A short Course for

zTree

Slides and Concept by Dennis Kubitza

briq Institute on Behavior & Inequality

15. Mai 2018

Contents

Content Overview

1 Installation

2 Introduction

3 Simple Single Player Experiments

Content Overview

1 Installation

2 Introduction

3 Simple Single Player Experiments

Content Overview

2 Introduction

- General Information
  - What is z Tree ?
- How does it work ?

What is/does zTree:

- Software package for developing and carrying out economic experiments

What is/does zTree:

- Software package for developing and carrying out economic experiments
- Enables easy Creation of Experiments with GUI.

What is/does zTree:

- Software package for developing and carrying out economic experiments
- Enables easy Creation of Experiments with GUI.
- Capable of creating complex Experiments and Interaction through Programming

What is/does zTree:

- Software package for developing and carrying out economic experiments
- Enables easy Creation of Experiments with GUI.
- Capable of creating complex Experiments and Interaction through Programming
- Provides automatic Recording for Data and Payments.

Website and License

All information on zTree can be found at `http://www.ztree.uzh.ch/en.html` including:

- Download
- License Registration
- Documentation (Manuals)

Content Overview

2 Introduction

- General Information
- How does it work ?

How does zTree work ?

- zTree implements a Server/Client system.

How does zTree work ?

- zTree implements a Server/Client system.
- The Server Application is called zTree.
- The Client Application is called zLeaf.

How does zTree work ?

- zTree implements a Server/Client system.
- The Server Application is called zTree.
- The Client Application is called zLeaf.

In this course we will only consider a Setup on a single PC Nevertheless it is possible to launch it in a Lab and even Online.

Usage Example

Preparation

- Copy the Folder zTree, Paste it in the same directory
- Rename the Copied Folder as Example
- Download `https://github.com/DennisKubitza/DennisKubitza.github.io/raw/master/zTree/example1.ztt`
- Start zTree and open the .ztt file

Usage Example

Conducting the Experiment on your own PC

- Start a Client zLeaf
- Alt-Tab and open zTree
- Click on Background
- click on Run > Start Treatment
- Switch to the client
- Play the game

When you see the message Experiment done: Alt-Tab and choose zTree again.

- Click on Run >Clients Table
- Click on the upper Left Corner
- Click on Run > Leave Stage

Content Overview

1 Installation

2 Introduction

3 Simple Single Player Experiments

Goal

Task: We want to implement the example.
Goals:

- Understand the Usage of zTree
- Define local and global variables.
- Start first basic programming

Content Overview

3 Simple Single Player Experiments

- The Instructions
- The First Choice
- Final Screen
- Thank you for your attention

Create the instructions

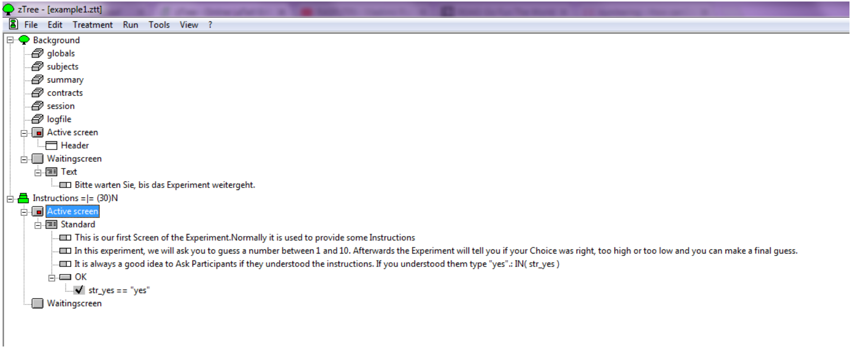
1. Click on Background
2. Choose Treatment > New Stage > Type the name and choose »no« for the timeout option
3. Click on Active Screen
4. Choose Treatment > New Box > New Standard Box > OK
5. Choose Treatment > New Item > Write some Instructions in the Label.
6. Create more Items with more Instructions.

Create the instructions

7. Create a last item:
   - Ask the participant if they understood everything in the label.
   - Click on Input
   - Type a VARIABLE name like »str_yes«
   - Type in Layout »!string«
   - Choose for MIN and MAX the value »3«
8. Choose Treatment > New Button > Click on Leave Stage yes > OK
9. Choose Treatment > New Checker
   - Type in the Condition »str_yes == "yes«<

Your are done.

