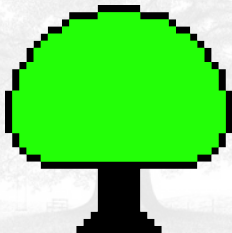


# z-Tree Programming: Part I

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

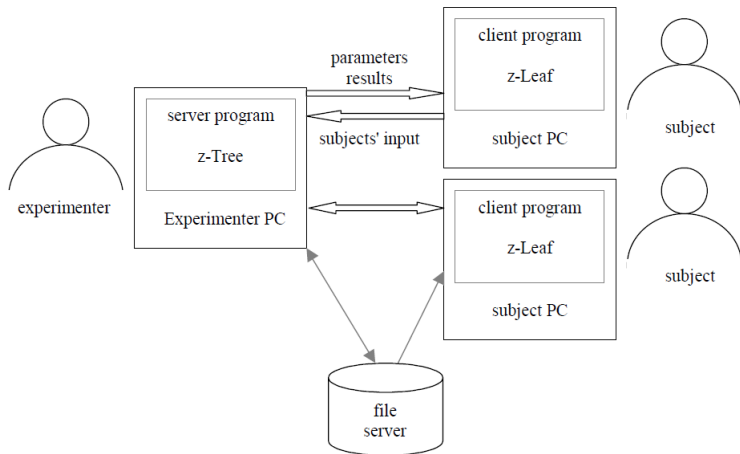
# What is z-Tree

- ▶ 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

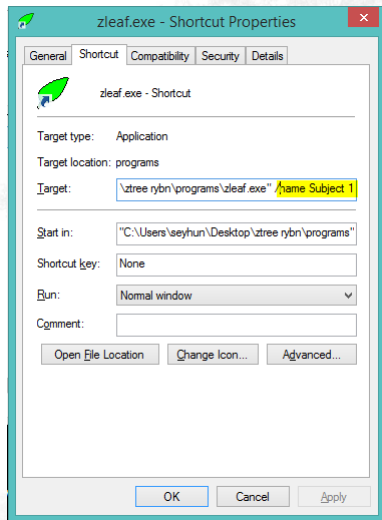
# Running zTree on your computer

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

# Running zTree on your computer

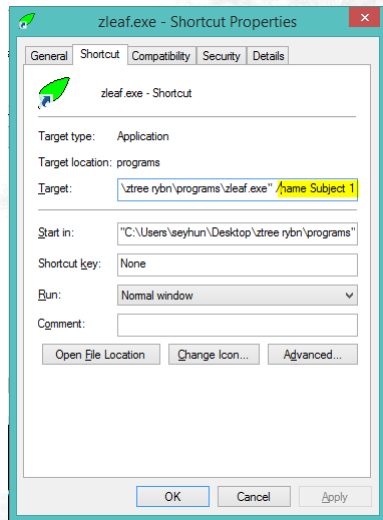
- ▶ Place *ztree.exe* and *zleaf.exe* in the same folder
- ▶ Run *ztree.exe*
- ▶ Run *zleaf.exe* (Only runs if *ztree.exe* is running)
  - ▶ Run from command line "*zleaf.exe /name SubjectName*" to give a name to the subject

# Running zTree on your computer



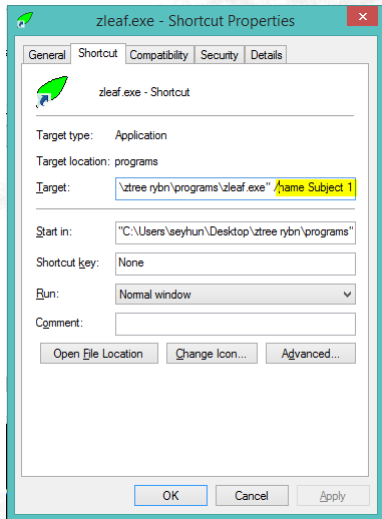
- ▶ Create a shortcut of zleaf

# Running zTree on your computer



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

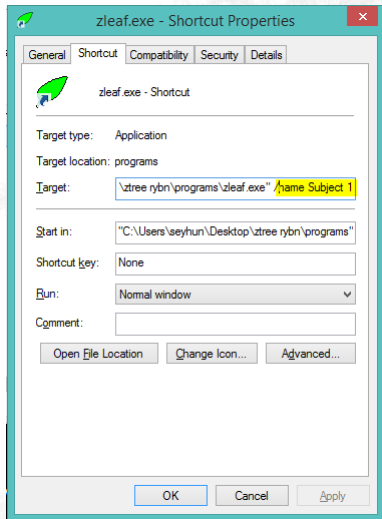
# Running zTree on your computer



- ▶ 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



- ▶ 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:

```
Command Prompt

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::hcb0: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.{C34DF010-3361-4100-BE31-395ACD7A560D}:

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>
```

