



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Manasa H. Bhimaraju</b>	<b>Project Number</b> <b>J1401</b>
<b>Project Title</b> <b>Low Cost Animated Teaching Tool for Study of Elements of the Periodic Table with an Interface for the Visually Impaired</b>	
<div><b>Objectives/Goals</b> Understanding elements and their atomic structure is a very interesting and important study. But, the logic involved is not straightforward. My goal is to come up with a solution that can help in teaching these complex concepts with visual effects that are easy to absorb. My goal is also to help visually impaired by adding sound. Finally I want my solution to be portable, extensible and low-cost and accessible to students with limited access to expensive resources or the Internet.</div> <div><b>Abstract</b> Materials: Raspberry Pi/Arduino, 2 Breadboards, 6V DC Motor, 8 LEDs, 8 resistors, Braille stickers. Programming: Elements, Exception Elements, Electron Configuration, Speech, Drawing the atom, Lighting up LEDs and beeping for valence electrons, Turning spinner/beeping for radioactive elements. Lewis Dot Notation: LEDs arranged in Lewis Dot Notation circuit on breadboard. Geiger Counter Simulation: Motor Circuit to simulate radioactivity level.</div> <div><b>Methods/Materials</b> Materials: Raspberry Pi/Arduino, 2 Breadboards, 6V DC Motor, 8 LEDs, 8 resistors, Braille stickers. Programming: Elements, Exception Elements, Electron Configuration, Speech, Drawing the atom, Lighting up LEDs and beeping for valence electrons, Turning spinner/beeping for radioactive elements. Lewis Dot Notation: LEDs arranged in Lewis Dot Notation circuit on breadboard. Geiger Counter Simulation: Motor Circuit to simulate radioactivity level.</div> <div><b>Results</b> The filling order of electrons is 1s2 2s2 2p6.. * Example: Oxygen: Atomic Number 8. Electron config: 1s2 2s2 2p4 with 6 electrons in valence shell. Lewis Dot Notation circuit: 6 LEDs light up (s1, s2, p1-1, p2-1, p3-1, p1-2) with 6 beeps. Exception Elements (20 in all): * The tool mentions #Exception#, so the student is aware * Example: Copper: Atomic Number 29. Based on the filling order, the configuration should be: 1s2 2s2 2p6 3s2 3p6 4s2 3d9. But in reality, an electron is transferred from 4s to the inner 3d, resulting in: 1s2 2s2 2p6 3s2 3p6 4s1 3d10. Lewis Dot Notation: The max of 8 valence electrons: s - 1st (s1), s - 2nd (s2), p1 - 3rd (p1-1), p2 - 4th (p2-1), p3 - 5th (p3-1), p1 - 6th (p1-2), p2 - 7th (p2-2), p3 - 8th (p2-3). Radioactive Elements: The Geiger counter is simulated</div> <div><b>Conclusions/Discussion</b> My tool best serves the goal as: 1. It has both visual/audio capabilities, making learning fun and effective. 2. It is low cost, compact and portable as it uses Raspberry Pi, and a few simple components, and addresses those who do not have access to expensive resources or Internet. 3. It helps the visually impaired by SHOWING them with words and sounds. Also as it is a programming project: 1. It shows it is much easier than a non-programming model to teach complicated concepts. 2. It, but for the breadboards part, can be put in a website and accessed from anywhere and free of cost, if Internet is available.</div>	
<b>Summary Statement</b> Low-cost Animated Teaching Tool for Study of Elements of the Periodic Table with an Interface for the Visually Impaired	
<b>Help Received</b> Perkins.org and Kentucky school of the blind for feedback. My grandfather, Professor P S Sarma for teaching electronic circuits. School Science teacher, Mr Scharmen. Donumvisi.org for teaching me to help the blind. My weekly science and programming classes. Rishabh Bhasin for teaching Arduino.	



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Anika Cheerla</b>	<b>Project Number</b> <b>J1402</b>
<b>Project Title</b> <b>Automated Diagnosis of Alzheimer's Using MRI Images</b>	
<b>Objectives/Goals</b> Over 60% of people with dementia die without ever being diagnosed as dementia patients. The reason for this is the lack of precise tools, and the extent of time and effort it takes to diagnose dementia. The purpose of this project was to create an automated tool that could accurately diagnose Alzheimer's. The tool would use neural networks to train itself to diagnose Alzheimer's. Neural networks model the human brain by finding patterns and making connections between the inputs and the output. This would provide a quick and precise result, and enable early detection of Alzheimer's. My tool would be an improvement from what doctors currently have to do: conduct many series of tests and brain scans, which consumes a lot of time, and come to a conclusion based largely on opinion.	
<b>Abstract</b> <b>Methods/Materials</b> I used the OASIS database of MRI scans and MATLAB to do my project. First, I extracted image features from the scans, such as the fractal dimension and the area of atrophy of the brain. Then, I combined those features with the patient's basic medical data. After that, I tested out combinations of different training algorithms, structures, numbers of hidden neurons, and amounts of data. I compared results from each classifier to find the best classifier. After creating the classifier, I built a GUI using the MATLAB toolkit where doctors would only have to upload an MRI scan and enter the patient's basic information to diagnose the disease.	
<b>Results</b> The neural network classifier with 3 chained stages, Bayesian Regulation Back propagation as a training algorithm, 12 # 16 hidden neurons, and image features combined with the patient's basic information, achieved the highest accuracy. This classifier reached a testing accuracy of 97%, and greater than 99% overall accuracy. When the neural network was not trained with image features, not trained with the MMSE feature, or trained with a different training algorithm, the accuracy reduced significantly. The two other neural network structures I tested, single-stage and hierarchical networks, could not obtain as high an accuracy as the 3 stage chained neural network.	
<b>Conclusions/Discussion</b> I was able to meet all my project goals. The classifier I created achieved a higher diagnostic accuracy than any previous method of Alzheimer's diagnosis. The GUI I built was functional and easy to use. My tool could be used in medical practice to change the current method of diagnosis.	
<b>Summary Statement</b> I created a tool that quickly and precisely diagnoses Alzheimer's using MRI Images and determines the early diagnosis of a patient more accurately than any previous methods.	
<b>Help Received</b> My brother taught me Matlab. My science teacher, Mrs. Iyer, helped me organize my ideas and develop my project.	



