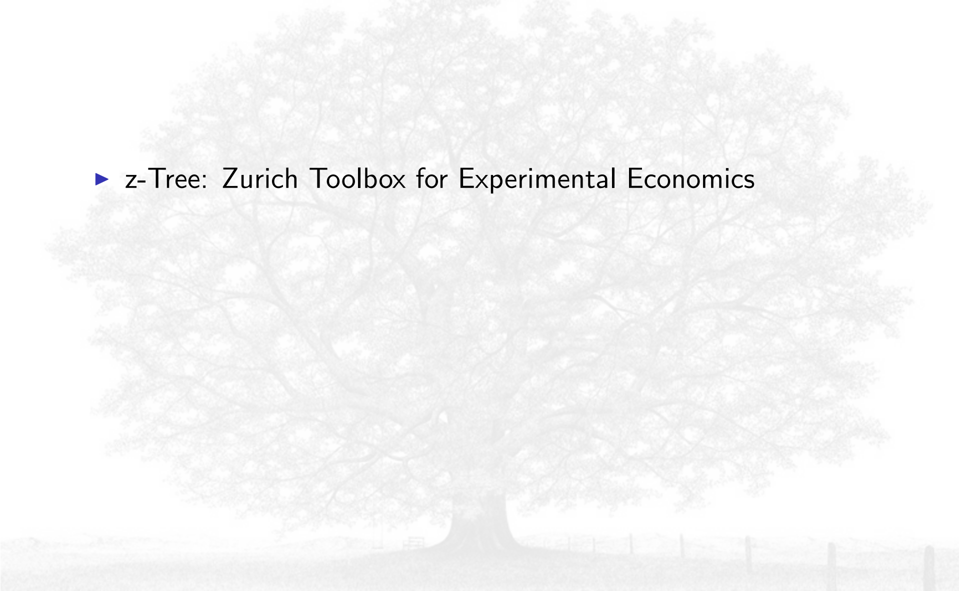
z-Tree Programming: Part I

Ali Seyhun Saral

University of Trento & CEEL

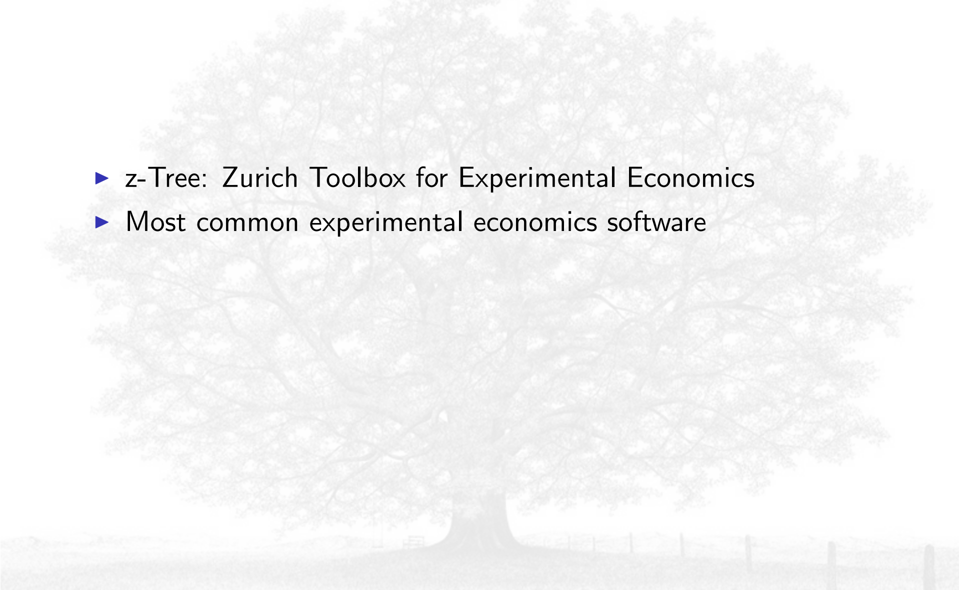
19.12.2016

What is z-Tree



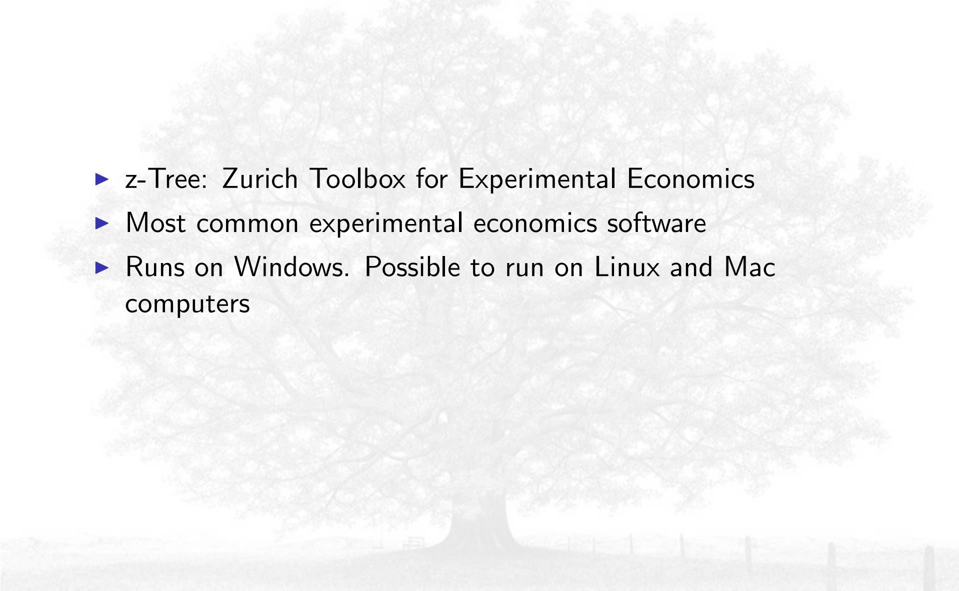
- z-Tree: Zurich Toolbox for Experimental Economics

What is z-Tree



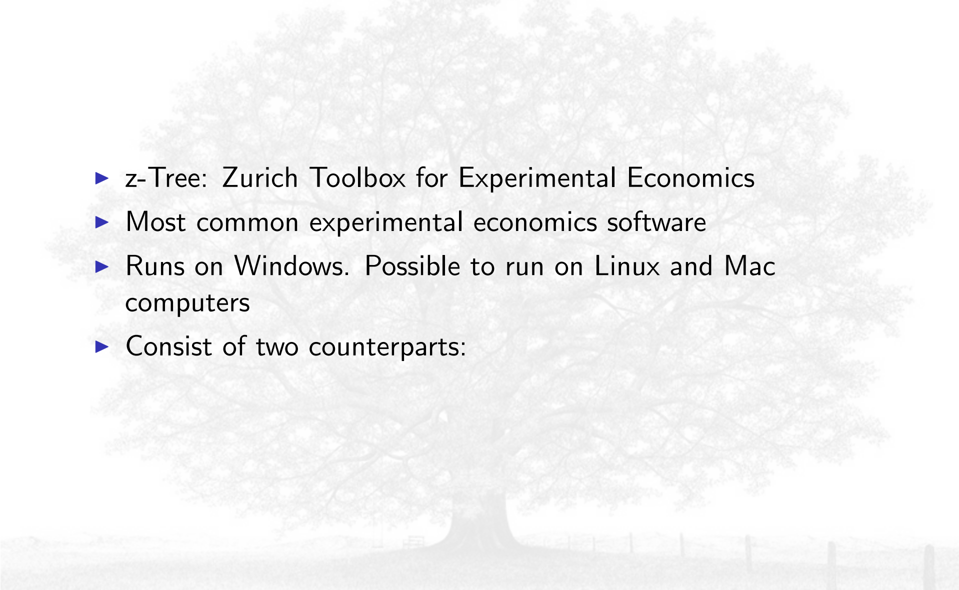
- z-Tree: Zurich Toolbox for Experimental Economics
- Most common experimental economics software

What is z-Tree



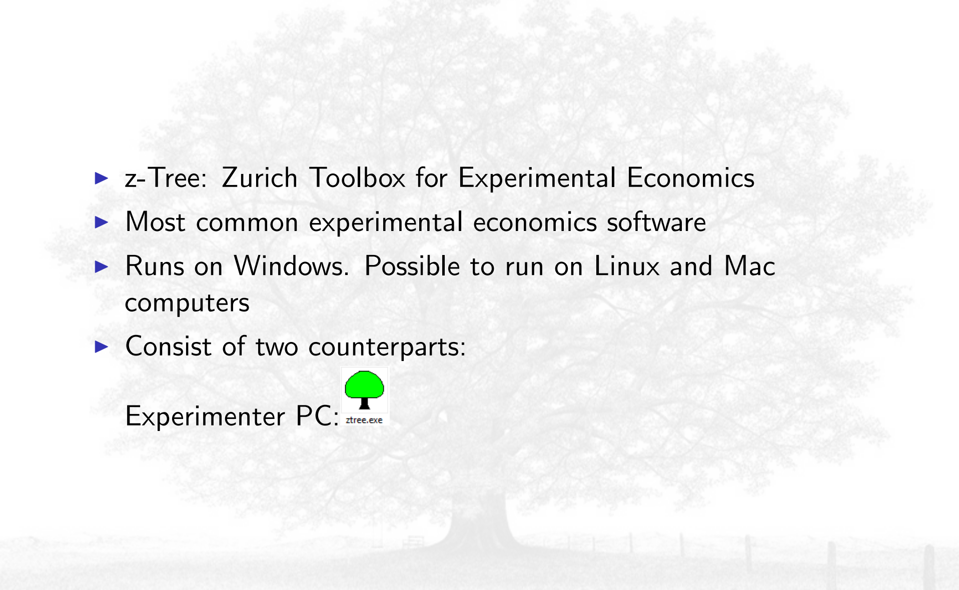
- z-Tree: Zurich Toolbox for Experimental Economics
- Most common experimental economics software
- Runs on Windows. Possible to run on Linux and Mac computers

What is z-Tree



- z-Tree: Zurich Toolbox for Experimental Economics
- Most common experimental economics software
- Runs on Windows. Possible to run on Linux and Mac computers
- Consist of two counterparts:

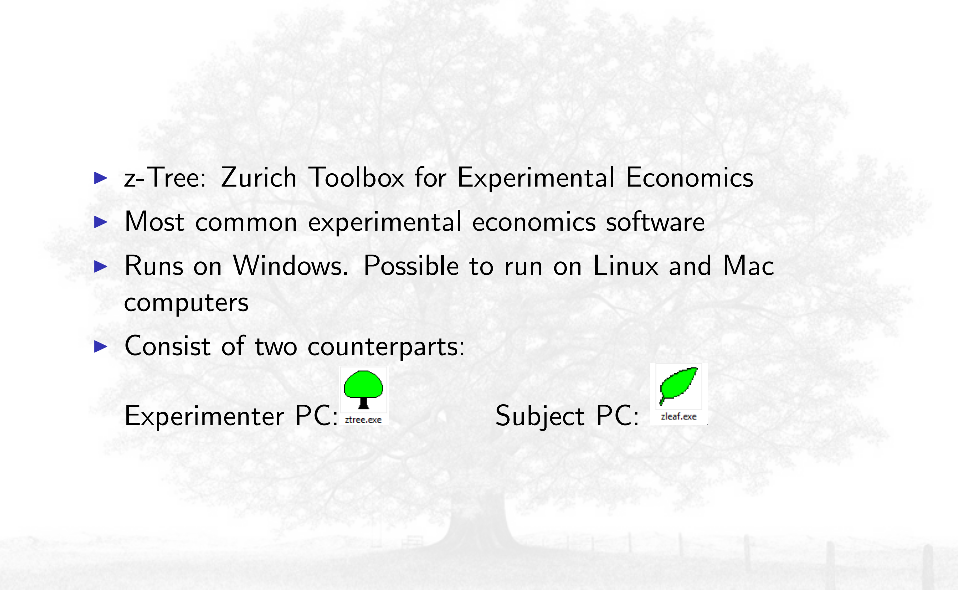
What is z-Tree



- z-Tree: Zurich Toolbox for Experimental Economics
- Most common experimental economics software
- Runs on Windows. Possible to run on Linux and Mac computers
- Consist of two counterparts:

  Experimenter PC: ztree.exe

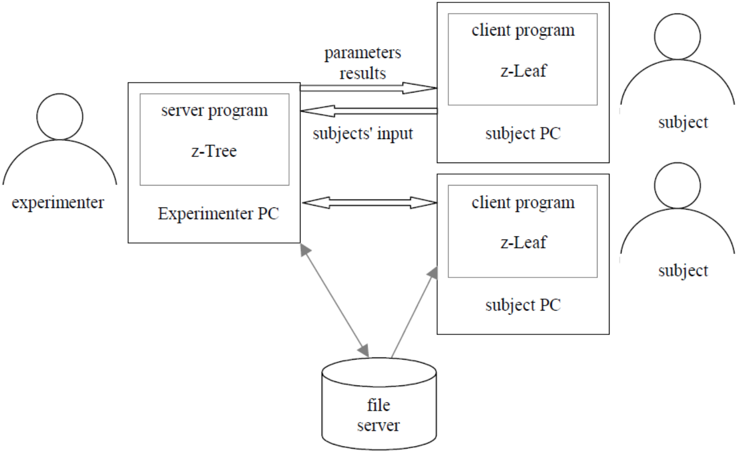
What is z-Tree



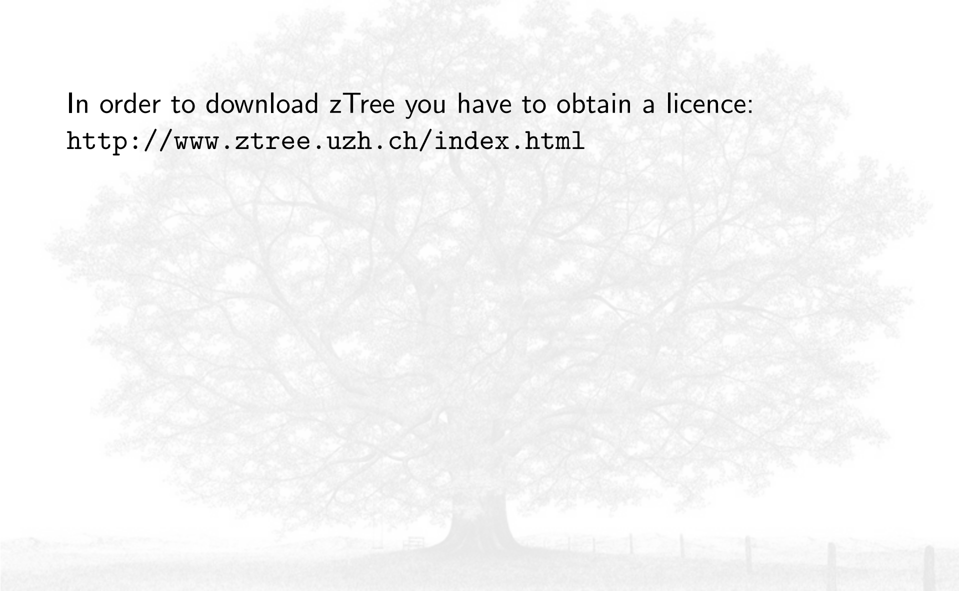
- z-Tree: Zurich Toolbox for Experimental Economics
- Most common experimental economics software
- Runs on Windows. Possible to run on Linux and Mac computers
- Consist of two counterparts:

  Experimenter PC: ztree.exe Subject PC: zleaf.exe

z-Tree Architecture



How to get z-Tree



In order to download zTree you have to obtain a licence:

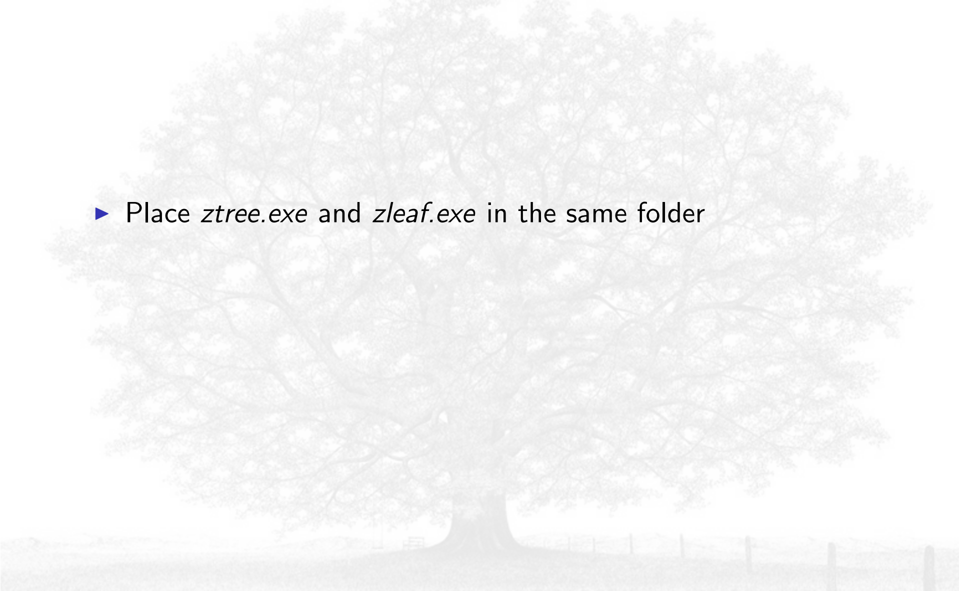
`http://www.ztree.uzh.ch/index.html`

How to get z-Tree

In order to download zTree you have to obtain a licence:
http://www.ztree.uzh.ch/index.html

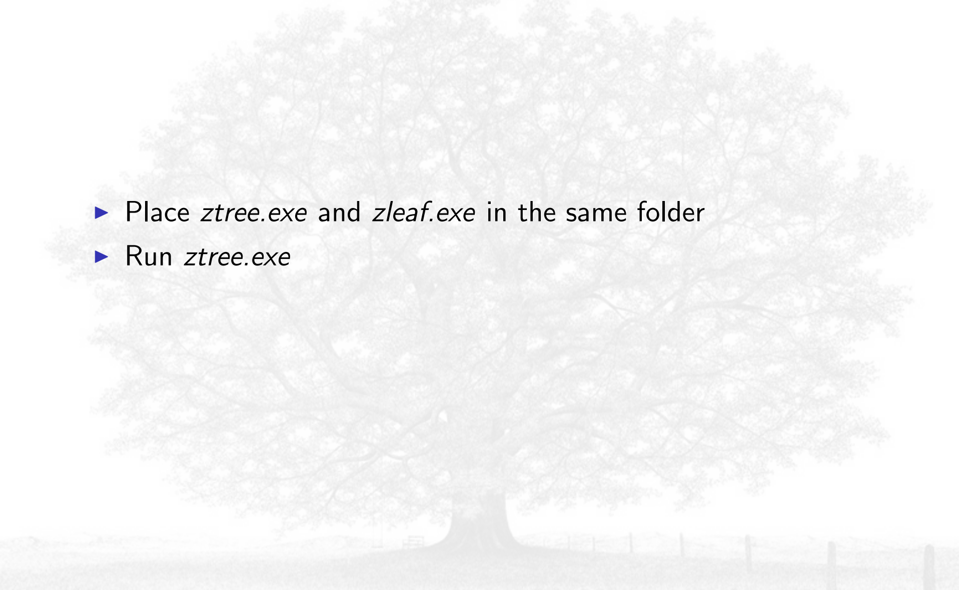
In return for the license YOU undertake to mention the name of the Department of Economics of the University of Zurich as well as the name z-Tree and to cite the following article in all publications in which results of experiments conducted with the Software are published:
Urs Fischbacher, z-Tree: Zurich Toolbox for Ready-made Economic Experiments, Experimental Economics 10(2), 171-178.

Running zTree **on your computer**



- Place *ztree.exe* and *zleaf.exe* in the same folder

Running zTree **on your computer**

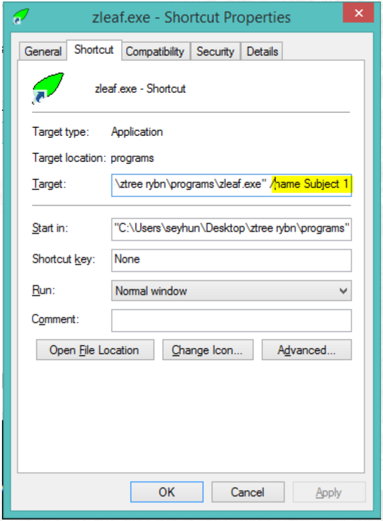


- Place *ztree.exe* and *zleaf.exe* in the same folder
- Run *ztree.exe*

Running zTree **on your computer**

- Place *ztree.exe* and *zleaf.exe* in the same folder
- Run *ztree.exe*
- Run *zleaf.exe* (Only runs if *ztree.exe* is running)
    - Run from command line ”zleaf.exe /name *SubjectName*” to give a name to the subject

Running zTree **on your computer**



zleaf.exe - Shortcut Properties

zleaf.exe - Shortcut

Target type: Application

Target location: programs

Target: \ztree rybn\programs\zleaf.exe" /name Subject 1

Start in: "C:\Users\seyhun\Desktop\ztree rybn\programs"

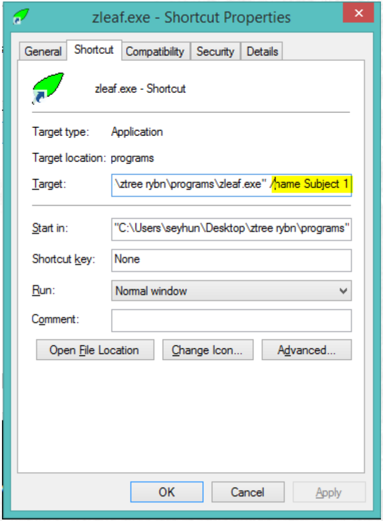
Shortcut key: None

Run: Normal window

Comment:

- ▶ Create a shortcut of zleaf

Running zTree **on your computer**


zleaf.exe - Shortcut Properties

zleaf.exe - Shortcut

Target type: Application

Target location: programs

Target: \ztree rybn\programs\zleaf.exe" /name Subject 1

Start in: "C:\Users\seyhun\Desktop\ztree rybn\programs"

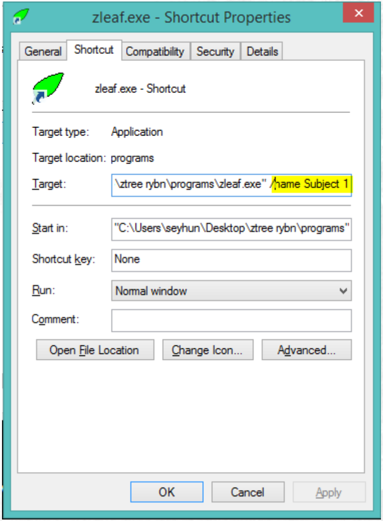
Shortcut key: None

Run: Normal window

Comment:

- Create a shortcut of zleaf
- Add /`name NameOfTheSubject` at the end of the target part

Running zTree **on your computer**



Target type: Application

Target location: programs

Target: \ztree rybn\programs\zleaf.exe" /name Subject 1

Start in: "C:\Users\seyhun\Desktop\ztree rybn\programs"

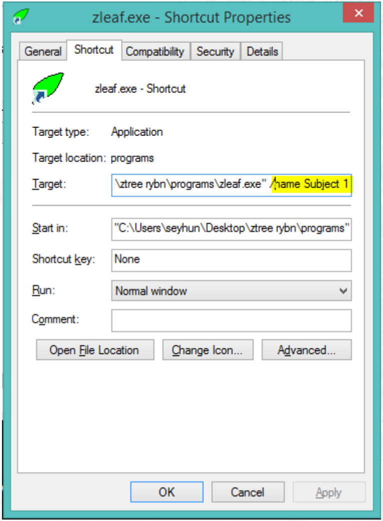
Shortcut key: None

Run: Normal window

Comment:

- Create a shortcut of zleaf
- Add /`name NameOfTheSubject` at the end of the target part
- Change the name for each instance

Running zTree **on your computer**



zleaf.exe - Shortcut

Target type: Application

Target location: programs

Target: \ztree rybn\programs\zleaf.exe" /name Subject 1

Start in: "C:\Users\seyhun\Desktop\ztree rybn\programs"

Shortcut key: None

Run: Normal window

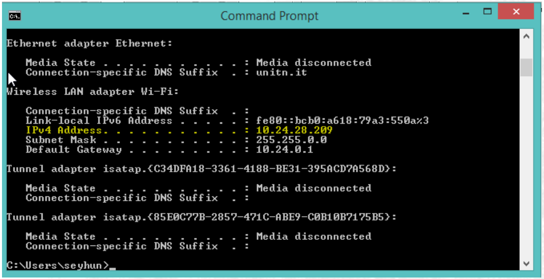
Comment:

- Create a shortcut of zleaf
- Add /`name NameOfTheSubject` at the end of the target part
- Change the name for each instance

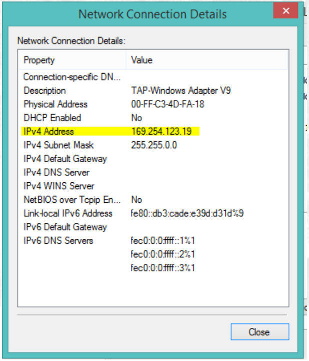
Please note!

There are some tools and tricks to open multiple z-leafs at once. We will discuss about them later.

Running zTree **on a lab**:



```
Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : unitn.it

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::bcb0:a618:79a3:550a%3
   IPv4 Address. . . . . . . . . . . : 10.24.28.209
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 10.24.0.1

Tunnel adapter isatap.{C34DFA18-3361-4188-BE31-395ACD7A568D}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{85E0C77B-2857-471C-ABE9-C0B10B7175B5}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\seyhun>_
```

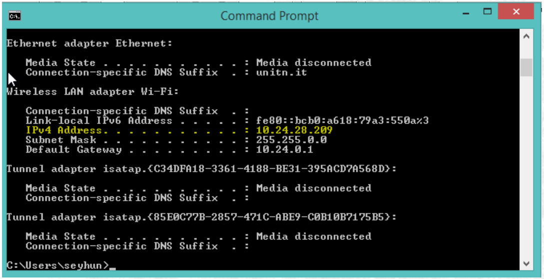


Network Connection Details:

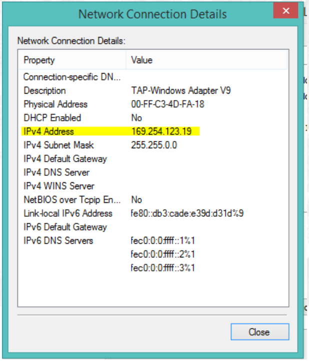
Property	Value
Connection-specific DN...	
Description	TAP-Windows Adapter V9
Physical Address	00-FF-C3-4D-FA-18
DHCP Enabled	No
IPv4 Address	169.254.123.19
IPv4 Subnet Mask	255.255.0.0
IPv4 Default Gateway	
IPv4 DNS Server	
IPv4 WINS Server	
NetBIOS over Tcpip En...	No
Link-local IPv6 Address	fe80::db3:cade:e39d:d31d%9
IPv6 Default Gateway	
IPv6 DNS Servers	fec0:0:0:ffff::1%1 fec0:0:0:ffff::2%1 fec0:0:0:ffff::3%1

- Run zTree.exe on experimenter pc

Running zTree **on a lab**:



```
Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : unitn.it

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::bcb0:a618:79a3:550a%3
   IPv4 Address. . . . . . . . . . . : 10.24.28.209
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 10.24.0.1

Tunnel adapter isatap.{C34DFA18-3361-4188-BE31-395ACD7A568D}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{85E0C77B-2857-471C-ABE9-C0B10B7175B5}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\seyhun>_
```

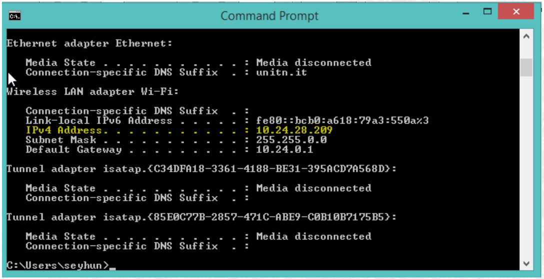


Network Connection Details:

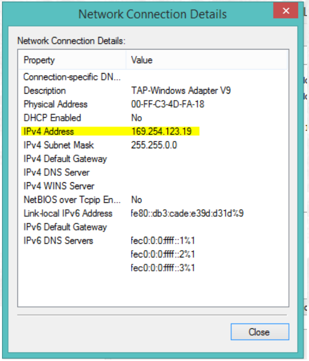
Property	Value
Connection-specific DN...	
Description	TAP-Windows Adapter V9
Physical Address	00-FF-C3-4D-FA-18
DHCP Enabled	No
IPv4 Address	169.254.123.19
IPv4 Subnet Mask	255.255.0.0
IPv4 Default Gateway	
IPv4 DNS Server	
IPv4 WINS Server	
NetBIOS over Tcpip En...	No
Link-local IPv6 Address	fe80::db3:cade:e39d:d31d%9
IPv6 Default Gateway	
IPv6 DNS Servers	fec0:0:0:ffff::1%1 fec0:0:0:ffff::2%1 fec0:0:0:ffff::3%1

- Run zTree.exe on experimenter pc
- Get the ip adress of experimenter's pc write `ipconfig` on `command window` or Open "Network and Sharing Center" $->$ Click *LAN connection* $->$ Click *Details*