- ▶ Run zTree.exe on experimenter pc

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:fff::1%1 fec0:0:0:fff::2%1 fec0:0:0:fff::3%1

Close

# Running zTree on a lab:

```
Command Prompt

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::hcb0: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.{C34DF010-3361-4100-BE31-395ACD7A560D}:

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:fff::1%1 fec0:0:0:fff::2%1 fec0:0:0:fff::3%1

Close

- ▶ Run zTree.exe on experimenter pc
- ▶ Get the ip address 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:

```
Command Prompt

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::hcb0: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.{C34DF010-3361-4100-BE31-395ACD7A560D}:

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\sseyhun>
```

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:fff::1%1 fec0:0:0:fff::2%1 fec0:0:0:fff::3%1

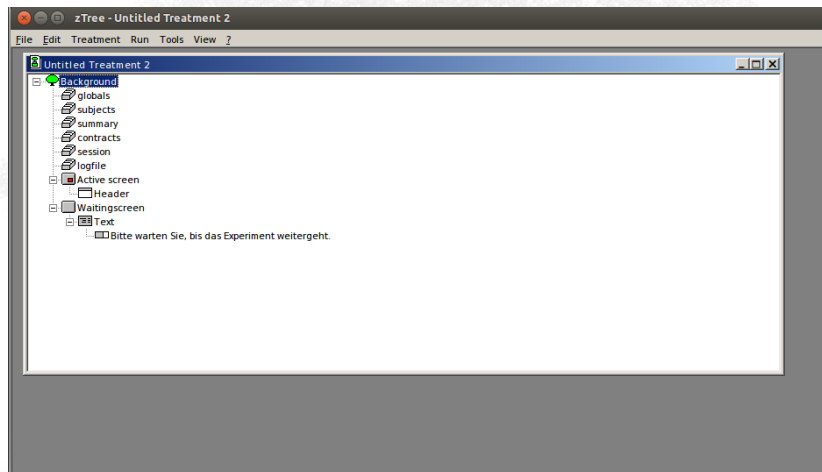
Close

- ▶ Run zTree.exe on experimenter pc
- ▶ Get the ip address 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

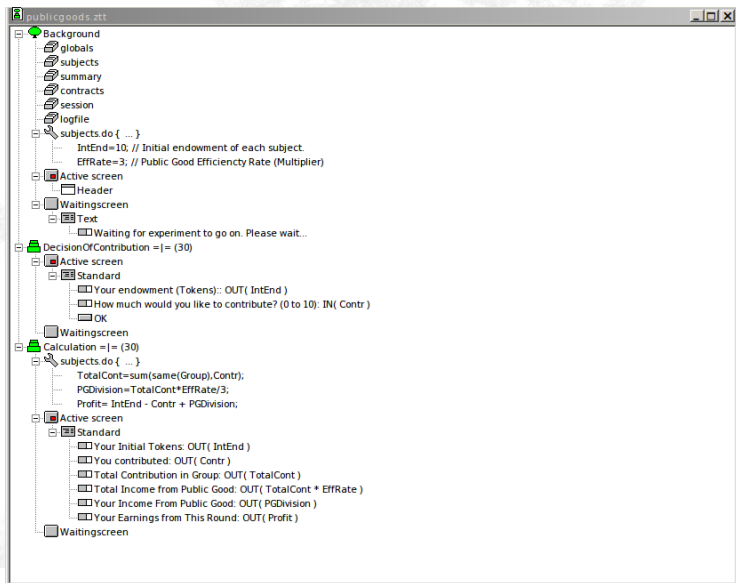
A large, leafy tree stands in a field with a fence in the foreground. The scene is captured in a soft, slightly faded style, giving it a gentle and inviting appearance. The tree's canopy is dense and rounded, and its trunk is thick and textured. The background shows a flat landscape under a pale sky.

Let's try together

# First look at z-Tree



# Structure of a Treatment file (\*.ztt)



# Structure of a Treatment file (\*.ztt)

The screenshot displays the Z-tree software interface for a file named 'publicgoods.ztt'. The interface is organized into a hierarchical tree structure. A large bracket on the right side of the screen groups the top three main sections: 'Background', 'DecisionOfContribution =| = (30)', and 'Calculation =| = (30)'. The 'Background' section includes folders for 'globals', 'subjects', 'summary', 'contracts', 'session', and 'logfile', along with a 'subjects.do { ... }' script containing parameters like 'IntEnd=10' and 'EffRate=3'. The 'DecisionOfContribution' section features an 'Active screen' with a 'Standard' screen and a 'WaitingScreen'. The 'Calculation' section includes a 'subjects.do { ... }' script with formulas for 'TotalCont', 'PGDivision', and 'Profit', followed by an 'Active screen' with a 'Standard' screen displaying various output variables and a 'WaitingScreen'.

