# CALIFORNIA STATE SCIENCE FAIR
## 2010 PROJECT SUMMARY

**Name(s)**
Justin J. Arevalo

**Project Number**
**J1601**

**Project Title**
## Some Reasons a Computer Slows Down

**Abstract**

**Objectives/Goals**
To find some reasons why computers slow down.

**Methods/Materials**
The materials I mostly needed was a stopwatch and a computer
I first installed the OS and timed the startup time, When that was done I installed applications to my computer and timed the startup time with application, and finally i got a virus and timed the start up time for that

**Results**
I got the results I was looking for the OS by itself was had a fast start up time. The start up time for the PC with installed applications took a little bit longer and finally the PC with a virus took about three minutes.

**Conclusions/Discussion**
I based my conclusion on my hypothesis, and my hypothesis was If you do not scan and update your computer#s antivirus, then your PC will not be protected from new viruses. The result of my project indicates that my hypothesis was correct. This is true because when the Symantec antivirus was disabled, and I went to the website and clicked on a link and it slowed down my system and made my computer not respond. Then I enabled the Symantec antivirus and it gave me the virus that affected my computer.

**Summary Statement**
It's about the the affects of applications and viruses on a computer.

**Help Received**
Dad supplied equipments for project.

# CALIFORNIA STATE SCIENCE FAIR
# 2010 PROJECT SUMMARY

**Name(s)**
Alex Badawi

**Project Number**
**J1602**

**Project Title**
## Statistics and M&Ms

**Abstract**

**Objectives/Goals**
How many tablets of blue, orange, brown, etc. of M&M's will there be in one bag, and what is my chances to select a color every time I pik a bunch of M&M's?

**Methods/Materials**
a-big bag of M&M's
b- several plastic cups
c-journal
d-calculator

Put the M$M colors in each cup.
Count how many colors there are in each cup.
Count all M&M's to find out how much the total is.
Find how many percent chances you will get if you pick any color of M&M's
Make a pie chart or bar graph.
Pick one M&M from a bag or cup and see what color you get.

**Results**
The first time I did my experiment I counted the different color of M&M. When I counted the blue, orange, green, and brown they had the same numbers, but then I found out that red has a less number then the other colors and yellow is the least number of all colors.

**Conclusions/Discussion**
There is a lot more information available on the internet than is possible to list in one paper and more is becoming available everyday. With this information, it is hoped that teachers will get a good start in the right direction, and then they will be able to explore more resources on their own.

**Summary Statement**
Statistics on the number of M&M's in one bag.

**Help Received**
Sistter helped type report and decorate my board.

# CALIFORNIA STATE SCIENCE FAIR
# 2010 PROJECT SUMMARY

**Name(s)**

Drew L. Bent

**Project Number**

**J1603**

**Project Title**

## Exploring Rule Variations in Conway's Game of Life: Programming for the Computer and iPhone Platforms

**Abstract**

**Objectives/Goals**

Conway's Game of Life is a cellular automaton, which is a mathematical simulation that demonstrates the phenomenon of how complex patterns can be formed by a few simple rules. The purpose of the project is to investigate how changing the rules will affect the population of cells as a function of generation in the Game of Life. It is hypothesized that there is a rule set different from Conway's that has a larger variety of evolution patterns formed from simple starting arrangements. Such a rule set would be an improved version of Conway's according to his standards, in which he tried to find a set that created unpredictable population behavior.

**Methods/Materials**

The Game of Life was programmed on the iPhone platform using the iPhone Software Development Kit and for the computer platform using Adobe's Flash software. In one experiment, rule variations were tested against the original rules using each of 12 pentominoes as starting arrangements. Population was recorded at each generation and the population evolution was categorized into four different population outcomes (rapidly increasing, dying off, steady, and oscillating) for each pentomino under each set of rules. In the second experiment, a new parameter was used with the original rules that caused each cell to die after staying alive for a specific number of consecutive generations (called a death parameter).

**Results**

In the first experiment, the results show that different sets of rules lead to different distributions of population outcomes. Two of the tested rule variations have a more even distribution of population outcomes (standard deviations of ~2.12 and ~1.58) than Conway's original rules do (~2.55). In the second experiment, the generation numbers at which the average populations of the pentominoes for each rule set start to noticeably decrease (G') are correlated with the values for the death parameter (D). The bigger the D value, the longer it takes for the average population to decrease, with G' becoming more similar as the death rates get larger.