Check what you did.



```
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    Active screen
        Header
    Waitingscreen
        Text
            Bitte warten Sie, bis das Experiment weitergeht.
Instructions =|= (30)N
    Active screen
        Standard
            This is our first Screen of the Experiment.Normally it is used to provide some Instructions
            In this experiment, we will ask you to guess a number between 1 and 10. Afterwards the Experiment will tell you if your Choice was right, too high or too low and you can make a final guess.
            It is always a good idea to Ask Participants if they understood the instructions. If you understood them type "yes".: IN( str_yes )
            OK
                ✓ str_yes == "yes"
    Waitingscreen
```

What did we do ?

2 We created a new Stage. Every Participant has to go through each stage. We can define a Time when the Treatment automatically ends. But we said »no«

4 We created a new box that can be displayed. A Stage can consist of more than 1 box. We can choose which boxes are displayed for which participant.

7 We defined a Input Field AND a Variable
  - The Input Field accepts Strings as we typed »!string«
  - The Input Field accepts Strings of length 3 as we set MIN and MAX to 3
  - We saved the Input as a Variable with name »str_yes«
  - This variable can only be accessed in the Descendent Branch of the Tree

8 We Defined a Button. Buttons can put a Player in the next Stage.

9 We Defined a Check for the button.
  - The Check defines a simple Program
  - Every time the Parent is pushed, the Program is executed
  - Only if the Condition evaluates as »true«, the button is executed.
  - We can use all variables, that are defined above the checker in the Tree Hierachy

Content Overview

3 Simple Single Player Experiments
- The Instructions
- The First Choice
- Final Screen
- Thank you for your attention

The First Choice

1. Click on Instructions
2. Choose Treatment > New Stage > Type the name »First_Choice« and choose »no« for the timeout option
3. Click on logfile
4. Choose Treatment > New Programm > Choose Subjects > Write in PROGRAMS

   ```
   int_hidden_number = rounddown(11*random(), 1);
   int_first_guess = 0;
   str_first_guess_was = “";
   ```
5. Create a new Standard Box

The First Choice

6. Create a new Item
   - Type as Label: »Guess a number between 0 and 10«
   - Click on Input
   - Type a VARIABLE name like »int_first_guess«
   - Type in LAYOUT»1«
   - Choose for MIN the value »0«
   - Choose for MAX the value »10«
7. Choose Treatment > New Button > Click on Leave Stage yes > OK

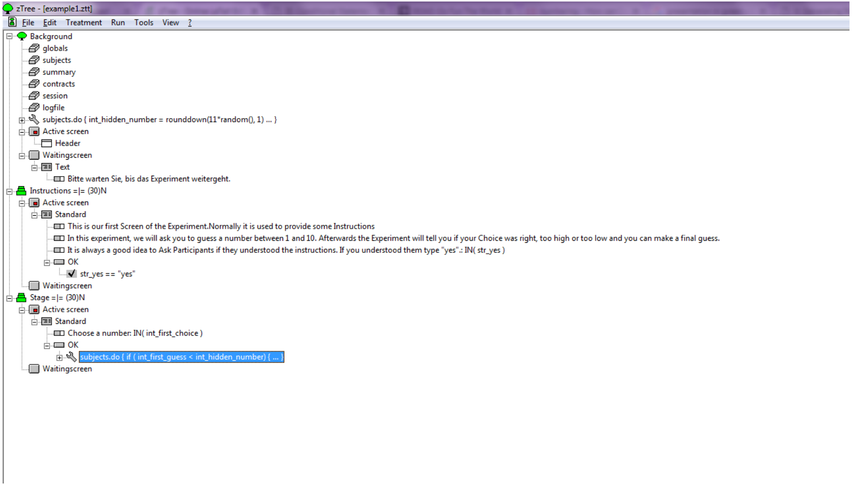
The First Choice

9. Choose Treatment > New Programm > Write in PROGRAMS:

```
if ( int_first_guess < int_hidden_number) {
str_first_guess_was = "too low";
}
elsif ( int_first_guess > int_hidden_number){
str_first_guess_was = "too high";
}
else {
str_first_guess_was = “ on point";
}
```

Your are done.

