Simulation and Statistical Exploration of Data (Let's Make a Deal – The Monty Hall Problem), using the new ClassPad300-technology

Ludwig Paditz, University of Applied Sciences Dresden (FH), Germany

Statement of the Problem

The **Monty Hall problem** involves a classical game show situation and is named after Monty Hall*), the long-time host of the TV game show *Let's Make a Deal*. There are three doors labeled 1, 2, and 3. A car is behind one of the doors, while goats are behind the other two:

The rules are as follows:
1. The player selects a door.
2. The host selects a different door and opens it.
3. The host gives the player the option of switching from the original choice to the remaining closed door.
4. The door finally selected by the player is opened and he either wins or loses.

*) cp. Internet: http://www.math.uah.edu/stat/games/games6.xml

By the help of the **ClassPad300** we will simulate n (say n = 100) such game show situations to see, if the option of switching the door is a good or a bad option to improve the chance to win. The experiments help our students better to understand the randomness and the statistic methods of every day life.

Monty Hall was born on August 25, 1924 in Canada. Hall attended the University of Manitoba, graduating in 1945, and served in the Canadian Army during World War II. He immigrated to the United States in 1955, and for the next few years worked in radio and television for the NBC and CBS networks. In 1963, Hall began as emcee for the game show *Let's Make a Deal*, the role that would make him famous. Except for brief interruptions, *Let's Make a Deal* ran on various US television networks for the next 23 years. The show was notorious for the outrageous costumes of the contestants.
In addition to his show business fame, Monty Hall is now immortalized for the probability problem that bears his name. The controversy concerning the problem began with a column by Marilyn Vos Savant in the Sunday newspaper supplement *Parade*.