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Adlae H. D'Orazio</b>	<b>Project Number</b> <b>J1403</b>
<b>Project Title</b> <b>False Positive and False Negative Paradox</b>	
<b>Abstract</b> <b>Objectives/Goals</b> Testing for diseases sometimes gives false positive or false negative results. Even for tests that are 99% accurate, it was hypothesized that, in the testing of a population, there would be more false positives than true positives and more false negatives than true negatives for rare diseases. The objectives of this project were to see if the hypothesis was correct and to determine the conditions causing false test results to be greater than true test results. <b>Methods/Materials</b> Calculations were done using a population of 100,000 with various disease rates and test accuracies. An Excel spreadsheet was programmed to cover a wide range of possibilities. Experiments were run to simulate the calculations and to check if the calculations worked for a population as low as 200. The experiments involved rolling dice, sometimes 2 and sometimes 3, to match disease rates and test accuracies. Results were tallied and analyzed. <b>Results</b> Calculations showed that there is a greater likelihood of getting more false positives than true positives if the disease is rare and more false negatives than true negatives if the disease is common. The experiments verified the calculations and showed that even a population as low as 200 had a similar result. <b>Conclusions/Discussion</b> The detailed hypothesis, which guessed that greater false tests would occur for rare diseases, was found correct for false positives but incorrect for false negatives (because more false negatives occurred for common diseases). The work from this project adds to the knowledge of doctors and health officials who are concerned about scaring people with false positive results or failing to correctly notify people with false negatives.	
<b>Summary Statement</b> The conditions causing the paradox of more false positive than true positive disease test results and more false negative than true negative disease test results were investigated.	
<b>Help Received</b> My dad checked my math and helped me write the abstract in the proper format.	



# CALIFORNIA STATE SCIENCE FAIR

## 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Elias B. Gilbert</b>	<b>Project Number</b> <b>J1404</b>
<b>Project Title</b> <b>Watch the Curve: Using Statistical Modeling to Predict the Next Baseball Pitch</b>	
<div><b>Objectives/Goals</b> The objective was to create computer software that would use a statistical model to accurately predict the next pitch to be thrown in a baseball game.</div> <div><b>Methods/Materials</b> I used data from Tim Lincecum and the statistical programming language R to create 9 models that each took different approaches to guessing pitches. The first guessed randomly among the pitches. The second always guessed the most common pitch thrown. The third guessed based on the proportions that each pitch was thrown. The 4th and 5th were the same as the previous two but they guessed based on the pitch before instead of the whole game. The 6th and 7th were the same as the 4th and 5th but guessed based on the pitch count. The 8th and 9th models guessed based on both the pitch before and the pitch count. I used a train/test split using data from one game to predict pitches in a different game. Then, I used different numbers of training games to improve the predictions.</div> <div><b>Results</b> The first, and most basic model got the pitch correct 20% of the time. The best performing model, model 8, got the pitch correct 32% of the time with a 1/1 train/test split. Then, when I used data from 3 games to guess the pitches in another game (a 3/1 train-test split), the model got the pitch correct 37% of the time.</div> <div><b>Conclusions/Discussion</b> Several things were learned from this project. First, games have predictability. My model was able to get the pitch correct 37% of the time. Also, I found that both the pitch count and the pitch before affect the next pitch to be thrown. Third, using a larger training dataset gives you a more accurate prediction of the next pitch. This model could continue to be made better by adding more factors, and a bigger training dataset.</div>	
<b>Summary Statement</b> I used statistical modeling to use previous pitches to predict the next kind of pitch that a pitcher will throw in a baseball game.	
<b>Help Received</b> My dad, Greg Gilbert, gave me helpful discussions and for teaching me coding when I did not know how to do something. My mom, Ingrid Parker, helped me lay out my poster and edited my text.	



