Part IV

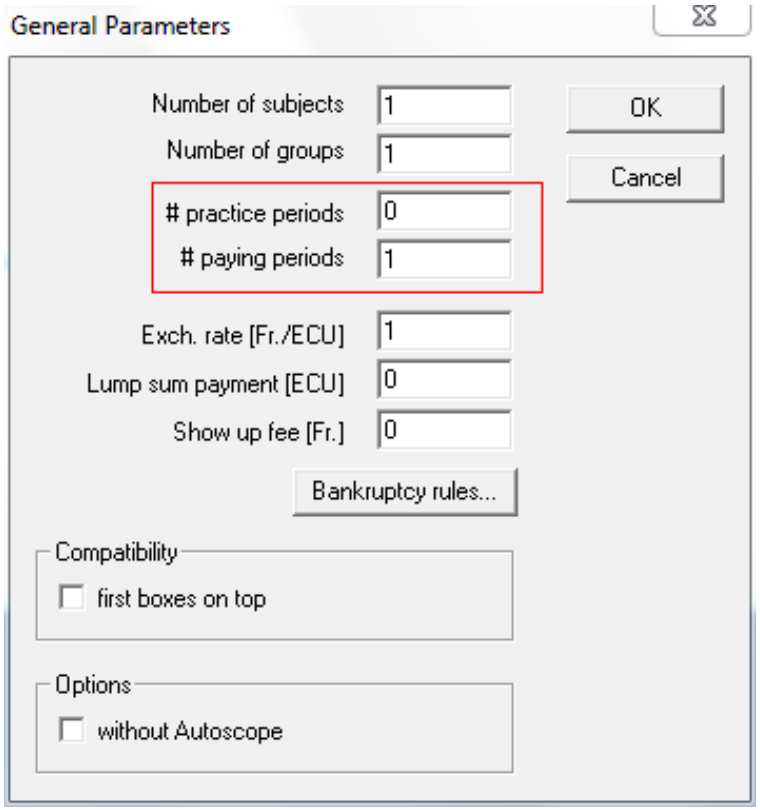
Repeated Two-player Games

Repeated Two-player Games

Symmetric games with finite horizon

Increase the number of periods:

- double-click on Background
- set the number of periods and trial periods



See prisoner_dilemma_2.ztt

Matching in repeated games - I

When you have more than 1 group and multiple periods, there are several possible matching procedures.

Option 1: from the menu, choose `Treatment` → `Matching`:

partner	groups remain the same throughout the whole treatment
stranger	groups are randomly formed at the beginning of each period
absolute stranger	players never meet more than once in the treatment (not always possible)

Matching in repeated games - I

When you have more than 1 group and multiple periods, there are several possible matching procedures.

Option 1: from the menu, choose `Treatment` → `Matching`:

- partner: groups remain the same throughout the whole treatment
- stranger: groups are randomly formed at the beginning of each period
- absolute stranger: players never meet more than once in the treatment (not always possible)

To **check the matching**, from the menu select `Treatment`→ `Parameter table`.

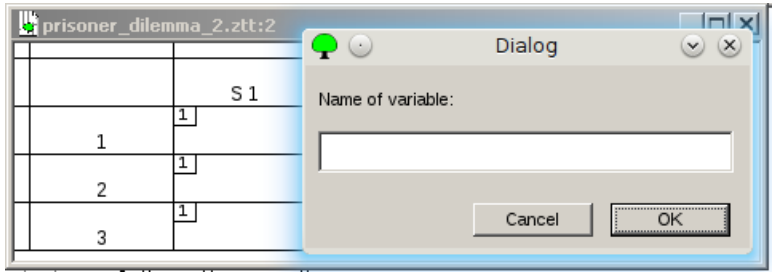
Matching in repeated games - II

Option 2: load the matching from an external table.

1. With MS Excel (or similar) create a tab-separated .txt file ("save as" → "other formats").

	A	B	C	D
1	1	1	2	2
2	1	2	1	2
3	1	2	2	1

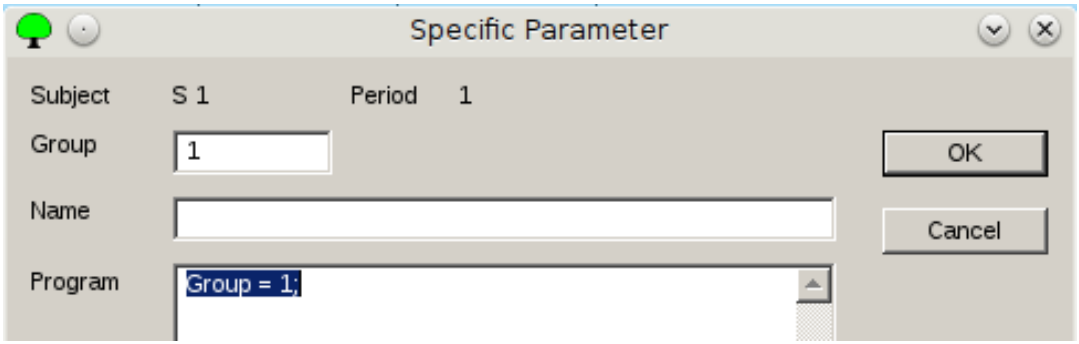
2. Then, from z-tree, open the `Parameter Table`
3. Select `Treatment` → `Import Variable Table`.
4. Set `Name of variable = Group`.