Marilyn Vos Savant has been listed in the *Guinness Book of World Records* Hall of Fame for her IQ score of 228. She is the author of the popular column "Ask Marilyn" in *Parade* magazine. Her column appears in over 350 newspapers, with a circulation of approximately 36 million, and a total readership of approximately 70 million. To students of probability, Marilyn is best known for her analysis of the Monty Hall problem, named for the television game show host Monty Hall. Over the years, she has analyzed a number of other mathematical (and probabilistic) problems as well.
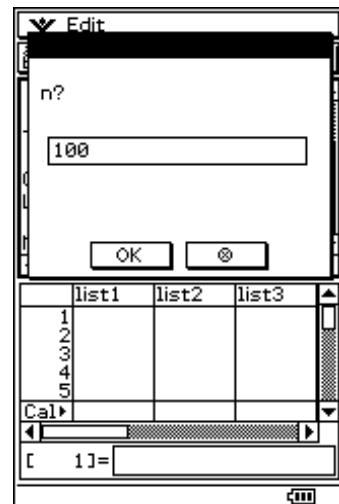
Screenshots:
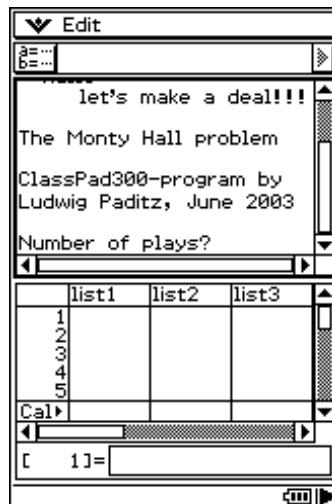
Install the "Paditz_02_06_2003.mcs" and "Paditz_02_06_2003.fls" files in the virtual Class-Pad300-emulator or type in the program-editor of your ClassPad300 the program, given at the end of this paper. Cp. http://www.informatik.htw-dresden.de/~paditz/Paditz_02_06_2003.zip
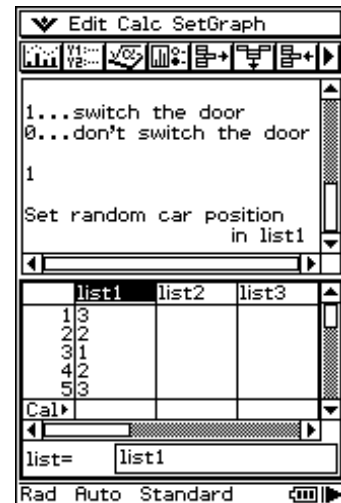
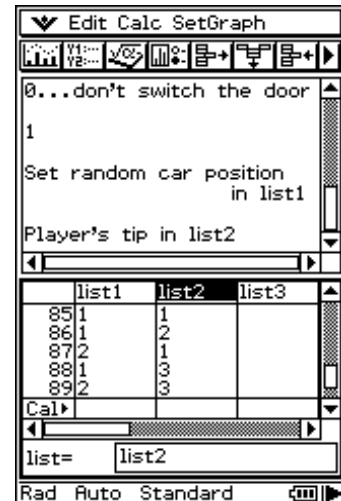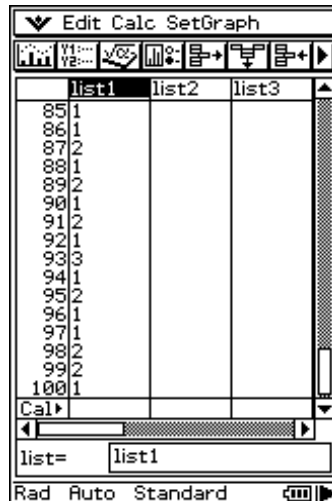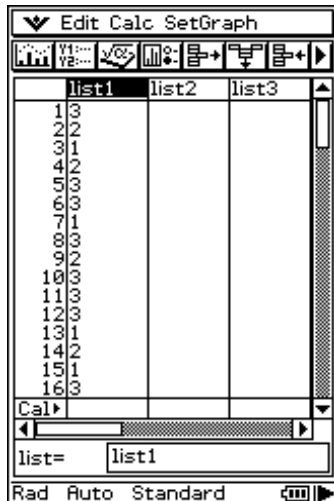Run the program "MontyHal" in the folder "main":
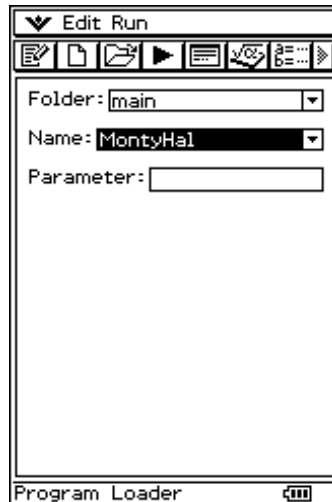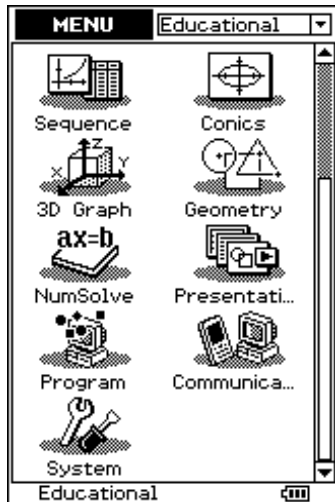
Start:

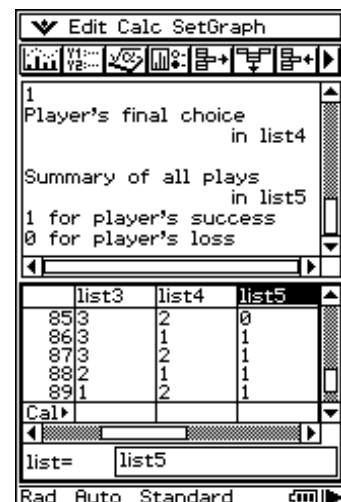Input the number of plays, you want to simulate (e.g. n=100):

