What is the best strategy for Master Mind?

MASTERMIND
The Challenging Game of Logic and Deduction
Break the hidden code! New sleek design with improved built-in storage tray and code shield!
MASTERMIND
For 2 players, Ages 8 to adult
Pressman

New Mexico
Supercomputing Challenge
Final Report
April 6, 2022

Team 48
Mountain Elementary

Team Members
Michael Petersen

Teacher
Mrs. Hawkins

Project Mentor
Mark Petersen

Executive Summary

My Supercomputing Challenge project is to figure out the best strategy for Master Mind. Master Mind is a game that helps kids learn about making and breaking secret codes. In real life, computers have to make a good enough code that other computers can't break it, for example to steal credit card numbers. I wrote a computer program that made a game of Master Mind. I tested my computer program to see what was the best strategy. I tested four strategies by having the computer play Master Mind 480 times . The best strategy for the computer was random guesses from the possible codes, but for humans the random strategy is really hard. The best strategy for humans is Add-On, because it is easier to keep track of than the random strategy, but is still pretty fast.

Statement Of The Problem

I'm trying to find what is the best strategy for Master Mind. I will write a computer program to help me test four strategies.

Hypothesis

I think the best strategy for Master Mind is "Add on", the third strategy. The code breaker adds one color to the last guess every turn until he or she gets knows all the colors.

Background

Mastermind is a game where one person is the code maker and the other person is the code breaker. The code maker makes a four or six digit code with different colored pegs. The code breaker tries to break the code by trying to mimic the code. The code maker grades the guess like this,
Grading Key: Red = right color, right location
White = right color, wrong location
and then the code breaker guesses again.

Master Mind is just a game, but in the real world computers have to make a code so hard that other computers can not break the code. This is called encryption and is used to keep crooks away from your credit card and private emails.

Procedure

1. Practice the real Mastermind game
2. Make the game by computer programming.
3. Practice with computer playing the human.
4. Test different strategies, like one color per turn, and random guess.
5. I will test the code a hundred times to see what the best strategy is. I will record which one wins the most.

