell the robot to go in a random corner, then they should pick a number out of the “corner hat” to determine exactly which corner to go to. Once Player 1 has executed their first move, Player 2 is up. Player 2 looks at their top card in the stack of “Player 2’s First Move”. If the top card is impossible to execute, then that card should be flipped upside down, and Player 2 will have to execute the instructions on the second card. For example, if both Player 1 and Player 2’s top card is “*Go on the center”*, then Player 2 will have to turn that card upside down and use the second card in the stack, since Player 1 is already occupying the center space. Play this game to completion, so that everyone sees exactly how to use their sets of instructions on their cards.

Teams should be paired, at random, with another team, and they should play two games, so that each team gets a chance to go first. They can also use the cards to mark their spots on the game board, or the teacher can give the students X’s and O’s. It is important that each team plays several games against unique opponents, so that the data that is generated during the game play is more accurate. In a perfect world, every team will get a chance to play every other team twice, once as Player 1 and once as Player 2. Each time they play a game, the teams should record the outcome on their Record Tally sheet. Since the teacher has copies of each team’s “program”, it will be easy to investigate any accusations of cheating. At the end of this lesson, the teacher will need to collect all of the Record Tally sheets and make enough copies so that to start off the next lesson, every team will have a copy of every other team’s Record Tally sheet.

Lesson 4 – Analyzing the Data (1 to 2 days)

Once students have played their games, it is time to analyze the data collected to decipher which robot was the best, and, in turn, what the best first move is for Player 1 in a game of Tic-Tac-Toe. Tell the students to pretend that they have been hired as a data analyst to dig through some data to determine the best first move in a game of Tic-Tac-Toe. To generate interest on what a data analyst does, the teacher can show a quick video on what a data analyst does, like this one: https://www.youtube.com/watch?v=jLmkhH_pbeE.

Hopefully, ***someone*** in the class has convinced their team that the best first move in Tic-Tac-Toe is ***not*** to go in the center, but rather to start in a random corner. If not, as mentioned previously, it is up to the teacher to make sure that at least one team is playing optimally, as seen below. As mentioned earlier, if no team seems to be playing with an optimal set of instructions, the teacher can choose a team to "coach", or just enter his or her own set of instructions. Either way, it is essential that one robot wins at least of couple of games. Obviously, most games will end in a tie.

Optimal game play for Player 1:

Move 1 – 1. Go in a free corner 2. Go in the center

Move 2 – 1. Go in the opposite corner to my last corner move 2. Go in a free corner

Move 3 – 1. Go for 3 in a row 2. Block 3 in a row 3. Go in a free corner 4. Go in a free space.

Move 4 – 1. Go for 3 in a row 2. Block 3 in a row 3. Go in a free corner 4. Go in a free space

Optimal game play for Player 2:

Move 1 – 1. Go in the center 2. Go in a free corner

Move 2 – 1. Block 3 in a row 2. If the other player holds opposite corners, go in a side 3. Go in the opposite corner of the other player's last corner move 4. Go in a free corner

Move 3 – 1. Go for 3 in a row 2. Block 3 in a row 3. Go in a free corner 4. Go in a free space

Move 4 – 1. Go for 3 in a row 2. Block 2 in a row 3. Go in a free corner 4. Go in a free space

Give each team copies of all of the Record Tally sheets from the previous lesson. As they dig through the data, they should collectively analyze which team's set of instructions were the best, using any criteria that they, as a team, desire. V After each team has analyzed their data and picked a winning robot, they should each be given a turn to informally announce whose robot they thought did the best job, and why. The students, at this point, should realize that the best sets of instructions started with Player 1 starting the game by going in the corner, and not in the center. It is a commonly held belief that to win at Tic-Tac-Toe, Player 1 should start in the center square, and it is this belief that we can challenge, with the use of our AI and our data analysis.

As a conclusion of the unit, the teacher should pose one simple, direct writing prompt to the students, which is: "*What is the best first move in Tic-Tac-Toe? How do you know?*" Although it is a seemingly simple, straight forward question, the teacher should expect a variety of reasons why the corner is the best first move. Some students may have analyzed that move enough to realize that it gives Player 1 more flexibility to eventually win the game. Others may simply say that since the best "bots" in the class chose the corner as the first move, then it must be the best first move, and I think this is a more than acceptable answer.