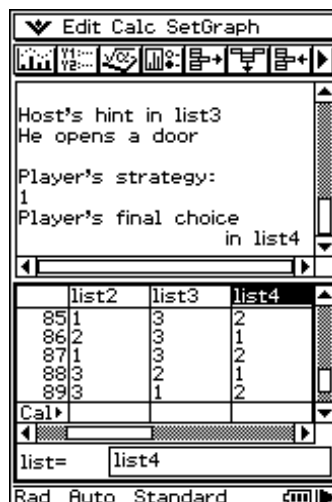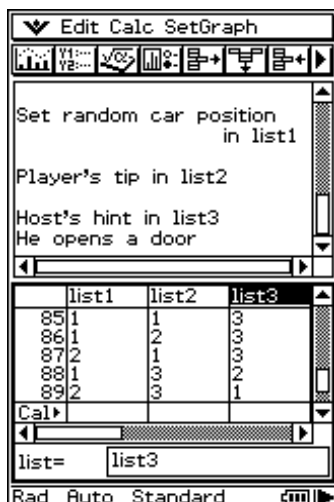Input the strategy of your plays:

Screen 1:

```
▼ Edit
Number of plays?
100

Strategy?

1...switch the door
0...don't switch the door
```

	list1	list2	list3
1			
2			
3			
4			
5			

Cal▶
[1]=

Screen 2:

```
▼ Edit
s?

1
        OK      ⊗
```

	list1	list2	list3
1			
2			
3			
4			
5			

Cal▶
[1]=

Screen 3:

```
▼ Edit Calc SetGraph
1...switch the door
0...don't switch the door

1

Set random car position
                in list1
```

	list1	list2	list3
1	3		
2	2		
3	1		
4	2		
5	3		

Cal▶
list=   list1
Rad Auto Standard

You see the list1 of simulated car-positions (in n=100 plays). The program is in pause-mode, you can consider the full list-editor-window:

Screen 4:

```
▼ Edit Calc SetGraph
```

	list1	list2	list3
1	3		
2	2		
3	1		
4	2		
5	3		
6	3		
7	1		
8	3		
9	2		
10	3		
11	3		
12	3		
13	1		
14	2		
15	1		
16	3		

Cal▶
list=   list1
Rad Auto Standard

Screen 5:

```
▼ Edit Calc SetGraph
```

	list1	list2	list3
85	1		
86	1		
87	2		
88	1		
89	2		
90	1		
91	2		
92	1		
93	3		
94	1		
95	2		
96	1		
97	1		
98	2		
99	2		
100	1		

Cal▶
list=   list1
Rad Auto Standard

Screen 6:

```
▼ Edit Calc SetGraph
0...don't switch the door

1

Set random car position
                in list1

Player's tip in list2
```

	list1	list2	list3
85	1	1	
86	1	2	
87	2	1	
88	1	3	
89	2	3	

Cal▶
list=   list2
Rad Auto Standard

In list2 you see the player's random choice, in list3 the host's hint.
list4 shows the final choice of the player and list5 shows the success or loss.

Screen 7:

```
▼ Edit Calc SetGraph
Set random car position
                in list1

Player's tip in list2

Host's hint in list3
He opens a door
```

	list1	list2	list3
85	1	1	3
86	1	2	3
87	2	1	3
88	1	3	2
89	2	3	1

Cal▶
list=   list3
Rad Auto Standard

Screen 8:

```
▼ Edit Calc SetGraph
Host's hint in list3
He opens a door

Player's strategy:
1
Player's final choice
                in list4
```

	list2	list3	list4
85	1	3	2
86	2	3	1
87	1	3	2
88	3	2	1
89	3	1	2

Cal▶
list=   list4
Rad Auto Standard

Screen 9:

```
▼ Edit Calc SetGraph
1
Player's final choice
                in list4

Summary of all plays
                in list5
1 for player's success
0 for player's loss
```

	list3	list4	list5
85	3	2	0
86	3	1	1
87	3	2	1
88	2	1	1
89	1	2	1

Cal▶
list=   list5
Rad Auto Standard

list6 contains the cumulative sums of n=100 plays. Here we have 70 times success by the help of the strategy 1 (i.e. strategy 0 gives only 30 times success in n=100 plays).

