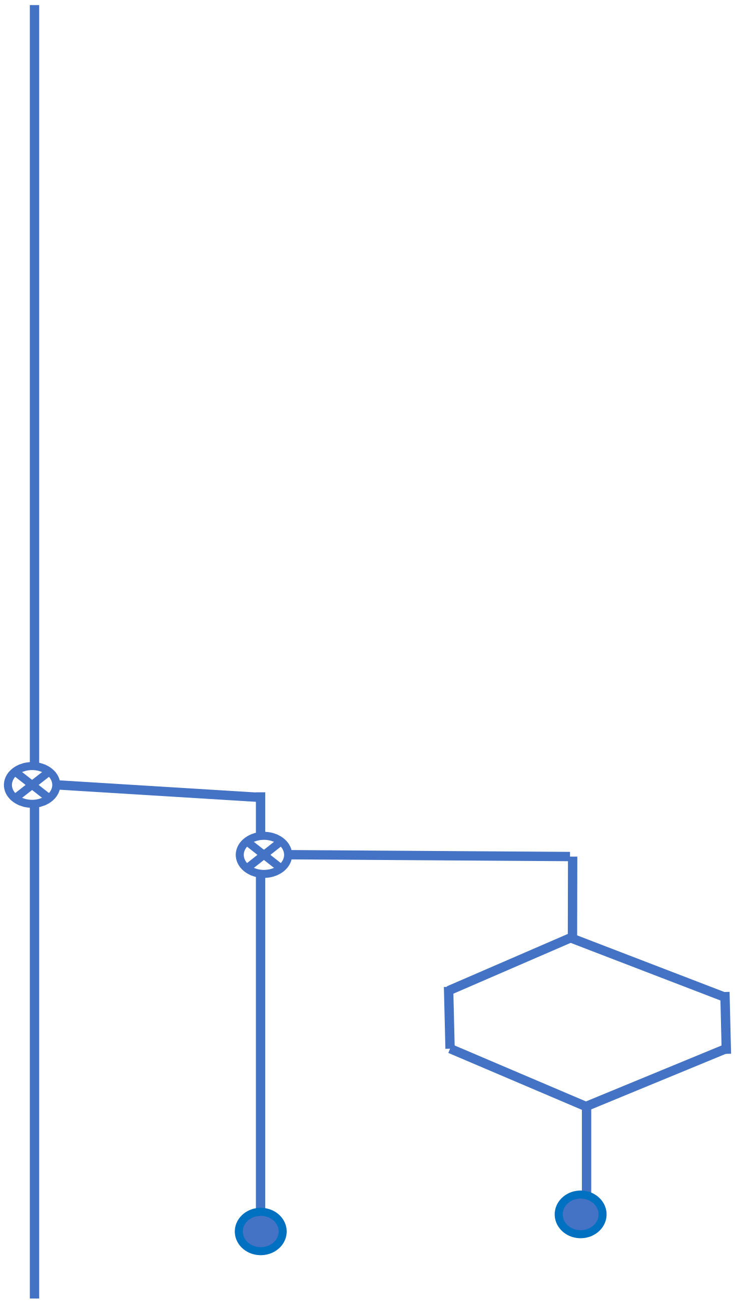

Rock-Paper-Scissors

Silicon Valley Forth
Interest Group
April 27, 2024
Bill Ragsdale

Today We’ll Cover

- Programming a game.
- Systematic Testing.
- Strategy development.

The Game

Two players alternate.

Each plays an object: rock, paper, scissors.

Each object can win, lose or tie to the other.

The Game

Two players alternate.
Each plays an object: rock, paper, scissors.
Each object can win, lose or tie to the other.

Rock breaks scissors.
Scissors cut paper.
Paper wraps rock.

The Game

If the play by Allen specifies a column.

If the play by Betsy specifies a row.

Then the win/loss/tie and winner is given by an integer in the following table:

The Game

If the play by Allen specifies a column.

If the play by Betsy specifies a row.

Then the win/loss/tie and winner is given by an integer in the following table:

ROCK	**PAPER**	**SCISSORS**	
Tie-0	Win-1	Lose-2	**ROCK**
Lose-3	Tie-0	Win-4	**PAPER**
Win-5	Lose-6	Tie-0	**SCISSORS**

Pseudocode

Create logic for win/loss/tie by player and by object.

Create report elements: who & why.

Create a playing process.

Report process results of one match.

Report process for a sequence of matches.

An automatic play process.

A statistical choice of play.

A strategy to win against a biased player.

Array For Items and Players

From two plays select the game result.

Alan's choice is the column.