Check what you did.



```
Background
    globals
    subjects
    summary
    contracts
    session
    logfile
    subjects.do { int_hidden_number = rounddown(11*random(), 1) ... }
    Active screen
        Header
    Waitingscreen
        Text
            Bitte warten Sie, bis das Experiment weitergeht.
Instructions =|= (30)N
    Active screen
        Standard
            This is our first Screen of the Experiment.Normally it is used to provide some Instructions
            In this experiment, we will ask you to guess a number between 1 and 10. Afterwards the Experiment will tell you if your Choice was right, too high or too low and you can make a final guess.
            It is always a good idea to Ask Participants if they understood the instructions. If you understood them type "yes".: IN( str_yes )
            OK
                str_yes == "yes"
    Waitingscreen
Stage =|= (30)N
    Active screen
        Standard
            Choose a number: IN( int_first_choice )
            OK
                subjects.do { if ( int_first_guess < int_hidden_number) { ... }
    Waitingscreen
```

What did we do ?

4 We defined a program, that sets 3 variables with values of a random number, 0 and an empty string.

- These Variables are saved in the subjects table
- These Variables can be accessed and changed throughout the whole game

7 We created an input field as before, but

- We overwrote the variable int_first_guess
- We did not define a string
- By telling layout to be 1, we required that the entered value is a multiple of 1.
- By setting min and max we set boundries on the input value, not the length.

What did we do ?

9 We replaced the Checker with a more complex program:

- The Programm implements a case distinction.
- The block in { } is only executed iff the condition after »if« hols
- if it does not hold it checks the condition of elsif
- if elsif does not hold it executes else.
- The programm overwrote in each case str_first_guess

Note: You can write as many elif(...)... Statemants as you want after an if clause. You can write at most one else Statement.

Content Overview

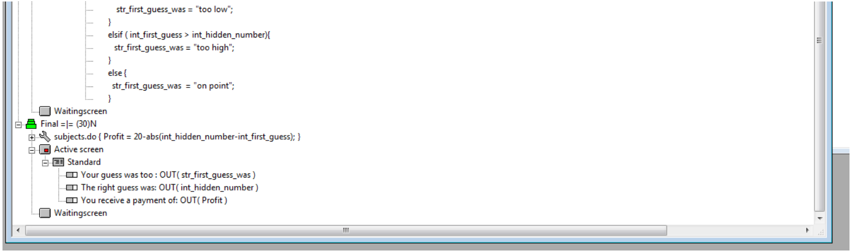
3 Simple Single Player Experiments

- The Instructions
- The First Choice
- Final Screen
- Thank you for your attention

The Final Screen

1. Click on First Choice
2. Choose Treatment > New Stage > Type the name »Final_Stage« and choose »no« for the timeout option
3. Choose Treatment > New Programm > Choose the Table subjects and type as program: Profit = 20 - abs(int_hidden_number - int_first_guess);
4. Create a new Standard Box
5. Create an Item
   - Type in as LABEL: Your choice was
   - Type in as VARIABLE »str_first_guess_was«
   - Type in as LAYOUT
6. Do the Same for the variables »int_hidden_number« and »Profit«

Check what you did.



```
      str_first_guess_was = "too low";
    }
    elsif ( int_first_guess > int_hidden_number){
      str_first_guess_was = "too high";
    }
    else {
     str_first_guess_was  = "on point";
    }
  Waitingscreen
Final =|= (30)N
  subjects.do { Profit = 20-abs(int_hidden_number-int_first_guess); }
  Active screen
    Standard
      Your guess was too : OUT( str_first_guess_was )
      The right guess was: OUT( int_hidden_number )
      You receive a payment of: OUT( Profit )
  Waitingscreen
```

What did we do ?

4 We defined a program, that is executed at the begin of the stage. The programm overwrites the special Variable »Profit«.

- »Profit« is a Variable that is integrated in zTree. You do not need to specify it like before.
- »Profit« is responsible for calculating the Payoff for each Stage.

6 We show the Particpant the Variables, that we have stored in the Subjects table

Content Overview

3 Simple Single Player Experiments

- The Instructions
- The First Choice
- Final Screen
- Thank you for your attention

Any Questions???

Exercise

Copy the Folder zTree again, rename it to Exercise1 and implement following game:

- The game generates a hidden random number between 0 and 1
- The Player can enter an integer between 1 and 10.
- If the player chooses a number smaller than 5 he receives his Input as Profit.
- ... If the player chooses a number bigger than 5 he receives the double amount if the hidden random number is smaller than 0.5
- ... Else he receives nothing.