5. Press OK and select the matching table (tab separated .txt file).

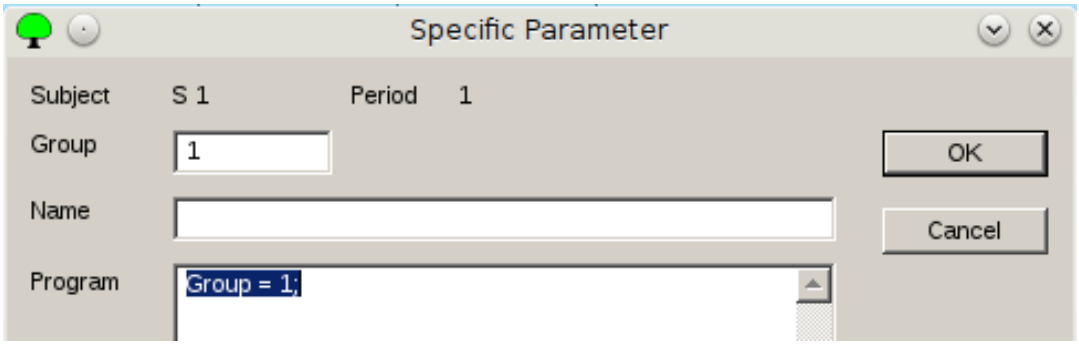
Matching in repeated games - III

If you double-click on one of the cells of the parameter table, you'll see a program defining the `Group` variable, for a specific subject in a given period.



Matching in repeated games - III

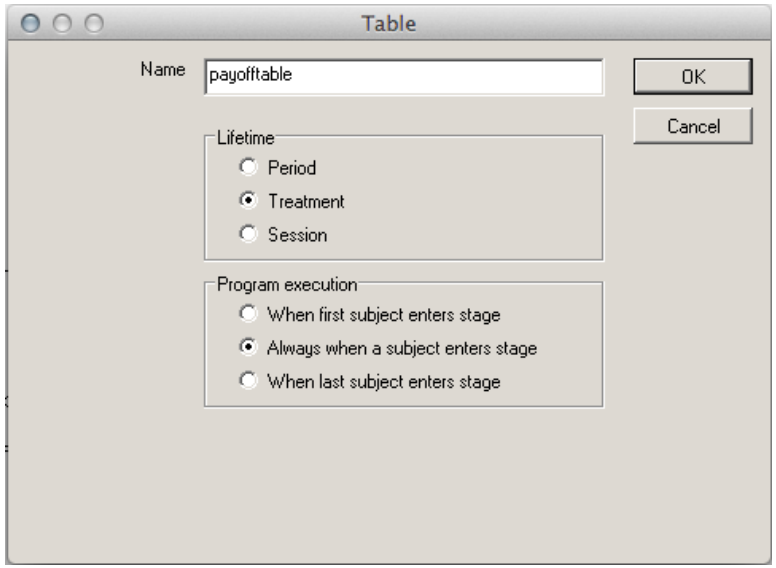
If you double-click on one of the cells of the parameter table, you'll see a program defining the `Group` variable, for a specific subject in a given period.



Programs in the parameter table are executed *after* those in the background.

Changing parameters across periods -1

1. in z-Tree, select the `Background` and choose `Treatment`$\rightarrow$ `New table`.

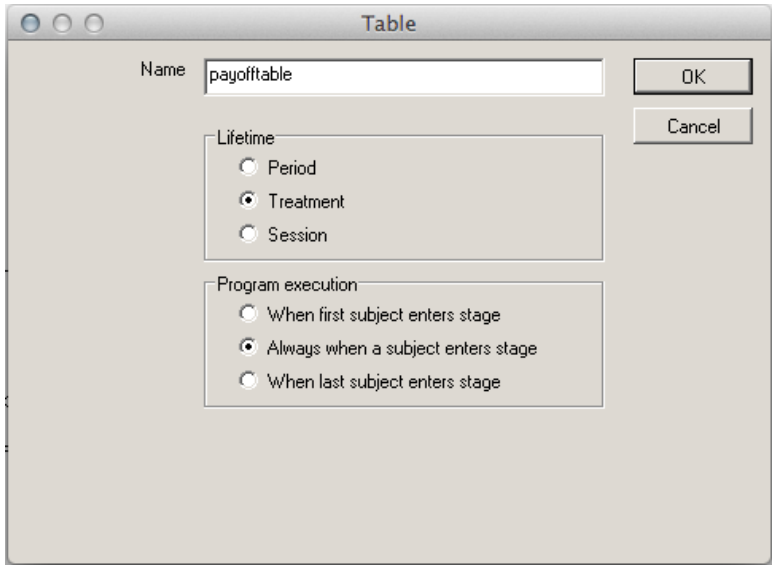


2. create a new table (with Excel, or similar programs)

	A	B	C	D	E	F
1	payofftable	Period	r	t	s	p
2	payofftable	1	3	5	0	1
3	payofftable	2	3	6	0	1
4	payofftable	3	4	5	0	1

Changing parameters across periods -1

1. in z-Tree, select the `Background` and choose `Treatment`$\rightarrow$ `New table`.



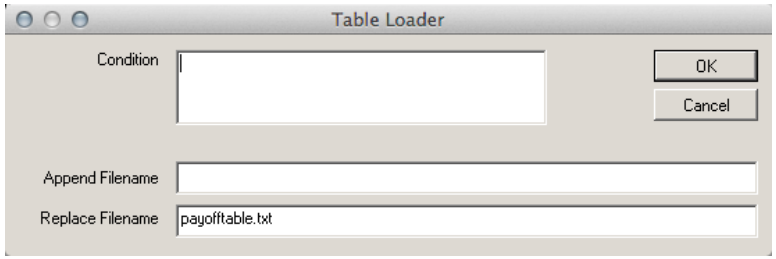
2. create a new table (with Excel, or similar programs)

	A	B	C	D	E	F
1	payofftable	Period	r	t	s	p
2	payofftable	1	3	5	0	1
3	payofftable	2	3	6	0	1
4	payofftable	3	4	5	0	1

3. save the table as a *tab delimited* .txt file
4. the first column reports the name of the table you want to load.