Running zTree **on a lab**:



```
Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : unitn.it

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::bcb0:a618:79a3:550a%3
   IPv4 Address. . . . . . . . . . . : 10.24.28.209
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 10.24.0.1

Tunnel adapter isatap.{C34DFA18-3361-4188-BE31-395ACD7A568D}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{85E0C77B-2857-471C-ABE9-C0B10B7175B5}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\seyhun>_
```



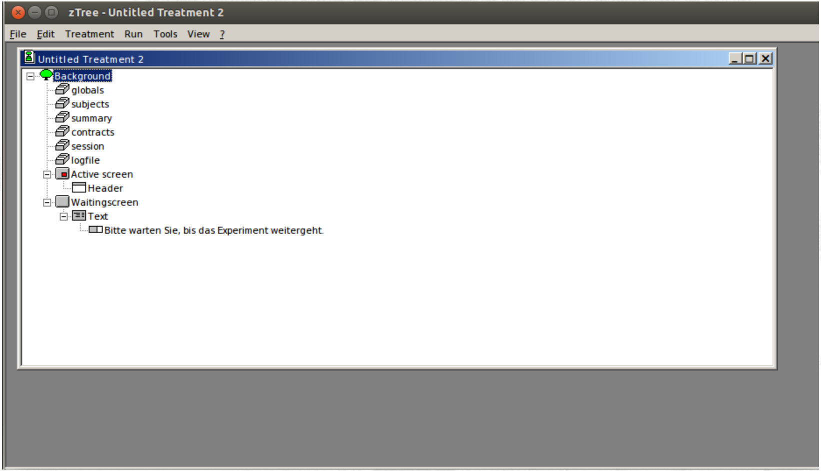
Network Connection Details:

Property	Value
Connection-specific DN...	
Description	TAP-Windows Adapter V9
Physical Address	00-FF-C3-4D-FA-18
DHCP Enabled	No
IPv4 Address	169.254.123.19
IPv4 Subnet Mask	255.255.0.0
IPv4 Default Gateway	
IPv4 DNS Server	
IPv4 WINS Server	
NetBIOS over Tcpip En...	No
Link-local IPv6 Address	fe80::db3:cade:e39d:d31d%9
IPv6 Default Gateway	
IPv6 DNS Servers	fec0:0:0:ffff::1%1 fec0:0:0:ffff::2%1 fec0:0:0:ffff::3%1

- Run zTree.exe on experimenter pc
- Get the ip adress of experimenter's pc write `ipconfig on command window` or Open "Network and Sharing Center" − > Click *LAN connection* − > Click *Details*
- Add `/server "ip number"` at the end of zleaf command

Let's try together

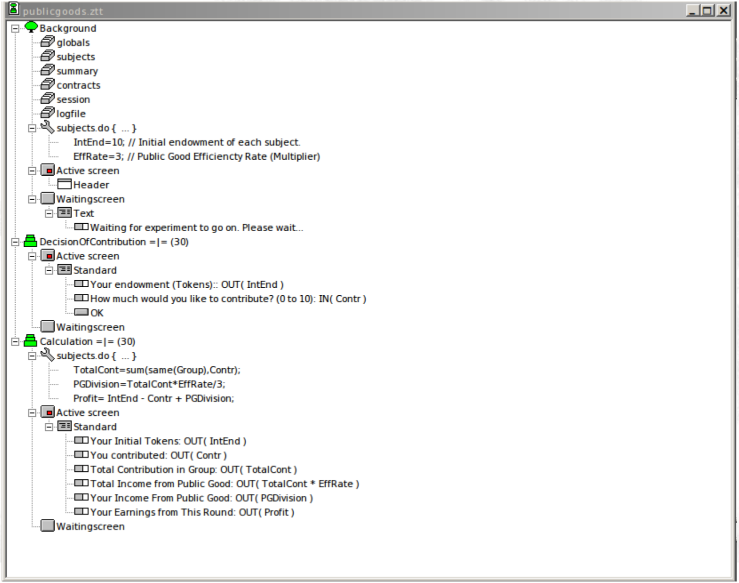
First look at z-Tree



zTree - Untitled Treatment 2
File Edit Treatment Run Tools View ?
Untitled Treatment 2

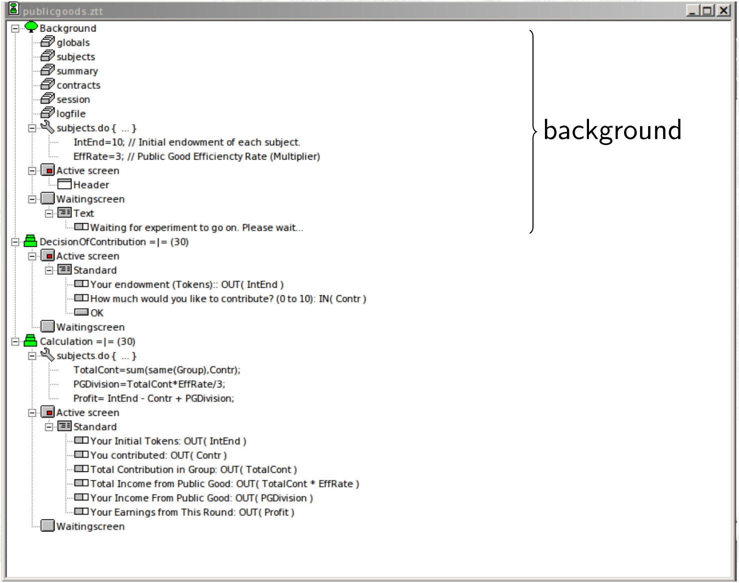
```
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    Active screen
        Header
    Waitingscreen
        Text
            Bitte warten Sie, bis das Experiment weitergeht.
```

Structure of a Treatment file (*.ztt)



```
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    subjects.do { ... }
        IntEnd=10; // Initial endowment of each subject.
        EffRate=3; // Public Good Efficiencty Rate (Multiplier)
    Active screen
        Header
    Waitingscreen
        Text
            Waiting for experiment to go on. Please wait...
DecisionOfContribution =|= (30)
    Active screen
        Standard
            Your endowment (Tokens):: OUT( IntEnd )
            How much would you like to contribute? (0 to 10): IN( Contr )
            OK
    Waitingscreen
Calculation =|= (30)
    subjects.do { ... }
        TotalCont=sum(same(Group),Contr);
        PGDivision=TotalCont*EffRate/3;
        Profit= IntEnd - Contr + PGDivision;
    Active screen
        Standard
            Your Initial Tokens: OUT( IntEnd )
            You contributed: OUT( Contr )
            Total Contribution in Group: OUT( TotalCont )
            Total Income from Public Good: OUT( TotalCont * EffRate )
            Your Income From Public Good: OUT( PGDivision )
            Your Earnings from This Round: OUT( Profit )
    Waitingscreen
```

Structure of a Treatment file (*.ztt)

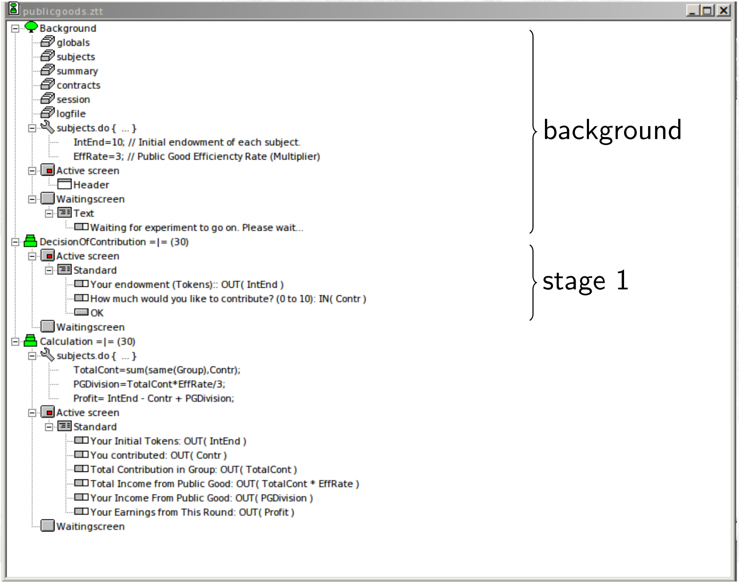


publicgoods.ztt

```
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    subjects.do { ... }
        IntEnd=10; // Initial endowment of each subject.
        EffRate=3; // Public Good Efficiencty Rate (Multiplier)
    Active screen
        Header
    Waitingscreen
        Text
            Waiting for experiment to go on. Please wait...
DecisionOfContribution =|= (30)
    Active screen
        Standard
            Your endowment (Tokens):: OUT( IntEnd )
            How much would you like to contribute? (0 to 10): IN( Contr )
            OK
    Waitingscreen
Calculation =|= (30)
    subjects.do { ... }
        TotalCont=sum(same(Group),Contr);
        PGDivision=TotalCont*EffRate/3;
        Profit= IntEnd - Contr + PGDivision;
    Active screen
        Standard
            Your Initial Tokens: OUT( IntEnd )
            You contributed: OUT( Contr )
            Total Contribution in Group: OUT( TotalCont )
            Total Income from Public Good: OUT( TotalCont * EffRate )
            Your Income From Public Good: OUT( PGDivision )
            Your Earnings from This Round: OUT( Profit )
    Waitingscreen
```

$\left.\vphantom{\begin{array}{c}a\\a\\a\\a\\a\\a\\a\\a\\a\\a\\a\\a\\a\end{array}}\right\}$ background

Structure of a Treatment file (*.ztt)

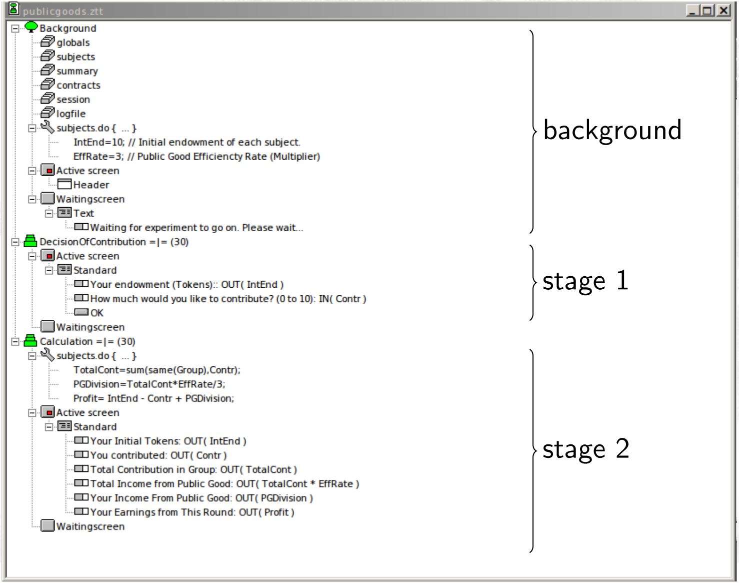


```
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    subjects.do { ... }
        IntEnd=10; // Initial endowment of each subject.
        EffRate=3; // Public Good Efficiencty Rate (Multiplier)
    Active screen
        Header
    Waitingscreen
        Text
            Waiting for experiment to go on. Please wait...
DecisionOfContribution =|= (30)
    Active screen
        Standard
            Your endowment (Tokens):: OUT( IntEnd )
            How much would you like to contribute? (0 to 10): IN( Contr )
            OK
    Waitingscreen
Calculation =|= (30)
    subjects.do { ... }
        TotalCont=sum(same(Group),Contr);
        PGDivision=TotalCont*EffRate/3;
        Profit= IntEnd - Contr + PGDivision;
    Active screen
        Standard
            Your Initial Tokens: OUT( IntEnd )
            You contributed: OUT( Contr )
            Total Contribution in Group: OUT( TotalCont )
            Total Income from Public Good: OUT( TotalCont * EffRate )
            Your Income From Public Good: OUT( PGDivision )
            Your Earnings from This Round: OUT( Profit )
    Waitingscreen
```

background

stage 1

Structure of a Treatment file (*.ztt)



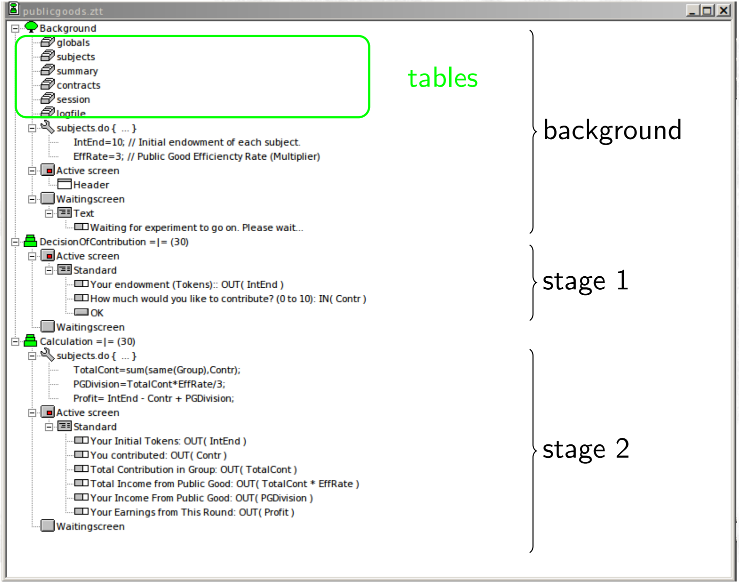
```
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    subjects.do { ... }
        IntEnd=10; // Initial endowment of each subject.
        EffRate=3; // Public Good Efficiencty Rate (Multiplier)
    Active screen
        Header
    Waitingscreen
        Text
            Waiting for experiment to go on. Please wait...
DecisionOfContribution =|= (30)
    Active screen
        Standard
            Your endowment (Tokens):: OUT( IntEnd )
            How much would you like to contribute? (0 to 10): IN( Contr )
            OK
    Waitingscreen
Calculation =|= (30)
    subjects.do { ... }
        TotalCont=sum(same(Group),Contr);
        PGDivision=TotalCont*EffRate/3;
        Profit= IntEnd - Contr + PGDivision;
    Active screen
        Standard
            Your Initial Tokens: OUT( IntEnd )
            You contributed: OUT( Contr )
            Total Contribution in Group: OUT( TotalCont )
            Total Income from Public Good: OUT( TotalCont * EffRate )
            Your Income From Public Good: OUT( PGDivision )
            Your Earnings from This Round: OUT( Profit )
    Waitingscreen
```

background

stage 1

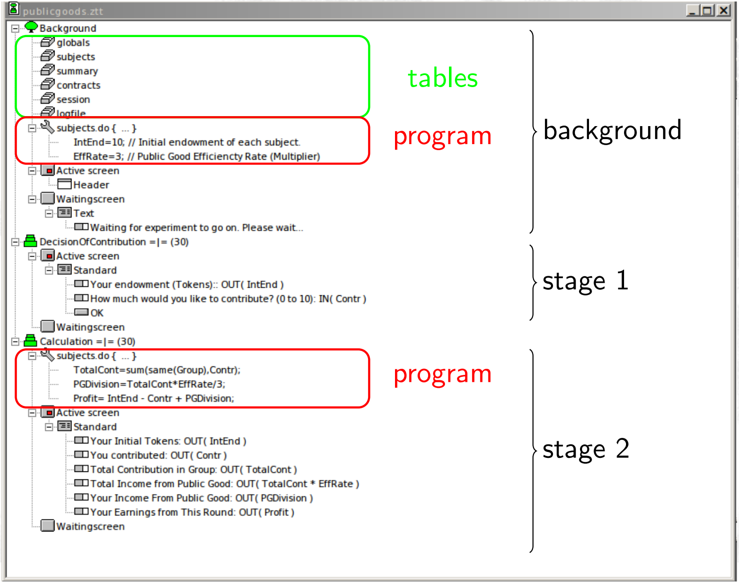
stage 2

Structure of a Treatment file (*.ztt)



```
publicgoods.ztt
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    subjects.do { ... }
        IntEnd=10; // Initial endowment of each subject.
        EffRate=3; // Public Good Efficiencty Rate (Multiplier)
    Active screen
        Header
    Waitingscreen
        Text
            Waiting for experiment to go on. Please wait...
DecisionOfContribution =|= (30)
    Active screen
        Standard
            Your endowment (Tokens):: OUT( IntEnd )
            How much would you like to contribute? (0 to 10): IN( Contr )
            OK
    Waitingscreen
Calculation =|= (30)
    subjects.do { ... }
        TotalCont=sum(same(Group),Contr);
        PGDivision=TotalCont*EffRate/3;
        Profit= IntEnd - Contr + PGDivision;
    Active screen
        Standard
            Your Initial Tokens: OUT( IntEnd )
            You contributed: OUT( Contr )
            Total Contribution in Group: OUT( TotalCont )
            Total Income from Public Good: OUT( TotalCont * EffRate )
            Your Income From Public Good: OUT( PGDivision )
            Your Earnings from This Round: OUT( Profit )
    Waitingscreen
```