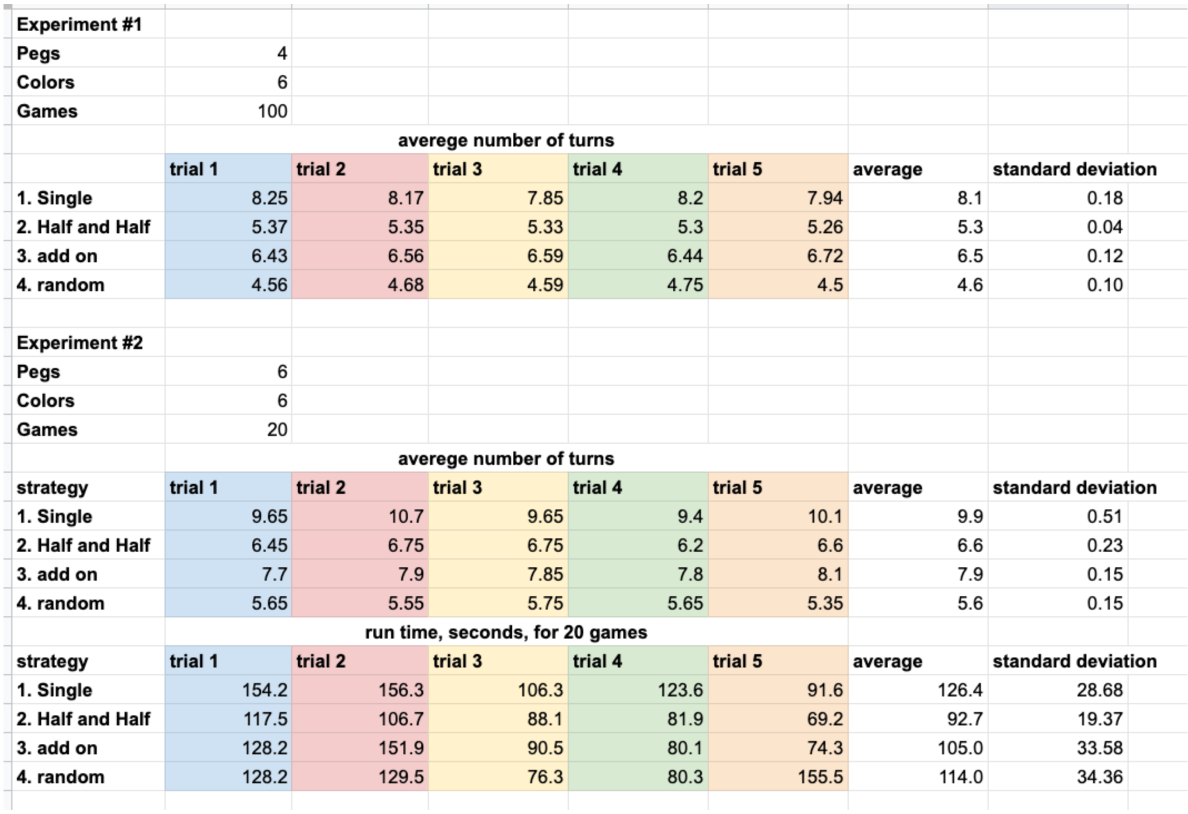
Results

I tested four strategies.

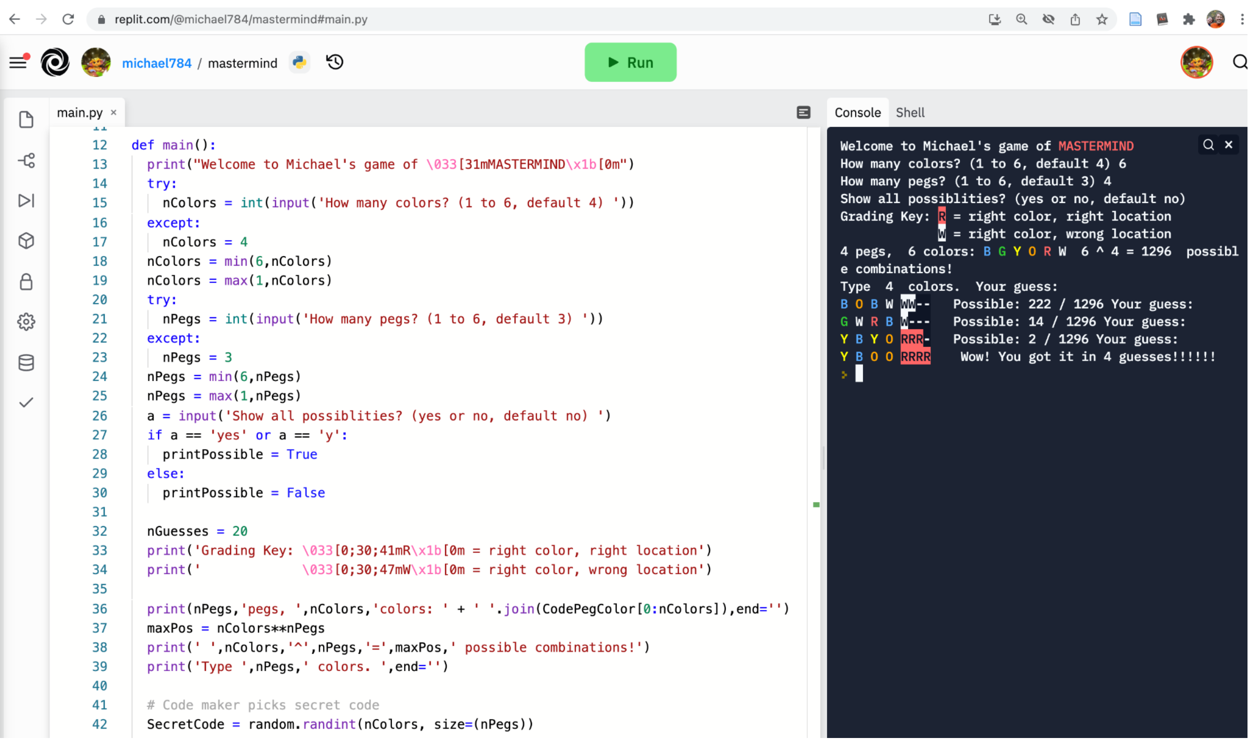
1. **single:** One color per turn
2. **half and half:** two colors per turn
3. **add on:** Add on one color to the last guess
4. **random:** different colors on every turn randomly chosen out of the of the combinations left.

For the first three strategies, once you know the colors you switch to random.