```
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)
├── DecisionOfContribution =| = (30)
│   ├── Active screen
│   │   └── Header
│   └── WaitingScreen
│       └── Text
│           └── Waiting for experiment to go on. Please wait...
├── 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



# Structure of a Treatment file (\*.ztt)

The screenshot displays the Z-tree software interface for a file named 'publicgoods.ztt'. The interface shows a hierarchical tree structure of the experiment's design. The tree is organized into several main sections, which are grouped by brackets on the right side of the screen:

- background**: This section includes the initial setup of the experiment, such as:
  - globals
  - subjects
  - summary
  - contracts
  - session
  - logfile
  - subjects.do { ... }
    - IntEnd=10; // Initial endowment of each subject.
    - EffRate=3; // Public Good Efficiency Rate (Multiplier)
- stage 1**: This section represents the first stage of the experiment, including:
  - DecisionOfContribution =| = (30)
    - Active screen
      - Standard
        - Your endowment (Tokens):: OUT( IntEnd )
        - How much would you like to contribute? (0 to 10): IN( Contr )
        - OK
      - WaitingScreen
        - Waiting for experiment to go on. Please wait...
- 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 Efficiency Rate (Multiplier)
- Active screen
  - Header
- Waiting screen
  - 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
    - Waiting screen
- 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 )
    - Waiting screen

# 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

tables

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 Efficiency Rate (Multiplier)
- Active screen
  - Header
  - Waitingsscreen
    - 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
    - Waitingsscreen
- 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 )
    - Waitingsscreen

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

tables

program

display

display

program

display

background

stage 1

stage 2

# 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	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

subjects table									
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	TimeOKD	TotalCont	PGDMision
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	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

subjects table									
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKD	TotalCont	PGDMision
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	TimeOKd	TotalCont	PGDMision
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

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	-	-

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	NumPeriods	RepeatTre	IntEnd	EffRate
1	1	0	10	3

subjects table									
Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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**

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	-	-

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

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	-	-

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

Period	NumPeriods	RepeatTre	IntEnd	EffRate
1	1	0	10	3

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	-	-

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*

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	-	-

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*

Period	NumPeriods	RepeatTre	IntEnd	EffRate
1	1	0	10	3

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	-	-

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

- ▶ **subjects**
  - ▶ *subject* specific variables
  - ▶ usually the most important
  - ▶ Default variables: *Period*, *Subject*, *Group*, *Profit*, *TotalProfit*, *Participate*
- ▶ **session:**

# Tables

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

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	-	-

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

- ▶ **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*

Period	NumPeriod	RepeatTre	IntEnd	EffRate
1	1	0	10	3

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	-	-

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

- ▶ **subjects**
  - ▶ *subject* specific variables
  - ▶ usually the most important
  - ▶ Default variables: *Period*, *Subject*, *Group*, *Profit*, *TotalProfit*, *Participate*
- ▶ **session:**
  - ▶ *session* specific variables
  - ▶ Contains the payment information

# Tables

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

Period	NumPeriods	RepeatTre	IntEnd	EffRate
1	1	0	10	3

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	-	-

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

- ▶ **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

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

Period	NumPeriods	RepeatTre	IntEnd	EffRate
1	1	0	10	3

Period	Subject	Group	Profit	TotalProfit	Participate	Contr	TimeOKd	TotalCont	PGDMision
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	-	-

Subject	FinalProfit	ShowUpFee	ShowUpFeeInv	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

- ▶ **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

Number of groups

# practice periods

# paying periods

Exch. rate [Fr./ECU]

Lump sum payment [ECU]

Show up fee [Fr.]