Structure of a Treatment file (*.ztt)



publicgoods.ztt

```
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    subjects.do { ... }
        IntEnd=10; // Initial endowment of each subject.
        EffRate=3; // Public Good Efficiencty Rate (Multiplier)
    Active screen
        Header
    Waitingscreen
        Text
            Waiting for experiment to go on. Please wait...
DecisionOfContribution =|= (30)
    Active screen
        Standard
            Your endowment (Tokens):: OUT( IntEnd )
            How much would you like to contribute? (0 to 10): IN( Contr )
            OK
    Waitingscreen
Calculation =|= (30)
    subjects.do { ... }
        TotalCont=sum(same(Group),Contr);
        PGDivision=TotalCont*EffRate/3;
        Profit= IntEnd - Contr + PGDivision;
    Active screen
        Standard
            Your Initial Tokens: OUT( IntEnd )
            You contributed: OUT( Contr )
            Total Contribution in Group: OUT( TotalCont )
            Total Income from Public Good: OUT( TotalCont * EffRate )
            Your Income From Public Good: OUT( PGDivision )
            Your Earnings from This Round: OUT( Profit )
    Waitingscreen
```

tables

program

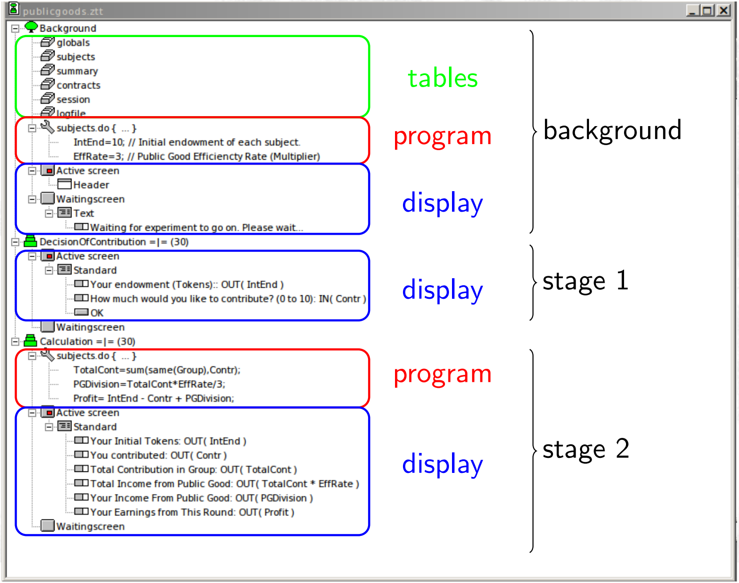
background

stage 1

program

stage 2

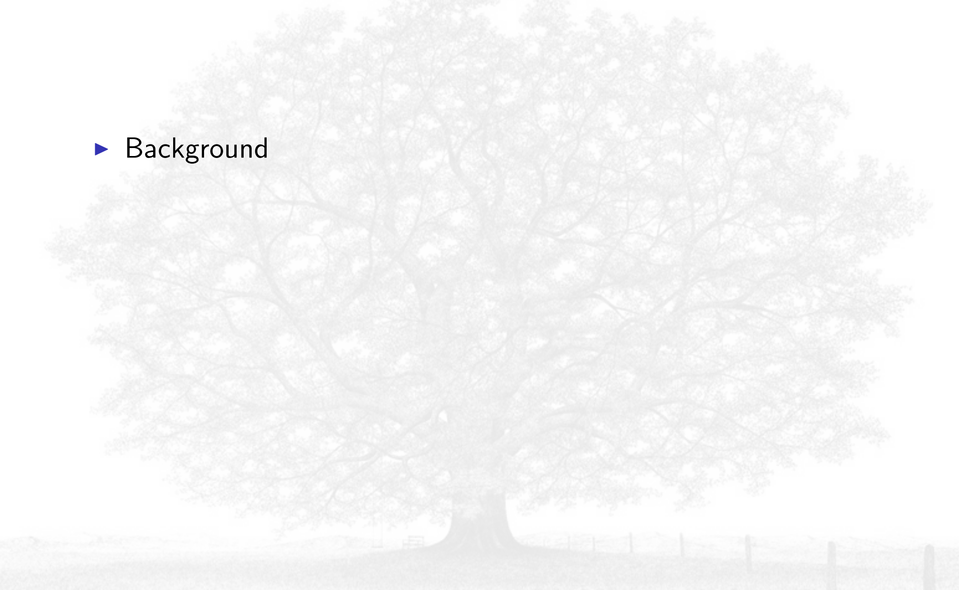
Structure of a Treatment file (*.ztt)



publicgoods.ztt

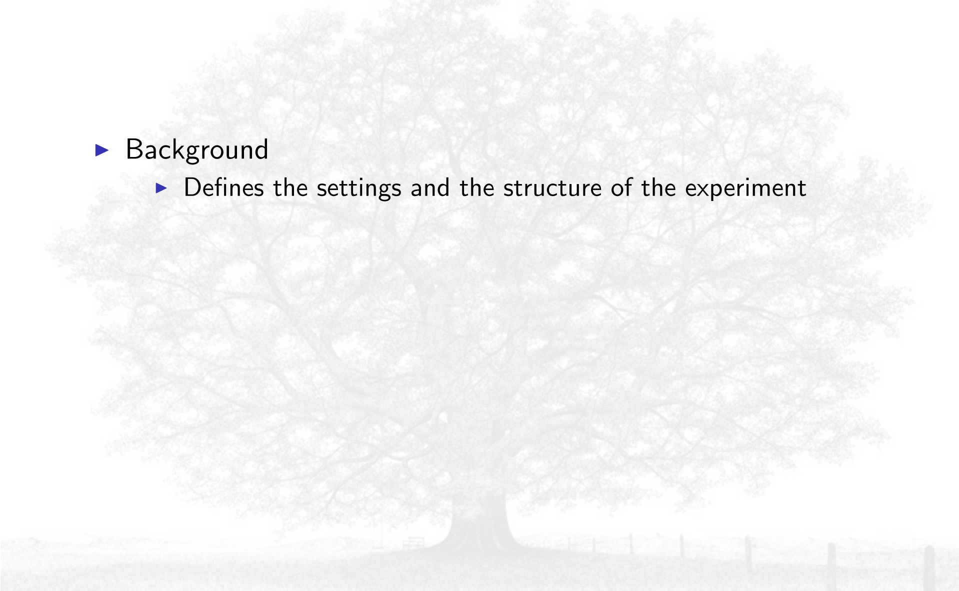
```
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    subjects.do { ... }
        IntEnd=10; // Initial endowment of each subject.
        EffRate=3; // Public Good Efficiency Rate (Multiplier)
    Active screen
        Header
    Waitingscreen
        Text
            Waiting for experiment to go on. Please wait...
DecisionOfContribution =|= (30)
    Active screen
        Standard
            Your endowment (Tokens):: OUT( IntEnd )
            How much would you like to contribute? (0 to 10): IN( Contr )
            OK
    Waitingscreen
Calculation =|= (30)
    subjects.do { ... }
        TotalCont=sum(same(Group),Contr);
        PGDivision=TotalCont*EffRate/3;
        Profit= IntEnd - Contr + PGDivision;
    Active screen
        Standard
            Your Initial Tokens: OUT( IntEnd )
            You contributed: OUT( Contr )
            Total Contribution in Group: OUT( TotalCont )
            Total Income from Public Good: OUT( TotalCont * EffRate )
            Your Income From Public Good: OUT( PGDivision )
            Your Earnings from This Round: OUT( Profit )
    Waitingscreen
```

Structure of z-Tree



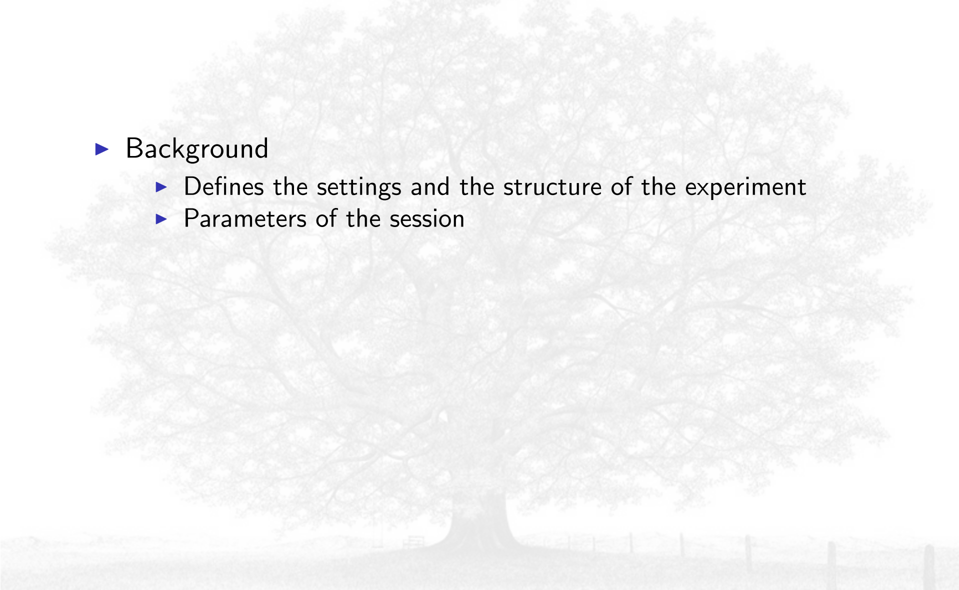
- Background

Structure of z-Tree



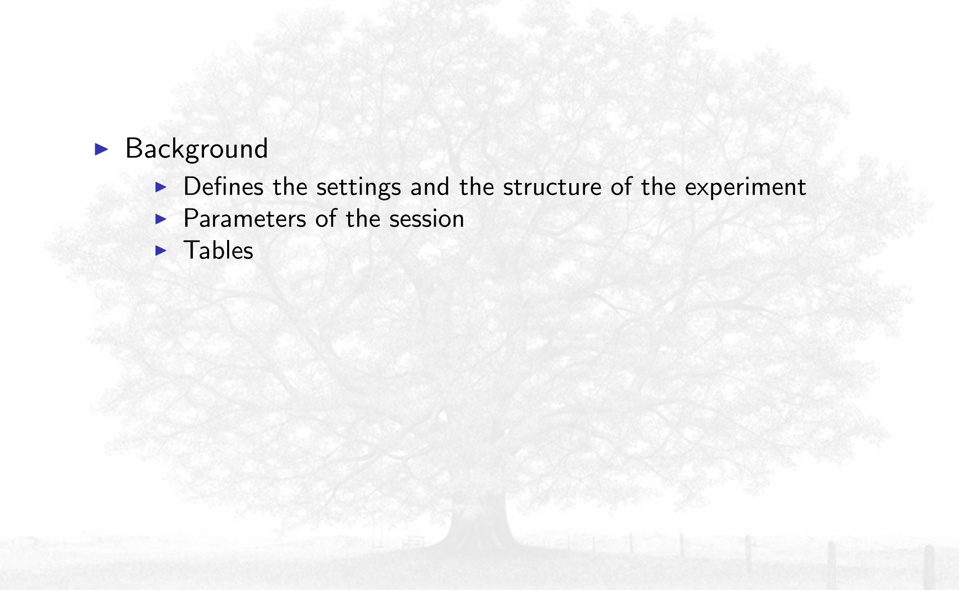
- Background
  - Defines the settings and the structure of the experiment

Structure of z-Tree



- Background
  - Defines the settings and the structure of the experiment
  - Parameters of the session

Structure of z-Tree

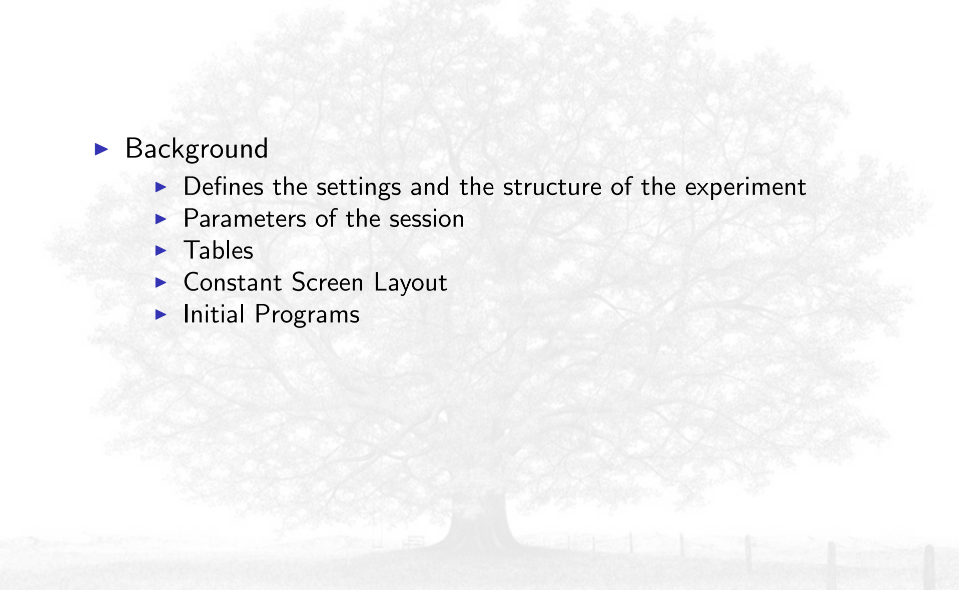


- Background
  - Defines the settings and the structure of the experiment
  - Parameters of the session
  - Tables