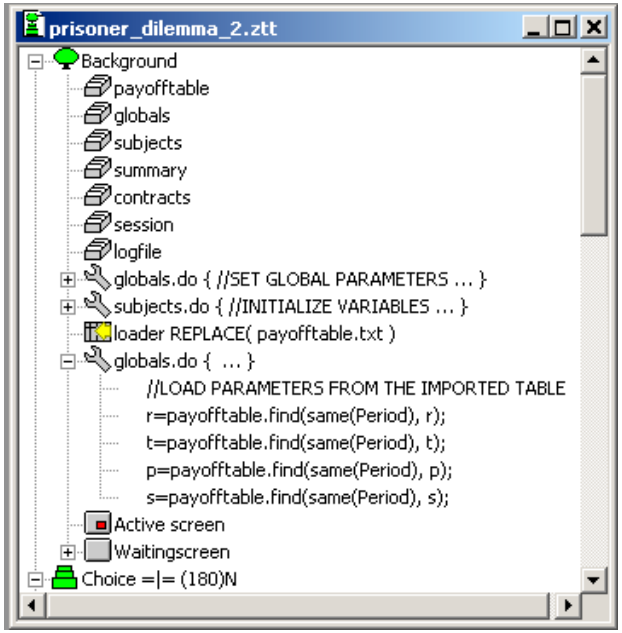
Changing parameters across periods - 2

5. in z-Tree, select the `Background` and choose `Treatment`$\rightarrow$ `New table loader`.



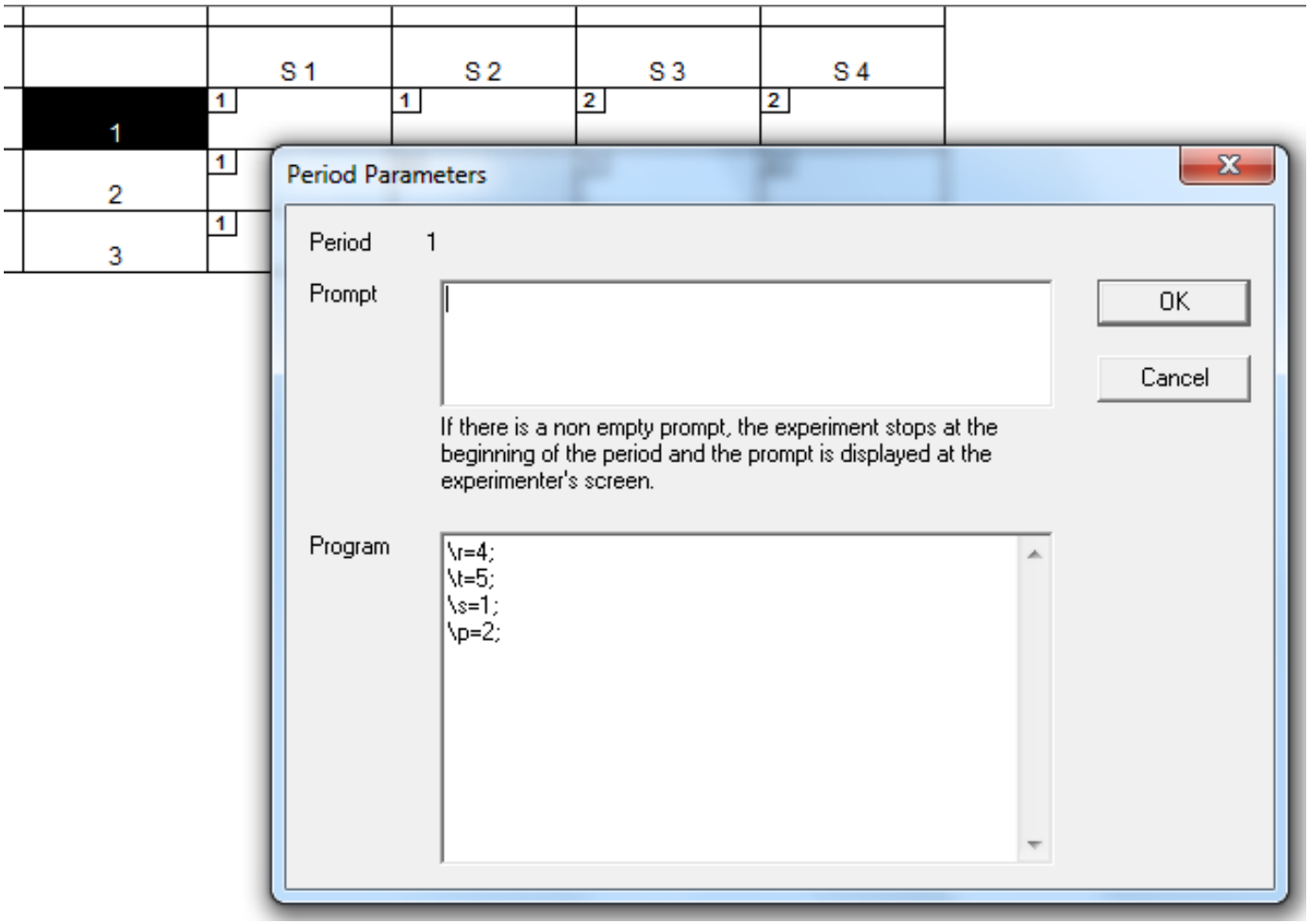
6. Append/Replace filename: name of your .txt file

7. in the Background, after the `Table Loader` create a new `Program`, to load the new parameters.



Changing parameters across periods - 3

An alternative possibility is setting parameters manually, adding programs in the `Parameter Table`.



Grid boxes and Container boxes - I

Now the payoff matrix of the game changes every period. You might want to show it on the subjects' screens.