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Paul C. Grosen</b>	<b>Project Number</b> <b>J1405</b>
<b>Project Title</b> <b>A Computer Program to Track and Count Particle Trails</b>	
<div><div><b>Objectives/Goals</b> The purpose of this project was to develop a fast and accurate computer program to track and count the trails of charged particles in a cloud chamber video.</div><div><b>Methods/Materials</b> Various cloud chambers were constructed using a container, isopropyl alcohol, dry ice, and a polonium-210 source. Videos were recorded of the resulting alpha particle trails in the alcohol vapor. Because the quality of these recordings was insufficient for processing, four videos were obtained from UC Santa Barbara's Senior Physics Lab. Next, an algorithm was developed to track and count moving and dissipating vapor trails in digital recordings. It was then implemented using OpenCV's Python package. Options were added to change the configuration of the counter to optimize for various alcohol vapor conditions within the cloud chamber. The program was then run on each of the four videos multiple times and the counts and times were recorded. In addition, multiple manual counts were performed on each video for comparison. Finally, "aided" manual counts were conducted to count the number of trails the program missed and the number that were counted more than once.</div><div><b>Results</b> The algorithm was able to correctly identify trails a large percentage of the time and tally a reasonable count. The program's count was, on average, within 0.7% of the best obtainable ("aided") count, whereas manually, on average, only 59% of the trails were counted, even with the video slowed to 3 FPS. However, on average, 14.1% of the trails counted by the program were duplicates and 14.8% were missed. The manual counts took, on average, about four times as long to complete as the program count.</div><div><b>Conclusions/Discussion</b> The computer program is substantially faster and more accurate than manual tallying. The low manual counts (which recorded less than 60% of the actual trails) were unexpected. With repeated manual attempts, it became clear that lowering the frame rate below 3 FPS resulted in a loss of ability to distinguish between old and new trails; consequently, more accurate manual counts were not achievable. When viewing a video that had been processed by the program that had rectangles highlighting the identified trails, it was clear that the computer program was able to identify more trails per frame than a human doing a manual count. The speed with which the program counted the trails was unsurprising.</div></div>	
<b>Summary Statement</b> In this project, a computerized image processing system was developed to reliably track and count charged particle trails in cloud chamber videos.	
<b>Help Received</b> Obtained cloud chamber videos from Dr. Deborah Fygenson at UCSB; Received advice on cloud chamber construction from Dr. Wing Howard To; General guidance from science teacher Ms. Kim Miller; General introduction to OpenCV from father, Dr. Mark Grosen; Help gluing board from mother	



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Jakob A. Hachigian-Kreutzer</b>	<b>Project Number</b> <b>J1406</b>
<b>Project Title</b> <b>Exploration of the Ulam Spiral</b>	
<div><div><b>Objectives/Goals</b> The Ulam Spiral is a way of visually displaying the prime numbers which shows some of the sequence relationships which exist across the whole set of natural numbers. My project focused on whether the patterns observed in the original Ulam Spiral would be seen in variant forms and most especially when the spiral became much larger than when it is typically displayed on a single page.</div><div><b>Methods/Materials</b> Windows PC loaded with a version of the Python Programming language which has the matplotlib package installed.</div><div><b>Results</b> My discovery was that patterns seen early within small versions of the Ulam spiral continue to exist even when the Ulam Spiral gets to be very large. Spiral sizes in excess of five million required hours for my computer to produce but could then be sampled to see the patterns continue to exist. These large spirals could then be searched by computer to find the areas of the spiral where the patterns were the longest and most frequently seen.</div><div><b>Conclusions/Discussion</b> My project showed that Ulam's discovery of the prime number spiral could be extended to a very large set and could possibly be mined to provide pattern areas which might be generalized to predict the existence of large prime numbers.</div></div>	
<b>Summary Statement</b> My project visually and computationally explores very large Ulam Prime number spirals to examine the consistency of the patterns seen when originally discovered.	
<b>Help Received</b> My prior year's school Scratch Programming instructor, Mr Lee Appelbaum introduced me to programming the Ulam Spiral in Scratch. I wrote and coded the Python programs used for this project but was assisted in the 2 lines of code which actually call the matplotlib scatter routine.	



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Jacob Kantor</b>	<b>Project Number</b> <b>J1407</b>
<b>Project Title</b> <b>Phi: The Golden Ratio</b>	
<b>Objectives/Goals</b> When creating a Fibonacci-like sequence of 100 numbers and then creating a ratio of the last number to the next-to-last number, which two starting numbers will produce a ratio that is closest to Phi, the Golden Ratio? My Hypothesis: The sequence whose two starting numbers have the closest starting ratio to Phi will end up closest to Phi by the end of the sequence because all Fibonacci-like sequences will approach Phi, and this one has a head start. In this experiment testing 144 pairs of numbers, that will be the sequence starting with 5 and 8, the pair whose starting ratio is closest to Phi.	
<b>Abstract</b> <b>Methods/Materials</b> 1. Create a spreadsheet using Microsoft Excel & the Arbitrary Precision Plug In (so that Excel does not cut off each number at 15 decimal places). 2. Select 144 pairs of numbers to test as follows: Every single combination of whole numbers between 1 and 12. (12X12=144 sequences). 3. Create a column in Excel that lists the Fibonacci-like sequence for each of the starting pairs until each sequence is 100 numbers long. (Formula = add the previous two numbers to get the next number in the sequence). 4. Find the ratio of the 100th number to the 99th number for each of the sequences. 5. Compare all of these final ratios to Phi to see which one is the closest. 6. Compare all of the starting ratios (2nd number divided by the 1st number) to see if the hypothesis is correct.	
<b>Results</b> I found that the sequence with starting numbers (1,12) had a final ratio ( $n_{100}/n_{99}$ ) that was closest to Phi. My hypothesis (that the pair of numbers with the closest starting ratio to Phi would be closest at the final ratio) was wrong. In fact, the ratio that was closest to Phi at ( $n_{100}/n_{99}$ ) had a starting ratio that was furthest from Phi. Although in my experiment, I focused on comparing the differences between the ratios and Phi, one of the most surprising results is that no matter the what the starting numbers were, all 144 of my sequences ended up within one-quadrillionth of Phi!	
<b>Conclusions/Discussion</b> Upon further analysis, I found that the pair of numbers with the highest starting ratio had the final ratio that was closest to Phi. By trying a few other higher number pairs, I confirmed that the higher the starting	
<b>Summary Statement</b> In my project, I investigated Phi, the Golden Ratio and the Fibonacci Series and built a model to prove which two starting numbers of a series produce a ratio closest to Phi.	
<b>Help Received</b> My dad helped locate and install the Arbitrary Precision Plug-In for Excel and learn how to write the formulas using special syntax so that the numbers would not be truncated at 15 decimals points. My mom helped me format and print the data from Excel.	