**Conclusions/Discussion**

The results from both experiments support the idea that the rules play an important role in creating different patterns in the Game of Life. The biggest contribution this work will give to the field of Conway's Game of Life is finding two rule variations that create more interesting and unpredictable patterns than Conway's when using simple starting arrangements.

**Summary Statement**

This project involves programming Conway's Game of Life for two platforms and exploring the effects of altering the rules to try to find a more desirable set of rules than the original ones.

**Help Received**

Parent helped with discussions about experimental design.

# CALIFORNIA STATE SCIENCE FAIR
# 2010 PROJECT SUMMARY

**Name(s)**
Andrew R. Dutcher

**Project Number**
**J1604**

**Project Title**
## Compressed for Time: Examining File Compression in Computers

**Abstract**

**Objectives/Goals**
My project's goal is to find out in which way files on computers can be most effectively compressed while retaining all the data inside them. I hypothesized that either the Bzip2 algorithm, an established entity in the world of compression, or an algorithm that began by rearranging the data of the file and ended with a function that took advantage of the orderliness of a file would be the most effective in compression.

**Methods/Materials**
To accomplish my objective, I built a computer program in C#, a programming language, that would be able to compress files in combinations of six different ways, and automatically cycle through all combinations of these methods for four different files and output the compressed file-sizes, in bytes. I then built a peripheral program that would process the raw numbers and return actual data.

**Conclusions/Discussion**
In the end, I found that the data support my hypothesis. Not only do many of the top algorithms closely resemble the Bzip2 algorithm, many other top algorithms also start with rearrangement functions and end with entropy functions.

**Summary Statement**
I am attempting to find the optimal form of losless file compression.

**Help Received**
Father helped with application coding.

# CALIFORNIA STATE SCIENCE FAIR
# 2010 PROJECT SUMMARY

**Name(s)**
Brian M. Ito-Kiley

**Project Number**
**J1605**

**Project Title**
## How to Win at Yahtzee

**Abstract**

**Objectives/Goals**
The objective is to find the most and least probable rolls in Yahtzee. With this information, I can use statistics to increase my chances of winning.

**Methods/Materials**
I used a pencil, paper, calculator, and 5 dice.
First, I used math to calculate the odds of rolling certain Yahtzee hands. Then I rolled 5 dice, 100 times, and recorded my results.

**Results**
The following results show the probability of Yahtzee hands (rolls), from most likely to occur, to least likely: a) ones, twos, threes, fours, fives, and sixes b) three-of-a-kind c) small straight d) full house e) large straight f) four-of-a-kind g) Yahtzee

**Conclusions/Discussion**
To minimize the frequency of "zero" points (incomplete Yahtzee hands)on your score card, this is my conclusion: First, try to get a Yahtzee. Next, go for a four-of-a-kind. Then (in descending order) try to get a large straight, a full house, a small stright, then a three-of-a-kind. You want to try to get the harder rolls first. You should always try for a roll that you're in a good position to achieve, but you should lean toward rolling a more improbable roll. The good thing about this strategy is that you can use the easier Yahtzee hands as a saftety net.

**Summary Statement**
To discover a winning strategy in the game of Yahtzee.

**Help Received**
Mother helped cut papers for display board.

# CALIFORNIA STATE SCIENCE FAIR
# 2010 PROJECT SUMMARY

**Name(s)**
Abraham P. Karplus

**Project Number**
**J1606**

**Project Title**
## A. I. Connect-Four

**Abstract**

**Objectives/Goals**
I am using the game Connect-Four to study artificial intelligence. I wanted to determine the effects of ply depth (the number of plies, which are turns by a single player, that the program looks ahead) and the effects of getting the first move on both the chance of winning and the time taken.

**Methods/Materials**
To do this, I wrote some computer programs in C to simulate games between computer-controlled players.

**Results**
I found that increasing the ply depth increases the chance of winning, as does getting the first move. I found that the effect of a single ply depth increase was greater than the effect of getting the first move. I found that when two identical players faced off, there would be more draws if both had even ply depths. I also found that increasing the ply depth by 1 make the program take about 7 times as long per move.