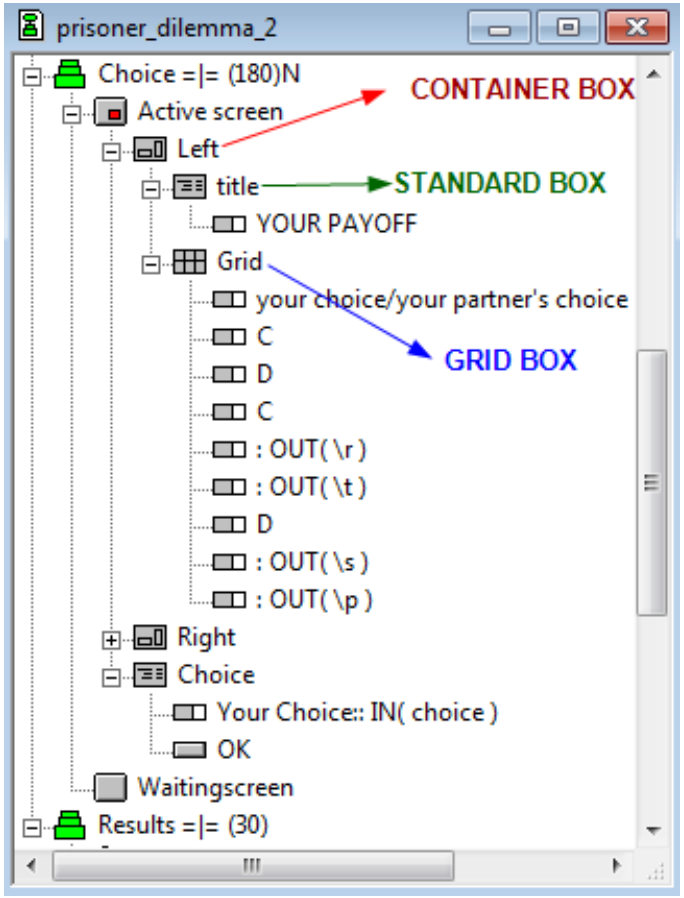
YOUR PAYOFF		
your choice/your partner's choice	C	D
C	3	0
D	5	1

Your Choice: D C

OK

YOUR PARTNER'S PAYOFF		
your choice/your partner's choice	C	D
C	3	5
D	0	1

Grid boxes and Container boxes - II



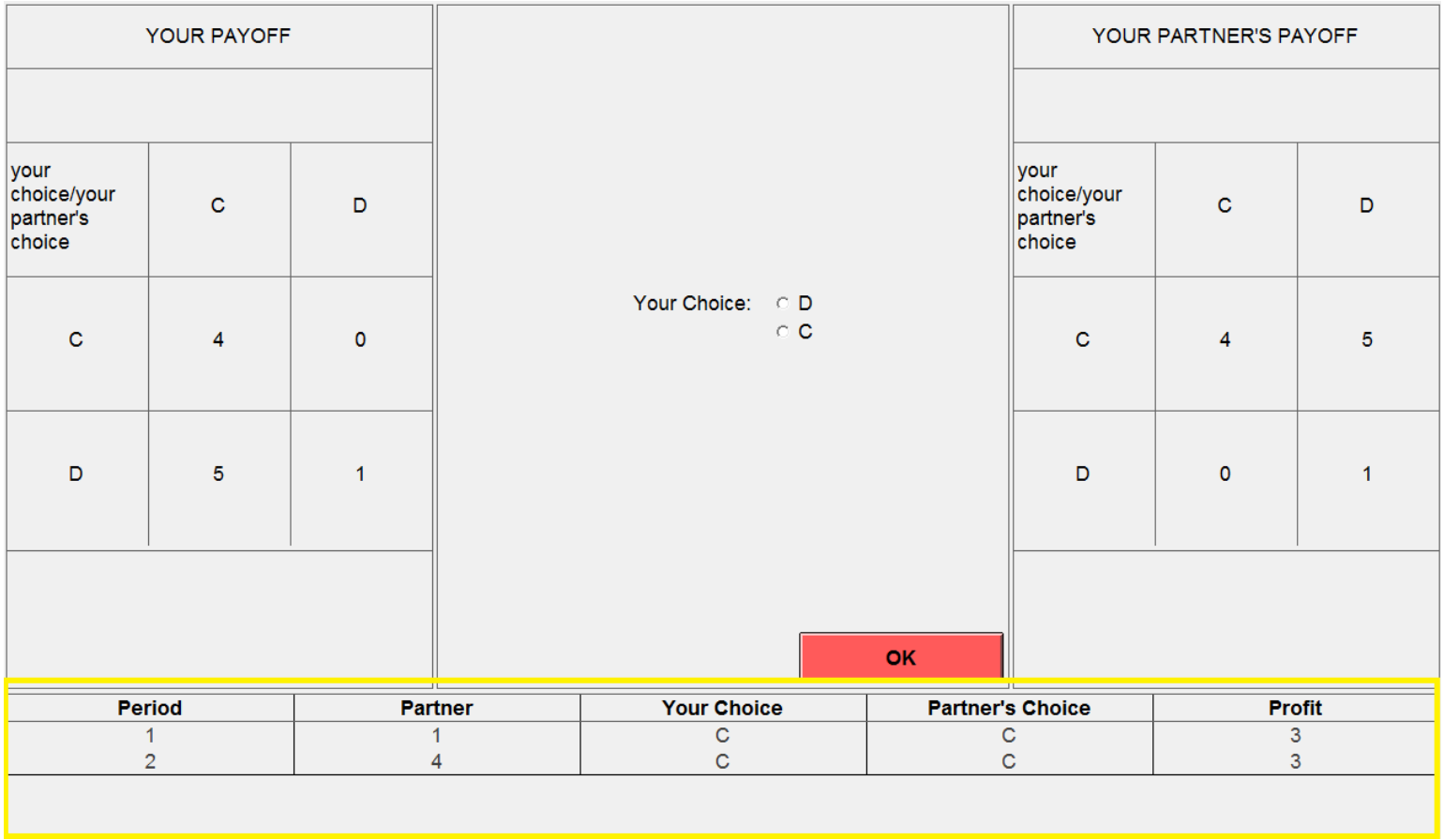
A `container box` is a box that can contain other boxes.