I did two Experiments because I thought that one should be easy and one should be hard. Experiment number one has 4 pegs and 6 colors and is exactly like the real master mind game. Experiment number two has 6 pegs and 6 colors and is much harder.



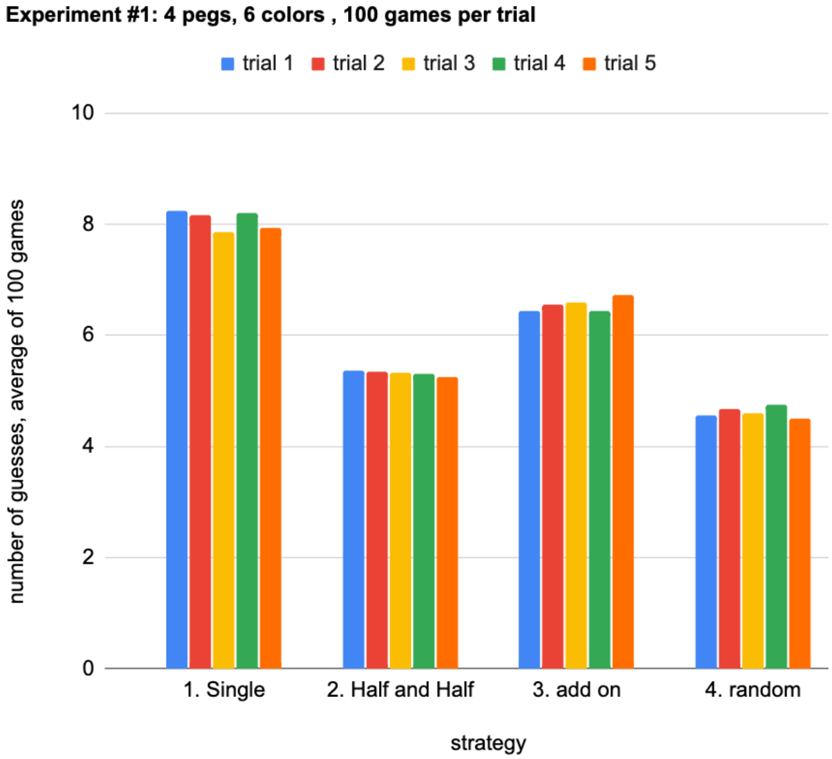
Experiment #1							
Pegs	4						
Colors	6						
Games	100						
			averege number of turns				
	trial 1	**trial 2**	**trial 3**	**trial 4**	**trial 5**	**average**	**standard deviation**
1. Single	8.25	8.17	7.85	8.2	7.94	8.1	0.18
2. Half and Half	5.37	5.35	5.33	5.3	5.26	5.3	0.04
3. add on	6.43	6.56	6.59	6.44	6.72	6.5	0.12
4. random	4.56	4.68	4.59	4.75	4.5	4.6	0.10
Experiment #2							
Pegs	6						
Colors	6						
Games	20						
			averege number of turns				
strategy	**trial 1**	**trial 2**	**trial 3**	**trial 4**	**trial 5**	**average**	**standard deviation**
1. Single	9.65	10.7	9.65	9.4	10.1	9.9	0.51
2. Half and Half	6.45	6.75	6.75	6.2	6.6	6.6	0.23
3. add on	7.7	7.9	7.85	7.8	8.1	7.9	0.15
4. random	5.65	5.55	5.75	5.65	5.35	5.6	0.15
			run time, seconds, for 20 games				
strategy	**trial 1**	**trial 2**	**trial 3**	**trial 4**	**trial 5**	**average**	**standard deviation**
1. Single	154.2	156.3	106.3	123.6	91.6	126.4	28.68
2. Half and Half	117.5	106.7	88.1	81.9	69.2	92.7	19.37
3. add on	128.2	151.9	90.5	80.1	74.3	105.0	33.58
4. random	128.2	129.5	76.3	80.3	155.5	114.0	34.36