Structure of z-Tree

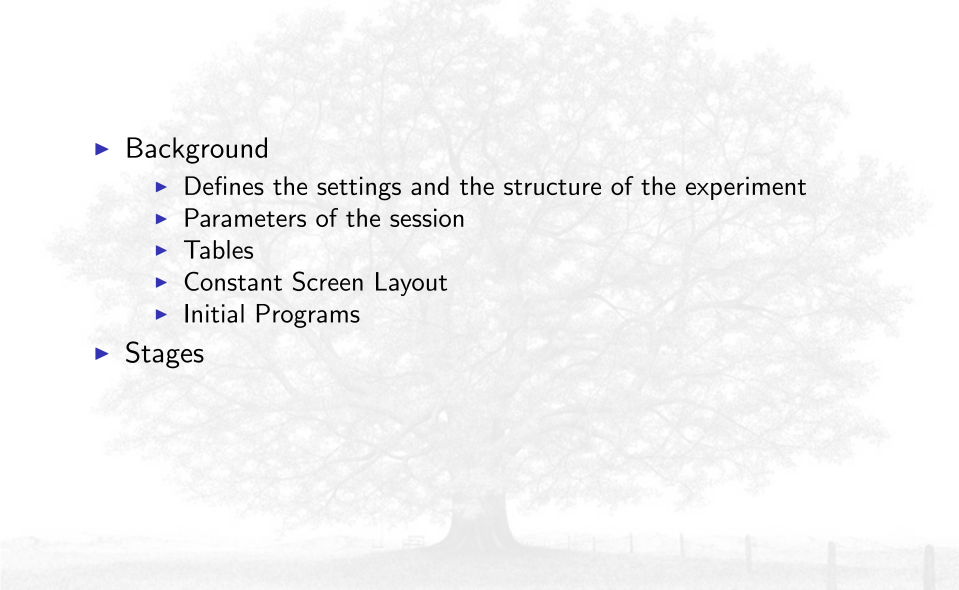
- Background
    - Defines the settings and the structure of the experiment
    - Parameters of the session
    - Tables
    - Constant Screen Layout

Structure of z-Tree



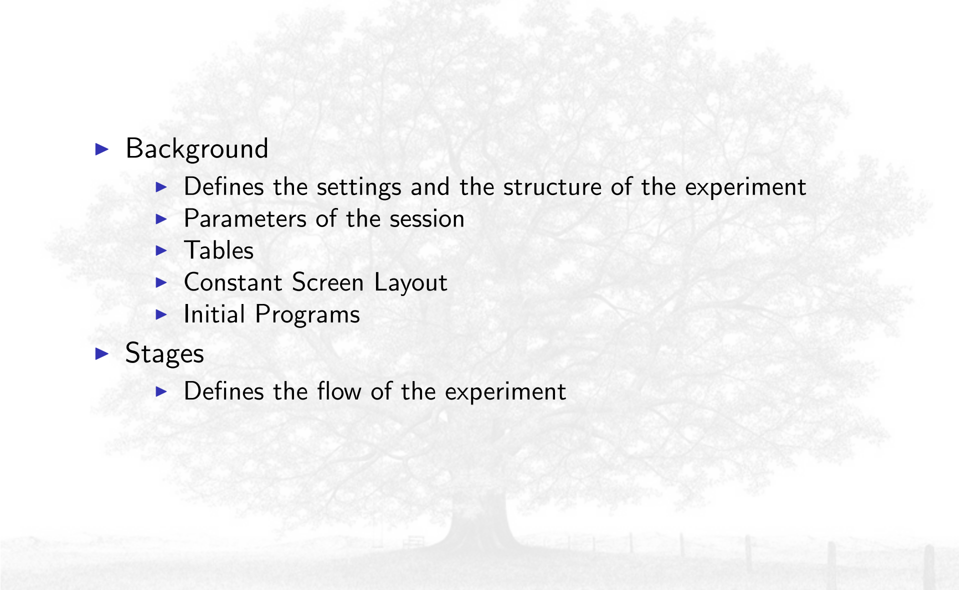
- Background
  - Defines the settings and the structure of the experiment
  - Parameters of the session
  - Tables
  - Constant Screen Layout
  - Initial Programs

Structure of z-Tree



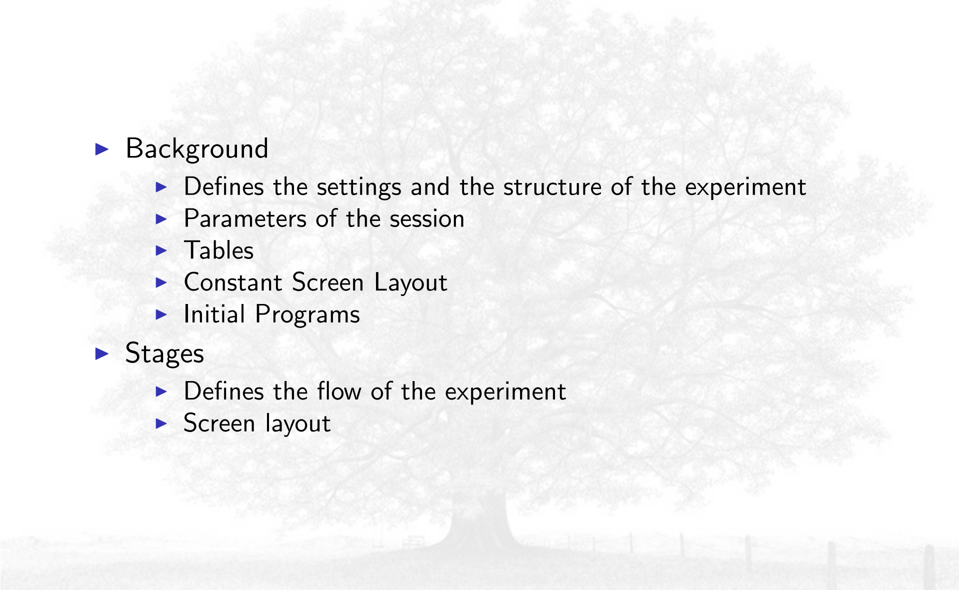
- Background
  - Defines the settings and the structure of the experiment
  - Parameters of the session
  - Tables
  - Constant Screen Layout
  - Initial Programs
- Stages

Structure of z-Tree



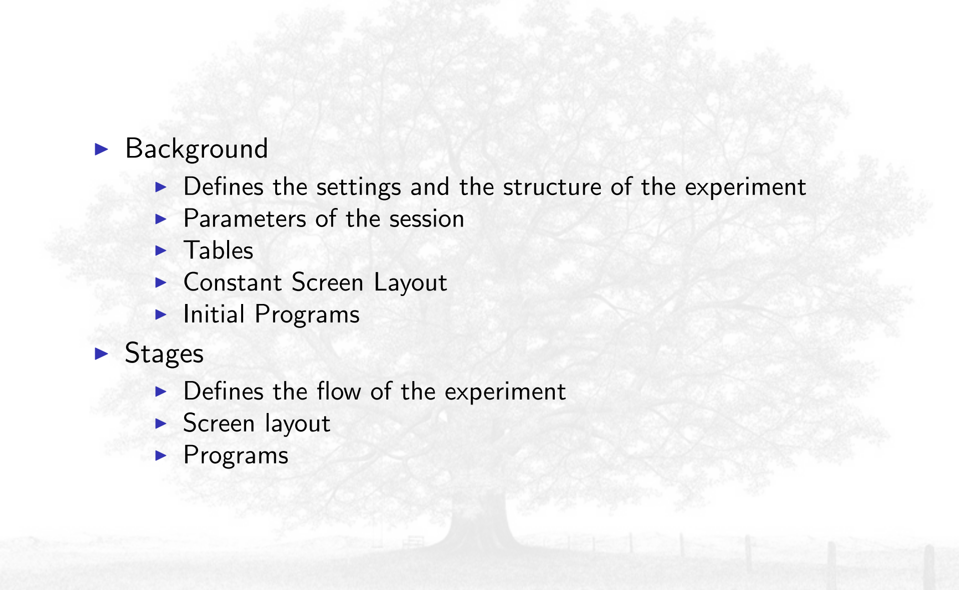
- Background
  - Defines the settings and the structure of the experiment
  - Parameters of the session
  - Tables
  - Constant Screen Layout
  - Initial Programs
- Stages
  - Defines the flow of the experiment

Structure of z-Tree



- Background
  - Defines the settings and the structure of the experiment
  - Parameters of the session
  - Tables
  - Constant Screen Layout
  - Initial Programs
- Stages
  - Defines the flow of the experiment
  - Screen layout

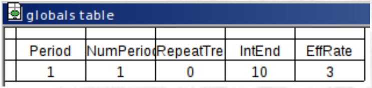
Structure of z-Tree



- Background
  - Defines the settings and the structure of the experiment
  - Parameters of the session
  - Tables
  - Constant Screen Layout
  - Initial Programs
- Stages
  - Defines the flow of the experiment
  - Screen layout
  - Programs

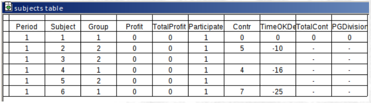
Tables

- Stores all the information.



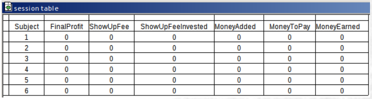
globals table

Period	NumPerio	RepeatTre	IntEnd	EffRate
1	1	0	10	3



subjects table

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

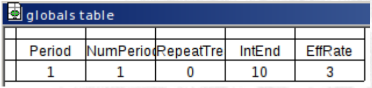


session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

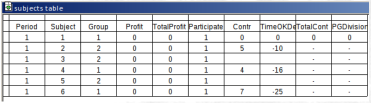
Tables

- Stores all the information.
- Six default. Three main:



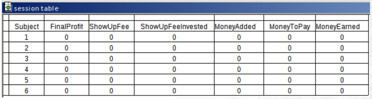
globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3



subjects table

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

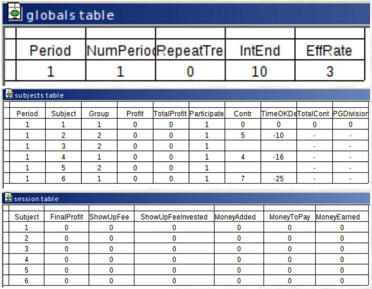


session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Tables

- Stores all the information.
- Six default. Three main:
  - **globals**



globals table

Period	NumPerio	RepeatTre	IntEnd	EffRate
1	1	0	10	3

subjects table

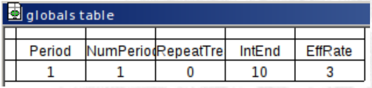
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

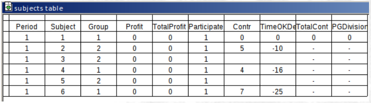
Tables

- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible



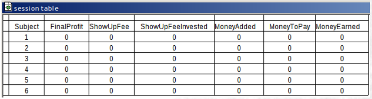
globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3



subjects table

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

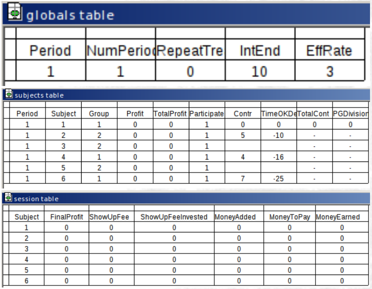


session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Tables

- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible
    - general parameters



globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

subjects table

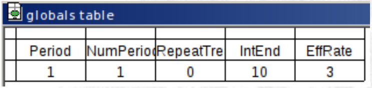
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

session table

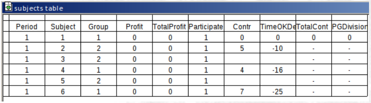
Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Tables

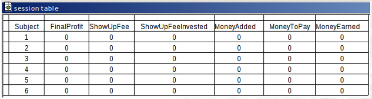
- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible
    - general parameters
    - Default variables: *Period*, *NumPeriods*, *RepeatTreatment*


globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3


subjects table

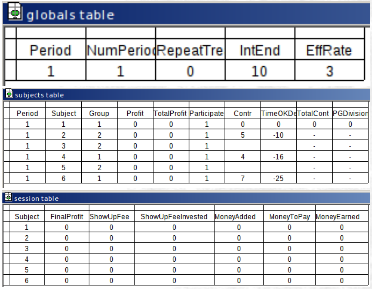
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-


session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Tables

- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible
    - general parameters
    - Default variables: *Period*, *NumPeriods*, *RepeatTreatment*
  - **subjects**



globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

subjects table

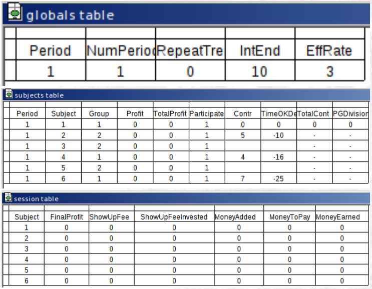
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Tables

- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible
    - general parameters
    - Default variables: *Period*, *NumPeriods*, *RepeatTreatment*
  - **subjects**
    - *subject* specific variables



globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

subjects table

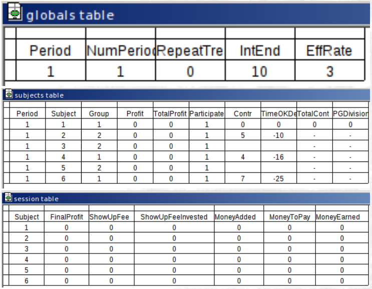
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Tables

- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible
    - general parameters
    - Default variables: *Period*, *NumPeriods*, *RepeatTreatment*
  - **subjects**
    - *subject* specific variables
    - usually the most important



globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

subjects table

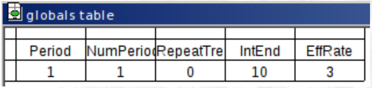
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

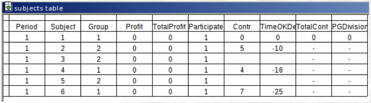
Tables

- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible
    - general parameters
    - Default variables: *Period*, *NumPeriods*, *RepeatTreatment*
  - **subjects**
    - *subject* specific variables
    - usually the most important
    - Default variables: *Period*, *Subject*, *Group*, *Profit*, *TotalProfit*, *Participate*