**Conclusions/Discussion**
My results agree with my hypothesis, though I did not expect the result with more draws at even ply depths. Each extra ply taking 7 times as long fits with the prediction based on branching width.

**Summary Statement**
I created an artificial-intelligence program to play Connect-Four and used it to test the effects of going first and looking further ahead in possible moves on the chance of winning and on the time taken.

**Help Received**
My father, Kevin Karplus, mentored me on this project.

# CALIFORNIA STATE SCIENCE FAIR
## 2010 PROJECT SUMMARY

**Name(s)**
Nikhil Lonberg

**Project Number**
**J1607**

**Project Title**
## The Random Fibonacci Sequence

**Abstract**

**Objectives/Goals**
The purpose of this project is to see if the nth root of the absolute value of the nth term in a random Fibonacci sequence approaches v (1.1319...). The project also resolves a discrepancy between two different published descriptions of the random Fibonacci sequence. The hypothesis of this project is that the nth root does approach v and the original description of the behavior of the random Fibonacci sequence, by Divakar Viswanath, is more correct.

**Methods/Materials**
The hypothesis is tested by generating a random Fibonacci sequence and determining the nth root of it to see if it converges on v. This experiment will also test to see if Mario Livio is right about the random Fibonacci sequence converging at around the hundredth number in the sequence. To simulate this, the experiment uses a Visual Basic computer program with a pseudo random number generator to choose the operators. An order to perform this experiment the computer needs Microsoft Visual Basic installed into Excel.

**Results**
The results show that the nth root of the nth term does eventually approach v, but very far into the sequence. Although the results vary because of the randomness of the sequence, the nth root typically becomes very close to v when n is approximately equal to 50,000.

**Conclusions/Discussion**
The nth root of the nth term approaches v well past the hundredth number in the sequence. This supports Viswanath#s conclusions on random Fibonacci sequences.

**Summary Statement**
This project explores and tests the behaviors of a random Fibonacci sequence, a variant of the Fibonacci sequence.

**Help Received**
Father made useful suggestions about programming.

# CALIFORNIA STATE SCIENCE FAIR
# 2010 PROJECT SUMMARY

**Name(s)**
Ayesha M. Rashed

**Project Number**
**J1608**

**Project Title**
## Turbo Charging Computer with Mathematical Algorithms

**Abstract**

**Objectives/Goals**
How does the computer compute and does the computing time depends upon the speed of the computer itself or the method used to solve a given problem.

**Methods/Materials**
Materials. I used the following things in this project: (1) Old and New Computers, (2) Stop Watch, (3) Logic Gate Simulator, (4) Scratch Programming Environment.

Method: To test my hypothesis I used a problem of adding N integer numbers starting from 1. There can be several methods we can use to program to solve this problem. I used following three methods:
M(1) Adding each number in a loop of N iterations.
M(2) Adding N+1 in a loop N/2 iterations.
M(3) Adding N integers using a pre-computed mathematical formula.

**Results**
All three methods were implemented in Scratch programs and run for N = 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000. The time it took to complete the computation has been noted and plotted on a line graph.

To compare the performance of the three methods on different computers following computers are used to test the speed of each method.
(1) Computer with Intel Core i7, 8 Core Processors
(2) Computer with Core 2 Quad Core, 4 Processors
(3) Computer with Core 2 Duo, 2 Processors
(4) Laptop with Intel Centrino Processor

Extended Experiments
Since the Scratch programs did not produce some sensible data to compare the speed of the different computers. I sought help from my father and implemented the same programs using Python environment on the following:
(1) Desktop Computer with Intel Core i7 (dad#s)
(2) iPhone 3Gs
(3) Vintage Laptop with Intel Pentium II

**Summary Statement**
Explore how basic logic gates are used to build computers and compare the mathematical methods to speedup computing.

**Help Received**
Father helped in programming in Python and younger brother helped in building the project board.

# CALIFORNIA STATE SCIENCE FAIR
# 2010 PROJECT SUMMARY

**Name(s)**
Brandon R. Sepulveda

**Project Number**
**J1609**

**Project Title**
## Cracking the Code: Analyzing Alphanumeric Combinations and Password Security

**Abstract**

**Objectives/Goals**
The objective of my science fair experiment is to determine the effects of variable alpha numeric combinations on password security.