Note: the *relative measures* of the boxes it contains are defined *w.r.t. the container box*, not to the whole screen.

A `grid box` presents items in a tabular form.

History box - I

To remind subjects about what happened in past periods, you can insert a `history box`.



YOUR PAYOFF

your choice/your partner's choice	C	D
C	4	0
D	5	1

Your Choice: D C

OK

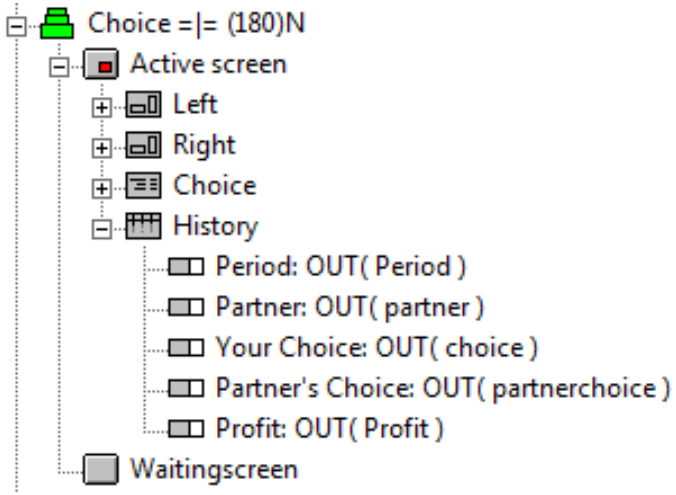
YOUR PARTNER'S PAYOFF

your choice/your partner's choice	C	D
C	4	5
D	0	1

Period	Partner	Your Choice	Partner's Choice	Profit
1	1	C	C	3
2	4	C	C	3

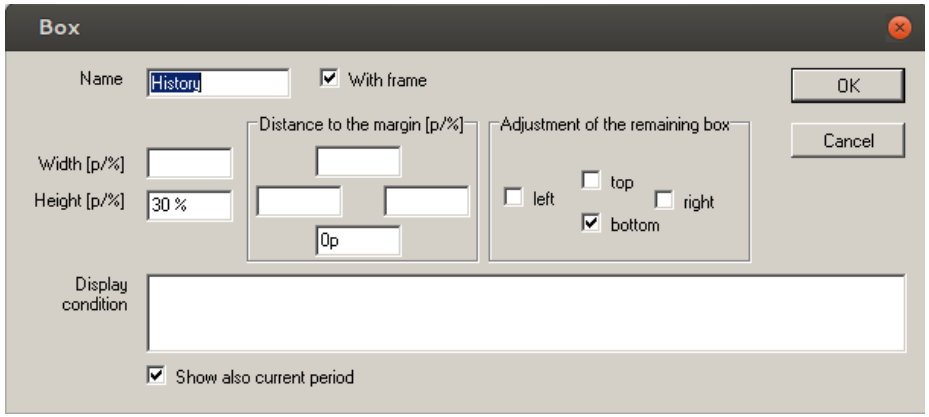
History box - II

see the Reference Manual, page 18.



- lists results from previous periods
- only takes variables form the `subjects` table
- a label row contains the labels

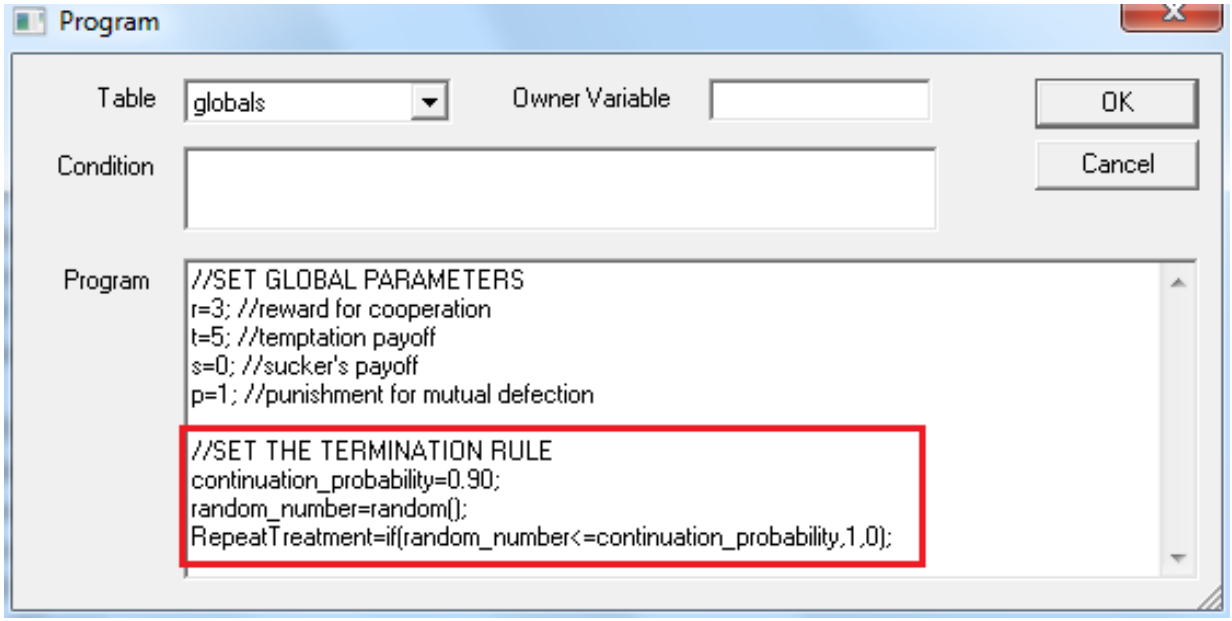
- if the table is too long, a scroll-bar appears



- Option: showing/not showing the current period.