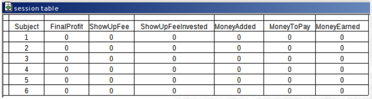
globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3



subjects table

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

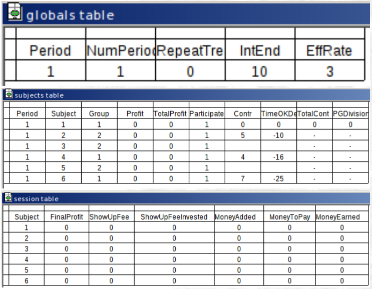


session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Tables

- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible
    - general parameters
    - Default variables: *Period*, *NumPeriods*, *RepeatTreatment*
  - **subjects**
    - *subject* specific variables
    - usually the most important
    - Default variables: *Period*, *Subject*, *Group*, *Profit*, *TotalProfit*, *Participate*
  - **session**:



globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

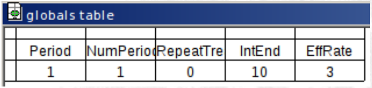
subjects table

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

session table

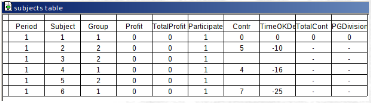
Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Tables



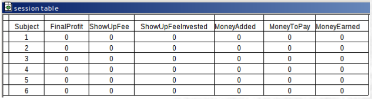
globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3



subjects table

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-



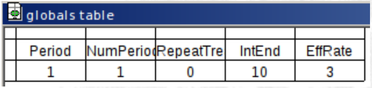
session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible
    - general parameters
    - Default variables: *Period*, *NumPeriods*, *RepeatTreatment*
  - **subjects**
    - *subject* specific variables
    - usually the most important
    - Default variables: *Period*, *Subject*, *Group*, *Profit*, *TotalProfit*, *Participate*
  - **session**:
    - *session* specific variables

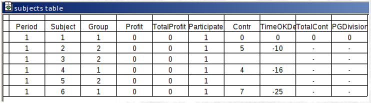
Tables

- Stores all the information.
- Six default. Three main:
    - **globals**
        - *globally* accessible
        - general parameters
        - Default variables: *Period*, *NumPeriods*, *RepeatTreatment*
    - **subjects**
        - *subject* specific variables
        - usually the most important
        - Default variables: *Period*, *Subject*, *Group*, *Profit*, *TotalProfit*, *Participate*
    - **session**:
        - *session* specific variables
        - Contains the payment information



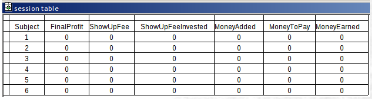
globals table

Period	NumPerio	RepeatTre	IntEnd	EffRate
1	1	0	10	3



subjects table

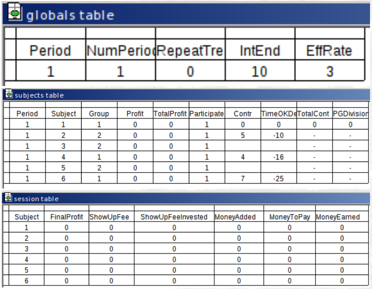
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-



session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Tables



globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

subjects table

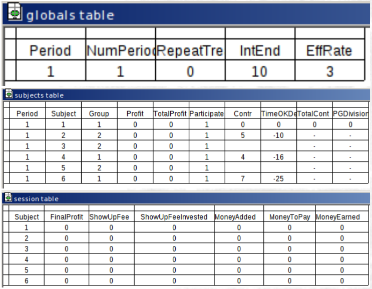
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible
    - general parameters
    - Default variables: *Period*, *NumPeriods*, *RepeatTreatment*
  - **subjects**
    - *subject* specific variables
    - usually the most important
    - Default variables: *Period*, *Subject*, *Group*, *Profit*, *TotalProfit*, *Participate*
  - **session**:
    - *session* specific variables
    - Contains the payment information
    - Gains importance in case of sessions with more than one treatment file

Tables



globals table

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

subjects table

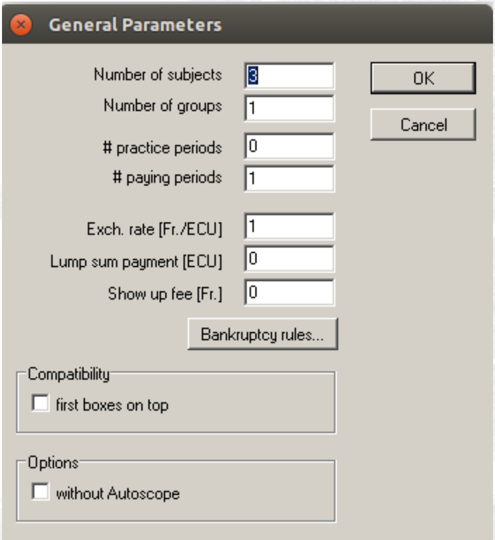
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKDe	TotalCont	PGDivision
1	1	1	0	0	1	0	0	0	0
1	2	2	0	0	1	5	-10	-	-
1	3	2	0	0	1			-	-
1	4	1	0	0	1	4	-16	-	-
1	5	2	0	0	1			-	-
1	6	1	0	0	1	7	-25	-	-

session table

Subject	FinalProfit	ShowUpFee	ShowUpFeeInvested	MoneyAdded	MoneyToPay	MoneyEarned
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

- Stores all the information.
- Six default. Three main:
  - **globals**
    - *globally* accessible
    - general parameters
    - Default variables: *Period*, *NumPeriods*, *RepeatTreatment*
  - **subjects**
    - *subject* specific variables
    - usually the most important
    - Default variables: *Period*, *Subject*, *Group*, *Profit*, *TotalProfit*, *Participate*
  - **session**:
    - *session* specific variables
    - Contains the payment information
    - Gains importance in case of sessions with more than one treatment file
    - Default variables:*Subject*, *FinalProfit*, *ShowUpFee*,*MoneyToPay*...

Background



General Parameters

Number of subjects: 3

Number of groups: 1

practice periods: 0

paying periods: 1

Exch. rate [Fr./ECU]: 1

Lump sum payment [ECU]: 0

Show up fee [Fr.]: 0

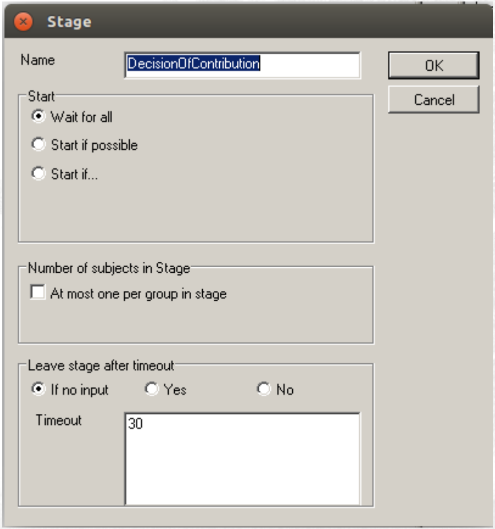
Compatibility

□ first boxes on top

Options

□ without Autoscope

Stage



Stage

Name: DecisionOfContribution

Start

● Wait for all

○ Start if possible

○ Start if...

Number of subjects in Stage

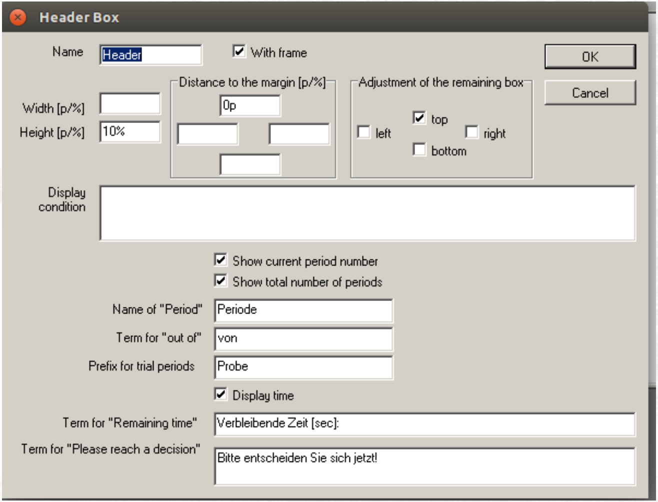
□ At most one per group in stage

Leave stage after timeout

● If no input ○ Yes ○ No

Timeout: 30

Header



Header Box

Name: Header

☑ With frame

Width [p/%]:

Height [p/%]: 10%

Distance to the margin [p/%]: 0p

Adjustment of the remaining box: □ left ☑ top □ right □ bottom

Display condition:

☑ Show current period number

☑ Show total number of periods

Name of "Period": Periode

Term for "out of": von

Prefix for trial periods: Probe

☑ Display time

Term for "Remaining time": Verbleibende Zeit [sec]:

Term for "Please reach a decision": Bitte entscheiden Sie sich jetzt!

Program

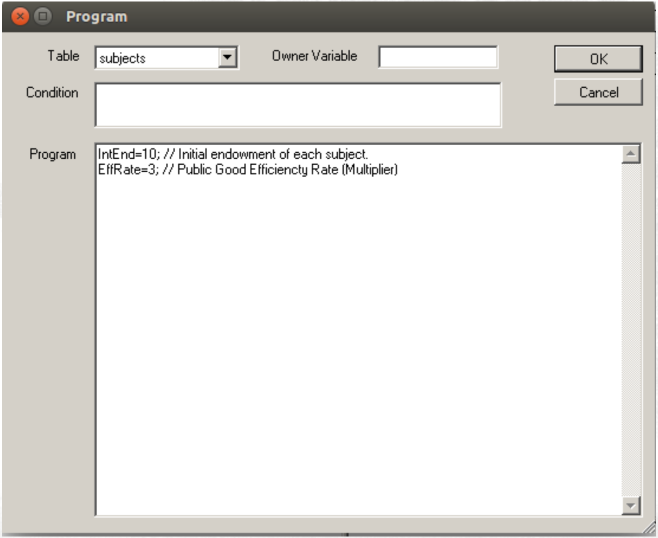


Table: subjects

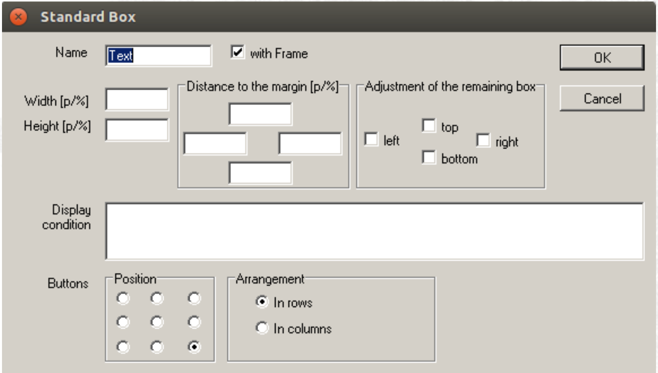
Owner Variable

Condition

Program

```
IntEnd=10; // Initial endowment of each subject.
EffRate=3; // Public Good Efficiencty Rate (Multiplier)
```

Standard Box



Standard Box

Name: Text

☑ with Frame

Width [p/%]

Height [p/%]

Distance to the margin [p/%]

Adjustment of the remaining box

□ left

□ top

□ right

□ bottom

Display condition

Buttons

Position

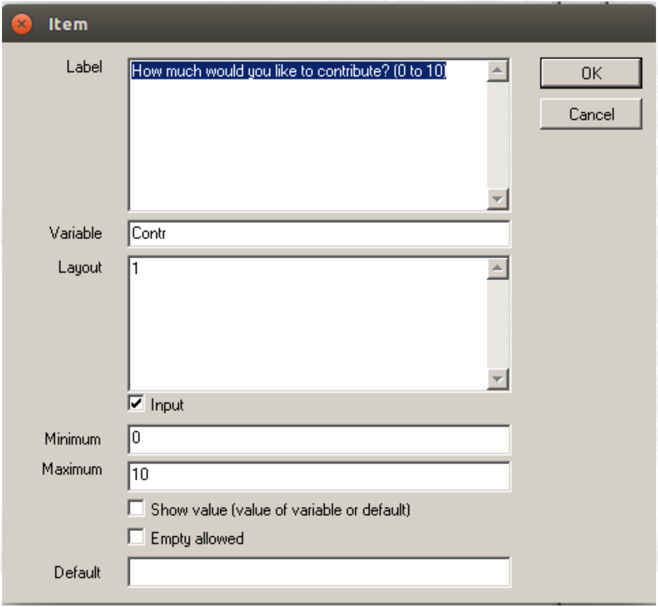
○ ○ ○
○ ○ ○
○ ○ ●

Arrangement

● In rows

○ In columns

Item



Label: How much would you like to contribute? (0 to 10)

Variable: Contr

Layout: 1

☑ Input

Minimum: 0

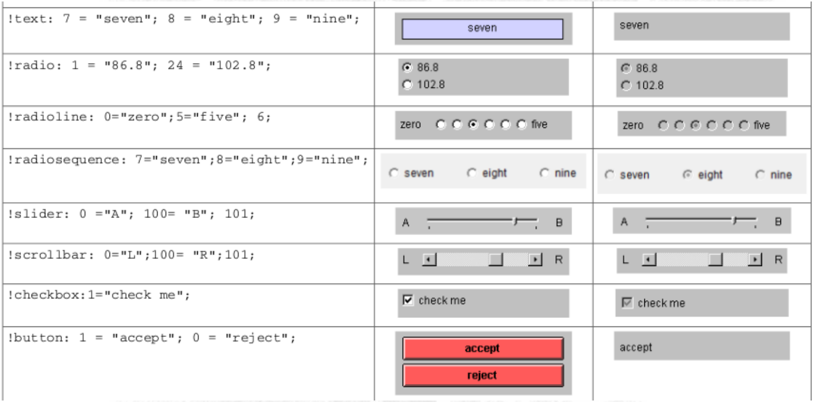
Maximum: 10

☐ Show value (value of variable or default)

☐ Empty allowed

Default:

Item



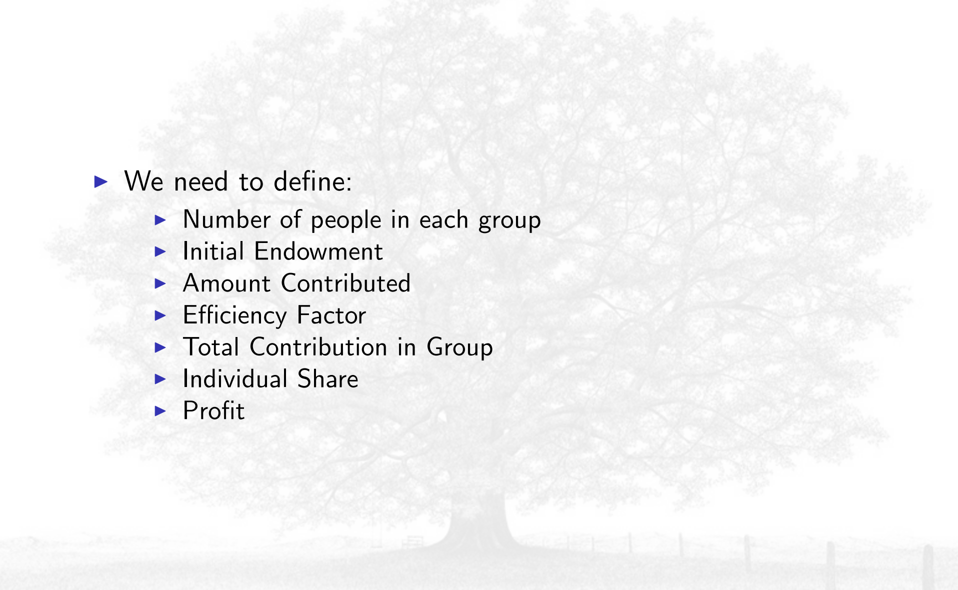
`!text: 7 = "seven"; 8 = "eight"; 9 = "nine";`	seven	seven
`!radio: 1 = "86.8"; 24 = "102.8";`	● 86.8 ○ 102.8	● 86.8 ○ 102.8
`!radioline: 0="zero";5="five"; 6;`	zero ○ ○ ● ○ ○ ○ five	zero ○ ○ ● ○ ○ ○ five
`!radiosequence: 7="seven";8="eight";9="nine";`	○ seven ○ eight ○ nine	○ seven ● eight ○ nine
`!slider: 0 ="A"; 100= "B"; 101;`	A B	A B
`!scrollbar: 0="L";100= "R";101;`	L R	L R
`!checkbox:1="check me";`	☑ check me	☑ check me
`!button: 1 = "accept"; 0 = "reject";`	accept reject	accept

Let’s start programming then...