**CALIFORNIA STATE SCIENCE FAIR  
2015 PROJECT SUMMARY**

<b>Name(s)</b> <b>Rachana Madhukara</b>	<b>Project Number</b> <b>J1408</b>
<b>Project Title</b> <b>Devising an Effective Way of Solving the Rubiks Cube</b>	
<div><div><b>Objectives/Goals</b> The purpose of my project is to devise a method of solving the Rubiks Cube that is not only simple to learn, but also solves the Rubiks Cube fairly quickly.</div><div><b>Methods/Materials</b> Paper, pencil, computer with an Internet connection and a 3x3 Rubiks cube were used. From my research online, I found out that the Fridrich Method was an effective method to solve the Rubiks cube. However, I realized that the second and third steps were way too hard to memorize and use. Hence, I developed new steps to replace them. By stitching all these steps together, I arrived at an optimal algorithmic sequence. This sequence is way easier to learn and memorize.</div><div><b>Results</b> The number of moves it takes to solve the Rubiks cube using my method is about 25 percent less compared to the Beginners Method.</div><div><b>Conclusions/Discussion</b> I conclude that if the number of operations to solve the Rubiks Cube are reduced, then the Rubiks Cube can be solved quickly and effectively.</div></div>	
<b>Summary Statement</b> Devising an easy to learn and memorize algorithm to solve the Rubiks Cube.	
<b>Help Received</b> My parents helped me prepare the display board.	





**CALIFORNIA STATE SCIENCE FAIR  
2015 PROJECT SUMMARY**

<b>Name(s)</b> <b>Alexander T. McDowell</b>	<b>Project Number</b> <b>J1409</b>
<b>Project Title</b> <b>Can Wind Chimes Generate Truly Random Numbers?</b>	
<div><div><b>Objectives/Goals</b> The sound of wind chimes can be used as a seed to generate truly random numbers.</div><div><b>Methods/Materials</b> A wind chime for recording sound, a fan to control wind speed, a recording device (iPhone), a program that can convert audio to numbers (Audacity), and the Java programming language. I put my iPhone next to the wind chimes to record the sound, and used a fan to control wind speeds. I recorded 60 samples of data, all around 20 seconds long with 4 different tests. First I rang the chimes manually, then I used the fan's 3 wind speeds. Next I ran the files through Audacity to take the Spectrum (or Fast Fourier Transform) of the sound file. These files each consisted of roughly 256 numbers. Then I ran the numbers through a program I created to seed a random number generator. Finally I analyzed the numbers through a program that I created to determine if the sequence were random by using the Chi Square Test.</div><div><b>Results</b> After going through all 60 files, the Chi Square Test returned that each of the sequences were random, and I compared these results to a control group that consisted of a sequence of truly random numbers and a non-random sequence (Fibonacci).</div><div><b>Conclusions/Discussion</b> I concluded that the random numbers created by the wind chimes were truly random.</div></div>	
<b>Summary Statement</b> Wind chimes can be used as a seed for a Truly Random Number generator.	
<b>Help Received</b>	



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Sebastian L. Mellen</b>	<b>Project Number</b> <b>J1410</b>
<b>Project Title</b> <b>MathSuite: A Free and Innovative Android App that Makes Dreaded Calculation Fun! Published and Available on Google Play</b>	
<div><b>Objectives/Goals</b> I set out to build a user-friendly suite of math calculators that can be used offline, &amp; downloaded at no cost from the Google Play Store. I wanted this app to help students &amp; teachers check their math &amp; physics work, by instantly solving commonly used mathematical equations.</div> <div><b>Methods/Materials</b> I have taught myself to code apps in the past; I intended to build on that experience to figure out how to approach this more ambitious project, with an eye to measuring &amp; minimizing the use of the user's Android device RAM. I started with MIT's AppInventor, experimenting with ways to build the code, and which functions to include. I created 28 different app builds of the app, some successful, some not, with each build including something new - a new line of code or a new function button. By the 17th build I had the basic code finished. Then I went on to edit the user interface and layout until I was satisfied with the functionality &amp; features of the app with the 28th build. Then I edited the underlying code in Eclipse &amp; Apk Tool, edited the Android Manifest XML, "signed" the app so as to be able to register it with Google, &amp; published it to the Google Play Store. At each stage I measured the efficiency of RAM use. As I created the app, I studied what calculators I thought would be useful for Junior &amp; High School students &amp; teachers, adding calculators &amp; functions accordingly</div> <div><b>Results</b> The app was completed &amp; published meeting all of my original design criteria, working on Android versions from 1.5 to the present version. In the end I developed the following components of what I call MathSuite: Calculators: Scientific (20-function), Pythagorean, Right Angles, Quadratic Equations, Midpoint Formulas, Distance Formulas, Degrees-to- Radians &amp; Radians-to-Degrees, + a Unit Circle Diagram, &amp; integrated Web browser directed to my school's Web site for student access. A number of people have now downloaded it, most from scanning the QR Code posted on my science project display board. I continue to tweak the design as a result of the feedback from those users.</div> <div><b>Conclusions/Discussion</b> Before the completion of MathSuite, no such app existed for Android devices. This app does not require an Internet connection once it is downloaded. It is visually pleasing, intuitive, fast &amp; accurate. It is a complete scientific &amp; 8-function equation calculator. I plan to continue adding equation functions, as requested by users.</div>	
<b>Summary Statement</b> I build MathSuite, a complete Scientific & multi-equation calculator not previously available to Android users, as a functional tool for teachers & students, to be used offline.	
<b>Help Received</b> My parents helped print & organize materials for board & binder. Otherwise, I was left to my own devices, as neither my parents nor my teacher are computer geeks. I am grateful for the free information on the WWW, & to the Google Groups Appinventor page & to Eclipse & MIT AppInventor for being	