**Methods/Materials**
In my experiment I am using four letters and or four numbers as the foundation of my investigation. I am using a password hacker called Free Word Excel Password Wizard which recovers encrypted word documents. I used Brute force recovery to crack a word document that I encrypted with one of the variable passwords. I used several different combinations of randomly generated passwords for each alpha-numeric code such as the combination of three letters and one number, two letters and two numbers, three numbers and one letter, four letters and four numbers.

**Results**
The results of my experiment show that the password with three letters and one number was the most secure and took on average 57.18 seconds to crack. The password of two letters and two numbers too an average time of 50.83 seconds to crack. The password of one letter and three numbers took 53.74 seconds to crack. The password of four letters took an average time of 52.04 seconds to crack. By far the least secure and fastest to crack was the password of four numbers on average was cracked at a time of 18.02 seconds.

**Conclusions/Discussion**
I found that my hypothesis was incorrect. I hypothesized the password with the most letters would be the hardest to crack based on the fact that there 456,976 possible combinations as compared to the 175,760 possibilities of a three letter one number password. In conclusion, people with passwords of more letters and a few numbers have the safer more secure passwords. In this dangerous cyber world where everything is computerized people need to be more informed on how to keep their private information private.

**Summary Statement**
Determining what combination of alpha numerics creates the strongest password.

**Help Received**
Mother helped put together board

# CALIFORNIA STATE SCIENCE FAIR
# 2010 PROJECT SUMMARY

**Name(s)**
Prem M. Talwai

**Project Number**
**J1610**

**Project Title**
## Chess Algorithms

**Abstract**

**Objectives/Goals**
The main goal of this project is to create an effective algorithm for finding the best move in a given chess position. This algorithm will consist of a series of step-by-step tests and avoids the use of intuition or the need for a theoretical chess background.

**Methods/Materials**
The materials I used were a chess set and chessboard, various positions from The Art of Planning in Chess and other chess books, Fritz 11 and ChessDiagrammer chess software, and a pencil or pen. In order to create my first algorithmic component, I analyzed various chess positions where tactical motifs, or types of tactics, were present and investigated the interactions and relationships between the pieces involved in the tactics. Geometrically defining these relationships enabled me to create a series of step-by-step methods a player can use to determine whether he can execute any tactics. For my second component, the Positional Evaluator, I studied annotated chess games and classified plans that were employed in those games under certain categories based on their purposes and goals. By examining these plans I was able to extract certain relationships between the pieces involved in the plan and other points and lines on the board. I again used these relationships to create various tests chess players can use to determine whether he should implement a certain type of plan. I tested my algorithm on many different instances from master games and in each instance a strong move was found.

**Results**
I successfully created an algorithm for finding the best move in any chess position that does not require the user to have any theoretical chess background or to implement his intuition.

**Conclusions/Discussion**
I concluded that there are certain spatial and functional relationships between the pieces on the chessboard that determine the positional and tactical features present in a given position. I fulfilled my goal by creating an algorithm for finding a strong move in a chess position that avoids the use of intuition.

**Summary Statement**
My project creates an understandable algorithm for finding the best move in a chess position that does not require the user to implement his intuition.

**Help Received**
No help was received.

# CALIFORNIA STATE SCIENCE FAIR
## 2010 PROJECT SUMMARY

**Name(s)**
Jacob A. Todd

**Project Number**
**J1611**

**Project Title**
# Location, Location, Location: The Impact of a Player's Starting Location in the Game of Risk

**Abstract**

**Objectives/Goals**
The objective was to discover the effect of a player's starting location on outcome in the game of RISK. I hypothesized that consolidation of a player's initial territories and involvement in one triangular relationship (where three players could all attack each other equally) would each increase the likelihood of winning by 10%.

**Methods/Materials**
A computer program simulating the RISK game was developed and simulated 100,000 games. For each game, the simulation randomly assigned players their initial territories and went through all the steps of a RISK game, including setup, diplomacy, attacking, and fortification.