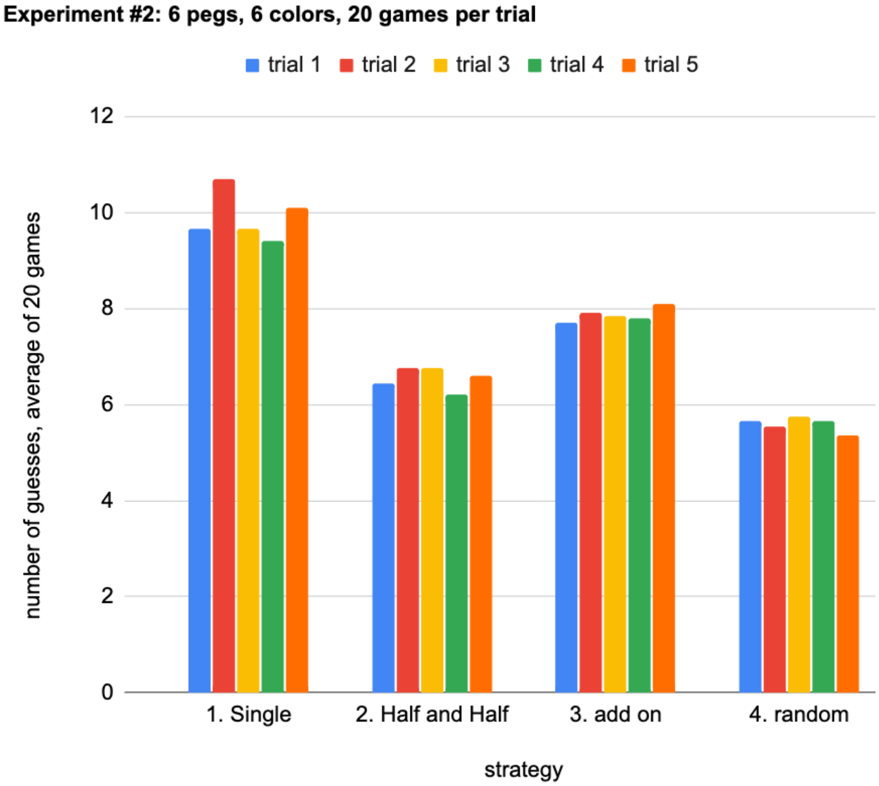


```python
def main():
  print("Welcome to Michael's game of \033[31mMASTERMIND\x1b[0m")
  try:
    nColors = int(input('How many colors? (1 to 6, default 4) '))
  except:
    nColors = 4
  nColors = min(6,nColors)
  nColors = max(1,nColors)
  try:
    nPegs = int(input('How many pegs? (1 to 6, default 3) '))
  except:
    nPegs = 3
  nPegs = min(6,nPegs)
  nPegs = max(1,nPegs)
  a = input('Show all possiblities? (yes or no, default no) ')
  if a == 'yes' or a == 'y':
    printPossible = True
  else:
    printPossible = False

  nGuesses = 20
  print('Grading Key: \033[0;30;41mR\x1b[0m = right color, right location')
  print('             \033[0;30;47mW\x1b[0m = right color, wrong location')

  print(nPegs,'pegs, ',nColors,'colors: ' + ' '.join(CodePegColor[0:nColors]),end='')
  maxPos = nColors**nPegs
  print(' ',nColors,'^',nPegs,'=',maxPos,' possible combinations!')
  print('Type ',nPegs,' colors. ',end='')

  # Code maker picks secret code
  SecretCode = random.randint(nColors, size=(nPegs))
```