Compatibility

first boxes on top

Options

without Autoscope

# Stage

**Stage**

Name

OK

Cancel

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



# Header

**Header Box**

Name   With frame OK

Width [p/%]  Distance to the margin [p/%]  Adjustment of the remaining box

Height [p/%]     left  top  right

bottom

Display condition

Show current period number  
 Show total number of periods

Name of "Period"

Term for "out of"

Prefix for trial periods

Display time

Term for "Remaining time"

Term for "Please reach a decision"

Cancel

# Program

Program

Table: subjects

Owner Variable:

OK

Cancel

Condition:

Program: 

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

# Standard Box

**Standard Box**

Name   with Frame OK

Width [p/]  Distance to the margin [p/]  Adjustment of the remaining box Cancel

Height [p/]

left  top  right

bottom

Display condition

Buttons

Position

Arrangement

In rows

In columns

# Item

**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)

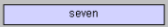

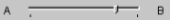
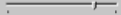


Empty allowed

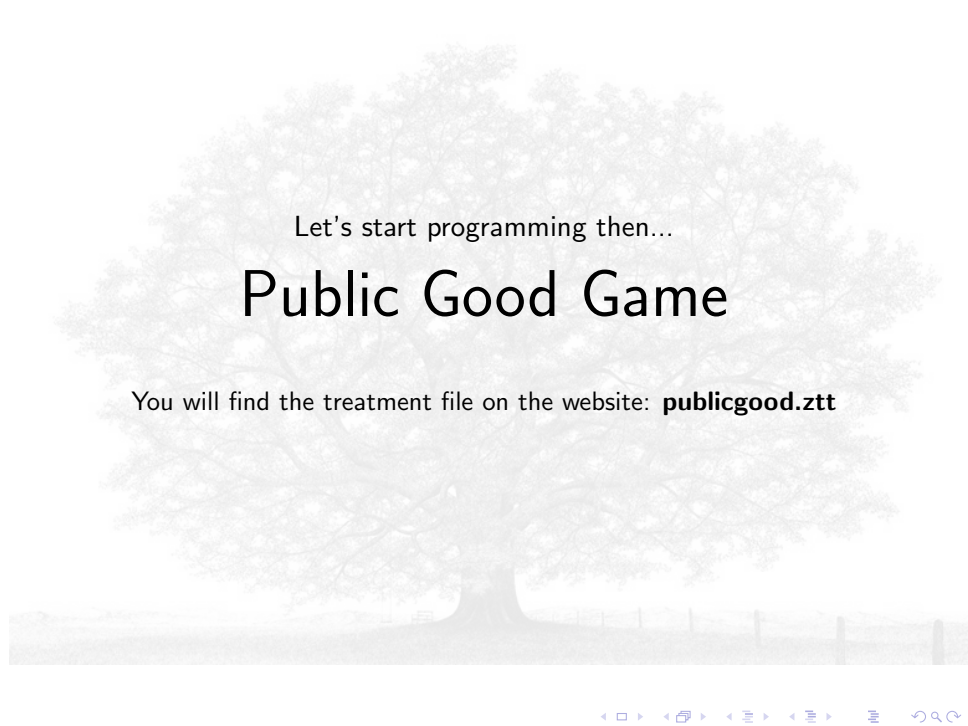
Default:

OK

Cancel

# Item

<code>!text: 7 = "seven"; 8 = "eight"; 9 = "nine";</code>		
<code>!radio: 1 = "86.8"; 24 = "102.8";</code>	<input checked="" type="radio"/> 86.8 <input type="radio"/> 102.8	<input checked="" type="radio"/> 86.8 <input type="radio"/> 102.8
<code>!radioline: 0="zero";5="five"; 6;</code>	zero <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> five	zero <input type="radio"/> <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> <input type="radio"/> five
<code>!radiosequence: 7="seven";8="eight";9="nine";</code>	<input type="radio"/> seven <input type="radio"/> eight <input type="radio"/> nine	<input type="radio"/> seven <input checked="" type="radio"/> eight <input type="radio"/> nine
<code>!slider: 0 = "A"; 100= "B"; 101;</code>	A  B	A  B
<code>!scrollbar: 0="L";100= "R";101;</code>	L  R	L  R
<code>!checkbox:1="check me";</code>	<input checked="" type="checkbox"/> check me	<input checked="" type="checkbox"/> check me
<code>!button: 1 = "accept"; 0 = "reject";</code>	<input type="button" value="accept"/> <input type="button" value="reject"/>	<input type="button" value="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**  $\text{IntEnd}$
  - ▶ Amount Contributed **subjects**  $\text{Cont}$
  - ▶ 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:

The screenshot shows a ZTC program editor window titled "publicgoods.ztc". The interface displays a hierarchical tree structure of the program's components. The tree is organized as follows:

- 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

# 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!