Public Good Game

You will find the treatment file on the website: **publicgood.ztt**

Public Good Game

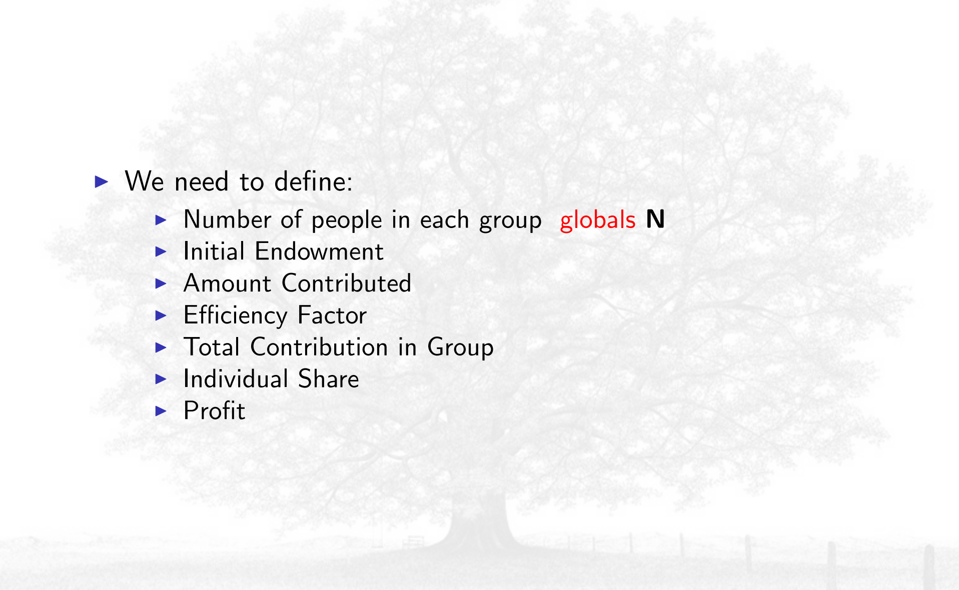


- We need to define:
  - Number of people in each group
  - Initial Endowment
  - Amount Contributed
  - Efficiency Factor
  - Total Contribution in Group
  - Individual Share
  - Profit

Public Good Game

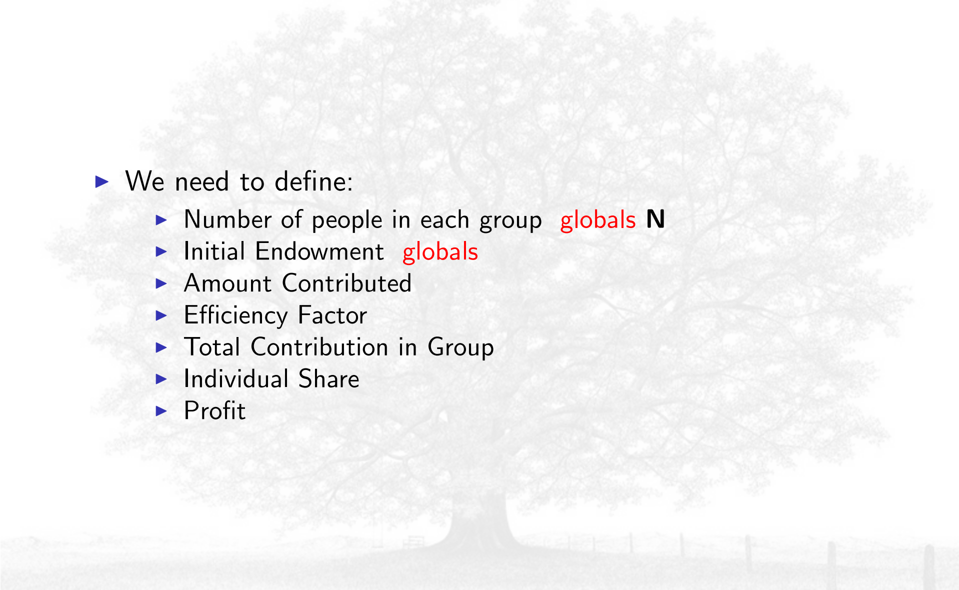
- We need to define:
  - Number of people in each group globals
  - Initial Endowment
  - Amount Contributed
  - Efficiency Factor
  - Total Contribution in Group
  - Individual Share
  - Profit

Public Good Game



- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment
  - Amount Contributed
  - Efficiency Factor
  - Total Contribution in Group
  - Individual Share
  - Profit

Public Good Game

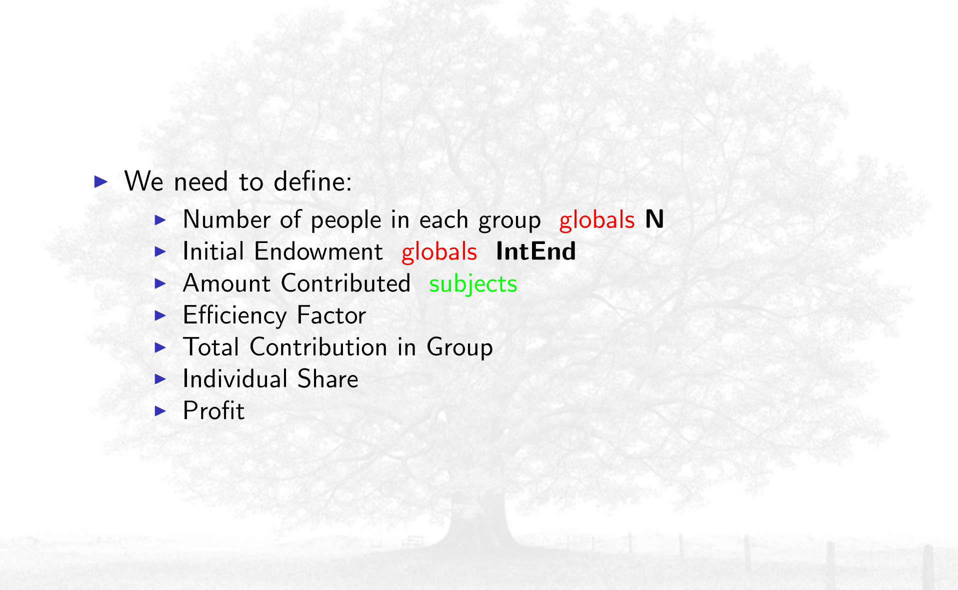


- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals
  - Amount Contributed
  - Efficiency Factor
  - Total Contribution in Group
  - Individual Share
  - Profit

Public Good Game

- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals **IntEnd**
  - Amount Contributed
  - Efficiency Factor
  - Total Contribution in Group
  - Individual Share
  - Profit

Public Good Game

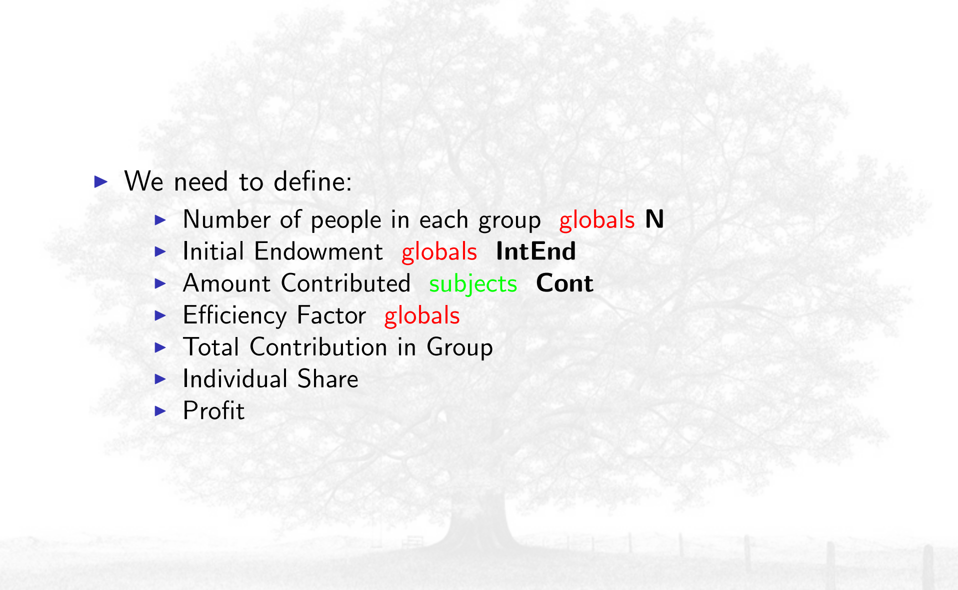


- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals **IntEnd**
  - Amount Contributed subjects
  - Efficiency Factor
  - Total Contribution in Group
  - Individual Share
  - Profit

Public Good Game

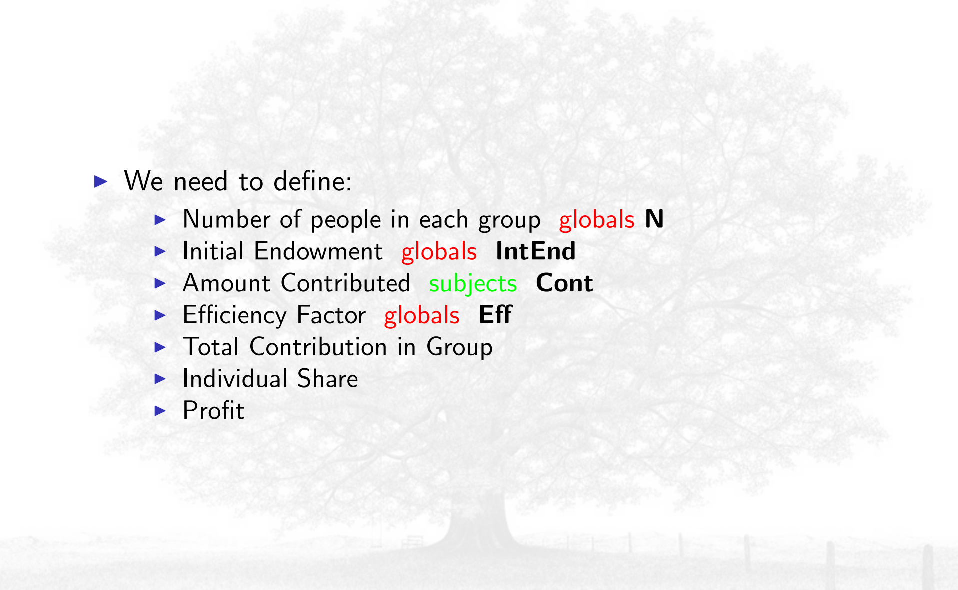
- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals **IntEnd**
  - Amount Contributed subjects **Cont**
  - Efficiency Factor
  - Total Contribution in Group
  - Individual Share
  - Profit

Public Good Game



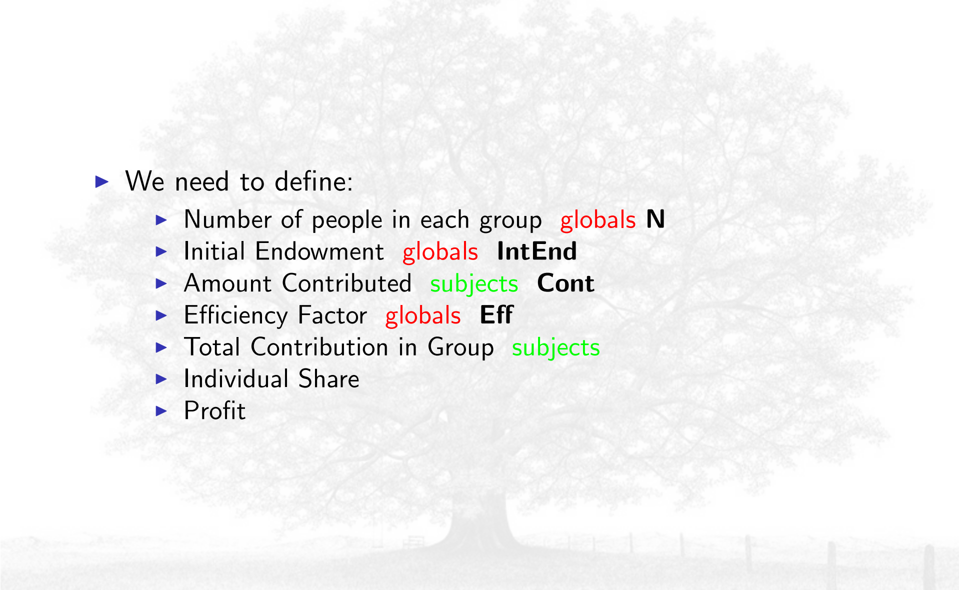
- We need to define:
    - Number of people in each group globals **N**
    - Initial Endowment globals **IntEnd**
    - Amount Contributed subjects **Cont**
    - Efficiency Factor globals
    - Total Contribution in Group
    - Individual Share
    - Profit