**CALIFORNIA STATE SCIENCE FAIR  
2015 PROJECT SUMMARY**

<b>Name(s)</b> <b>Margaret A. Mendez</b>	<b>Project Number</b> <b>J1411</b>
<b>Project Title</b> <b>Elevator Modification in Virtual Reality with the Use of Logic Gates and Adders</b>	
<div><div><b>Objectives/Goals</b> My objective was to create a Minecraft mod that creates an instant elevator with the help of logic gates and adders.</div><div><b>Methods/Materials</b> Materials: PC; Full Version of Minecraft; Binary; Logic Gates; Adders. Methods: I used the engineering process. I also used methods such as systems integration, trial and error, and build and test.</div><div><b>Results</b> I successfully created a compact elevator that is perfect for modding.</div><div><b>Conclusions/Discussion</b> Because of my project I have more knowledge on computers than I thought I would receive. I found that logic gates and adders can be used in real life to create switches, and build the components of all electronics.  Next year I plan to learn the modification process so that I can create the actual mod.</div></div>	
<b>Summary Statement</b> I successfully created a compact elevator in Minecraft with logic gates and adders that will soon be coded into a modification program.	
<b>Help Received</b> Mentor: George Morgan; Support: Mom and Dad	



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Jaydah M. Mills</b>	<b>Project Number</b> <b>J1412</b>
<b>Project Title</b> <b>Blind Gaming</b>	
<div><b>Objectives/Goals</b> Can I make a video game that blind and sighted players can enjoy?</div> <div><b>Methods/Materials</b> Computer Version of GameMaker: Studio 25 blindfolded test subjects and 25 subjects not blindfolded</div> <div><b>Results</b> My question was "Can I make a video game that blind and sighted players can enjoy?" By the research I have collected so far, I am now certain that I can make a video game of that nature. The graph shows that a high amount of tested players, both blind and sighted, enjoyed the game.</div> <div><b>Conclusions/Discussion</b> I proved that my hypothesis was correct and found out that creating a video game takes lots of time and effort. I learned that sometimes even the smallest mistakes can crash a game. I almost broke my whole game just because I added an extra comma. Coding is very detailed. Just to add random rocks falling, the coding would look like this: "random_instance(20, room_length-20),-20, OBJ_ROCK;" Because video game design is so detailed, it takes a lot of time. Even simple looking games can be more complicated than they look. One thing I loved about the project was that I was able to learn programming. When I grow older, I want to create video games. This project has helped me learn the basics of coding and will help me later on. I also loved how I was able to create a video game at such a young age. Not many kids my age can say they made a video game for the blind. I also would like to be able to have more time to draw the game art. A lot of my art was drawn before I installed pressure sensitivity to my drawing tablet. I would love to draw colorful art and make it look nice. My project helped humanity by giving the blind something fun to do and future aspects of the game could help to develop skills. Overall, I loved the project and wish that I could do it again because it was so much fun.</div>	
<b>Summary Statement</b> Developing a computer game that is fun and educational for the blind.	
<b>Help Received</b> I went to various discussion forums and looked for information on programming.	



**CALIFORNIA STATE SCIENCE FAIR  
2015 PROJECT SUMMARY**

<b>Name(s)</b> <b>Ignacio A. Morales</b>	<b>Project Number</b> <b>J1413</b>
<b>Project Title</b> <b>Can a Lego Robot Guard Your Treasure?</b>	
<div><div><b>Objectives/Goals</b> My objectives during this experiment were to learn to program a Lego robot and program it to patrol around an area and defend it against intruders by executing a series of actions.</div><div><b>Methods/Materials</b> First, you must build a Lego robot from a Lego EV3 Mindstorms set. Second, you must learn to program your robot using a computer. Then, you must program your robot to protect an area by using different command "blocks". After that, test your robot to confirm that it can detect intruders that remain still and execute the programmed commands. Lastly, start robot to patrol the protected area and execute 16 tests.</div><div><b>Results</b> From my research and testing, I discovered that a Lego robot cannot successfully guard your treasure.</div><div><b>Conclusions/Discussion</b> I learned that Lego robots would not be effective guarding treasure from intruders that move inside the protected area at higher speeds. Although I could not prove my hypothesis, I did learn many things while performing this project.</div></div>	
<b>Summary Statement</b> My project is about programming a Lego robot to patrol an area and execute a series of actions to defend the area.	
<b>Help Received</b> None	