Betty's choice is the row.

```
CREATE Outcome    ( 0 to 6 )
\  rock    paper   scissors       for Allen
   0 C,    1 C,    2 C,   \ rock for Betty
   3 C,    0 C,    4 C,   \ paper for Betty
   5 C,    6 C,    0 C,   \ scissors for Betty
```

A Sample Game

```
Establish the game result from two inputs.

0 CONSTANT rock     1 CONSTANT paper
2 CONSTANT scissors

: a-game
  result-selection show-result show-reason ;

: result-selection
     3 * + Outcome + c@  log-game-result  ;

paper rock a-game     \ play play result

[see]  Allen wins: paper wraps rock  ok
```

Log Game Winner

Increment a value for a match winner.

```
: log-game-result
        dup to Result  1 swap
   case 0 of +to Ties        endof
        1 of +to Allen       endof
        2 of +to Betty       endof
        3 of +to Betty       endof
        4 of +to Allen       endof
        5 of +to Allen       endof
        6 of +to Betty       endof
                          endcase ;
```

Display The Game Winner

```
Give a message selected by the game result.

: show-result
 Result case  cr
  0 of ." Allen and Betty tie."  endof
  1 of ." Allen wins: "          endof
  2 of ." Betty wins: "          endof
  3 of ." Betty wins: "          endof
  4 of ." Allen wins: "          endof
  5 of ." Allen wins: "          endof
  6 of ." Betty wins: "          endof
                                 endcase ;
```

Display The Reason

From the game result give a message why.

```
: show-reason
    Result case
   0 of    ( silent here )         endof
   1 of ." paper wraps rock "     endof
   2 of ." rock breaks scissors “ endof
   3 of ." paper wraps rock "     endof
   4 of ." scissors cut paper "   endof
   5 of ." rock breaks scissors " endof
   6 of ." scissors cut paper "   endof
                                endcase ;
```

Sample Play

```
paper rock a-game    \ play play result

show-result Allen wins:

show-reason paper wraps rock  ok
```

Strategy Development

Your opponent must have a bias or you have an ‘edge’.

When you win, do more if it.

When you lose do less of it.

Vary your play to avoid alerting your opponent.

[This is identical to card counting at a casino.]

Pseudocode

```
Setup your probability for each of the three choices.  All equal to start.
Your opponent picks a choice at random but with a bias.
You make a choice according to your  table of probabilities.
On a win, increase the future probability of that choice.
On a loss, decrease the future probability of that choice.
```

Probability Support For Betty

Begin with equal probability of choices.

```
VARIABLE B-Rock     \ Probability of a choice
VARIABLE B-Paper
VARIABLE B-Scissors


: setup-Betty   \ all equal probabilities
 333 dup dup B-Rock ! B-Paper ! B-Scissors !
;
```

Allen’s Choice, 50% ROCK,

If a random number is greater than 500 play ROCK (0).

Else if it is greater than 250 play PAPER (1).

Else play SCISSORS (2).

```
: Allen-Strategy-Play
  1000 random dup
   500 > if drop ROCK else
   250 > if PAPER else SCISSORS then then ;
```

Betty’s Choice

```
Random number under 1000.
If greater than prob(Paper+Scissors)
    play ROCK.

If greater than prob(Scissors) play PAPER
   else play SCISSORS.
```

```
: Betty-Strategy-Play
1000 random  dup
 B-Paper @ B-Scissors @ +  >
 if drop ROCK
  else B-Scissors @ >
      if PAPER  else SCISSORS then then   ;
```

Betty’s Adjustments, +/- ½%

Adjust probability of a choice up or down by win/loss.

```
: Betty-update
  B-play case  0 of B-rock     endof \ Betty’s last play
               1 of B-paper    endof
               2 of B-scissors endof  endcase
   dup @  Result                     \ Betty's outcome


   case 0 of  1000  endof     \ no adjustment
        1 of   995  endof     \ decrease by ½%
        2 of  1005  endof     \ increase by ½%
        3 of  1005  endof
        4 of   995  endof
        5 of   995  endof
        6 of  1005  endof  1000  endcase
          1000 */ swap ! ;     \ apply adjustment
```

Results, all in percentages

	Ties	Allen	Betty	Rock	Paper	Scissors	
Match 1	0	100	0	33	33	33	
Match 100	31	33	36	32	36	31	
Match 500	34	29	35	35	45	19	A
Match 1,000	37	30	32	32	56	10	
Match 1,200	31	31	36	19	76	3	
Match 1,500	34	28	36	B 28	71	0	
Match 1,600	30	31	38	5	94	0	
Match 1,700	30	28	C 41	0	100	0	

A First Betty decreases Scissors, to avoid losses.

B Then Betty decreases Rock, to avoid ties.

C Due to ties the final is Betty 41% to Allen 28%.

This illustrates the Law of Large Numbers.

Conclusions

- Game play was very simple; select a win/loss/tie from a table.

Conclusions

- Game play was very simple; select a win/loss/tie from a table.
- User input and scoring was simple.

Conclusions

- Game play was very simple; select a win/loss/tie from a table.
- User input and scoring was simple.
- Even against a very gross player developing a strategy took many matches.

Conclusions

- Game play was very simple; select a win/loss/tie from a table.
- User input and scoring was simple.
- Even against a very gross player developing a strategy took many matches.
- The law of large numbers is why casinos stay in business: A small edge over a large number of plays.