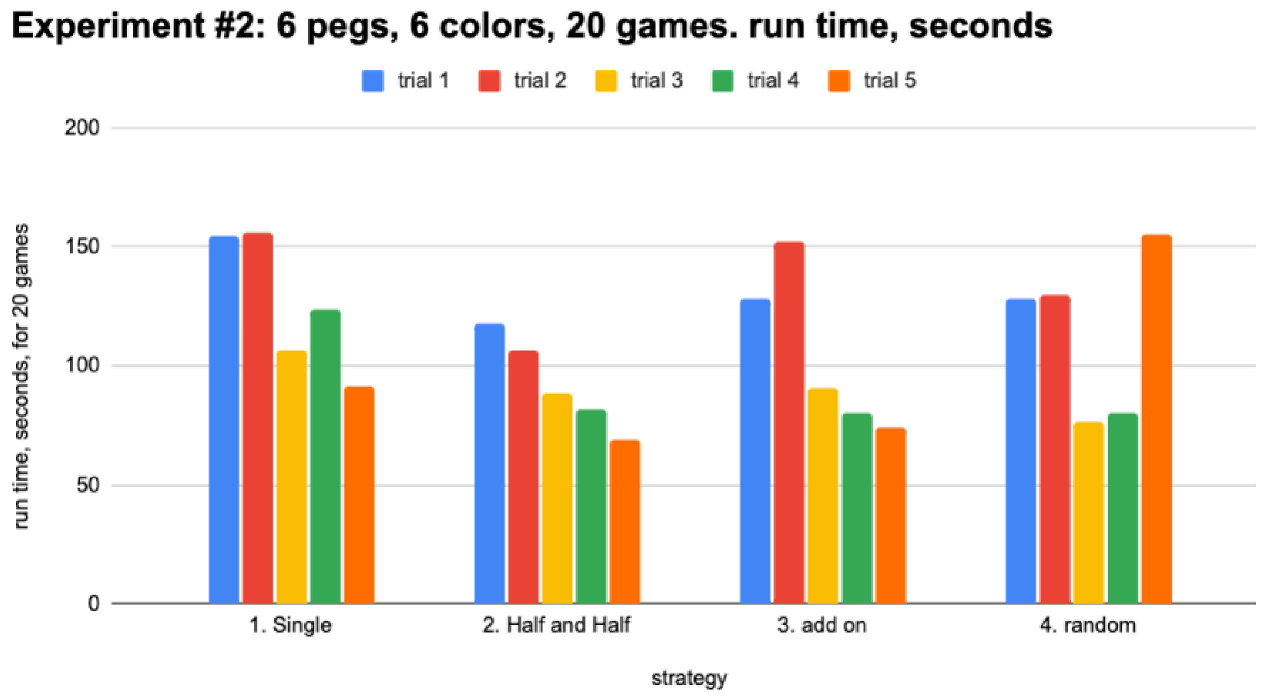
```
Welcome to Michael's game of MASTERMIND
How many colors? (1 to 6, default 4) 6
How many pegs? (1 to 6, default 3) 4
Show all possiblities? (yes or no, default no)
Grading Key: R = right color, right location
             W = right color, wrong location
4 pegs,  6 colors: B G Y O R W  6 ^ 4 = 1296  possibl
e combinations!
Type  4  colors.  Your guess:
B O B W WW--   Possible: 222 / 1296 Your guess:
G W R B W---   Possible: 14 / 1296 Your guess:
Y B Y O RRR-   Possible: 2 / 1296 Your guess:
Y B O O RRRR    Wow! You got it in 4 guesses!!!!!!
>
```

Results: number of guesses





Results: run time



Conclusions

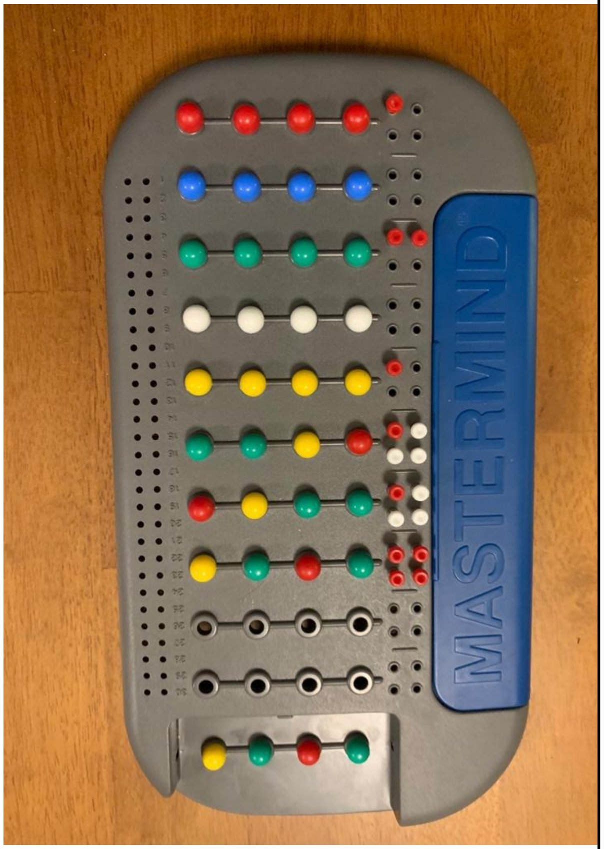
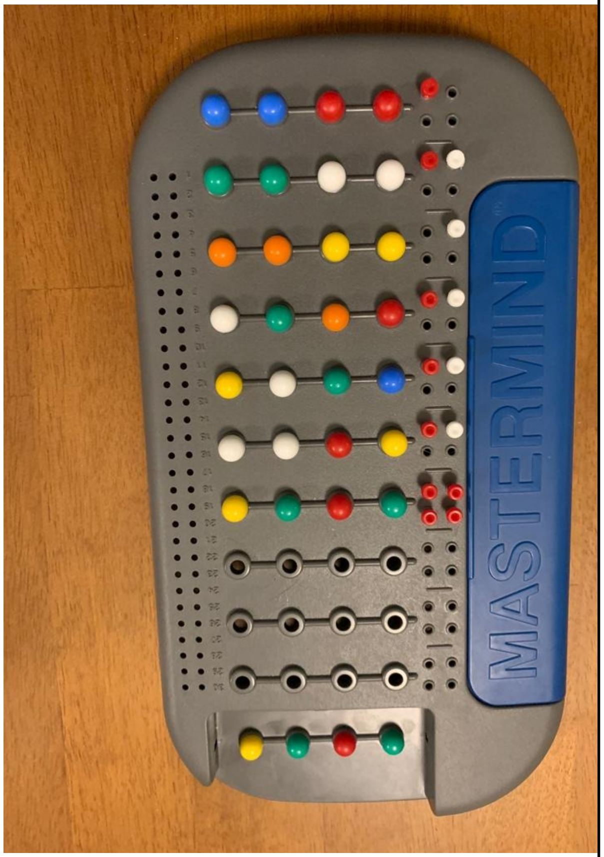
In the end the random strategy beat them all! My hypothesis was wrong. The worst strategy was single because it wasted the most guesses, even though it is the easiest for a human. The half-and-half strategy come in second place, but it is the second hardest for a human. I think that the add-on strategy was the best for humans because it is easier to keep track of than random or half-and-half, and faster than the single strategy. For the computer random is the best because it can keep track of all the combinations, but a human can't do that!