Public Good Game



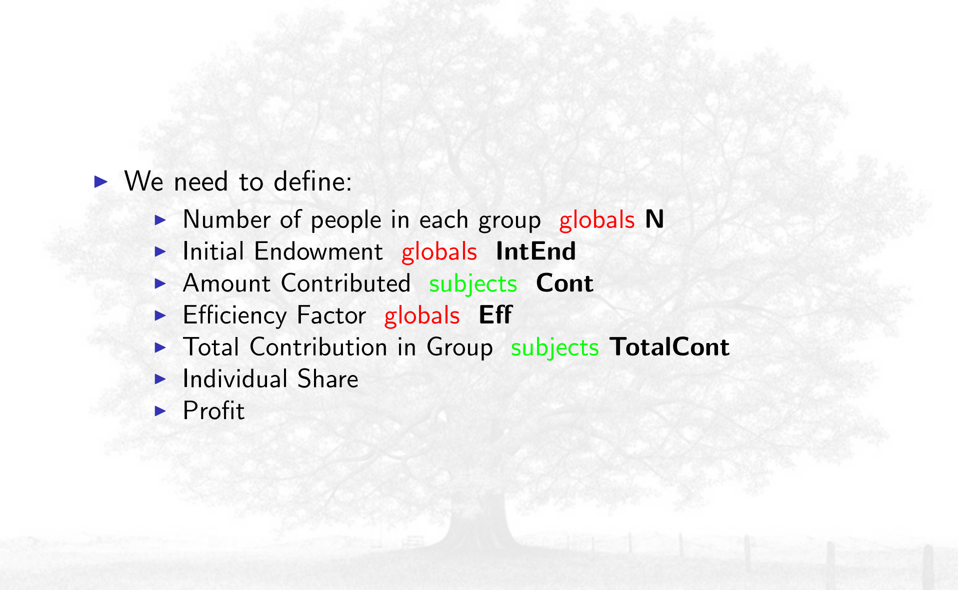
- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals **IntEnd**
  - Amount Contributed subjects **Cont**
  - Efficiency Factor globals **Eff**
  - Total Contribution in Group
  - Individual Share
  - Profit

Public Good Game



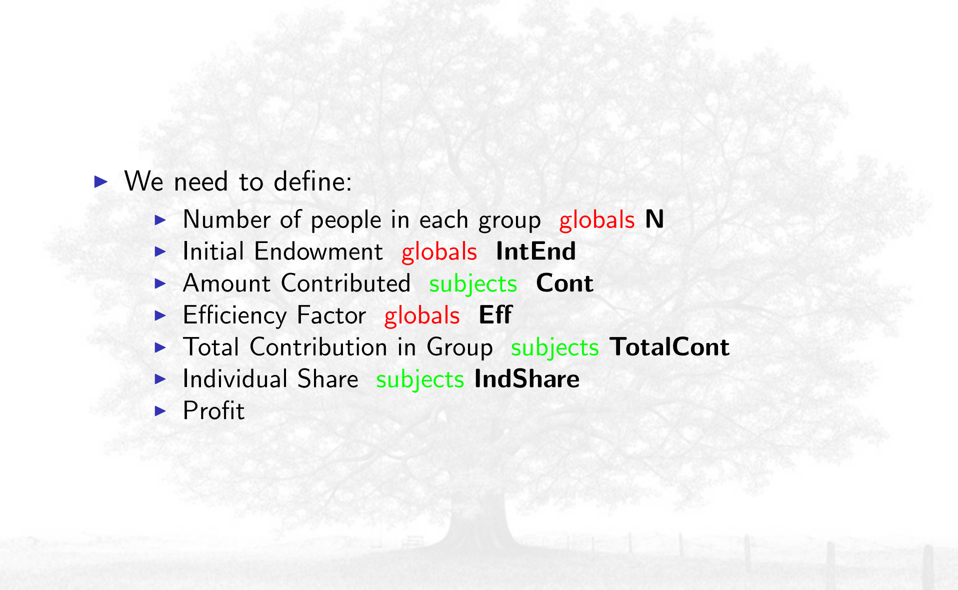
- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals **IntEnd**
  - Amount Contributed subjects **Cont**
  - Efficiency Factor globals **Eff**
  - Total Contribution in Group subjects
  - Individual Share
  - Profit

Public Good Game



- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals **IntEnd**
  - Amount Contributed subjects **Cont**
  - Efficiency Factor globals **Eff**
  - Total Contribution in Group subjects **TotalCont**
  - Individual Share
  - Profit

Public Good Game

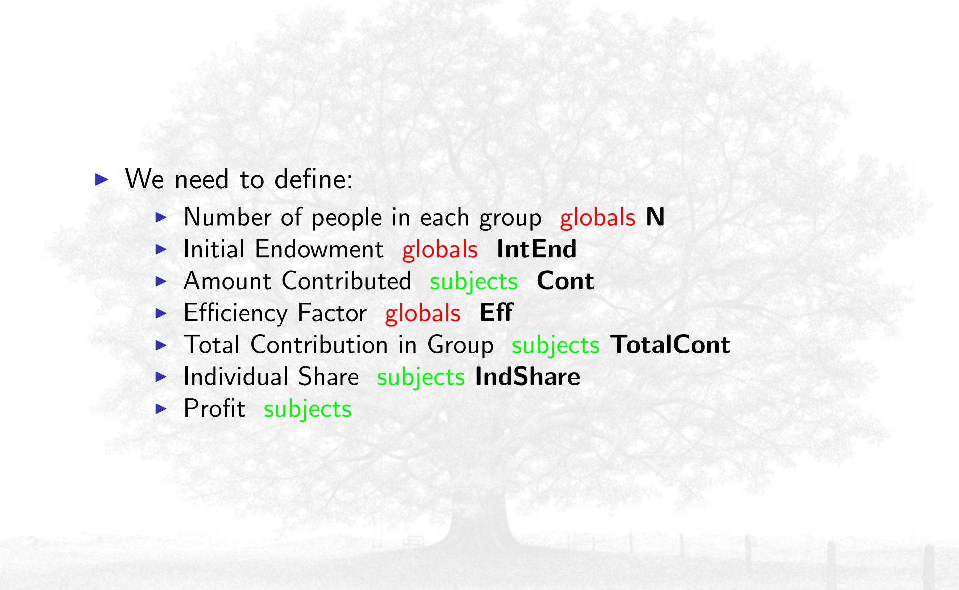


- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals **IntEnd**
  - Amount Contributed subjects **Cont**
  - Efficiency Factor globals **Eff**
  - Total Contribution in Group subjects **TotalCont**
  - Individual Share subjects **IndShare**
  - Profit

Public Good Game

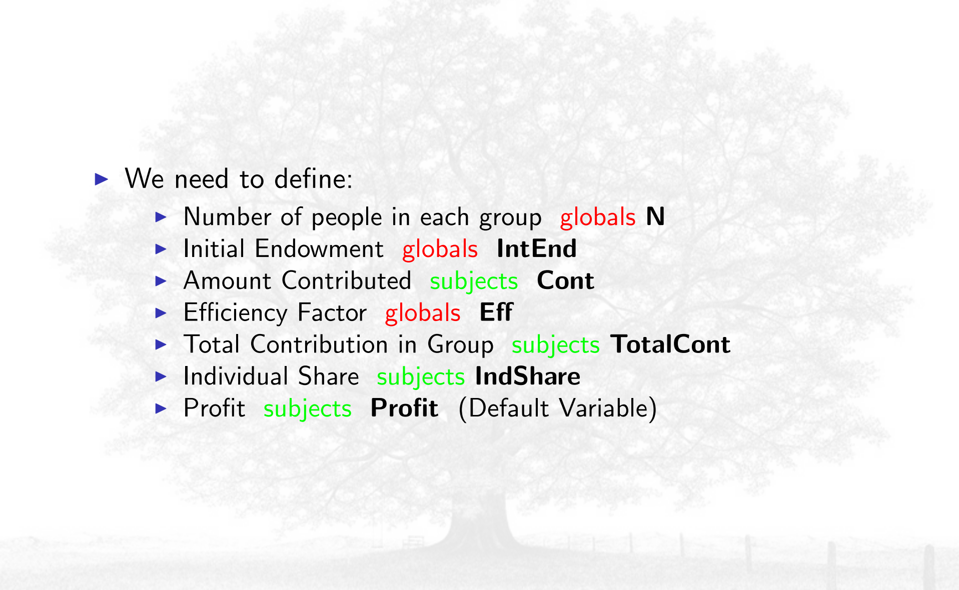
- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals **IntEnd**
  - Amount Contributed subjects **Cont**
  - Efficiency Factor globals **Eff**
  - Total Contribution in Group subjects **TotalCont**
  - Individual Share subjects **IndShare**
  - Profit

Public Good Game



- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals **IntEnd**
  - Amount Contributed subjects **Cont**
  - Efficiency Factor globals **Eff**
  - Total Contribution in Group subjects **TotalCont**
  - Individual Share subjects **IndShare**
  - Profit subjects

Public Good Game

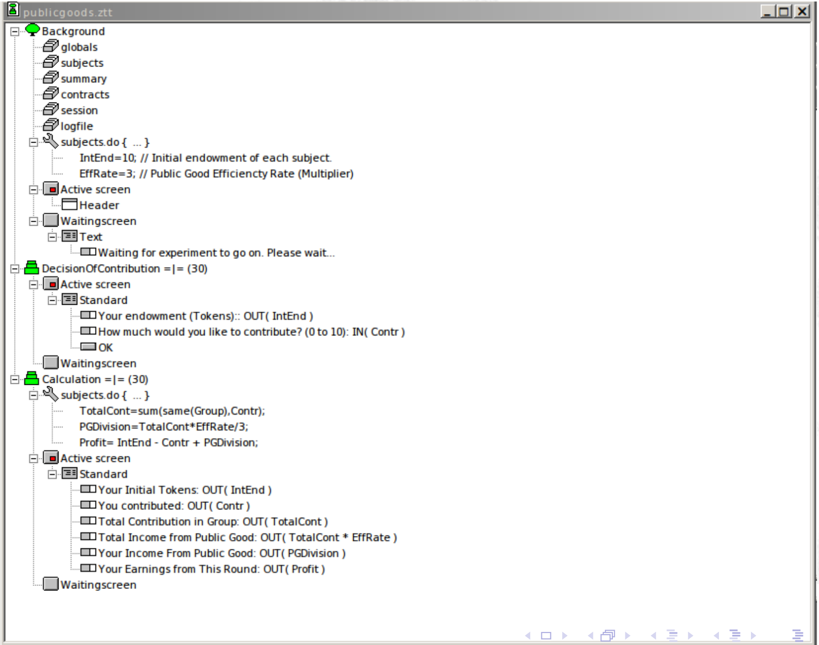


- We need to define:
  - Number of people in each group globals **N**
  - Initial Endowment globals **IntEnd**
  - Amount Contributed subjects **Cont**
  - Efficiency Factor globals **Eff**
  - Total Contribution in Group subjects **TotalCont**
  - Individual Share subjects **IndShare**
  - Profit subjects **Profit** (Default Variable)

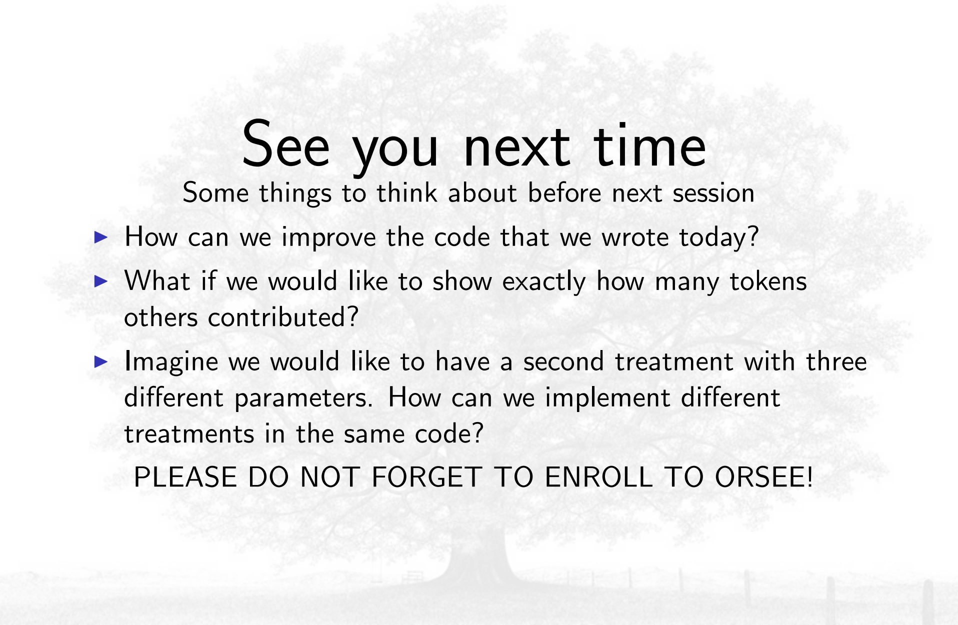
Some Table Functions

average(x), average(a, x)	Average of the values.
count(), count(a)	Number of records in the table
find(x), find(a, x)	The first value of the variable
maximum(x), maximum(a, x)	Maximum of the (found) values.
median(x), median(a, x)	Median of the (found) values.
minimum(x), minimum(a, x)	Minimum of the (found) values.
product(x), product(a, x)	Product of the (found)values.
sum(x), sum(a, x)	Sum of the (found) values.

Program should look like this:



```
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    subjects.do { ... }
        IntEnd=10; // Initial endowment of each subject.
        EffRate=3; // Public Good Efficiencty Rate (Multiplier)
    Active screen
        Header
    Waitingscreen
        Text
            Waiting for experiment to go on. Please wait...
DecisionOfContribution =|= (30)
    Active screen
        Standard
            Your endowment (Tokens):: OUT( IntEnd )
            How much would you like to contribute? (0 to 10): IN( Contr )
            OK
    Waitingscreen
Calculation =|= (30)
    subjects.do { ... }
        TotalCont=sum(same(Group),Contr);
        PGDivision=TotalCont*EffRate/3;
        Profit= IntEnd - Contr + PGDivision;
    Active screen
        Standard
            Your Initial Tokens: OUT( IntEnd )
            You contributed: OUT( Contr )
            Total Contribution in Group: OUT( TotalCont )
            Total Income from Public Good: OUT( TotalCont * EffRate )
            Your Income From Public Good: OUT( PGDivision )
            Your Earnings from This Round: OUT( Profit )
    Waitingscreen
```



See you next time

Some things to think about before next session

- How can we improve the code that we wrote today?
- What if we would like to show exactly how many tokens others contributed?
- Imagine we would like to have a second treatment with three different parameters. How can we implement different treatments in the same code?

PLEASE DO NOT FORGET TO ENROLL TO ORSEE!