**Results**
The four territories in Australia outplayed every other territory and continent by far. The average Australian player controlled 20.5 territories at the end of a game, while the overall average per player was 5.2 territories. Players whose initial territories were non-contiguous did twice as well as players with contiguous initial territories. Players starting in continents with no triangular relationships (Australia and South America) averaged approximately 12 territories held at the end of the game, while players starting in continents that were in one triangular relationship averaged approximately 3 territories. The results did not support the hypothesis.

**Conclusions/Discussion**
Starting in Australia is more likely to lead to a successful outcome. The relative size of the continents may have impacted results. Australia is a small, isolated continent with four territories and one corridor of attack. The benefits of starting in Australia appear to outweigh the hypothesized benefits of starting with consolidated territories and the presence of a triangular relationship.

**Summary Statement**
This experiment tested the impact of starting location on a player's chances of winning in the game of RISK.

**Help Received**
Computer program troubleshooting assistance by Gerald Fitzgerald.

# CALIFORNIA STATE SCIENCE FAIR
# 2010 PROJECT SUMMARY

**Name(s)**
Jennie Werner

**Project Number**
**J1612**

**Project Title**
## Mysterious Number 6174

**Abstract**

**Objectives/Goals**
To determine if a formula can be derived to find the number iterations it takes to reach zero or a Kaprekar Constant with out going through the Kaprekar Routine.

**Methods/Materials**
Write a program to find the number of iterations it takes for numbers 1 to 10,000 to hit 0 or a Kaprekar Constant. Then create a graph that shows the number of iterations it takes each number to complete the Kaprekar routine to get a visual representation of the patterns. After, import the data from the program into Excel. Use Excel functions to find patterns between the starting number and the number of iterations.

**Results**
1. Based on the difference found in the first iteration of the Kaprekar Routine the number of iterations it takes to reach zero or a Kaprekar Constant can be predicted. If the difference is a Kaprekar Constant, the order of the digits matter, otherwise the order of the digits in the difference does not matter. Since all two, three and four digit numbers all lead to one of 43 numbers, once the difference for those numbers is found, one can find the number of iterations for all the two through four digit numbers.

2. For two digit numbers based on the difference of the digits alone, the number of iterations can be predicted. If the difference is 0, 1, 3 or 4, add 1 to the difference to get the number of iterations. For numbers with a difference of 5 or 7, subtract 2 to find the difference. For 6 and 8 divide by 2 to find the number of iterations, and if the difference is 2, the number of iterations will be 6.

3. The sum of the digits in the differences will always be zero or a multiple of 9 for two to four digit numbers. For three digit numbers the middle digit in the difference is always a nine, and the sum of the other two digits is nine, which mean the sum is 18, except for the numbers that lead to zero.

4. The sum of the digits in the Kaprekar Constants for three and four digit numbers is always a multiple of 9.

**Conclusions/Discussion**
A generic formula was not found to derive the number of iterations it takes for a number to reach zero or a Kaprekar Constant. However several patterns were established that allow one to find the number of iterations needed without going through the Kaprekar Series for each number.

**Summary Statement**
A program was written with Scratch and patterns were found in the Kaprekar Numbers.

**Help Received**
My technology teacher, Mr. Appelbaum, gave me the idea for my project, and introduced me to Scratch, free programing software that I used.

# CALIFORNIA STATE SCIENCE FAIR
## 2010 PROJECT SUMMARY

**Name(s)**
Yousuf Soliman

**Project Number**
**J1699**

**Project Title**
**Go! Go! Gadget!**

**Abstract**

**Objectives/Goals**
This project is designed for senior and disabled individuals by building a robot that is able to pick up objects and controlled by a wireless PS2 controller.

**Methods/Materials**
A robot with motors to be able to move and a motor for the claw was designed and built. It was then programmed in JAVA to be remote controlled. The robot was then tried repeatedly and results were recorded.

**Results**
The robot was able to move forward, backwards, be able to pick up an object, and release it on command. It was also able to hold a first aid kit and it also has a small light in the front.

**Conclusions/Discussion**
The robot worked well and almost flawlessly. It was able to open and close the claw using one motor, this was very efficient and helped me have more functions for the robot. The robot was helpful to seniors and was able to pick up certain objects for them.

**Summary Statement**
This project is to build a robot that is remote controlled to help senior and disabled people.

**Help Received**
Teacher helped teach me JAVA.