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Liana C. Mueller</b>	<b>Project Number</b> <b>J1414</b>
<b>Project Title</b> <b>Eye Cue: Creating a Computer Game for the Blind</b>	
<div><div><b>Objectives/Goals</b> The purpose of my project is to design a computer game that can be played successfully by sighted and blind players. My hypothesis is, if the correct audio cues are used then visually impaired and sighted people can play the same game successfully.</div><div><b>Methods/Materials</b> The computer program GameMaker was used to create the game on a PC, with the program WaveEditor for sound editing and the program Paint to create the sprites. The game is a maze that simulates a city street. During the design and making of the game multiple game objects were created, tested and rejected. The finished game contains four levels of increasing difficulty and one bonus round. The game was tested by sighted and blind players. All players tested the game with the computer display covered and sighted players also tested with the computer display uncovered.</div><div><b>Results</b> The results show that 50% of all sighted players, playing with the computer display covered, completed all four levels of the game. A total of 43% of all blind players completed all four levels of the game. The average time to complete all four levels when the display was covered was 11 minutes.</div><div><b>Conclusions/Discussion</b> My hypothesis was supported because I was able to create a computer game that could be played successfully by sighted and blind players. Some players were able to complete all four levels of the game and some were not. Even though the times to complete the game varied I think all of the tests were successful.</div></div>	
<b>Summary Statement</b> The purpose of my project was to create a computer game that could be played successfully by sighted and blind players.	
<b>Help Received</b> My dad helped me learn how to calculate and store the game time, scores and how to make a timer for the game. He helped me learn how to destroy a specific object when there are multiple objects. He helped me with the traffic light logic and fixing some bugs in the code.	



**CALIFORNIA STATE SCIENCE FAIR  
2015 PROJECT SUMMARY**

<b>Name(s)</b> <b>Kaylen Ng</b>	<b>Project Number</b> <b>J1415</b>
<b>Project Title</b> <b>Experimental Theory of Using Bubbles as a Sustainable Source of Air Underwater</b>	
<b>Abstract</b> <b>Objectives/Goals</b> The purpose of this experiment is to find how much air is in a water bubble and if this is enough to supply a human. The hypothesis is if size and pressure of bubbles at 0.3 to 3.6 meters are predicted to assume the bubble's volume of air, then the results will show that the amount of air in bubbles will be enough to sustain a human being because of the amount of dissolved air in water. This study is important because current ocean exploration and recreational snorkeling is limited to the amount of air a human can hold. <b>Methods/Materials</b> Finding bubble diameter at 0.3 and 0.6 meters, predictions were made for each 0.3-meter progressive change up to 3.6 meters. Taking those numbers, the formulas for surface tension and volume were used to find the liters of air within each bubble and how many bubbles were needed per breath at each depth. <b>Results</b> With each depth, bubble diameter dropped 39.5%. At 0.3 and 0.6 meters, air sustainability with bubbles was most sufficient, needing only 21 and 30 bubbles respectively <b>Conclusions/Discussion</b> Through measurement of bubbles from 0.3 to 3.6 meters, it was determined that bubbles are a sustainable source of air for humans underwater.	
<b>Summary Statement</b> Calculating the number of bubbles needed per breath at different depths underwater to sustain a human.	
<b>Help Received</b>	





**CALIFORNIA STATE SCIENCE FAIR  
2015 PROJECT SUMMARY**

<b>Name(s)</b> <b>William M. Pramanik</b>	<b>Project Number</b> <b>J1416</b>
<b>Project Title</b> <b>Stock Market + Risk Management = College Paid</b>	
<div><div><b>Objectives/Goals</b> Within the financial markets, I am comparing momentum indicators with oscillating indicators. Then I created programs with multiple time frames to try to improve their results.</div><div><b>Methods/Materials</b> 1. Tradestation 2. Computer 3. Babypips.com  I picked three indices and six indicators for the daily charts. After that I optimized the exit strategy with the 6 indicators from 2000 - 2005. Then I used those values from the optimization in charts from 2005 - 2015. Then I compared those results. Lastly I wrote programs applying multiple time frames trying to improve the previous results.</div><div><b>Results</b> On the Volatility Index, the Oscillating indicators did better, but on the trending markets such as the Bond Barkly Index and the NASDAQ, the Momentum Indicators outperformed the Oscillators. Also, when I applied the Multiple time frames, the programs did worse because the multiple time frames prevented the shorter term indicators from getting early entries.</div><div><b>Conclusions/Discussion</b> In conclusion I am very surprised that the indicators with the multiple time frames didn't do as well as the single time frame indicators. Also I would conclude from from these results the Momentum Indicators are better during the trending markets but the Oscillators do better during the non-trending markets.</div></div>	
<b>Summary Statement</b> I am comparing stock market indicators and trying to improve the results with multiple time frames.	
<b>Help Received</b> None	