Indefinite number of periods

1. Define a treatment with a single period.
2. In the `Background`, create a new program which runs on the `globals` table. In this program, you can set the variable `RepeatTreatment`, as follows:



Random matching in programs - I

Hard to use the `parameter table` or to import an external matching table.

$\Rightarrow$ Matching **in a program** in the `Background`, running on the `globals` table (it runs only once, at the beginning of each period).

- generate a different random number (rand) for each subject;
- generate the variable rank, which sorts subjects according to the variable rand;
- group together subject having consecutive ranks.

subjects table

Period	Subject	Group	Profit	TotalProfit	Participate	i	rand	rank
1	1	2	0	0	1	2	0.3314207	3
1	2	2	0	0	1	2	0.6126384	4
1	3	1	0	0	1	2	0.2980133	2
1	4	1	0	0	1	2	0.1228111	1

Random matching in programs - II

```
//CREATE GROUPS
i=1;
repeat{
  i=i+1;
  subjects.do{
    rand=random();
  }
  subjects.do{
    rank=subjects.count(rand<=:rand);
  }
}
while(subjects.sum(Subject)!=subjects.sum(rank)&i<10);
subjects.do{
  Group=roundup(rank/2,1);
}
```

Example: random_matching_program.ztt

do statement and loops

`do` **statement**:
see page 62 of the Tutorial

With `do{commands}` the commands are executed *for all records* in the current table.

For example, with `subjects.do{commands}`
we specify that the commands should be executed for all records in the `subjects` table.

do statement and loops

`do` **statement**:
see page 62 of the Tutorial

With `do{commands}` the commands are executed *for all records* in the current table.

For example, with `subjects.do{commands}`
we specify that the commands should be executed for all records in the `subjects` table.

`repeat{commands} while (condition)` **statement**

The commands are executed. *Then* it is checked whether the condition is TRUE, and as long as it is the commands are repeated.

do statement and loops

`do` **statement**:
see page 62 of the Tutorial

With `do{commands}` the commands are executed *for all records* in the current table.

For example, with `subjects.do{commands}`
we specify that the commands should be executed for all records in the `subjects` table.

`repeat{commands} while (condition)` **statement**

The commands are executed. *Then* it is checked whether the condition is TRUE, and as long as it is the commands are repeated.

Loops can be left with the key combination Ctrl+Alt+F5

Different games in different periods I

When you don't know in advance the length of a treatment, you can:

1. **randomize**:

```
//LOAD PARAMETERS FROM THE IMPORTED TABLE
n=max(1,roundup(3*random(),1));
//random number equal to 1, 2 or 3.
r=payofftable.find(Period==\n, r);
t=payofftable.find(Period==\n, t);
p=payofftable.find(Period==\n, p);
s=payofftable.find(Period==\n, s);
```

Different games in different periods II

2. **rotate:**

```
//LOAD PARAMETERS FROM THE IMPORTED TABLE
n=if(mod(Period,3)==0,3,mod(Period,3));
//number taking value 1, 2 or 3, in turn.
r=payofftable.find(Period== \n, r);
...
```

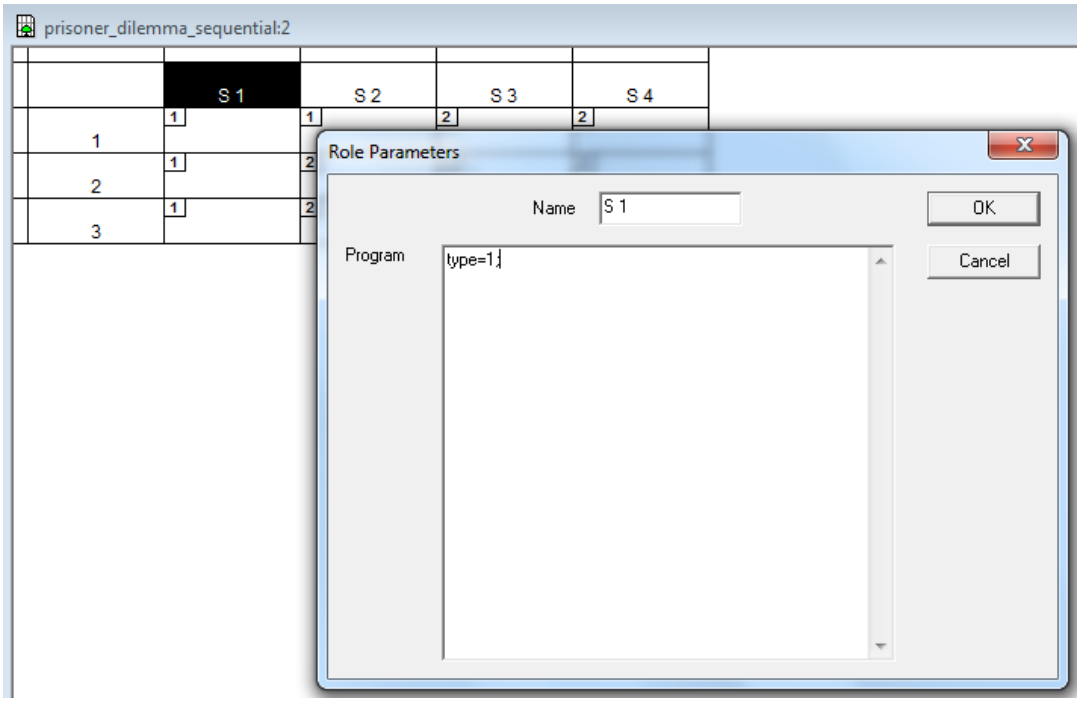
Note: setting parameters in the Parameter Table is less flexible.