Figure 3

Record Tally

Team Name: ____

Team Members: ____

AS PLAYER 1:

The first choice of our first move was: ______________________________________

The first choice of our second move was: _____________________________________

WINS	**TIES**	**LOSSES**

Appendix 1 – Implementing Teaching Standards

As stated in the rationale, this unit focuses more on the 8 mathematical practices as described in the Common Core State Standards, more than it focuses on specific 6th and 7th grade standards. It is for this reason that a great time to do this unit with your students would be after any standardized testing is finished for the year. Teachers often feel burned out with “test prep” this time of year, and so this unit steps away from detailed standards, and focuses on the bigger picture of critical thinking, analyzing, and teamwork.

Since the students have to collect and analyze the data from all of the games being played, they will also be learning about the 7th grade CCSS 7.SP.C.8.C: *Design and use a simulation to generate frequencies for compound events.*

Notes

1. *Piedmont Open Middle Details*. (n.d.). Accessed September 22, 2014, from Greatschools.org: http://www.greatschools.org/north-carolina/charlotte/1278-Piedmont-Open-Middle-School/details/#Students

2. Turing, A.M., "Computing Machinery and Intelligence," *Mind* 59 (1950): 433-460.

3. *Home Page of The Loebner Prize in Artificial Intelligence.* (n.d.). Accessed November 12, 2014 from Loebner.net: http://www.loebner.net/Prizef/loebner-prize.html

4. *Answer Sheet*. (n.d.). Accessed October 22, 2014, from Washingtonpost.com: http://www.washingtonpost.com/blogs/answer-sheet/wp/2014/10/24/teacher-spends-two-days-as-a-student-and-is-shocked-at-what-she-learned/

5. *Math Practice*. (n.d.). AccessedSeptember 11, 2014, from Corestandards.org: http://www.corestandards.org/Math/Practice/

6. Richard C. Overbaugh, L. S. (n.d.). *Bloom's Taxonomy*. Accessed October 6, 2014, from Old Dominion University: http://ww2.odu.edu/educ/roverbau/Bloom/blooms_taxonomy.htm

7. Vega, V. (n.d.). *Project-Based Learning Research Review*. Accessed from Edutopia.org: http://www.edutopia.org/pbl-research-learning-outcomes

8. Kay, A. (n.d.). *Kay*. Accessed November 1, 2014, from USAToday.com: http://usatoday30.usatoday.com/money/jobcenter/workplace/kay/2011-05-23-employers-value-creativity-teamwork_N.htm

9. Zaslavsky, Claudia. *Tic Tac Toe: And Other Three-In-A Row Games from Ancient Egypt to the Modern Computer*. New York: Crowell, 1982

List of Materials for Classroom Use

1. Numbered Tic-Tac-Toe boards (Figure 1)
2. Programming Cards for Player1 and Player 2 (Figure 2)
3. X's and O's for the students to practice with. (Not included)
4. Records Tally Sheet (Figure 3)
5. Index Cards

Reading List for Students

Definition of Artificial Intelligence. (n.d.). Retrieved October 10, 2014, from Techterms.com: http://www.techterms.com/definition/artificial_intelligence. This is a very basic definition of Artificial Intelligence. It could be read together as a class at the beginning of the unit.

Tic-Tac-Toe. (n.d.). Retrieved August 20, 2014, from Wikipedia.org: http://en.wikipedia.org/wiki/Tic-tac-toe. This entry on Wikipedia gives everything one would need to know about Tic-Tac-Toe. It includes the history of the game, strategy, variations of the game, and an explanation of how many outcomes are possible in any given game.

Bibliography for Teachers

Beck, J. *Combinatorial Games : Tic-Tac-Toe Theory*. Cambridge: Cambridge University Press, 2008. Although a bit "heavy", this book does a great job delving into the world of game theory, and how figuring out all possible combinations of outcomes is key to creating good game theory.

Maggi Savin-Baden, C. H. *Foundations of Problem-Based Learning.* Maidenhead: Society for Research into Higher Education & Open University Press, 2004. This book does an excellent job of explaining why doing problem-based tasks is important in the classroom. The book also explains how to sucessfully manage lessons that involve problem-based learning.

S. Orr, C. C., "A Compact Strategy for Numerical Tic-Tac-Toe," *Journal of Recreational Mathematics.* (1995): 161-171. This article explains exactly why going in the center is not the best move in Tic-Tac-Toe. It also explains how to react when your opponent makes each move.

Shi, Z. *Advanced Artificial Intelligence.* Hackensack, NJ: World Scientific, 2011. Although a bit long (632 pages), this book provides an extremely detailed explanation of what advanced artificial intelligence really is. It reads like a textbook, so it is easy to reference what you are looking for, without having to read the entire book.