# CALIFORNIA STATE SCIENCE FAIR

## 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Rashi Ranjan</b>	<b>Project Number</b> <b>J1417</b>
<b>Project Title</b> <b>Infant Car Alert System</b>	
<b>Objectives/Goals</b> Since 1990, over 747 children (on average 31 per year) have died from heat stroke when left alone in a vehicle. This is an alarming situation, so I decided to solve this problem, by designing a low-cost programmable device called iCAS. The objective of this device is to save infants when left in the car by tracking their presence and alerting via text message and alarm once the temperature exceeds the limit.	
<b>Abstract</b> <b>Methods/Materials</b> I used an Arduino development board attached with four sensors and a switch. When the baby is placed in the car seat, iCAS is activated due to a switch. The temperature sensor senses the ambient temperature inside the car and my code checks if it is above the threshold (82 degrees Fahrenheit) set. When the temperature is out of range, the Doppler sensor, which uses microwave frequency signals to detect a moving object, checks if a living body is in the car seat. If conditions are not met, the program goes back to checking for the temperature. As soon as both conditions are met, the alarm is sounded and a text message is sent, utilizing the GSM Cellular Shield, to all emergency contacts.	
<b>Results</b> My device was tested in a hot car with a model baby. (I shook my hand to activate the Doppler Sensor.) My computer was connected to the device to capture the output of all the sensors. All 5 trials showed that as soon as the temperature was out of range and motion was detected, the text message was sent and the alarm sounded. An external thermometer was also placed in the car to make sure the sensor was reliably sensing the ambient temperature.	
<b>Conclusions/Discussion</b> My final solution satisfies all the objectives. iCAS is \$48, a price that fits the budget of an average parent and a price any parent should be willing to pay for the safety of their child. The device is small and can be attached to any car seat without expert help. As of now, phone numbers must be hardcoded in the sketch, but in the future, I will make it easy for users to update the phone number using a website or app.	
<b>Summary Statement</b> The Infant Car Alert System is an inexpensive, programmable, reliable device to prevent child deaths due to heat stroke in vehicles.	
<b>Help Received</b> Father helped in making GSM Shield send text message; Teacher provided mentoring and guidance	



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Sanjana V. Shah</b>	<b>Project Number</b> <b>J1418</b>
<b>Project Title</b> <b>Text Compression Algorithm Based on Popularity Using Global English Dictionary</b>	
<b>Objectives/Goals</b> The objective of this project is to invent a new algorithm that encodes texts found in documents, books, and novels, using popularity of words selected from English dictionary. The goal of this project is to demonstrate that the new algorithm can create a compressed file, whose size is smaller than the one produced by standard compression software such as gzip and bzip2.	
<b>Abstract</b> <b>Methods/Materials</b> Compression ratios highly depend on extent to which the redundancies are detected and the ways in which they are encoded in the output. Since traditional schemes such as Huffman encoding and data dictionaries use the frequency of words that are found only within the document, storing these dictionary values in the output file increases the compression overhead. In contrast, this project uses a word list of 86,600 most popular English words, and encodes every word in the document with its popularity value. Higher the popularity, smaller is the encoded value. These popular words are stored in a global dictionary, accessible by any document that needs to be encoded or decoded. Since this dictionary resides outside the final document, it not only reduces the file size significantly, but also allows the dictionary to be more adaptable to adding/removing/rearranging popular words over time. The version number of the dictionary is stored in the header of the encoded file. The encoded file is then compressed using gzip and bzip2. When decoding, the output file is compared byte by byte to the original file to guarantee that there was no data loss.	
<b>Results</b> This algorithm reduces file sizes of books and novels by 20-30% during the encoding phase. When compressed, it further reduces the file size by 8% over the file size produced by gzip. Statistics show that about 75% of the words got encoded with popularity codes. With specialized domain specific dictionaries, the compressed file sizes of medical transcripts got reduced by 15% over gzip.	
<b>Conclusions/Discussion</b> The major contribution of this work is the use of popularity word dictionary and its adaptability to rapidly changing popularity of English words. In addition to its use in improving the compression ratios, this algorithm allows storing the dictionary globally, thereby sharable by all users for any type of documents. The novel approach in this project is to combine the two areas - the need to compress a file and the availability of popular words in English literature.	
<b>Summary Statement</b> The purpose of this project was to develop a new text compression algorithm that uses popularity word list, and to demonstrate that the compression ratios are better than standard gzip software.	
<b>Help Received</b> My math teacher, Mr. Chang, guided me during the project. Local city library helped me learn Java.	



# CALIFORNIA STATE SCIENCE FAIR 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Andrew M. Shaheen</b>	<b>Project Number</b> <b>J1419</b>
<b>Project Title</b> <b>The Geometry of Banking a Basket</b>	
<div><div><b>Objectives/Goals</b> Have you ever wondered where the best spot to shoot a bank-shot on a Basketball court is? In my experiment, I hypothesized that if I used Algebra and Geometry to find the relative probability of making a bank-shot from different positions on a basketball court and then make a scale model of a basket and backboard to find the relative probability to physically test the bank-shot, then I would be able to predict the best position on the basketball court for a real player to make a bank-shot.</div><div><b>Abstract</b></div><div><b>Methods/Materials</b> I first used math and geometry to find different lengths on the court such as backboard length and length from the backboard to the player the apostrophe s position. I then used these lengths to find the relative probability from the 30, 45, 60, and 90 degree positions. I then built a scale model using poster boards for the court, a wrapping paper tube for a ramp and a miniature basketball and tested it from the 0, 30, 60, and 90 degree positions. The data was used to find the relative probability from each position.</div><div><b>Results</b> For my calculations, the highest relative probability was at the 90 degree position. For my scale model, the 30 degree position had the highest relative probability of 3.03.</div><div><b>Conclusions/Discussion</b> I was able to find the relative probability of making a bank-shot with Math and Geometry and by using a scale model. This project can help basketball players to know where the best position to shoot a bank-shot on the court is.</div></div>	
<b>Summary Statement</b> My project is about using geometry to find the best position on a basketball court to shoot a bank-shot.	
<b>Help Received</b> My dad taught me to use excel spreadsheets	