list7 to list9 are summery-lists: each digit shows what happens:
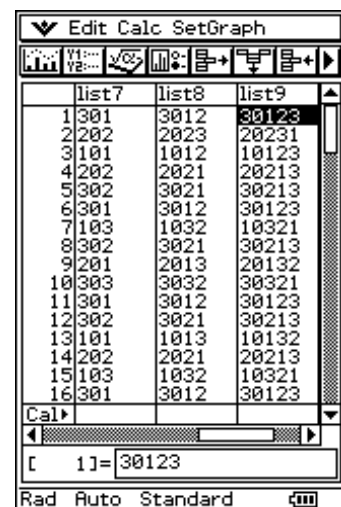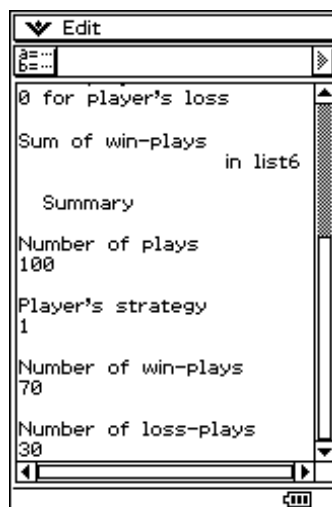
E.g. 30123 in list9 shows: car-position 3 (next digit without meaning), player's choice is position 1, the host ("showmaster") opens the door 2 and finally the player switch to door 3 and so he wins with strategy 1!

E.g. 20231 in list9 shows: car-position 2 (next digit without meaning), player's choice is position 2, the host ("showmaster") randomly opens the door 3 and finally the player switch to door 1 and so he doesn't win with strategy 1!

Program-text:

{ } ⇒ list1
{ } ⇒ list2
{ } ⇒ list3
{ } ⇒ list4
{ } ⇒ list5
{ } ⇒ list6
{ } ⇒ list7
{ } ⇒ list8
{ } ⇒ list9
DispListEditor
ClrText
SetFix 0

Print "Hallo"
Print "   let's make a deal!!!"
Print " "
Print "Hallo"
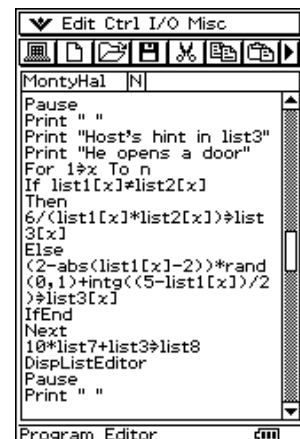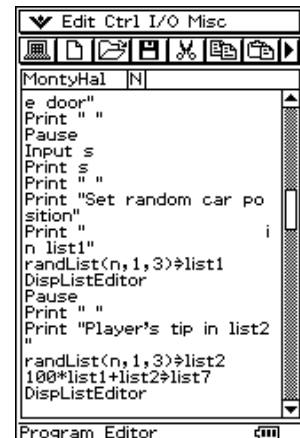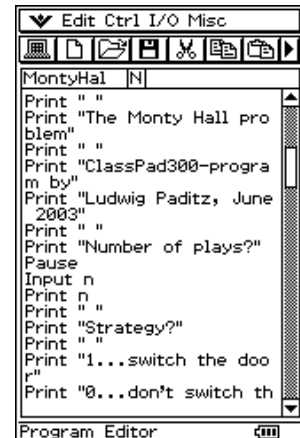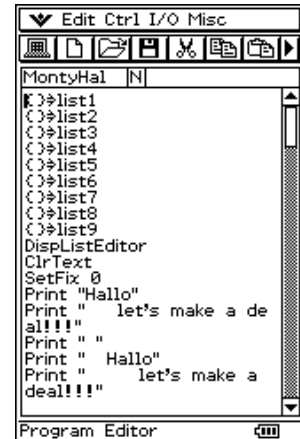Print "   let's make a deal!!!"
Print " "