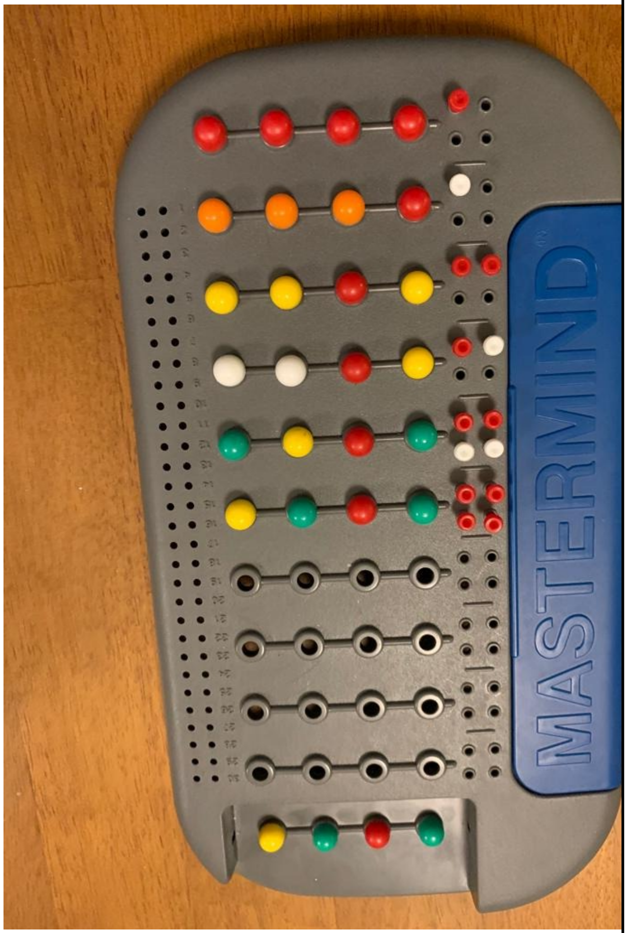
Bibliography:

Mastermind (board game) - Wikipedia
https://en.wikipedia.org/wiki/Mastermind_(board_game)

How to play Mastermind | Official Rules | UltraBoardGames
https://www.ultraboardgames.com/mastermind/game-rules.php

Strategies

1. single: 1 color per turn	**2. half & half:** 2 colors per turn
	

3. add on: Add on 1 color to the last guess  	**4. random:** different colors on every turn randomly chosen out of the of the combinations left.