Games with asymmetric players - I

Suppose now you want to let subjects play the prisoner dilemma *sequentially*.

2 types of players: 1 (first mover) and 2 (second mover)$\Rightarrow$ `type` is a subject variable.

If types remain fixed across periods, you can set them in the `Parameter Table`.



Games with asymmetric players - II

You can also assign types with a **program** in the `Background` (on the `globals` table).

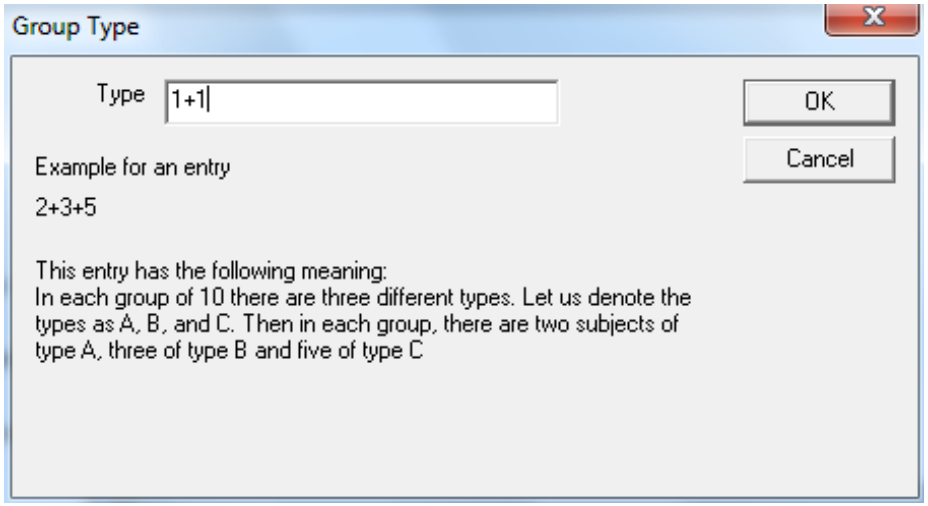
```
numsubjects=subjects.maximum(Subject);
subjects.do{
  type=if(Subject<=\numsubjects/2,1,2);
}
```

Games with asymmetric players - II

You can also assign types with a **program** in the `Background` (on the `globals` table).

```
numsubjects=subjects.maximum(Subject);
subjects.do{
  type=if(Subject<=\numsubjects/2,1,2);
}
```

With types, you can use the `Absolute typed strangers` matching procedure (*Ref.Man., page 32*). From the menu, select `Treatment` → `Matching`

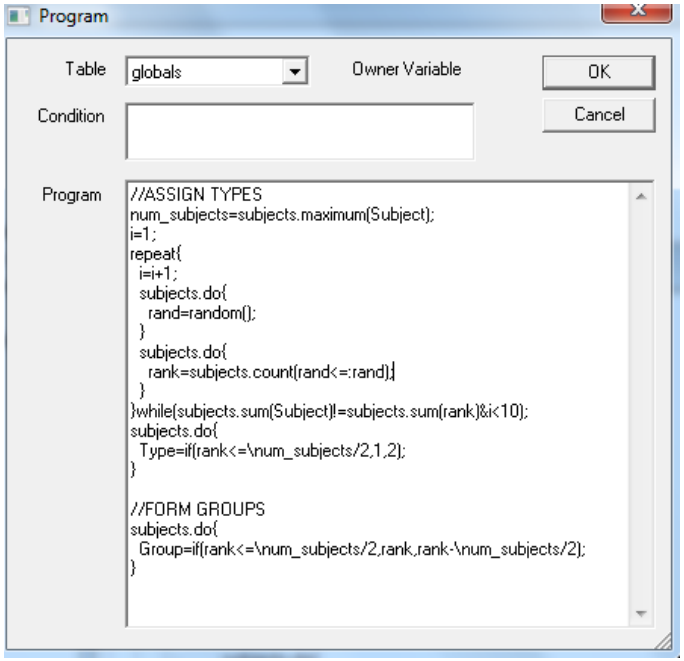


Assign types and groups randomly

See assign_types.ztt

It is possible to set types randomly at the beginning of each period, with a program in the `Background`.

1. rank subjects according to some random variable
2. set `type=1` for the first half of the subjects, `type=2` for the others
3. form groups picking the first subject from the first half, and the second subject from the second half.

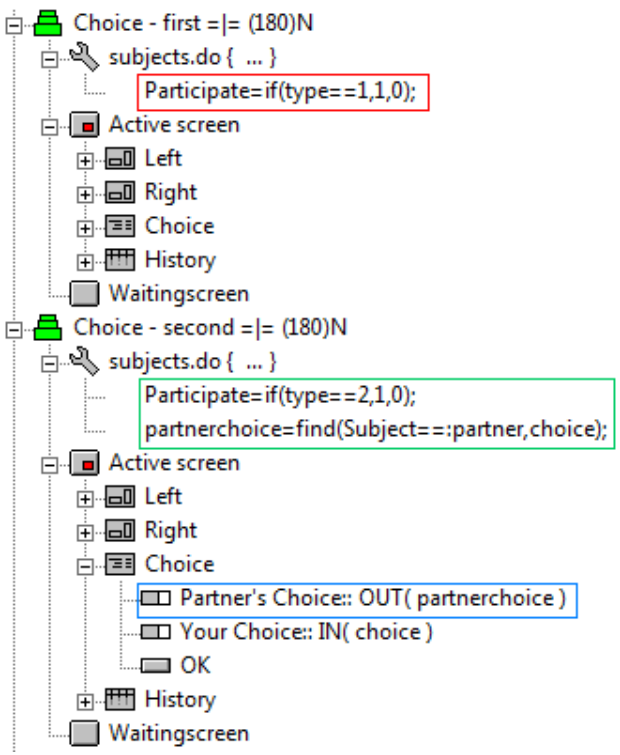


subjects table

Period	Subject	Group	Profit	TotalProfit	Participate	rand	rank	Type
3	1	1	0	0	1	0.7331797	3	2
3	2	1	0	0	1	0.2467885	1	1
3	3	2	0	0	1	0.4047647	2	1
3	4	2	0	0	1	0.9164408	4	2

The Participate variable

Once you have defined the types, you can modify the structure of the program. The `Participate` variable in the `subjects table` determines whether a subjects enters a stage or not.