Print "The Monty Hall problem"
Print " "
Print "ClassPad300-program by"
Print "Ludwig Paditz, June 2003"
Print " "

Print "Number of plays?"
Pause
Input n
Print n
Print " "
Print "Strategy?"
Print " "
Print "1…switch the door"
Print "0…don't switch the door"
Print " "
Pause
Input s
Print s
Print " "
Print "Set random car position"
Print "   in list1"
randList(n,1,3) ⇒ list1
DispListEditor
Pause

Print " "
Print "Player's tip in list2"
randList(n,1,3)) ⇒ list2
100 × list1 + list2 ⇒ list7
DispListEditor
Pause

Print " "
Print "Host's hint in list3"
Print "He opens a door"

For 1 ⇒ x To n
If list1[x] ≠ list2[x]
Then
6 / (list1[x] × list2[x]) ⇒ list3[x]
Else

(2-abs(list1[x] − 2)) × rand(0,1) + intg((5 - list1[x]) / 2) ⇒ list3[x]
IfEnd
Next
10 × list7 + list3 ⇒ list8
DisplListEditor
Pause

Print " "
Print "Player's strategy:"
Print s
Print "Player's final choice"
Print "   in list4"

If s=1
Then
6 – list2 – list3 ⇒ list4
Else
list2 ⇒ list4
IfEnd

10 × list8 + list4 ⇒ list9
DisplListEditor
Pause

Print " "
Print "Summary of all plays"
Print "   in list5"
Print "1 for player's success"
Print "0 for player's loss"

For 1 ⇒ x To n
If list4[x] = list1[x]
Then
1 ⇒ list5[x]
Else
0 ⇒ list5[x]
IfEnd
Next
DisplListEditor
Pause

Print " "
Print "Sum of win-plays"
Print "   in list6"
Cuml(list5) ⇒ list6
DisplListEditor
Pause

Print " "
Print "   Summary"
Print " "
Print "Number of plays"
Print n
Print " "
Print "Player's strategy"
Print s
Print " "
Print "Number of win-plays"
Print list6[n]
Print " "
Print "Number of loss-plays"
Print n-list6[n]

Stop


```
MontyHal  N
Print "Player's strategy:"
Print s
Print "Player's final choic
e"
Print "                   i
n list4"
If s=1
Then
6-list2-list3⇒list4
Else
list2⇒list4
IfEnd
10*list8+list4⇒list9
DisplListEditor
Pause
Print " "
Print "Summary of all pla
ys"
Print "                   i
Program Editor
```



```
MontyHal  N
n list5"
Print "1 for player's succ
ess"
Print "0 for player's loss
"
For 1⇒x To n
If list4[x]=list1[x]
Then
1⇒list5[x]
Else
0⇒list5[x]
IfEnd
Next
DisplListEditor
Pause
Print " "
Print "Sum of win-plays"
Print "                   i
n list6"
Program Editor
```



```
MontyHal  N
cuml(list5)⇒list6
DisplListEditor
Pause
Print " "
Print "  Summary"
Print " "
Print "Number of plays"
Print n
Print " "
Print "Player's strategy"
Print s
Print " "
Print "Number of win-play
s"
Print list6[n]
Print " "
Print "Number of loss-pla
ys"
Print n-list6[n]
Program Editor
```



```
MontyHal  N
Print "  Summary"
Print " "
Print "Number of plays"
Print n
Print " "
Print "Player's strategy"
Print s
Print " "
Print "Number of win-play
s"
Print list6[n]
Print " "
Print "Number of loss-pla
ys"
Print n-list6[n]
Stop
Program Editor
```

The Monty Hall problem and the probability theory:

events: A_i = { the car is behind door i }, i = 1, 2, 3 .