# CALIFORNIA STATE SCIENCE FAIR

## 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Samyak P. Surti, IV</b>	<b>Project Number</b> <b>J1420</b>
<b>Project Title</b> <b>Lane Merge Algorithm for Autonomous Cars</b>	
<div><div><b>Objectives/Goals</b><p>Traffic jams are caused in a lane merge when people overly slow down when they are approaching the merge. As the people in front start to slow down people behind them also start to get backed up causing everybody to squeeze together like an accordion being pushed from one side. But in the case of an autonomous car if the proper algorithm is implied to accelerate on entry to the merge point it can avoid traffic jams and still maintain a safe distance between each car. So my goal is to create an algorithm that will allow autonomous cars to successfully execute a lane merge under certain assumptions.</p></div><div><b>Methods/Materials</b><p>The method I intend to use is a computer simulation of a lane merge scenario that includes only autonomous cars.</p></div><div><b>Results</b><p>Initially my algorithm was not working for some cases because I was first approximating the distance between each car which allowed for some faults in the program. To fix those faults, I used another approach of finding the actual distance between each car for every delta t the program was running for. This helped me better understand the type of angles the autonomous cars would be turning at, and the speeds they would be going at in real life on the road. Through the course of figuring out my algorithm, I realized that it is correlated with the Venturi Effect. In the Venturi Effect it states that if the space within the pipe is smaller, the velocity within the pipe will be more so that it can maintain streamline motion.</p></div><div><b>Conclusions/Discussion</b><p>My original intuition that I had to the solution of the problem was demonstrated to work in several cases for autonomous cars. Accelerating at the merge point is the best way to avoiding a traffic jam on the road. Not only does it prevent traffic jams, but it also keeps a safe distance between consecutive autonomous cars. I am inspired to continue my research in the subject of autonomous cars. Even though in most cases my algorithm works, there are still some cases that aren't covered. I will continue working on these problems concerning autonomous cars and making the algorithm more intelligent and safer for real life use. I see that my vision about the future of transportation , as shared by an article in National Geographic, is slowly turning into a reality.</p></div></div>	
<b>Summary Statement</b> <p>My project describes using a computer visual how autonomous cars can successfully perform a lane merge.</p>	
<b>Help Received</b> <p>My dad helped me get familiarized with the programming language that I used for my project.</p>	



# CALIFORNIA STATE SCIENCE FAIR

## 2015 PROJECT SUMMARY

<b>Name(s)</b> <b>Patrick Z. Yu</b>	<b>Project Number</b> <b>J1421</b>
<b>Project Title</b> <b>Developing Efficient Algorithms for Self-Navigating Vehicles</b>	
<div><b>Objectives/Goals</b><p>The purpose of this study is to develop efficient algorithms for self navigating vehicles to maneuver around roads. To simulate a self driving car, I created an optimized algorithm and programmed it on a self-driving robot to navigate through different kinds of mazes. I used two other existing algorithms for comparison.</p><b>Methods/Materials</b><p>The two existing algorithms are the Random Algorithm and the Right-Left Algorithm, and the optimized version of the Right-Left is the Loopable Right-Left Algorithm. I spent the most time on developing the Loopable Algorithm and then blended this looped feature with the existing Right-Left Algorithm. To prove the new algorithm was superior to the two existing ones, I used 3 mazes to test the robot. The first maze contained 2 three-way intersections. The second maze with 2 four-way intersections and 1 three-way intersection. The third maze contained 3 three-way intersections, 1 four way intersection, and a loop. I placed the robot in the three mazes and ran it through 6 times for each algorithm, compiling a total of 18 data points per maze and recording the runtimes after each run. I plotted the averages and standard deviations from the runtimes to determine the optimal algorithm.</p><b>Results</b><p>After 6 trials, I compared the runtimes. The Loopable Right-Left Algorithm produced the best results, solving all three mazes in shortest time by completing Mazes 1, 2 and 3 in 7, 15 and 22 seconds respectively. The runner-up was the Right-Left, which produced the identical runtimes as the Loopable Right-Left for the first two mazes, but was not able to solve Maze 3. The worst algorithm of the three was the Random Algorithm, completing the three mazes in 15, 32 and 72 seconds respectively.</p><b>Conclusions/Discussion</b><p>The results clearly demonstrate that the Loopable Right-Left Algorithm is superior to the other two in terms of stability, efficiency and speed. After sensing four consecutive identical turns in a certain direction, the robot knows it will be stuck in a rectangular loop. To exit out of this loop, the rules are reversed, allowing the robot to successfully exit so it doesn't follow the same path. The Right-Left Algorithm yielded similar runtimes for the first two mazes, but did not run as well as the Loopable Right-Left due to its incapability to recognize loop patterns. Finally, the Random Algorithm performed the worst, as it chooses a random direction without evaluating it.</p></div>	
<b>Summary Statement</b> <p>This project's goal is to create an optimal algorithm for robots to navigate all kinds of mazes with the fastest times and best stability.</p>	
<b>Help Received</b> <p>My mentor Dr. Ismail helped in developing my test procedure and reducing the data as well as helping me debug the algorithms. My brother helped me edit my abstract and presentation.</p>	