- If this variable has value 1 then the subjects enters the stage, i.e., the corresponding `Active Screen` appears on the subjects computer screen.
- Programs in the stage are executed *for all* subjects.

See prisoner_dilemma_sequential.ztt

Program execution

At the beginning of each period, **programs** are executed in the following **order**:

1. Standard variables (Subject, Period, etc.) are set.
2. Programs in `Background`, in the order they appear in the .ztt file

Program execution

At the beginning of each period, **programs** are executed in the following **order**:

1. Standard variables (Subject, Period, etc.) are set.
2. Programs in `Background`, in the order they appear in the .ztt file
3. Programs in the `Parameter Table`

cells	Subject programs (in current period) in the subjects table
top row	Role program in subjects table
first column	Period program in globals table

Program execution

At the beginning of each period, **programs** are executed in the following **order**:

1. Standard variables (Subject, Period, etc.) are set.
2. Programs in `Background`, in the order they appear in the .ztt file
3. Programs in the `Parameter Table`

cells	Subject programs (in current period) in the subjects table
top row	Role program in subjects table
first column	Period program in globals table

4. Programs of the first stage

Exercise - battle of the sexes

The "battle of the sexes" is a two-player asymmetric game.

Rob (row player) and Clara (column player) want to go out together tonight. She prefers the box, he is a fan of ballet.

	box	**ballet**
box	3,5	0,0
ballet	0,0	5,3

Exercise - battle of the sexes

The "battle of the sexes" is a two-player asymmetric game.

Rob (row player) and Clara (column player) want to go out together tonight. She prefers the box, he is a fan of ballet.

	box	**ballet**
box	3,5	0,0
ballet	0,0	5,3

They will go out together every Tuesday, as long as their engagement lasts.

	cooperate	**defect**
cooperate	m, hc	l, l
defect	l, l	hr, m

Exercise - details - I

- **Indefinitely repeated** game. There is a 10% probability that Rob and Clara break up before next Tuesday.
- **Partner matching** (couples remain the same throughout the treatment). Types are also fixed across periods.
- $l = 0$, $m = 3$, while *hc* **and** *hr* **parameters vary** randomly across periods:
  - sometimes the box match is particularly important: hc=8, hr=5;
  - sometimes the ballet company is very good: hc=5, hr=8;
  - otherwise hc=5, hr=5;

Exercise - details - II

- Simultaneous moves.
- Show a `history box` and two `grid boxes` to display the payoff matrices.
- You can use the file prisoner_dilemma_sequential.ztt as a starting point.

Exercise - screen-shot

YOUR PAYOFF

your choice/Clara's choice	box	ballet
box	3	0
ballet	0	5

Hi Rob, make your choice:
- ballet
- box

OK

CLARA'S PAYOFF

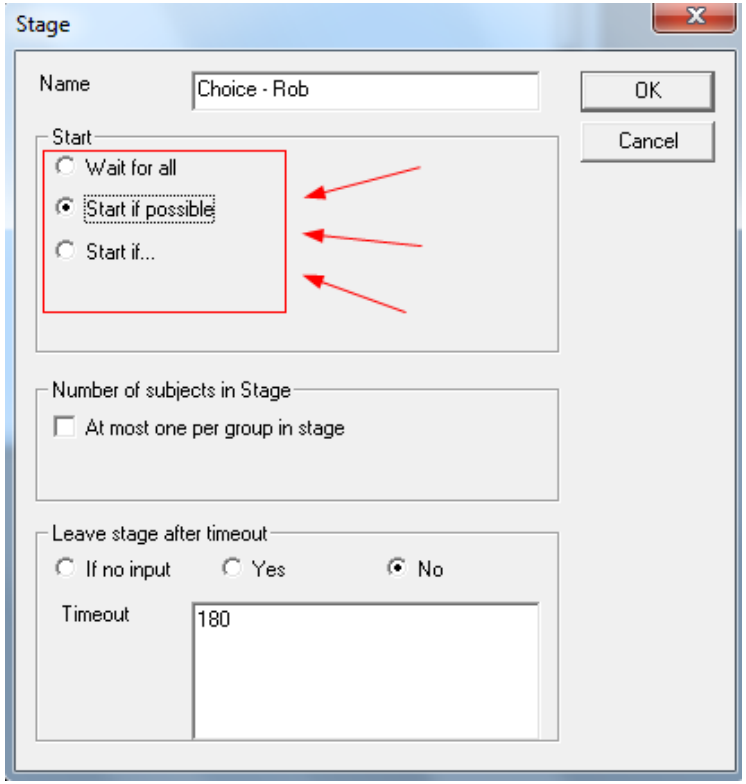
your choice/Claras's choice	box	D
box	8	0
ballet	0	3

Period	Partner	Your Choice	Clara's Choice	Profit
1	4	ballet	ballet	8
2	4	ballet	ballet	8
3	4	ballet	box	0
4	4	box	box	3

Exercise - hint

Create 3 stages:

1. Rob's choice stage
2. Clara's choice stage
3. Results



To let the two players play simultaneously, set Rob's choice stage and Clara's choice stage to "`start if possible`".

Solution: battle_of_sexes.ztt