M_j = { the host opens the door j }, j = 1, 2, 3 .

probabilities: $P(A_i)$ = 1/3 for all i

The player chooses the door 1 (without loss of generality)

Conditioned probabilities:

$P(M_2 / A_1) = 1 / 2$ $P(M_3 / A_1) = 1 / 2$

$P(M_2 / A_2) = 0$ $P(M_3 / A_2) = 1$

$P(M_2 / A_3) = 1$ $P(M_3 / A_3) = 0$

Formula of total probability:

$P(M_2) = P(M_2 / A_1) \times P(A_1) + P(M_2 / A_2) \times P(A_2) + P(M_2 / A_3) \times P(A_3)$
   $= 1 / 6 + 0 + 1 / 3 = 1 / 2 ,$

analogous

$P(M_3) = P(M_3 / A_1) \times P(A_1) + P(M_3 / A_2) \times P(A_2) + P(M_3 / A_3) \times P(A_3)$
   $= 1 / 6 + 1 / 3 + 0 = 1 / 2 .$

Bayes' formula:

$P(A_1 / M_2) = P(M_2 / A_1) \times P(A_1) / P(M_2) = (1 / 2) \times (1 / 3) / (1 / 2) = 1 / 3,$

$P(A_3 / M_2) = P(M_2 / A_3) \times P(A_3) / P(M_2) =$ $(1) \times (1 / 3) / (1 / 2) = 2 / 3,$

$P(A_2 / M_2) = 0.$

The mathematical background of the simulation:

RandList(n,1,3) generates a list of n discrete uniformly distributed random numbers between 1 and 3: the random position of the car (stored in list1).

Now again, **RandList(n,1,3)** generates a independent list of n discrete uniformly distributed random numbers between 1 and 3: the random choice of the player (stored in list2).

If **a := list1[x] not equal k := list2[x]** then the host opens the door with the number

$$m := f(a,k) = 6 / (a * k) = 6/(list1[x]*list2[x])$$

because in every case list1[x]*list2[x]*list3[x]=6 (6 = 1 * 2 * 3).

If **a := list1[x] equal k := list2[x]** then the host randomly opens one of the other doors with the number

$$m := f(a,k) = (2 - |a - 2|) * rand(0,1) + intg((5 - a) / 2)$$

$$= (2 - abs(list1[x] - 2)) * rand(0,1) + intg((5 - list1[x]) / 2) ,$$

cp. the followig table:

car-door a list1	player's door k list2	m = f(a, k) host's door list3	w (s = 0) player's strategy list4	w (s = 1) player's strategy list4
1	1	2 v 3	1	3 v 2
1	2	3	2	1
1	3	2	3	1
2	1	3	1	2
2	2	1 v 3	2	3 v 1
2	3	1	3	2
3	1	2	1	3
3	2	1	2	3
3	3	1 v 2	3	2 v 1

Contact: paditz@informatik.htw-dresden.de

References:

[1] Fryska, M.: **Bee or Car?** workshop 2000, Amman, Jordan, Proceedings of the Internat. Conf. on Mathematics Education into the 21st Century: Mathematics For Living cp. http://math.unipa.it/~grim/Jworkshopsabstract

[2] Paditz, L.: **Zwei Ziegen und ein Auto** (german program for CASIO FX 2.0PLUS), 2002, cp. http://www.informatik.htw-dresden.de/~paditz/Casio_Ziegen.pdf

[3] Paditz, L.: Simulation and Statistical Exploration of Data (Let's Make a Deal – The Monty Hall Problem), using the new ClassPad300-technology, 2003, Abstract, cp. http://www.informatik.htw-dresden.de/~paditz/Abstract_Paditz2003.pdf